Global Constraint Catalog, Volume II, Time-Series Constraints

Ekaterina Arafailova, Nicolas Beldiceanu, Rémi Douence, Mats Carlsson, Pierre Flener, Justin Pearson, María Andreína Francisco Rodríguez, Helmut Simonis

Global Constraint Catalog
Volume II
Time-Series Constraints

Ekaterina Arafailova, Nicolas Beldiceanu[1], Rémi Douence
IMT Atlantique (LS2N-CNRS), Nantes, France

Mats Carlsson
RISE SICS, Kista, Sweden

Pierre Flener, Justin Pearson
Uppsala University, Uppsala, Sweden

María Andreína Francisco Rodríguez,
School of Computing, National University of Singapore

Helmut Simonis
Insight Centre for Data Analytics, University College Cork, Ireland

Abstract: First this report presents a restricted set of finite transducers used to synthesise structural time-series constraints described by means of a multi-layered function composition scheme. Second it provides the corresponding synthesised catalogue of structural time-series constraints where each constraint is explicitly described in terms of automata with registers.

Keywords: constraint programming, global constraint, finite transducer, automaton with registers, reversible automaton, glue matrix, sharp bounds, parametrised invariants, meta-data, ontology, sequential pattern mining, minimum description length.

2018-09-17

[1]Corresponding author, Email: nicolas.beldiceanu@imt-atlantique.fr

Contents

Preface

"Efficiency can only be attained through generality."
– Jean-Louis Laurière, [15]

This second volume of the Global Constraint Catalogue [10] is devoted to time-series constraints. Within the context of Constraint Programming, time-series constraints go back to the work of Goldin and Kanellakis [14]. This volume contains 719 constraints, which are explicitly described in terms of automata with registers [7]. Checkers and propagators for all these constraints were synthesised [8] from 22 transducers [20, 12, 16, 19] given in the second chapter of this volume.

As in the first volume, the global constraints described in this second volume are not only accessible to humans, who can read the catalogue when searching for some information. It is also available to machines, which can read and interpret it. This is why there also exists an electronic version of this catalogue where one can get, for all time-series constraints, a complete description in terms of metadata used in the first volume. In fact, unlike the first volume, *all the metadata* of the electronic version, as well as *all text and figures* of this second volume, were automatically generated. While this second volume is by no means supposed to contain all possible time-series constraints, it contributes in the context of time-series constraints to the *systematic reconstruction* of the Global Constraint Catalogue that we have previously advocated [11]. This reconstruction is based on the following methodology:

- First reuse, adapt or come up with abstractions, which allow to concisely represent structures and properties of time series as abstract combinatorial objects. In our context these abstractions essentially correspond to:
  1. Transducers where letters of the output alphabet are interpreted as phase letters indicating the different steps for recognising pattern occurrences denoted by regular expressions.[1]
  2. Transducers glue matrices expressing the relationship between the prefix, the suffix and the full sequence passed to a transducer.
  3. Properties associated with regular expressions corresponding to fragments of the input language of our transducers.

[1] Note that an approach developed by R. Alur named *qualitative regular expressions* is presented in [1]. This approach focusses essentially on the mining aspect: given a fixed time series, the question is how to efficiently evaluate a quantitative regular function. However, it does not consider the question of generating a time series verifying a conjunction of quantitative regular functions.

- Second, create from these abstract combinatorial objects a database of concrete combinatorial objects.
- Third, synthesise concrete code for various technologies, languages, tasks from this database. In this context, correctness and efficiency [15] of the synthesised code are essentially side product of:
  - The correctness of the formulae of our database, which is itself based on well-formed abstractions.
  - The generality behind our abstract combinatorial objects.

The time-series catalogue is done in the following way:

- All time-series constraints are now defined in a *compositional way* from a few basic constituents, i.e., patterns, features, aggregators, and predicates, which completely define the meaning of a constraint, where patterns are defined using regular expressions.
- Constraint names are now constructed in a systematic way as the *concatenation* of pattern name, feature name, and aggregation or predicate name.
- Given a pattern p, checkers and constraints are now *systematically synthesised* from a transducer that, given an input sequence over the input alphabet $\{<, =, >\}$, compares two adjacent values of a time series and determines an output sequence over an output phase alphabet describing how to recognise the occurrences of p.
- For each time-series constraint associated with a pattern p, the generation of an automaton with registers is completely driven by the transducer associated with the pattern p, as well as by *decoration tables* describing for each phase letter of the output alphabet of the transducers how to generate register updates. Code optimisation is ensured by using decoration tables that depend on properties of the pattern, of the feature, and of the aggregator associated with the time-series constraint.
- Lower and upper bounds of characteristics of time series that appear in the restriction slot of a time-series constraint are synthesised from a *few parameterised formulae* that only depend on a restricted set of characteristics of the regular expression associated with the pattern [6].
- Parametrised glue matrices are provided for each transducer that corresponds to reversible time-series constraints. A concrete glue matrix is given for each reversible time-series constraint.
- Linear invariants are systematically obtained by applying the Farkas Lemma [13] to the automata with registers that were synthesised. They consist of *linear constraints typically linking consecutive register values*, e.g., see the legend of the second automaton of the MAX_MAX_PEAK, MAX_RANGE_DECREASING, MAX_RANGE_INCREASING, MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE,

MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE, MIN_MAX_PEAK, MIN_WIDTH_PLAIN, MIN_WIDTH_PLATEAU, NB_BUMP_ON_DECREASING_SEQUENCE, NB_DIP_ON_INCREASING_SEQUENCE, NB_GORGE, NB_PEAK, NB_SUMMIT, NB_VALLEY, and NB_ZIGZAG constraints, which are generated even with non-linear register updates.

- Last but not least, time-series constraints were used for generating time series verifying a conjunction of constraints both in the context of Constraint Programming [3] and in the context of Linear Programming. The latter gives a linear reformulation [2].

- In the context of sequential pattern mining, time-series constraint checkers can be used to identify and extract patterns from fixed sequences. Specialised code was synthesised for SICStus, C and Java. While the time-series catalogue may need to be extended in order to capture more patterns, having a possibly large set of fixed time-series constraints is a natural safeguard to prevent overfitting when dealing with few sequences, at a price of not finding patterns that are not covered by the catalogue.

- SICStus code is synthesised and MiniZinc code is available. The latter allows using time-series constraints on many plate forms such as Choco, Gecode, OR-tools, Cplex or Gurobi and is available from the Electronic Constraint Catalogue in Appendix A of this document.

The catalogue contains the following types of figures, which were all synthesised by computer programs producing Ti*k*Z [21] code:

- Figures representing the logo of each pattern.

- Figures for visualising the transducer associated with each pattern. The following scheme was consistently used over all transducers:

  - An arc labelled by the output symbol **out**, **maybe**, **found**, or **in** is respectively coloured in red, orange, blue, and violet.
  - If all input transitions of a state have the same colour, then the state uses this colour, otherwise it is coloured in black.
  - The name of a state is preceded by all input symbols of the transitions that enter this state. We use special characters for representing combinations of input symbols, e.g. $\leq$ for $<$ and $=$.

- Figures giving time series illustrating lower and upper bounds of the `VALUE` argument of a time-series constraint.

- Figures illustrating each constraint on a relevant time series.

- Figures providing the synthesised automaton with registers associated with each constraint. Such automata use the same graphical scheme as the transducers from which they were generated.

- Figures displaying the linear and non-linear invariants linking the result variables of two time-series constraints.

In order to see PDF annotations and attached files, you are advised to use Adobe Acrobat Reader with the PDF version of the catalogue.[2] If you do not see on your screen a small yellow bullet at the beginning of this paragraph, you are using a PDF viewer that does not fully support PDF annotations.
To get started the reader should first consult Chapter 2 which presents a short overview of the available constraint classes of this volume, their naming conventions and their arguments, and some of their potential uses. Then to get a more precise idea of the different objects used in this volume the reader may look at Chapter 3, which defines the notions of a pattern, of a feature, of an aggregator, of a seed transducer and of a glue matrix used throughout this document, as well as the conventions used for drawing time series.
The initial work on synthesising automata with registers from transducers was done by N. Beldiceanu, R. Douence, and H. Simonis. The generation of the catalogue was done by H. Simonis except, for most examples and figures, which were handled by N. Beldiceanu. The code generation part dealing with constraint checkers was rewritten by M. Carlsson for getting optimised checkers that do not use any constraint. The creation of dedicated decoration tables that depend on properties of patterns, features, and aggregators was done by E. Arafailova under the supervision of N. Beldiceanu and R. Douence [3]. The bounds based on regular expression characteristics [6], the among implied constraints [4], the linear [5] and non-linear invariants linking the results variables of two time-series constraints, and the definition of constant size finite automata encoding a restriction on the result variable of a time-series constraint were done by E. Arafailova under the supervision of N. Beldiceanu; the generation of the corresponding code was done by H. Simonis. The generation of linear invariants linking consecutive register values of a register automaton [17, 3], the generation of parametrised glue matrices [9, 2], and the generation of seed transducers from their regular expressions [18] were done by M. A. Francisco Rodríguez under the supervision of P. Flener and J. Pearson.

The authors have benefited from the following support:

- E. Arafailova and R. Douence were supported by the European H2020 FETPROACT-2014 project “GRACeFUL” (project number 640954).
- From 2014 to 2018 N. Beldiceanu was partially supported by the Gaspard Monge Programme for Optimisation and Operations Research (PGMO), as well as by the European H2020 FETPROACT-2014 project “GRACeFUL”. In 2018 N. Beldiceanu was also partially supported by the HYDDA project.
- M. Carlsson was supported by SICS.
- P. Flener, J. Pearson and M. A. Francisco Rodríguez were supported by grants 2011-6133 and 2012-4908 of the Swedish Research Council (VR).

[2] Since we are using the LaTeX packages pdfcomment and attachfile and since most PDF viewers do not support PDF annotations or attachments.

- During 2014 H. Simonis was partially supported by a senior researcher chair by the Région Pays de la Loire and by the EU FET grant ICON (project number 284715). From 2014 to 2016 H. Simonis was also supported by Science Foundation Ireland under Grant Number SFI/10/IN.1/I3032. The INSIGHT Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

Readers may send their suggestions via email to the corresponding author with `catalogue` as subject.

Cork, Ireland, Nantes, France, Uppsala, Sweden, September 2018
— EA, NB, RD, MC, PF, MAFR, JP, HS

1

Describing Global Constraints

Contents

This chapter, taken from the first volume of the global constraint catalogue [10], recalls how to describe a global constraint and its arguments.

Since global constraints have to receive their arguments in some form, no matter whether we use the graph-based or automaton-based description, we start by describing the abstract data types that we use in order to specify the arguments of a global constraint. These abstract data types are not related to any specific programming language like Caml, C, C++, Java, or Prolog. If one wants to focus on a specific language, then one has to map these abstract data types to the data types that are available in the considered language. Second we describe all the restrictions that one can impose on the arguments of a global constraint. Finally, we show how to use these ingredients in order to declare the arguments of a global constraint.

1.1 Basic data types

We use the following *basic data types*:

- `atom` corresponds to an atom. Predefined atoms are `MININT` and `MAXINT`, which respectively correspond to the *smallest integer* and *largest integer*.
- `int` corresponds to an *integer value*.
- `dvar` corresponds to a *domain variable*. A *domain variable* V is a variable that will be assigned an *integer* value taken from an initial finite set of integer values

denoted by dom(V). $\underline{V}$ and $\overline{V}$ respectively denote the minimum and maximum values of dom(V).

- `fdvar` corresponds to a *possibly unbounded domain variable*. A *possibly unbounded domain variable* is a variable that will be assigned an *integer* value from an initial finite set of integer values denoted by dom(V) or from $]-\infty, +\infty[$.
- `sint` corresponds to a *finite set of integer values*.
- `svar` corresponds to a *set variable*. A *set variable* V is a variable that will be assigned to a *finite set* of integer values. Its *lower bound* $\underline{V}$ denotes the set of integer values that for sure belong to V, while its *upper bound* $\overline{V}$ denotes the set of integer values that may belong to V. Let dom(V) = $\{\mathbf{v_1}, \ldots, \mathbf{v_n}, v_{n+1}, \ldots, v_m\}$ be a shortcut for combining the lower and upper bounds of V in a single notation:
  - **Bold** values designate those values that only belong to $\underline{V}$.
  - *Italic* values indicate those values that belong only to $\overline{V}$.
- `mint` corresponds to a *multiset of integer values*.
- `mvar` corresponds to a *multiset variable*. A *multiset variable* is a variable that will be assigned to a *multiset of integer values*.
- `real` corresponds to a *real number*.
- `rvar` corresponds to a *real number variable*. A *real number variable* is a variable that will be assigned a *real number* taken from an initial finite set of intervals. A real number is usually represented by an interval of two floating-point numbers.

1.2 Compound data types

We use the following *compound data types*:

- `list`(T) corresponds to a list of elements of the type T, where T is a basic or compound data type.
- `collection`($A_1, A_2, \ldots, A_n$) corresponds to a collection of ordered items, where each item consists of $n > 0$ attributes $A_1, A_2, \ldots, A_n$. Each attribute is an expression of the form `a` $- T$, where `a` is the *name* of the attribute and T the *type* of the attribute (a basic or compound data type). All names of the attributes of a given collection should be distinct, and different from the keyword `key`, which corresponds to an implicit[1] attribute. The value of the `key` attribute is the position of an item within the collection. The first item of a collection is associated with position 1.

[1] This attribute is not explicitly defined.

The following notation is used for instantiated arguments:

- A *list of elements* $e_1, e_2, \ldots, e_n$ is denoted by $[e_1, e_2, \ldots, e_n]$.
- A *finite set of integers* $i_1, i_2, \ldots, i_n$ is denoted by $\{i_1, i_2, \ldots, i_n\}$.
- A *collection of* n *items*, each item having m attributes, is denoted by $\langle \mathtt{a}_1 - v_{11} \ldots \mathtt{a}_m - v_{1m},\ \mathtt{a}_1 - v_{21} \ldots \mathtt{a}_m - v_{2m},\ \ldots\ , \mathtt{a}_1 - v_{n1} \ldots \mathtt{a}_m - v_{nm} \rangle$. Each item is separated from the previous item by a comma. If the items of the collection involve a single attribute $\mathtt{a}_1$, then $\langle v_{11}, v_{21}, \ldots, v_{n1} \rangle$ can be used as a shortcut for $\langle \mathtt{a}_1 - v_{11},\ \mathtt{a}_1 - v_{21},\ \ldots\ , \mathtt{a}_1 - v_{n1} \rangle$.
- *Item* i *of a collection* c is denoted by $\mathtt{c}[i]$.
- The *value of the attribute* a *of the* i^{th} *item of a collection* c is denoted by $\mathtt{c}[i].\mathtt{a}$. Note that, within an arithmetic expression, we can use the shortcut $\mathtt{c}[i]$ when the collection c involves a single attribute.
- The *number of items* of a collection c is denoted by $|\mathtt{c}|$.

1.3 Restrictions

When defining the arguments of a global constraint, it is often the case that one needs to express additional conditions that refine the type declarations of its arguments. For this purpose we provide *restrictions* that allow the specification of these additional conditions.
Currently the list of restrictions is:

- `in_list(Arg, ListAtoms)`
  - `Arg` is an argument of the type `atom`,
  - `ListAtoms` is a non-empty list of distinct atoms.

  This restriction enforces `Arg` to be one of the atoms specified in `ListAtoms`.
- `in_list(Arg, Attr, ListIntOrAtom)`
  - `Arg` is an argument of the type `collection`,
  - `Attr` is an attribute of the type `int` or `atom` of `Arg`,
  - If `Attr` is an attribute of the type `int`, then `ListIntOrAtom` is a non-empty list of distinct integers; if `Attr` is an attribute of the type `atom`, then `ListIntOrAtom` is a non-empty list of distinct atoms.

  This restriction enforces, for all items of `Arg`, the attribute `Attr` to take its value within `ListIntOrAtom`.
- `in_attr(Arg1, Attr1, Arg2, Attr2)`
  - `Arg1` is an argument of the type `collection`,

  - `Attr1` is an attribute of the type `dvar` or `int` of `Arg1`,
  - `Arg2` is an argument of the type `collection`,
  - `Attr2` is an attribute of the type `int` of `Arg2`.

  Let $\mathcal{S}_2$ denote the set of values assigned to the `Attr2` attributes of the items of `Arg2`. This restriction enforces the following condition: for all items of `Arg1`, the attribute `Attr1` takes its value in the set $\mathcal{S}_2$.

- `distinct(Arg, Attrs)`

  - `Arg` is an argument of the type `collection`,
  - `Attrs` is an attribute of the type `int` or `dvar`, or a (possibly empty) list of distinct attributes of the type `int` or `dvar` of `Arg`.

  For each pair of distinct items of `Arg`, this restriction enforces that there be at least one attribute specified by `Attrs` with two distinct values. If `Attrs` is the empty list, then all items of `Arg` should be distinct.

- `increasing_seq(Arg, Attrs)`

  - `Arg` is an argument of the type `collection`,
  - `Attrs` is an attribute of the type `int` or a list of distinct attributes of the type `int` of `Arg`.

  Let n and m respectively denote the number of items of `Arg`, and the number of attributes of `Attrs`. For item i of `Arg` let t_i denote the tuple of values $\langle v_{i,1}, v_{i,2}, \ldots, v_{i,m} \rangle$ where $v_{i,j}$ is the value of attribute j of `Attrs` of item i of `Arg`. The restriction enforces a strict lexicographical ordering on the tuples $t_1, t_2, \ldots, t_n$.

- `non_increasing_size(Arg, Attr)`

  - `Arg` is an argument of the type `collection`,
  - `Attr` is an attribute of the type `collection` of `Arg`.

  This restriction enforces for each pair of consecutive items `Arg`$[i]$, `Arg`$[i+1]$ that the number of items of `Arg`$[i]$.`Attr` is greater than or equal to the number of items of the collection `Arg`$[i+1]$.`Attr`.

- `required(Arg, Attrs)`

  - `Arg` is an argument of the type `collection`,
  - `Attrs` is an attribute or a list of distinct attributes of `Arg`.

  This restriction enforces all attributes denoted by `Attrs` to be explicitly used within all items of `Arg`.

  This restriction is usually systematically used for every attribute of a collection. It is not used when some attributes may be implicitly defined according to other

attributes. In this context, we use the `require_at_least` restriction, which we now introduce.

- `require_at_least(Atleast, Arg, Attrs)`
  - `Atleast` is a positive integer,
  - `Arg` is an argument of the type `collection`,
  - `Attrs` is a non-empty list of distinct attributes of `Arg`. The length of this list should be strictly greater than `Atleast`.

  This restriction enforces that at least `Atleast` attributes of the list `Attrs` be explicitly used within all items of `Arg`.

- `same_size(Arg, Attr)`
  - `Arg` is an argument of the type `collection`,
  - `Attr` is an attribute of the type collection of `Arg`.

  This restriction enforces that all collections of `Attr` have the same number of items.

- $\texttt{Term}_1$ `Comparison` $\texttt{Term}_2$
  - $\texttt{Term}_1$ is a *term*. A *term* is an expression that can be evaluated to one or possibly several integer values.
  - `Comparison` is one of the comparison operators $\leq, \geq, <, >, =, \neq$.
  - $\texttt{Term}_2$ is a *term*.

  Let $v_{1,1}, v_{1,2}, \ldots, v_{1,n_1}$ and $v_{2,1}, v_{2,2}, \ldots, v_{2,n_2}$ be the values respectively associated with $\texttt{Term}_1$ and $\texttt{Term}_2$. The restriction $\texttt{Term}_1$ `Comparison` $\texttt{Term}_2$ enforces $v_{1,i}$ `Comparison` $v_{2,j}$ to hold for every $i \in [1, n_1]$ and every $j \in [1, n_2]$.

  A *term* is one of the following expressions:

  - e, where e is an integer. The corresponding value is e.
  - $|c|$, where c is an argument of the type `collection`. The value of $|c|$ is the number of items of the collection c.
  - `first`$(c.a)$: `first`$(c.a)$ denotes the value assigned to the attribute a of the first item of the collection c. It is equal to 0 if the collection is empty.
  - `last`$(c.a)$: `last`$(c.a)$ denotes the value assigned to the attribute a of the last item of the collection c. It is equal to 0 if the collection is empty.
  - `sum`$(c.a)$: `sum`$(c.a)$ denotes the sum of the values assigned to the attribute a of the collection c; it is equal to 0 if the collection is empty.
  - `sum`(ℓ): `sum`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the sum of the values assigned to the attributes a_i of the collections c_i (with $i \in [1, n]$).

- `range`$(c.a)$: `range`$(c.a)$ denotes the difference between the maximum value and minimum value plus one of the values assigned to the attribute a of c; it is equal to 0 if the collection is empty.
- `range`(ℓ) : `range`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the difference between the maximum value and minimum value plus one of the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).
- `minval`$(c.a)$: `minval`$(c.a)$ denotes the minimum of the values assigned to the attribute a of the collection denoted by c: it is equal to 0 if the collection is empty.
- `minval`(ℓ) : `minval`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the minimum of the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).
- `maxval`$(c.a)$: `maxval`$(c.a)$ denotes the maximum of the values assigned to the attribute a of c; it is equal to 0 if the collection is empty.
- `maxval`(ℓ) : `maxval`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the maximum of the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).
- `nval`$(c.a)$: `nval`$(c.a)$ denotes the number of distinct values over the values assigned to the attribute a of c, it is equal to 0 if the collection is empty.
- `nval`(ℓ) : `nval`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the number of distinct values over the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).
- `prod`$(c.a)$: `prod`$(c.a)$ denotes the product of the values assigned to the attribute a of c, it is equal to 1 if the collection is empty.
- `prod`(ℓ) : `prod`(ℓ), where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the product of the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).
- `pfdc`$(c.a)$: `pfdc`$(c.a)$, where `pfdc` is a pure functional dependency constraint of the form `pfdc`(v, col) that computes a value v from a collection of variables col, and where $c.a$ is a collection with attribute a denotes the `pfdc` of the values assigned to the attribute a of c; it is equal to 0 if the collection c is empty.
- `pfdc`(ℓ) : `pfdc`(ℓ), where `pfdc` is a pure functional dependency constraint of the form `pfdc`(v, col) that computes a value v from a collection of variables col, and where ℓ is a list of collection attributes, each of the form $c_i.a_i$ (with $i \in [1, n]$), is the `pfdc` of the values assigned to the attributes a_i of the c_i (with $i \in [1, n]$).

  - t, where t is an argument of the type `int`. The value of t is the value of the corresponding argument.
  - v, where v is an argument of the type `dvar`. The value of v is the value assigned to variable v. Note that restrictions are defined on the ground instance of a global constraint.
  - s, where s is an argument of the type `sint` or `svar`. The values denoted by s are all the values of the corresponding set.
  - $c.a$, where c is an argument of the type `collection`, and a an attribute of c of the type `int` or `dvar`. The values denoted by $c.a$ are all the values corresponding to attribute a for the items of c. When $c.a$ designates a domain variable we consider the value assigned to that variable.
  - $c.a$, where c is an argument of the type `collection`, and a an attribute of c of the type `sint` or `svar`. The values denoted by $c.a$ are all the values belonging to the sets corresponding to attribute a for the items of c. When $c.a$ designates a set variable we consider the values that belong to that set.
  - t_1 `op` t_2, where t_1 and t_2 are *terms* and `op` one of the operators $+$, $-$, $*$, $/$ or mod.[2] Let $\mathcal{V}_1$ and $\mathcal{V}_2$ denote the sets of values respectively associated with the terms t_1 and t_2. The set of values associated with t_1 `op` t_2 is $\mathcal{V}_{12} = \{v : v = v_1 \texttt{ op } v_2, v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2\}$.

- $|t|$, where t is a *term*. Let $\mathcal{V}$ denote the set of values associated with the term t. The set of values associated with $|t|$ is $\mathcal{V} = \{v : v = |val|, val \in \mathcal{V}\}$.
- $\min_{v \in [\ell,u]}(t)$, where v is a variable that occurs in the term t, and ℓ, u are two terms. The value of $\min_{v \in [\ell,u]}(t)$ is the smallest value among all possible values of the term t where v is assigned a value in the interval $[\ell, u]$.
- $\max_{v \in [\ell,u]}(t)$, where v is a variable that occurs in the term t, and ℓ, u are two terms. The value of $\max_{v \in [\ell,u]}(t)$ is the largest value among all possible values of the term t where v is assigned a value in the interval $[\ell, u]$.
- Using the $\vee$ connector we can express a disjunction between two restrictions.
- Using the $\Rightarrow$ connector we can express an implication between a restriction (or a conjunction of restrictions) and a restriction.
- We can also use a constraint C of the catalogue for expressing a restriction as long as that constraint is not defined according to the constraint under consideration. The constraint C should have a graph-based or automaton-based description so that its meaning is explicitly defined.

  For instance the AMONG(`NVAR`, `VARIABLES`, `VALUES`) constraint holds if `NVAR` is the number of variables of the collection `VARIABLES` that are assigned a value from the collection of values `VALUES`.

[2] / denotes a *floor division*, that is a division in which the result is rounded to the nearest integer that is smaller or equal.

Within the restriction slot it is possible to define local variables to designate intermediate expressions. This is done by using the `where` statement after the set of restrictions.

1.4 Declaring a global constraint

Declaring a global constraint consists of providing the following information:

- A *term* `constraint`$(A_1, A_2, \ldots, A_n)$, where `constraint` corresponds to the *name* of the global constraint and $A_1, A_2, \ldots, A_n$ to its *arguments*.
- A possibly empty *list of type declarations*, where each declaration has the form `type:type_declaration`; `type` is the *name* of the new type we define and `type_declaration` is a basic data type, a compound data type, or a previously defined type.
- An *argument declaration* $A_1{:}T_1, A_2{:}T_2, \ldots, A_n{:}T_n$ giving for each argument $A_1, A_2, \ldots, A_n$ of the global constraint `constraint` its type. Each type is a basic data type, a compound data type, or a type that was declared in the list of type declarations.
- A possibly empty *list of restrictions*, where each restriction is one of the restrictions described in Section 1.3.

1.5 Keywords

This section lists the keywords used in the time-series catalogue.

- **added**: Denotes the fact that, even though the same constant is added to all variables of the `VARIABLES` collection, the corresponding constraint still holds.
- **functional dependency**: A constraint that allows for representing a functional dependency between possibly several domain variables and a single domain variable. A sequence of variables $X_1, X_2, \ldots, X_n$ is said to functionally determine another variable Y if and only if each potential tuple of values of $X_1, X_2, \ldots, X_n$ is associated with exactly one potential value of Y (i.e. Y is a function of $X_1, X_2, \ldots, X_n$).
- **reversed**: Denotes the fact that, even though the variables of the `VARIABLES` collection are reversed, the corresponding constraint still holds.
- **reverse of a constraint**: A constraint which has a reverse constraint, where the reverse is defined in the following way. Consider two constraints $c(col, r_1, \ldots, r_n)$ and $c'(col, r_1, \ldots, r_n)$ for which, in both cases, the argument col is a collection of items that functionally determines all the other arguments $r_1, \ldots, r_n$.

  The constraint c' is the *reverse constraint* of constraint c if, for any collection of items col, we have the equivalence $c(col, r_1, \ldots, r_n) \Leftrightarrow c'(col^{rev}, r_1, \ldots, r_n)$,

where col^{rev} denotes the collection col where the items of the collection are reversed. When constraints c and c' are identical, we say that constraint c is its own reverse.

2

Overview of Time-Series Constraints

Contents

This chapter provides a brief overview of the available constraint classes of this volume. Each constraint class is described in the following way:

- We give the naming convention used to construct systematically the constraint names.
- We provide the common argument pattern of all constraints belonging to a class.
- We state the purpose of the constraints of a class.
- We illustrate the use of such constraints with an example where we compute some information from a fixed time series, and another example where we generate time series satisfying a given property.

No matter to which class they belong, all constraints of this volume have a finite time series $X = \langle x_1, x_2, \ldots, x_n \rangle$ as one of their arguments, where each x_i is a fixed integer or a domain variable (see Section 1.1). We derive from X the sequence of signature values $S = \langle s_1, s_2, \ldots, s_{n-1} \rangle$ via the signature constraints $(x_i < x_{i+1} \Leftrightarrow s_i = \text{`<'}) \wedge (x_i = x_{i+1} \Leftrightarrow s_i = \text{`='}) \wedge (x_i > x_{i+1} \Leftrightarrow s_i = \text{`>'})$ for all $i \in [1, n-1]$.

Names of constraints are systematically constructed using the following grammar, where terminals use lowercase that are shown in quotation marks, even if all constraint names will be capitalised.

CONDITION	::=	'all_equal' | 'decreasing' | 'increasing'
EXTREMEAGG	::=	'max' | 'min'
AGGREGATOR	::=	EXTREMEAGG | 'sum'
NONCONSTANTFEAT	::=	'max' | 'min' | 'range' | 'surf' | 'width'
FEATURE	::=	NONCONSTANTFEAT | 'one'
MONOTONICPATTERN	::=	'decreasing' | 'decreasing_sequence' | 'increasing' | 'increasing_sequence' | 'strictly_decreasing_sequence' | 'strictly_increasing_sequence'
CONSTANTPATTERN	::=	'decreasing_terrace' | 'increasing_terrace' | 'plain' | 'plateau' | 'proper_plain' | 'proper_plateau' | 'steady' | 'steady_sequence'
PATTERN	::=	MONOTONICPATTERN | CONSTANTPATTERN | 'bump_on_decreasing_sequence' | 'dip_on_increasing_sequence' | 'gorge' | 'inflexion' | 'peak' | 'summit' | 'valley' | 'zigzag'

An extended presentation of the semantics of patterns, features, and aggregators can be found in Chapter 3. Also, to get a more precise idea of how constraint arguments are described, refer to Chapter 1.

Note that in what follows, the 'range' feature is only used with monotonic patterns, and the 'min' feature is renamed 'height' when used together with constant patterns.

The time-series constraint catalogue contains four constraints classes respectively called *predicate*, *function*, *feature*, and *extreme*, which we now describe. All examples of this chapter use the 'peak' pattern, where a *peak* is a maximal occurrence of the regular expression '$< (< \mid =)^* (> \mid =)^* >$' in the sequence of signature values associated with a time series.

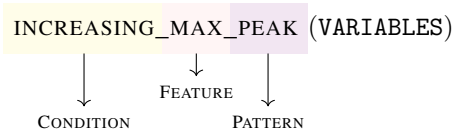
2.1 Predicate

The class[1] of constraints PREDICATENAME(VARIABLES) enforces a binary comparison operator to hold between the feature values of consecutive maximal pattern occurrences in a time series given as a collection VARIABLES of variables. The name of such a constraint is constructed in the following way:

PREDICATENAME ::= CONDITION '_' FEATURE '_' PATTERN

where CONDITION gives the binary comparison operator, i.e. 'all_equal', 'decreasing', 'increasing', used for comparing FEATURE values of consecutive maximal occurrences of PATTERN.

For example, using the 'increasing' condition, the 'max' feature, and the 'peak' pattern, we illustrate two possible uses of the constraint



namely:

- checking mode: *are all peaks of a given time series sorted by increasing maximum value?*

  For instance, INCREASING_MAX_PEAK$(\langle 5,1,2,3,2,2,2,3,4,6,1,1,6,1\rangle)$ holds since the signature of the time series $\langle 5,$ 1,2,**3**,2 $, 2,$ 2,3,4,**6**,1 $,$ 1,**6**,1 $\rangle$ contains three peaks highlighted in yellow, whose respective maximum values **3**, **6**, and **6** form an increasing sequence, namely $[3,6,6]$.

- generation mode: *generate all time series of five elements, all in the set* $\{0,1,2\}$*, where peaks are sorted by increasing maximum value.* Searching for all solutions to the constraints

$$\begin{cases} 0 \leq x_i \leq 2 \ \ (i \in [1,5]) \\ \text{INCREASING\_MAX\_PEAK}(\langle x_1,x_2,x_3,x_4,x_5\rangle) \end{cases}$$

  gives 241 solutions, 11 of which have two peaks, as shown in red below:

  - $\langle 0,1,0,1,0\rangle$
  - $\langle 0,1,0,2,0\rangle$
  - $\langle 0,1,0,2,1\rangle$
  - $\langle 0,2,0,2,0\rangle$
  - $\langle 0,2,0,2,1\rangle$
  - $\langle 0,2,1,2,0\rangle$
  - $\langle 0,2,1,2,1\rangle$
  - $\langle 1,2,0,2,0\rangle$
  - $\langle 1,2,0,2,1\rangle$
  - $\langle 1,2,1,2,0\rangle$
  - $\langle 1,2,1,2,1\rangle$

[1] The name of such class may sound a little bit weird but simply means that, unlike the other classes of time-series constraints, there is no functional dependency between arguments of such class of constraints.

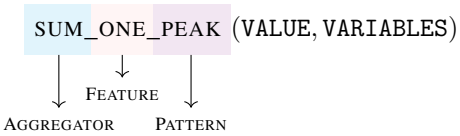
2.2 Function

The class of constraints FUNCTIONNAME(VALUE, VARIABLES) restricts VALUE to the aggregation of feature values of all pattern occurrences in a time series given as a collection VARIABLES of variables. The name of such a constraint is constructed in the following way:

FUNCTIONNAME ::= AGGREGATOR '_' FEATURE '_' PATTERN

where AGGREGATOR is the aggregator used to aggregate the FEATURE values of the maximal occurrences of PATTERN.

Using the 'one' feature, the 'sum' aggregator, and the 'peak' pattern, we illustrate two possible uses of the constraint



abbreviated as NB_PEAK where 'nb' is a shortcut for 'sum_one', namely:

- checking mode: *test whether* VALUE *can be restricted to the number of peaks of a given time series.*

  For instance, NB_PEAK(**3**, ⟨5, 1, 2, 3, 2, 2, 2, 3, 4, 5, 1, 1, 6, 1⟩) holds since the signature of the time series ⟨5, 1,2,3,2 , 2, 2,3,4,5,1 , 1,6,1 ⟩ contains three peaks highlighted in yellow, as stated by the first argument of NB_PEAK.

  Also, VALUE ∈ [0, 9] ∧ NB_PEAK(VALUE, ⟨5, 1, 2, 3, 2, 2, 2, 3, 4, 5, 1, 1, 6, 1⟩) sets VALUE to **3** since VALUE is functionally determined by the second argument of NB_PEAK.

- generation mode: *generate all time series of five elements, all in the set* {0, 1, 2}*, containing two peaks.* Searching for all solutions to the constraints

$$\begin{cases} 0 \leq x_i \leq 2 \ \ (i \in [1,5]) \\ \text{NB\_PEAK}(2, \langle x_1, x_2, x_3, x_4, x_5 \rangle) \end{cases}$$

  gives 13 solutions:

  - ⟨0, 1, 0, 1, 0⟩
  - ⟨0, 1, 0, 2, 0⟩
  - ⟨0, 1, 0, 2, 1⟩
  - ⟨0, 2, 0, 1, 0⟩
  - ⟨0, 2, 0, 2, 0⟩
  - ⟨0, 2, 0, 2, 1⟩
  - ⟨0, 2, 1, 2, 0⟩
  - ⟨0, 2, 1, 2, 1⟩
  - ⟨1, 2, 0, 1, 0⟩
  - ⟨1, 2, 0, 2, 0⟩
  - ⟨1, 2, 0, 2, 1⟩
  - ⟨1, 2, 1, 2, 0⟩
  - ⟨1, 2, 1, 2, 1⟩

2.3 Feature

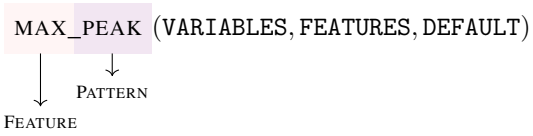
The class of constraints FEATURENAME(VARIABLES, FEATURES, DEFAULT) constraints the feature values of all maximal pattern occurrences in a time series given as a

collection VARIABLES of variables. More precisely, if position i of VARIABLES is not the first position where a maximal pattern occurrence is identified (even if this maximal pattern occurrence is not yet completed), then FEATURES$[i]$ is forced to the default value DEFAULT; otherwise, FEATURES$[i]$ is the feature value of the corresponding maximal pattern occurrence. The name of such a constraint is constructed in the following way:

FEATURENAME ::= FEATURE '_' PATTERN

where FEATURE and PATTERN respectively are the feature and the pattern we focus on. Within this class of constraints, the feature 'one' is renamed 'index' for reasons of readability.

Using the 'max' feature and the 'peak' pattern, we illustrate two possible uses of the constraint



namely:

- checking mode: *test whether the non-default values that occur in the* FEATURES *argument correspond to the maximum values of the different peaks of a time series; in addition check that the positions of these non-default values coincide with the positions where a peak is discovered, i.e. the first decrease after a sequence of increases.*

  For instance, MAX_PEAK $\left(\begin{array}{c}\langle 5,1,4,3,2,2,2,5,4,4,1,1,6,6,4,1\rangle,\\ \langle 0,0,4,0,0,0,0,5,0,0,0,0,0,6,0,0\rangle,\\ 0\end{array}\right)$ holds since the signature of the time series $\langle 5,$ 1,**4**,3,2 $, 2,$ 2,**5**,4,4,1 $,$ 1,6,**6**,4,1 $\rangle$ contains three peaks highlighted in yellow, whose maximum values **4**, **5**, and **6** where the three peaks are first identified, are positions 3, 8, and 14 as reported by the non-default values in the FEATURES argument. Note that the position where the third peak is identified corresponds to the last occurrence of value 6, namely the first decrease in the third peak.

  Also, with the same time series and the same default value as first and third arguments, MAX_PEAK$\langle 5,1,4,3,2,2,2,5,4,4,1,1,6,6,4,1\rangle$, FEATURES, $0)$ sets FEATURES to $\langle 0,0,4,0,0,0,0,5,0,0,0,0,0,6,0,0\rangle$, since FEATURES is functionally determined by the first and third arguments of MAX_PEAK.

- generation mode: *generate all time series of seven elements, all in the set* $\{0,1,2\}$*, with peaks of heights* 2 *and* 1 *at positions* 2 *and* 6*, and no peak for which the first decrease is located at position* 4. Searching for all solutions to the constraints

$$\left\{\begin{array}{l} 0 \leq x_i \leq 2 \ \ (i \in [1,7]) \\ 0 \leq f_i \leq 2 \ \ (i \in [1,7]) \\ \text{MAX\_PEAK}\left(\begin{array}{c}\langle x_1,x_2,x_3,x_4,x_5,x_6,x_7\rangle,\\ \langle f_1,f_2,f_3,f_4,f_5,f_6,f_7\rangle,\\ 0\end{array}\right) \\ f_2 = 2, f_4 = 0, f_6 = 1 \end{array}\right.$$

gives 12 solutions, all of them having two peaks located at positions 2 and 6, of heights 2 and 1, as well as no peak at position 4, as shown below:

- $\langle 0,2,0,0,0,1,0\rangle$
- $\langle 0,2,0,0,1,1,0\rangle$
- $\langle 0,2,0,1,1,1,0\rangle$
- $\langle 0,2,1,0,0,1,0\rangle$
- $\langle 0,2,1,0,1,1,0\rangle$
- $\langle 0,2,1,1,0,1,0\rangle$
- $\langle 1,2,0,0,0,1,0\rangle$
- $\langle 1,2,0,0,1,1,0\rangle$
- $\langle 1,2,0,1,1,1,0\rangle$
- $\langle 1,2,1,0,0,1,0\rangle$
- $\langle 1,2,1,0,1,1,0\rangle$
- $\langle 1,2,1,1,0,1,0\rangle$

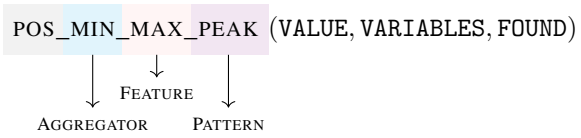
2.4 Extreme

The class of constraints EXTREMENAME(VALUE, VARIABLES, FOUND) constraints the maximum (or minimum) feature value VALUE of the maximal pattern occurrences in a time series given as a collection VARIABLES of variables. In addition it restricts the positions of those maximal pattern occurrences achieving the corresponding minimum or maximum feature values. More precisely, if position i is not the first position where a maximal pattern occurrence with a maximum (or minimum) feature value is identified (even if this pattern occurrence is not yet completed) then FOUND$[i]$ is 0; otherwise FOUND$[i]$ is 1. The name of such a constraint is constructed in the following way:

EXTREMENAME ::= 'POS_' EXTREMEAGG '_' NONCONSTANTFEAT '_' PATTERN

where EXTREMEAGG denotes the maximum or the minimum non-constant feature NONCONSTANTFEAT value over the maximal pattern occurrences of pattern PATTERN.

Using the 'max' feature, the 'peak' pattern, and the fact that we are interested in the minimum feature value, we illustrate two possible uses of the constraint



namely:

- checking mode: *test whether* VALUE *can be restricted to the altitude of the lowest peak of a time series; in addition check that the positions of the occurrences of* 1 *in* FOUND *coincide with the positions of all the lowest peaks.*

  For instance, POS_MIN_MAX_PEAK $\left(\begin{array}{c}4,\\ \langle 5,1,4,3,2,2,2,5,4,4,1,1,4,4,3,1\rangle,\\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle\end{array}\right)$

holds since the signature of the time series $\langle 5,$ 1,4,3,2 $, 2,$ 2,5,4,4,1 $,$ 1,4,4,3,1 $\rangle$ contains three peaks highlighted in yellow, where two of them are peaks that have the smallest maximum value, i.e. value 4. These two peaks are identified in positions 3 and 14 as reported by the two occurrences of 1 in the FOUND argument. Note that the position where the last peak is discovered corresponds to the last occurrence of value 4 in the time series since the first decrease takes place at this position.

Also, with the same time series as second argument, POS_MIN_MAX_PEAK(VALUE, $\langle 5,1,4,3,2,2,2,5,4,4,1,1,4,4,3,1\rangle$, FOUND) sets VALUE to 4 and FOUND to $\langle 0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle$, since VALUE and FOUND are both functionally determined by the second argument of POS_MIN_MAX_PEAK.

- generation mode: *generate all time series of seven elements, all in the set* $\{0,1,2\}$*, starting and ending with a '*1*', where the first decreases of peaks of minimum height are located in positions* 2 *and* 6.

$$\begin{cases} 0 \le v \le 2 \\ 0 \le x_i \le 2 \ \ (i \in [1,7]) \\ 0 \le f_i \le 1 \ \ (i \in [1,7]) \\ \text{POS\_MIN\_MAX\_PEAK}\left(\begin{array}{c} v, \\ \langle x_1,x_2,x_3,x_4,x_5,x_6,x_7\rangle, \\ \langle f_1,f_2,f_3,f_4,f_5,f_6,f_7\rangle \end{array}\right) \\ x_1 = 1, x_7 = 1 \\ f_2 = 1, f_6 = 1 \end{cases}$$

gives 17 solutions, with peaks of minimum height in positions 2 and 6, as shown below:

- $\langle 1,2,0,0,0,2,1\rangle$
- $\langle 1,2,0,0,1,2,1\rangle$
- $\langle 1,2,0,0,2,2,1\rangle$
- $\langle 1,2,0,1,1,2,1\rangle$
- $\langle 1,2,0,1,2,2,1\rangle$
- $\langle 1,2,0,2,0,2,1\rangle$
- $\langle 1,2,0,2,1,2,1\rangle$
- $\langle 1,2,0,2,2,2,1\rangle$
- $\langle 1,2,1,0,0,2,1\rangle$
- $\langle 1,2,1,0,1,2,1\rangle$
- $\langle 1,2,1,0,2,2,1\rangle$
- $\langle 1,2,1,1,0,2,1\rangle$
- $\langle 1,2,1,1,1,2,1\rangle$
- $\langle 1,2,1,1,2,2,1\rangle$
- $\langle 1,2,1,2,0,2,1\rangle$
- $\langle 1,2,1,2,1,2,1\rangle$
- $\langle 1,2,1,2,2,2,1\rangle$

3

Patterns and Corresponding Tables

Contents

Patterns focus on the *topological* aspect of subsequences of a time series. They are defined by two components:

- First, a regular expression over the alphabet $\{<, =, >\}$.
- Second, two non-negative integers b and a, that are intended to respectively delete a prefix and a suffix of the pattern that should be discarded for computing a characteristic of an occurrence of a pattern.

Given a pattern and a time series $x_1, x_2, \ldots, x_n$ of integer constants, called the *input values*, and forming the *input sequence*, a single integer is computed as follows:

I. Compare each pair of adjacent input values in order to build a sequence $s_1, s_2, \ldots, s_{n-1}$ of *signature values* over the alphabet $\{<, =, >\}$, as follows: $(x_i < x_{i+1} \Leftrightarrow s_i = \text{`} < \text{'}) \wedge (x_i = x_{i+1} \Leftrightarrow s_i = \text{`} = \text{'}) \wedge (x_i > x_{i+1} \Leftrightarrow s_i = \text{`} > \text{'})$. The signature values form the *signature sequence*.

II. Within the signature sequence, find *all* maximal words matching the regular expression associated with the pattern.

III. For each found pattern occurrence, discard its prefix and suffix of length b and a to obtain an integer sequence e for which we compute an integer *feature value*, so that we obtain a *feature sequence*. The features we currently consider are `one`, `width`, `surf`, `min`, `max`, and `range`, and correspond respectively to the value 1, to the number of elements of e, to $\sum_{i \in e} x_i$, to $\min_{i \in e} x_i$, to $\max_{i \in e} x_i$, and to $\max_{i \in e} x_i - \min_{i \in e} x_i$.

IV. Aggregate the values of the feature sequence into a single integer value. The aggregators we currently consider are taking the sum (`Sum`), taking the minimum (`Min`), and taking the maximum (`Max`). The feature `one` only makes sense with the `Sum` aggregator.

Definition 1 (*s*-occurrence, *i*-occurrence, *e*-occurrence)**.** *Given an input sequence $x_1, x_2, \ldots, x_n$, its signature sequence $S = s_1, s_2, \ldots, s_{n-1}$, a pattern $\langle r, b, a \rangle$, and a non-empty signature subsequence $s_i, s_{i+1}, \ldots, s_j$, with $1 \leq i \leq j \leq n-1$, forming a maximal word that matches r, the* s-occurrence *$(i..j)$ is the index sequence $i, i+1, \ldots, j$; the* i-occurrence *$[(i+b)..j]$ is the index sequence $i+b, \ldots, j$; and the* e-occurrence *$[[(i+b)..(j+1-a)]]$ is the index sequence $i+b, \ldots, j+1-a$.*

Figures representing time series use the following convention for denoting i-occurrences, s-occurrences and e-occurrences:

- Positions belonging to an i-occurrence are represented by a red circle $\bullet$.
- Positions belonging to a s-occurrence, but not to an i-occurrence, are represented by a circle $\circ$.
- Positions of an occurrence of pattern corresponding to an e-occurrence have a background coloured in yellow.

When an i-occurrence and an e-occurrence of two consecutive occurrences of a pattern coincide a red circle $\bullet$ is used.

Figures representing transducers and automata use the following conventions:

- An arrow coming from nowhere denotes an initial state.
- A double circle indicates an accepting state.
- The acceptance function of a register automaton is depicted by a box connected by dotted lines to each state of the register automaton.
- On the transitions of a transducer, a colon separates the consumed input letter(s) from the produced output letter(s).
- A transition labelled by the input letter $\leq$ is a shortcut for two transitions respectively labelled by the letters $<$ and $=$. The same convention is used for the letters $\geq$, $\gtrless$, $\gtreqqless$, which respectively represent a set of transitions, each of them labelled by a letter of $\{>,=\}$, $\{>,<\}$, $\{>,=,<\}$.

The name of a constraint is defined by the concatenation of the aggregator function, the feature name and the pattern. For each pattern, a so-called *seed transducer* recognises all pattern occurrences.

Definition 2 (seed transducer)**.** *A* seed transducer *is a transducer for which all states are accepting with input alphabet* $\{<,=,>\}$*, output alphabet, also-called phase alphabet,* $\{\mathbf{out}, \mathbf{maybe}_{\mathrm{b}}, \mathbf{out}_{\mathrm{r}}, \mathbf{found}, \mathbf{maybe}_{\mathrm{b}}, \mathbf{in}, \mathbf{out}_{\mathrm{a}}, \mathbf{found}_{\mathrm{e}}\}$.

Definition 3 (t-occurrence)**.** *Given an input sequence s over the input alphabet* $\{<,=,>\}$ *the* t-occurrence *of s wrt a given seed transducer corresponds to the indices of the phase letters of a maximum word in the generated output sequence that matches* '$\mathbf{maybe}_{\mathrm{b}}^{*}\mathbf{found}_{\mathrm{e}} \mid \mathbf{maybe}_{\mathrm{b}}^{*}\mathbf{found}(\mathbf{maybe}_{\mathrm{a}}^{*}\mathbf{in}^{+})^{*}$'.

Definition 4 (well-formedness)**.** *A seed transducer is* well formed *if (1) all output sequences it produces are recognised by the finite deterministic automaton depicted in Figure 3.1, and if (2) the t-occurrence for any input sequence s coincides with the i-occurrence of the pattern for s.*

Well-formedness of seed transducers guarantees that the automata with registers obtained by applying the decoration tables for replacing phase letters of the output alphabet by register updates compute the expected result [8].

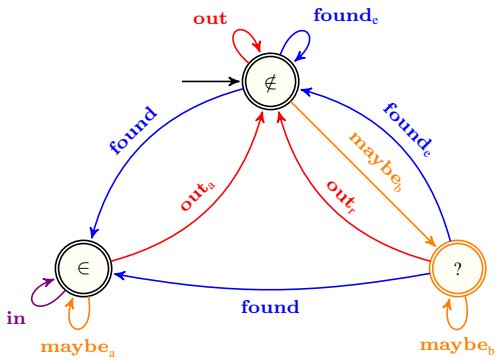


Figure 3.1: Automaton describing the output language of a well-formed transducer; states $\notin$, ?, $\in$ respectively mean that (1) we are outside an occurrence of pattern, (2) we are potentially inside an occurrence of pattern, (3) we are inside an occurrence of pattern that is not yet finished.

Following the presentation in [9, 2], we introduce the notion of glue matrix. A *glue matrix* for a reversible time-series constraint c specified by a seed transducer σ, an aggregator g and a feature f is a matrix indexed by the states of σ and the states of σ^{rev}, and parametrised by g and f, where σ^{rev} is a seed transducer of the reverse of constraint c. Consider the automaton $\mathcal{A}_\sigma$ (respectively $\mathcal{A}_\sigma^{\text{rev}}$) with registers R, C and D obtained by applying the decoration table 3.37 to the seed transducer σ (respectively σ^{rev}). Let $\mathcal{A}_\sigma$ reach state $\overrightarrow{Q}$ and register values $\left\langle \overrightarrow{C}, \overrightarrow{D}, \overrightarrow{R} \right\rangle$ on a prefix of a word w. Similarly let $\mathcal{A}_\sigma^{\text{rev}}$ reach state $\overleftarrow{Q}$ and register values $\left\langle \overleftarrow{C}, \overleftarrow{D}, \overleftarrow{R} \right\rangle$ on the reverse of the corresponding suffix of w. The value returned by $\mathcal{A}_\sigma$ on the entire word w is $\phi_g(\overrightarrow{R}, \overleftarrow{R}, \Gamma)$, where Γ is the entry of the glue matrix corresponding to row $\overrightarrow{Q}$ and column $\overleftarrow{Q}$. That entry contains an expression parametrised by g and f involving a subset of the registers $\overrightarrow{C}$, $\overrightarrow{D}$, $\overleftarrow{C}$, $\overleftarrow{D}$.

3.1 Patterns, Seed Transducers and Parametrised Glue Matrices

Feature f	id_f	$\min_f$	$\max_f$	ϕ_f	δ_f^i	$\hat{\delta}_f^i$
`one`	0	0	1	$\max$	1	-1
`width`	0	0	$n+1$	$+$	1	-1
`surf`	0	$-\infty$	$+\infty$	$+$	x_i	$-x_i$
`max`	$-\infty$	$-\infty$	$+\infty$	$\max$	x_i	n/a
`min`	$+\infty$	$-\infty$	$+\infty$	$\min$	x_i	n/a
`range`	0	0	$+\infty$	n/a	x_i	n/a

Table 3.1: Features: identity, minimum, and maximum values; the operators ϕ_f and δ_f^i recursively define the feature value v_u of a time series $x_\ell, \ldots, x_u$ by $v_\ell = \phi_f(\mathrm{id}_f, \delta_f^\ell)$ and $v_i = \phi_f(v_{i-1}, \delta_f^{i+1})$ for $i > \ell$, where δ_f^i is the contribution of x_i to v_u; $\hat{\delta}_f^i$ cancel out the contribution of x_i within a feature value so that this contribution is not counted twice; n and n/a respectively stand for the length of the time series and for *not available*.

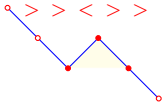
Aggregator g	ϕ_g	$\mathrm{default}_{gf}$
`Max`	$\max$	$\min_f$
`Min`	$\min$	$\max_f$
`Sum`	$+$	0

Table 3.2: Aggregators: operators and identity values relative a feature f.

Predicate θ	$\mathtt{default}_\theta$
$\leq$	$-\infty$
$=$	x (free variable)
$\geq$	$+\infty$

Table 3.3: Default value for a predicate θ used in the decoration table 3.38

3.1.1 BUMP_ON_DECREASING_SEQUENCE



Given a sequence S over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, an occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence of S which matches the regular expression '>><>>'. Part (A) and parts (B,C) of Figure 3.2 respectively depict the seed transducer associated with the BUMP_ON_DECREASING_SEQUENCE pattern as well as one example of its execution on a time series.

The pattern DIP_ON_INCREASING_SEQUENCE = '<<><<' is not the reverse of the pattern BUMP_ON_DECREASING_SEQUENCE even though '>><>>' is the mirror word of '<<><<'.[1] This is because the constants $b = 2$ and $a = 1$ (see Figure 3.2) used for trimming the regular expression '>><>>' are different, which makes the two corresponding *e*-occurrences different. As a consequence we do not have any glue matrix for the BUMP_ON_DECREASING_SEQUENCE pattern.

[1] To get the mirror of a word w over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, first reverse the word w, second swap the '>' with the '<'.

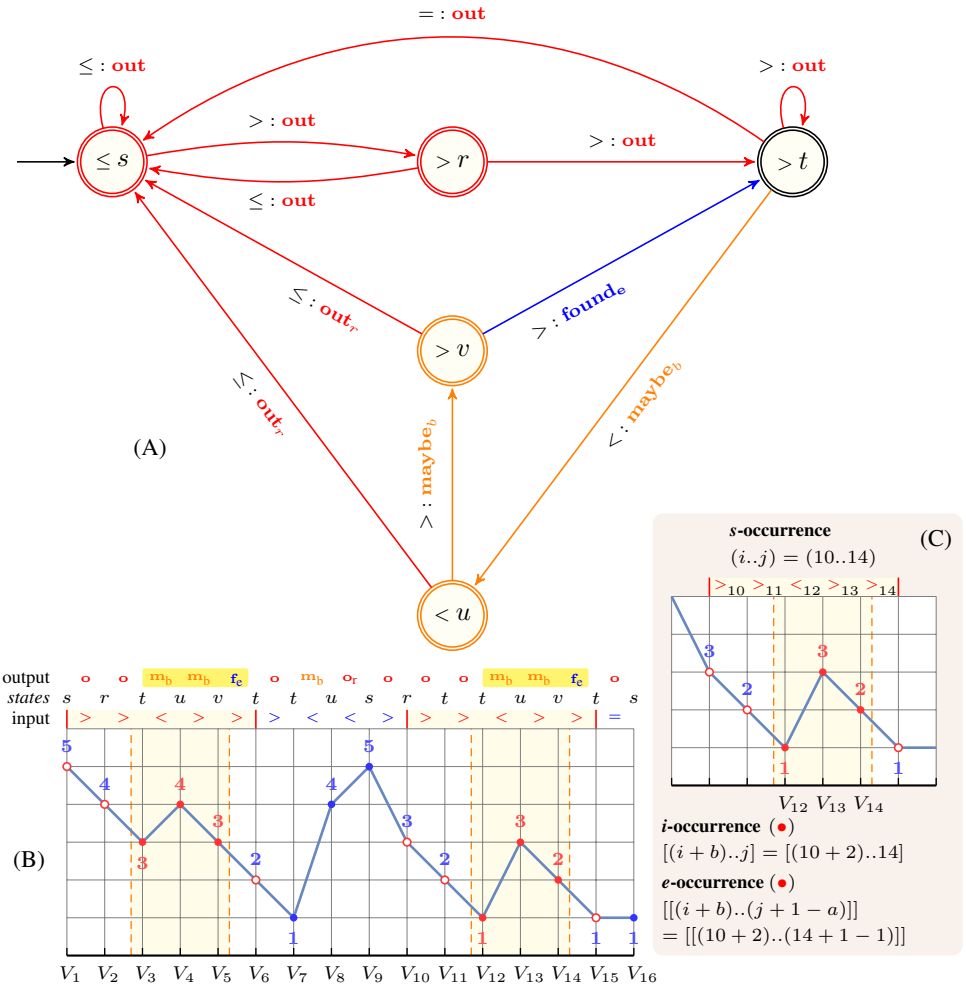


Figure 3.2: (A) Seed transducer for the BUMP_ON_DECREASING_SEQUENCE pattern: '>><>>' with $b = 2$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{o}_\mathrm{r}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{f}_\mathrm{e}$ are shortcut for $\mathbf{out}$, $\mathbf{out}_\mathrm{r}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{found}_\mathrm{e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

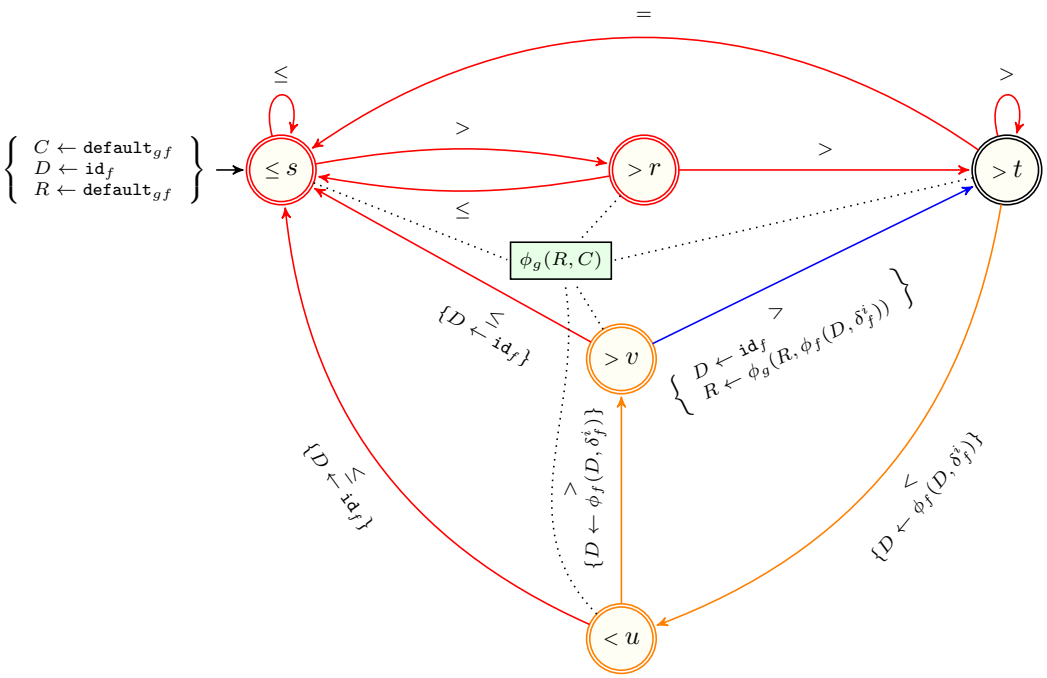


Figure 3.3: Parametrised automaton for any functional time-series constraints of the BUMP_ON_DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

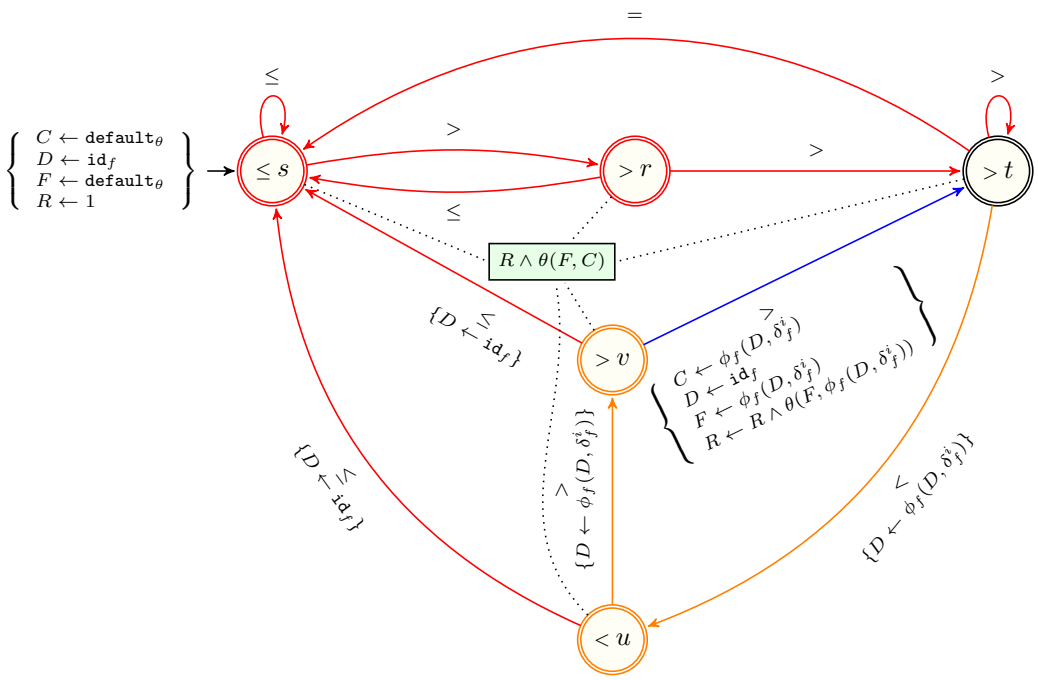


Figure 3.4: Parametrised automaton for any predicate time-series constraints of the BUMP_ON_DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

3.1.2 DECREASING

Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern DECREASING is the subsequence of S which matches the regular expression '>'. Part (A) and parts (B,C) of Figure 3.5 respectively depict the seed transducer associated with the DECREASING pattern as well as one example of its execution on a time series.

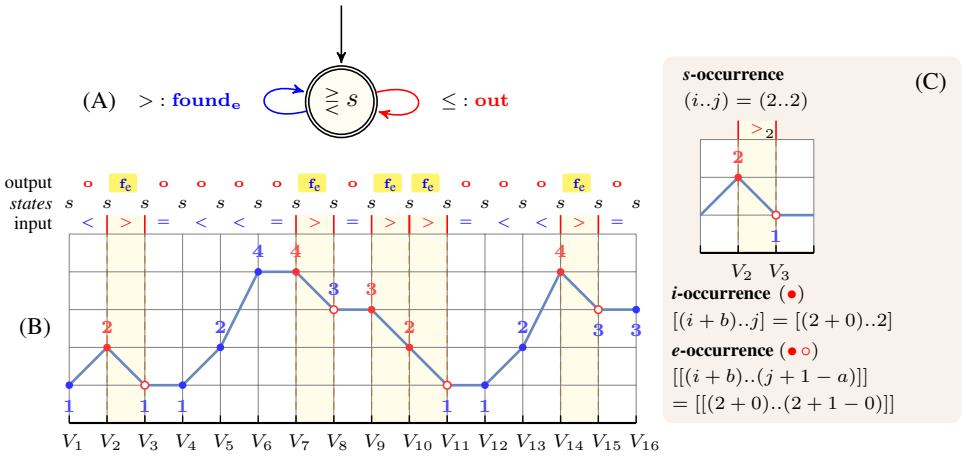


Figure 3.5: (A) Seed transducer for the DECREASING pattern: '>' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{f_e}$ are shortcut for $\mathbf{out}$, $\mathbf{found_e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

	s
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.4: Parametrised glue matrix for any $g_f_$DECREASING constraint; the state associated with each row corresponds to a state of the DECREASING transducer – see Figure 3.5, while the state associated with each column corresponds to a state of the reverse of the DECREASING transducer, i.e. the INCREASING transducer – see Figure 3.20.

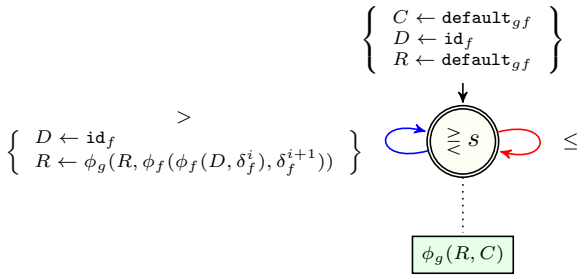


Figure 3.6: Parametrised automaton for any functional time-series constraints of the DECREASING pattern obtained by applying the decoration table 3.37 to the corresponding transducer

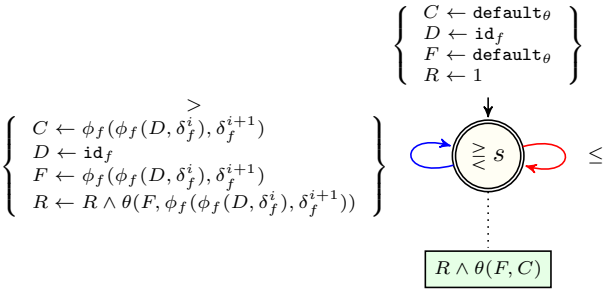
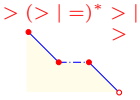


Figure 3.7: Parametrised automaton for any predicate time-series constraints of the DECREASING pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

3.1.3 DECREASING_SEQUENCE



Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence of S which matches the regular expression '$> (> \mid =)^* > \mid >$'. Part (A) and parts (B,C) of Figure 3.8 respectively depict the seed transducer associated with the DECREASING_SEQUENCE pattern as well as one example of its execution on a time series.

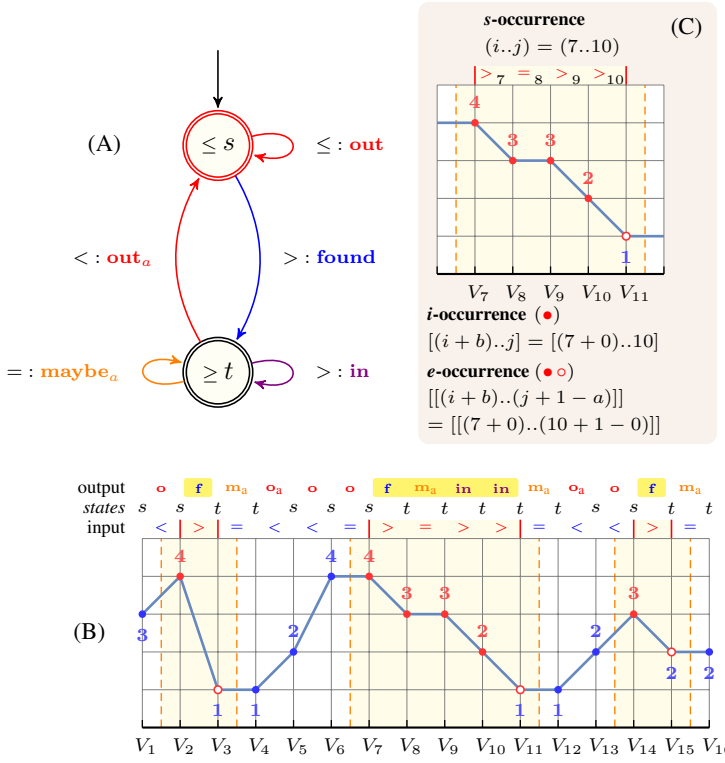


Figure 3.8: (A) Seed transducer for the DECREASING_SEQUENCE pattern: '$> (> \mid =)^* > \mid >$' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f** $\mathbf{m}_a$, $\mathbf{o}_a$ are shortcut for **out**, **found** $\mathbf{maybe}_a$, $\mathbf{out}_a$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

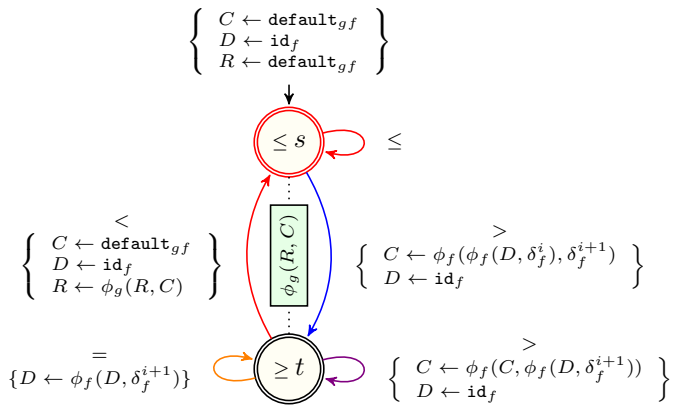


Figure 3.9: Parametrised automaton for any functional time-series constraints of the DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

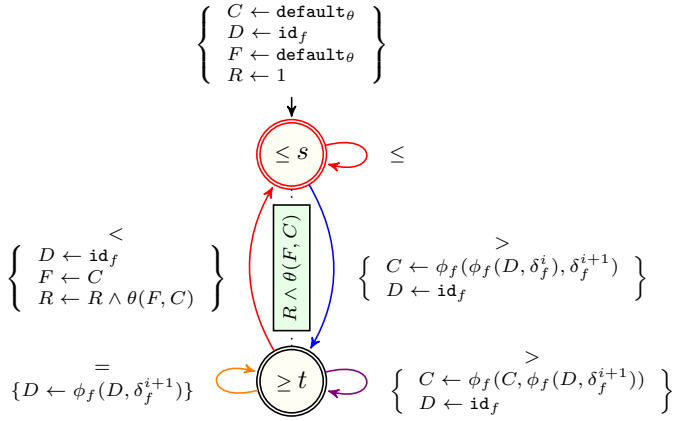
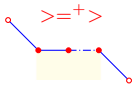


Figure 3.10: Parametrised automaton for any predicate time-series constraints of the DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \hat{\delta}_f^{i+1})$ M

Table 3.5: Parametrised glue matrix for any g_f_DECREASING_SEQUENCE constraint, where the cell annotation M stands for merging two existing decreasing sequences sharing an element; the state associated with each row corresponds to a state of the DECREASING_SEQUENCE transducer – see Figure 3.8, while the state associated with each column corresponds to a state of the reverse of the DECREASING_SEQUENCE transducer, i.e. the INCREASING_SEQUENCE transducer – see Figure 3.23.

3.1.4 DECREASING_TERRACE



Given a sequence S over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, an occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence of $S$ which matches the regular expression `>=+>'. Part (A) and parts (B,C) of Figure 3.11 respectively depict the seed transducer associated with the DECREASING_TERRACE pattern as well as one example of its execution on a time series.

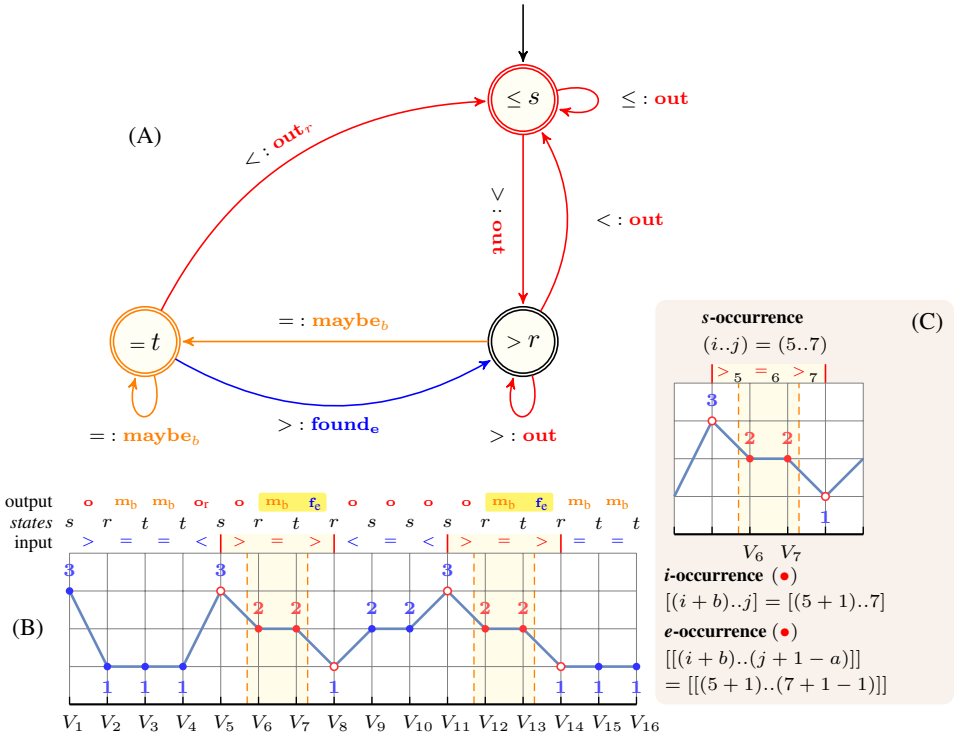


Figure 3.11: (A) Seed transducer for the DECREASING_TERRACE pattern: `>=+>' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{o}_\mathrm{r}$, $\mathbf{f}_\mathrm{e}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{out}_\mathrm{r}$, $\mathbf{found}_\mathrm{e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

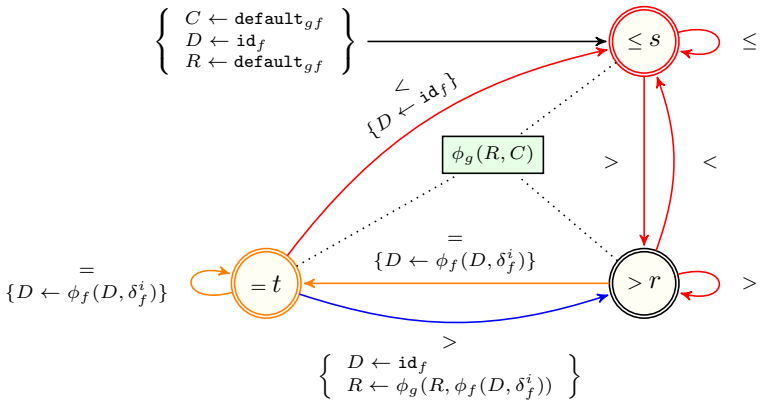


Figure 3.12: Parametrised automaton for any functional time-series constraints of the DECREASING_TERRACE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

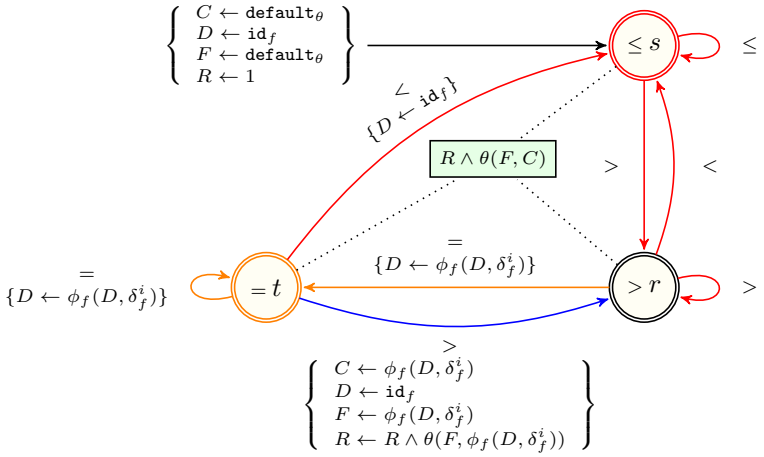
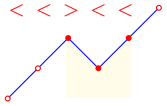


Figure 3.13: Parametrised automaton for any predicate time-series constraints of the DECREASING_TERRACE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.6: Parametrised glue matrix for any g_f_DECREASING_TERRACE constraint, where the cell annotation C stands for creating a new decreasing terrace occurrence from no existing decreasing terrace; the state associated with each row corresponds to a state of the DECREASING_TERRACE transducer – see Figure 3.11, while the state associated with each column corresponds to a state of the reverse of the DECREASING_TERRACE transducer, i.e. the INCREASING_TERRACE transducer – see Figure 3.26.

3.1.5 DIP_ON_INCREASING_SEQUENCE



Given a sequence S over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, an occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence of S which matches the regular expression '<<><<'. Part (A) and parts (B,C) of Figure 3.14 respectively depict the seed transducer associated with the DIP_ON_INCREASING_SEQUENCE pattern as well as one example of its execution on a time series.

The pattern BUMP_ON_DECREASING_SEQUENCE = '>><>>' is not the reverse of the pattern DIP_ON_INCREASING_SEQUENCE even though '<<><<' is the mirror word of '>><>>'.[2] This is because the constants $b = 2$ and $a = 1$ (see Figure 3.14) used for trimming the regular expression '<<><<' are different, which makes the two corresponding *e*-occurrences different. As a consequence we do not have any glue matrix for the DIP_ON_INCREASING_SEQUENCE pattern.

[2]To get the mirror of a word w over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, first reverse the word w, second swap the '>' with the '<'.

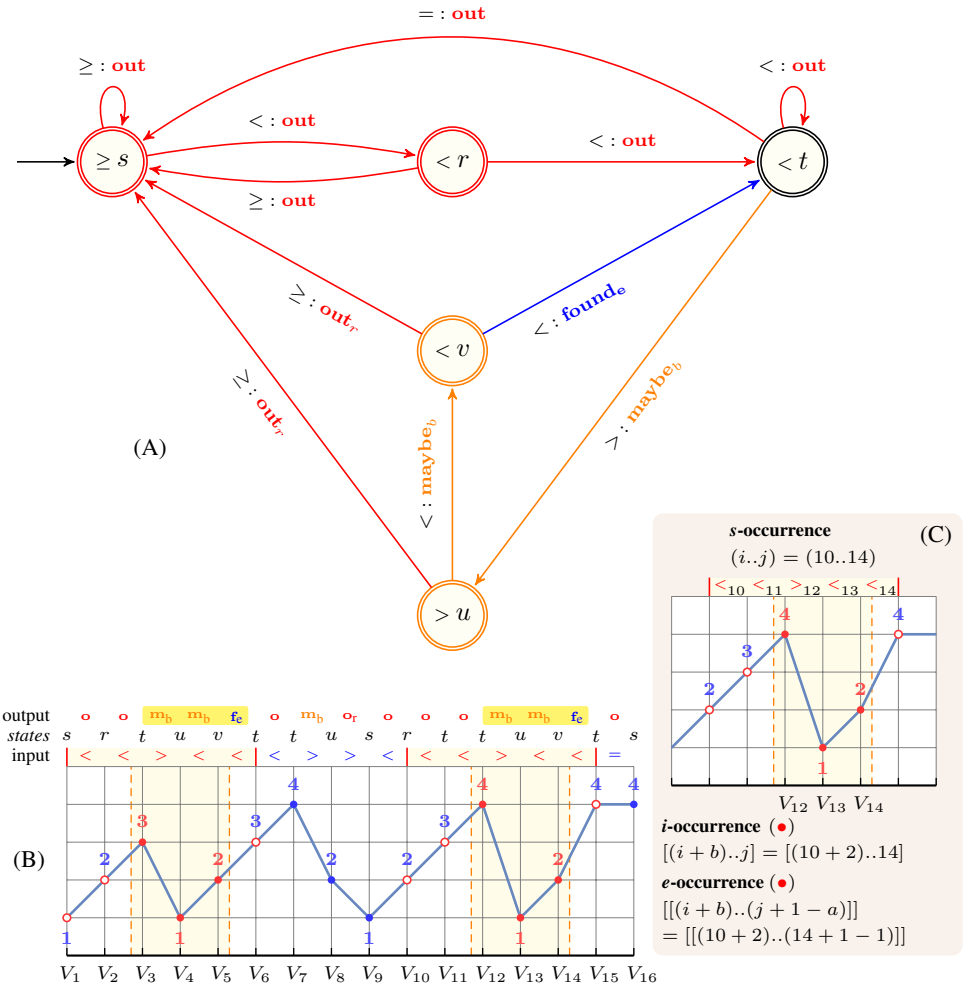


Figure 3.14: (A) Seed transducer for the DIP_ON_INCREASING_SEQUENCE pattern: '<<>><<' with $b = 2$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{o}_\mathrm{r}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{f}_\mathrm{e}$ are shortcut for $\mathbf{out}$, $\mathbf{out}_\mathrm{r}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{found}_\mathrm{e}$); (C) Illustrating the relation between the s-occurrence, the i-occurrence and the e-occurrence.

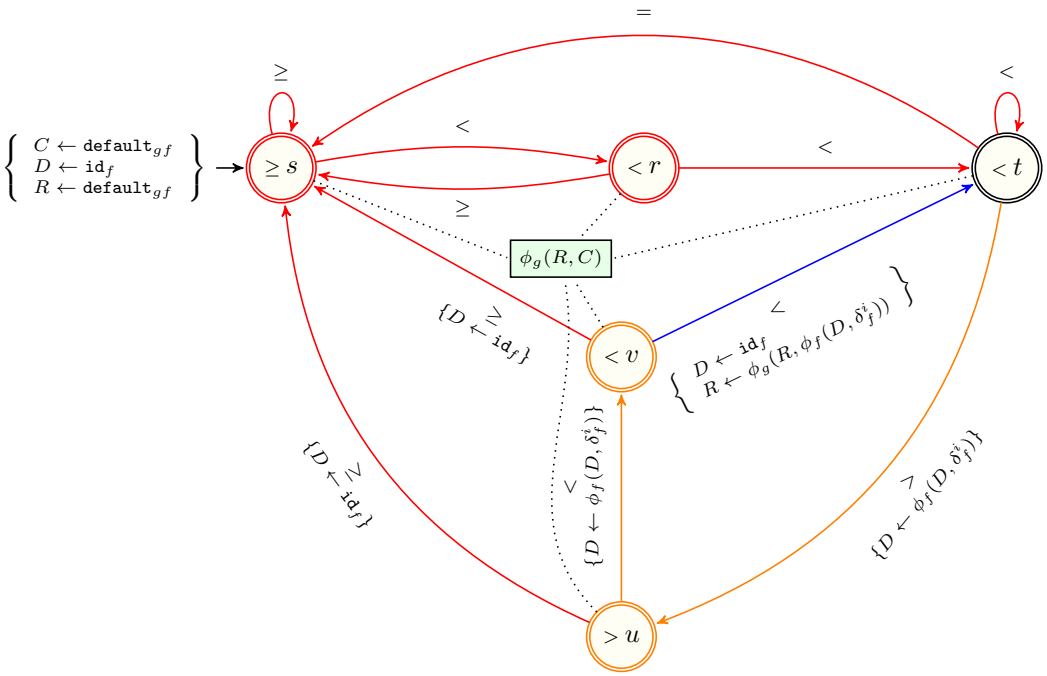


Figure 3.15: Parametrised automaton for any functional time-series constraints of the DIP_ON_INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

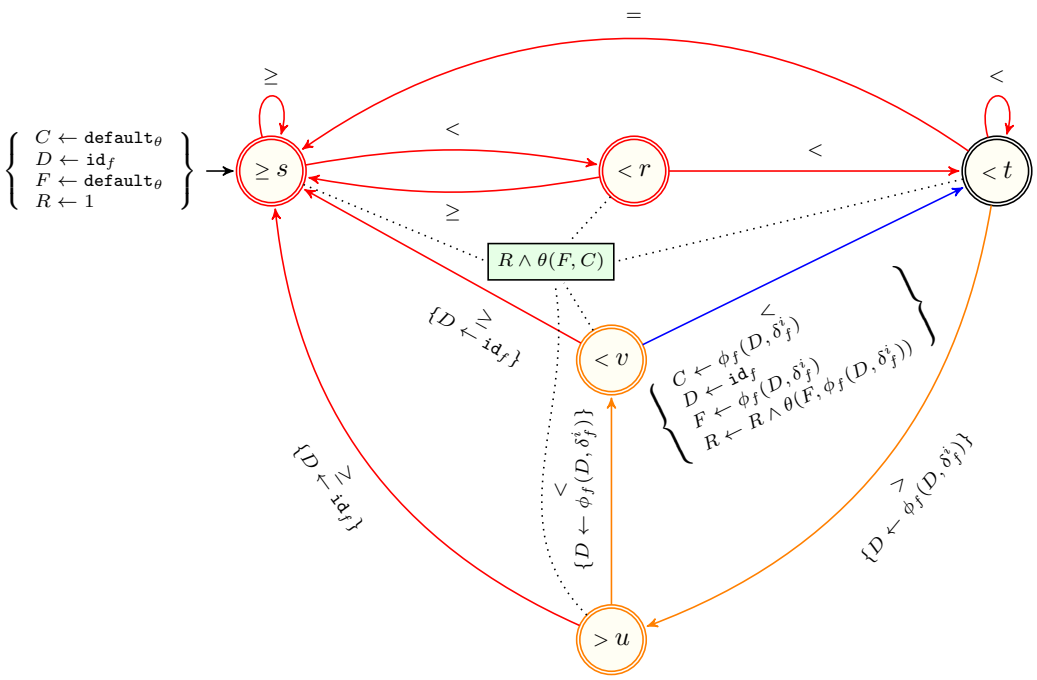
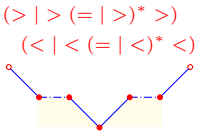


Figure 3.16: Parametrised automaton for any predicate time-series constraints of the DIP_ON_INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

3.1.6 GORGE



Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern GORGE is the *maximal* subsequence of S which matches the regular expression $\text{'}(> |(> (= | >)^* >))(< |(< (= | <)^* <))\text{'}$. Part (A) and parts (B,C) of Figure 3.17 respectively depict the seed transducer associated with the GORGE pattern as well as one example of its execution on a time series.

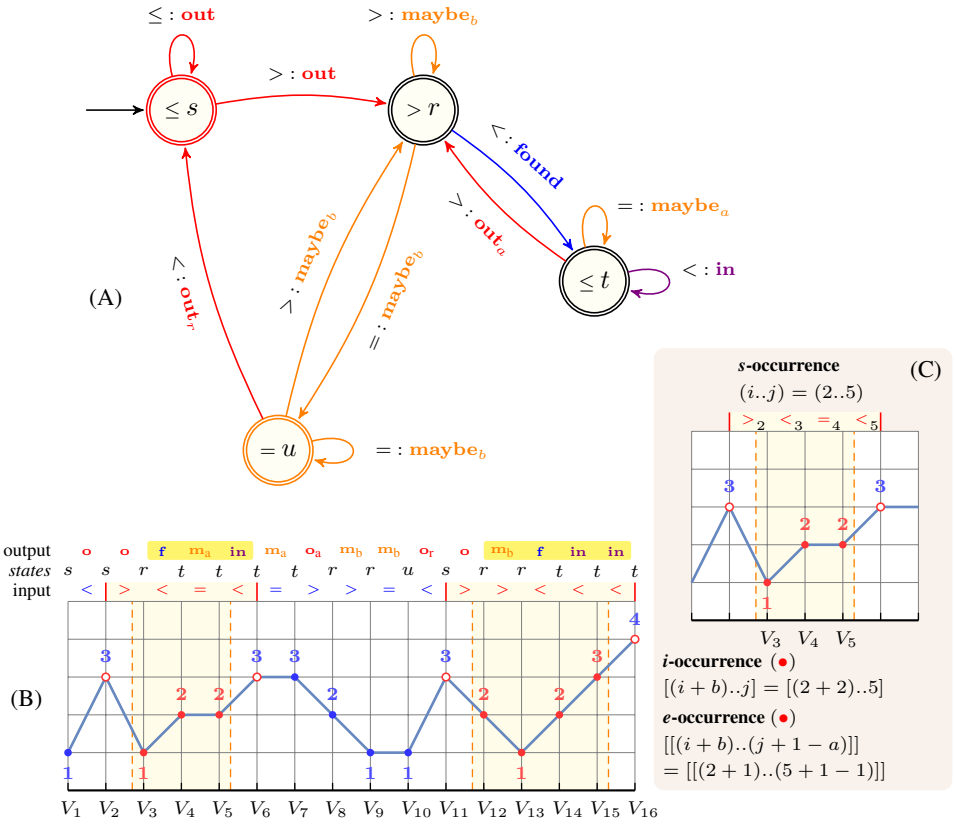


Figure 3.17: (A) Seed transducer for the GORGE pattern: $\text{'}(> |(> (= | >)^* >))(< |(< (= | <)^* <))\text{'}$ with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{m}_\text{a}$, $\mathbf{o}_\text{a}$, $\mathbf{m}_\text{b}$, $\mathbf{o}_\text{r}$ are shortcut for **out**, **found**, $\mathbf{maybe}_\text{a}$, $\mathbf{out}_\text{a}$, $\mathbf{maybe}_\text{b}$, $\mathbf{out}_\text{r}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

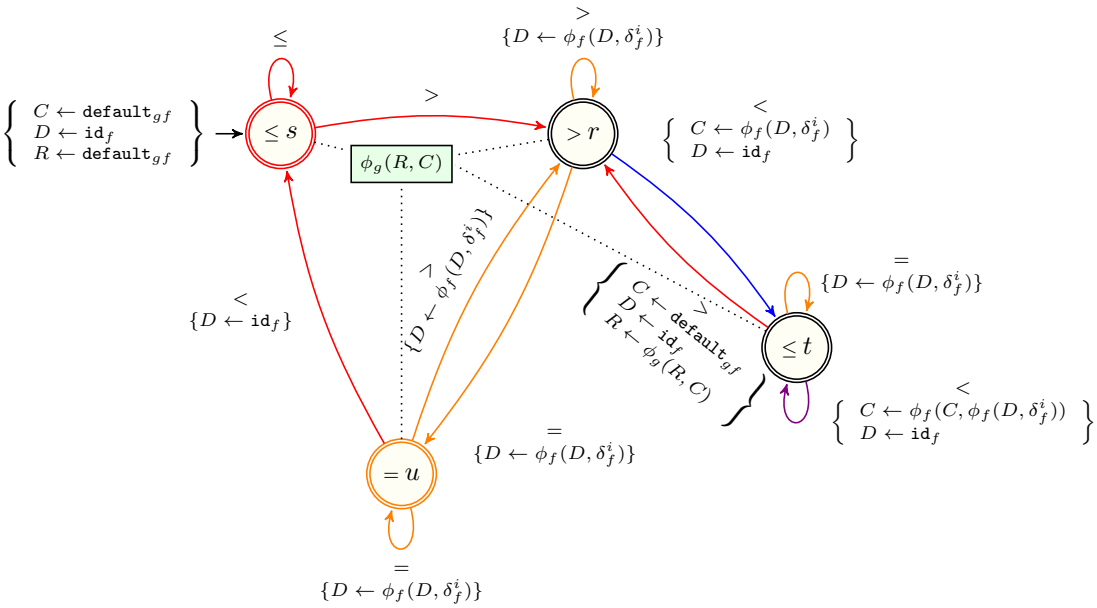


Figure 3.18: Parametrised automaton for any functional time-series constraints of the GORGE pattern obtained by applying the decoration table 3.37 to the corresponding transducer (transition $u \to r$ has the same register update as transition $r \to u$)

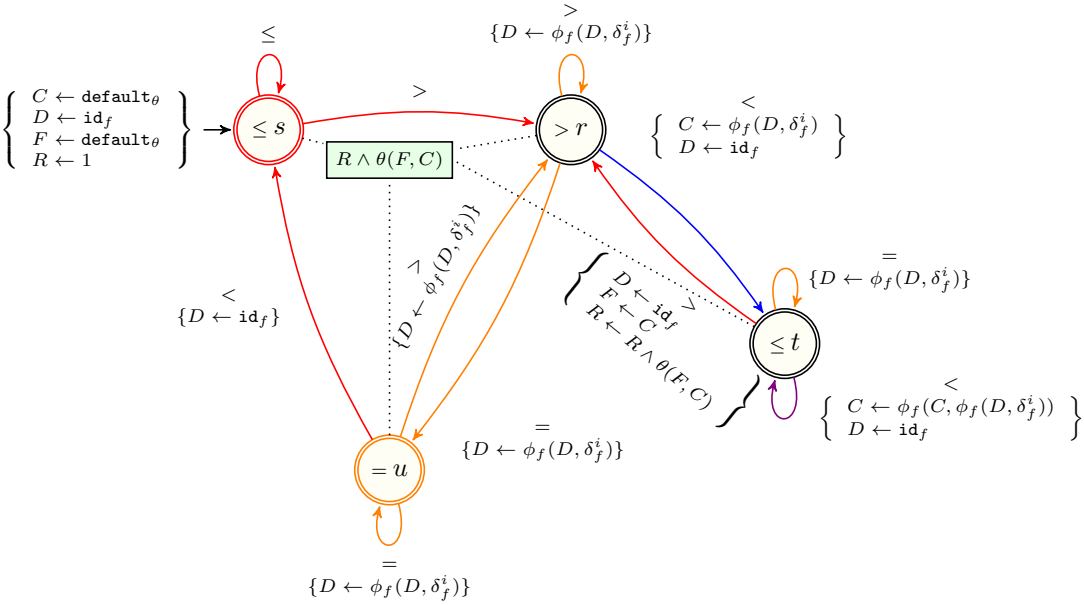


Figure 3.19: Parametrised automaton for any predicate time-series constraints of the GORGE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values; transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$.

	s	r	t	u
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L
u	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.7: Parametrised glue matrix for any $g_f_$GORGE constraint, where cell annotations have the following meaning: C stands for creating a new gorge occurrence from no existing gorge, R stands for extending to the right an existing gorge, L stands for extending to the left an existing gorge; the state associated with each row corresponds to a state of the GORGE transducer – see Figure 3.17, while the state associated with each column corresponds to a state of the reverse of the GORGE transducer, i.e. the GORGE transducer – see Figure 3.17.

3.1.7 INCREASING

Given a sequence S over the alphabet $\{\text{`<'}, \text{`=`}, \text{`>'}\}$, an occurrence of the pattern INCREASING is the subsequence of S which matches the regular expression `<'. Part (A) and parts (B,C) of Figure 3.20 respectively depict the seed transducer associated with the INCREASING pattern as well as one example of its execution on a time series.

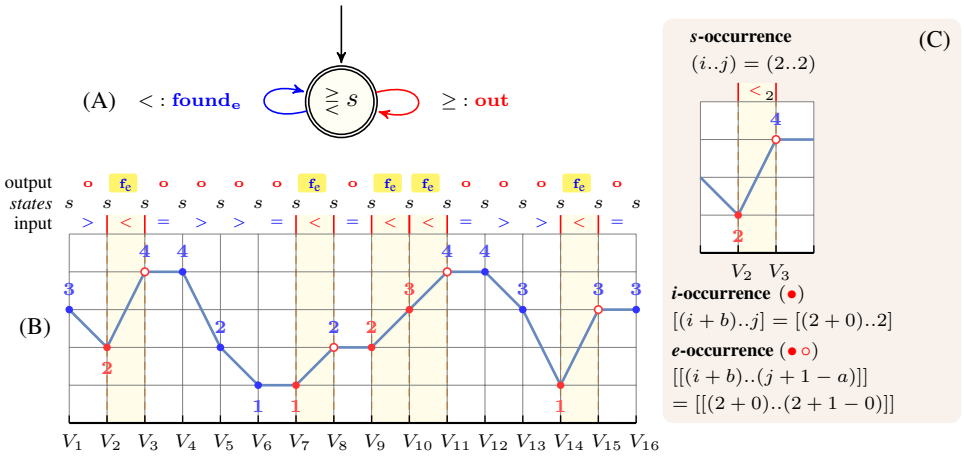


Figure 3.20: (A) Seed transducer for the INCREASING pattern: `<' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{f_e}$ are shortcut for $\mathbf{out}$, $\mathbf{found_e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

	s
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.8: Parametrised glue matrix for any $g_f_$INCREASING constraint; the state associated with each row corresponds to a state of the INCREASING transducer – see Figure 3.20, while the state associated with each column corresponds to a state of the reverse of the INCREASING transducer, i.e. the DECREASING transducer – see Figure 3.5.

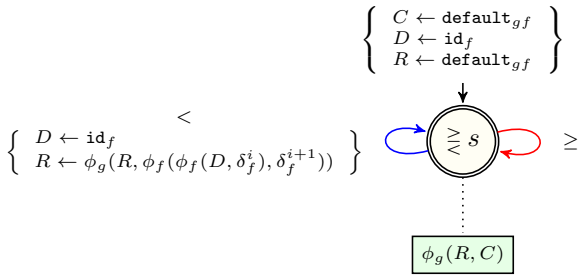


Figure 3.21: Parametrised automaton for any functional time-series constraints of the INCREASING pattern obtained by applying the decoration table 3.37 to the corresponding transducer

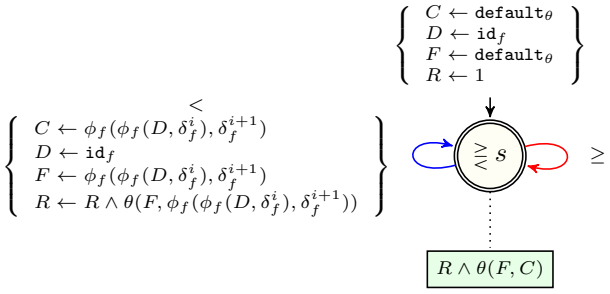
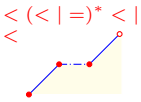


Figure 3.22: Parametrised automaton for any predicate time-series constraints of the INCREASING pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

3.1.8 INCREASING_SEQUENCE



Given a sequence S over the alphabet $\{\text{'}<\text{'}, \text{'}=\text{'}, \text{'}>\text{'}\}$, an occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence of S which matches the regular expression '$< (< \mid =)^* < \mid <$'. Part (A) and parts (B,C) of Figure 3.23 respectively depict the seed transducer associated with the INCREASING_SEQUENCE pattern as well as one example of its execution on a time series.

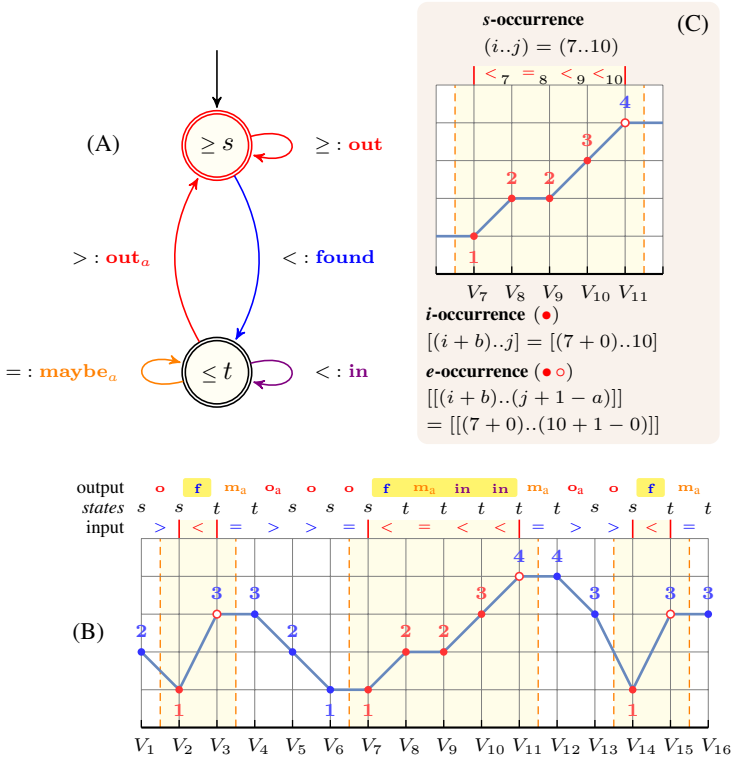


Figure 3.23: (A) Seed transducer for the INCREASING_SEQUENCE pattern: '$< (< \mid =)^* < \mid <$' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{m}_a$, $\mathbf{o}_a$ are shortcut for **out**, **found**, $\mathbf{maybe}_a$, $\mathbf{out}_a$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

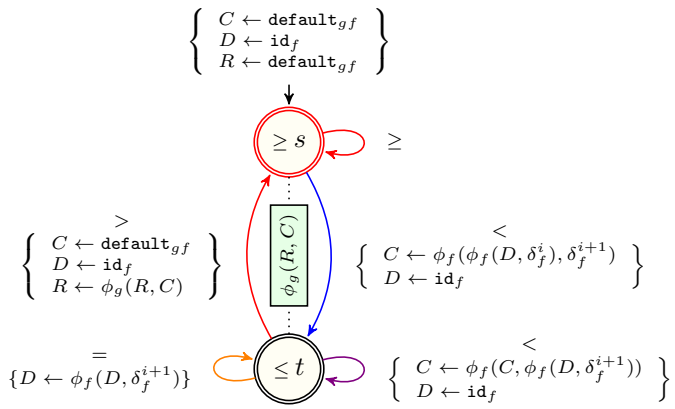


Figure 3.24: Parametrised automaton for any functional time-series constraints of the INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

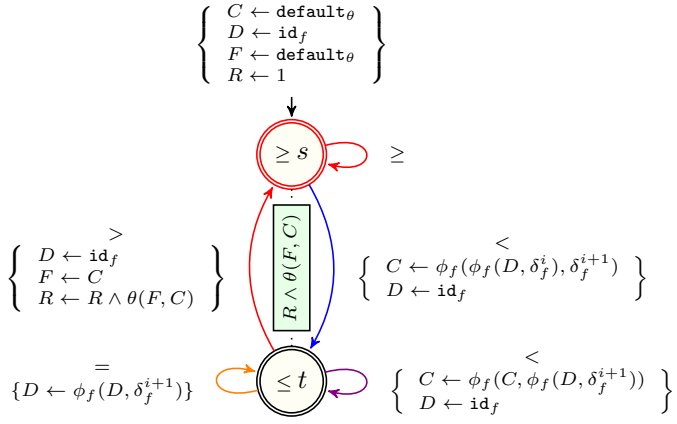
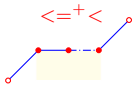


Figure 3.25: Parametrised automaton for any predicate time-series constraints of the INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \hat{\delta}_f^{i+1})$ M

Table 3.9: Parametrised glue matrix for any $g_f_$INCREASING_SEQUENCE constraint, where the cell annotation M stands for merging two existing increasing sequences sharing an element; the state associated with each row corresponds to a state of the INCREASING_SEQUENCE transducer – see Figure 3.23, while the state associated with each column corresponds to a state of the reverse of the INCREASING_SEQUENCE transducer, i.e. the DECREASING_SEQUENCE transducer – see Figure 3.8.

3.1.9 INCREASING_TERRACE



Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence of S which matches the regular expression '$<=^+<$'. Part (A) and parts (B,C) of Figure 3.26 respectively depict the seed transducer associated with the INCREASING_TERRACE pattern as well as one example of its execution on a time series.

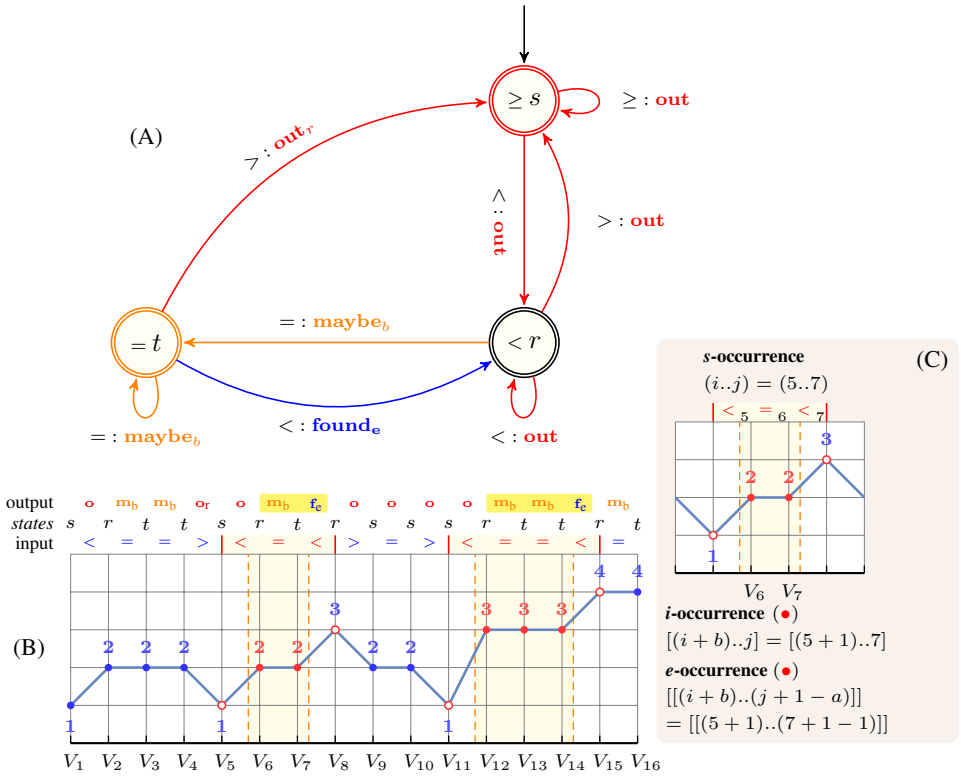


Figure 3.26: (A) Seed transducer for the INCREASING_TERRACE pattern: '$<=^+<$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{m_b}$, $\mathbf{o_r}$, $\mathbf{f_e}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe_b}$, $\mathbf{out_r}$, $\mathbf{found_e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

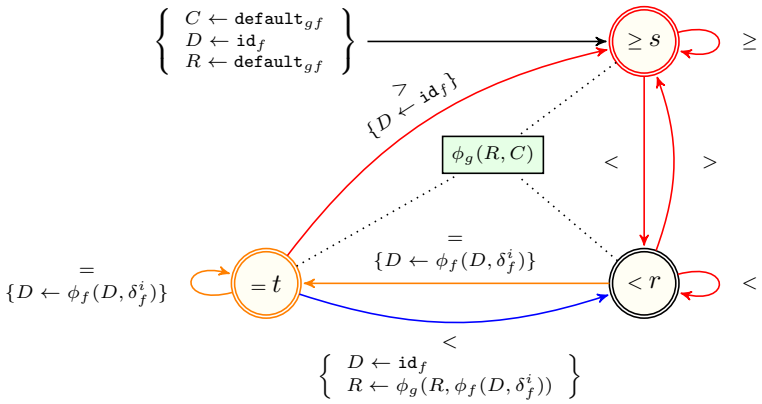


Figure 3.27: Parametrised automaton for any functional time-series constraints of the INCREASING_TERRACE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

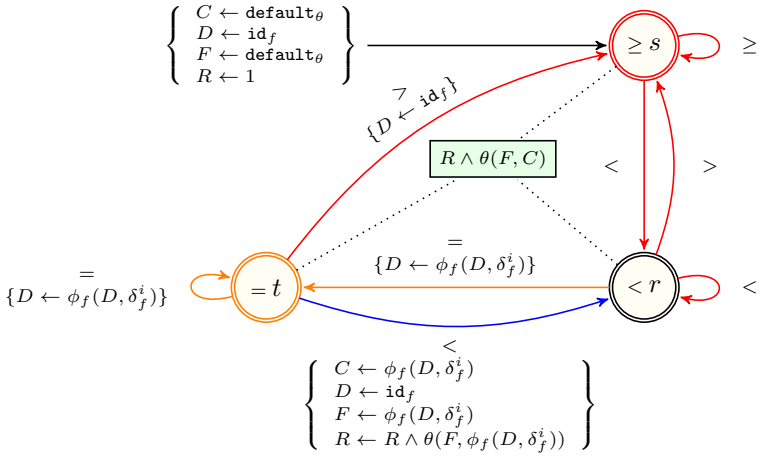
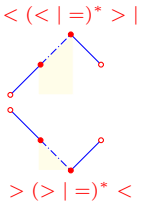


Figure 3.28: Parametrised automaton for any predicate time-series constraints of the INCREASING_TERRACE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.10: Parametrised glue matrix for any g_f_INCREASING_TERRACE constraint, where the cell annotation C stands for creating a new increasing terrace occurrence from no existing increasing terrace; the state associated with each row corresponds to a state of the INCREASING_TERRACE transducer – see Figure 3.26, while the state associated with each column corresponds to a state of the reverse of the INCREASING_TERRACE transducer, i.e. the DECREASING_TERRACE transducer – see Figure 3.11.

3.1.10 INFLEXION



Given a sequence S over the alphabet {'<', '=', '>'}, an occurrence of the pattern INFLEXION is the *maximal* subsequence of S which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'. Part (A) and parts (B,C) of Figure 3.29 respectively depict the seed transducer associated with the INFLEXION pattern as well as one example of its execution on a time series.

The pattern INFLEXION is not its own reverse as illustrated by the following example: when scanned from left to right, the regular expression '<<<>>>' contains the inflexion '<<<>', which does not coincide with the inflexion '<>>' obtained by performing a scan from right to left. As a consequence we do not have any glue matrix for the INFLEXION pattern.

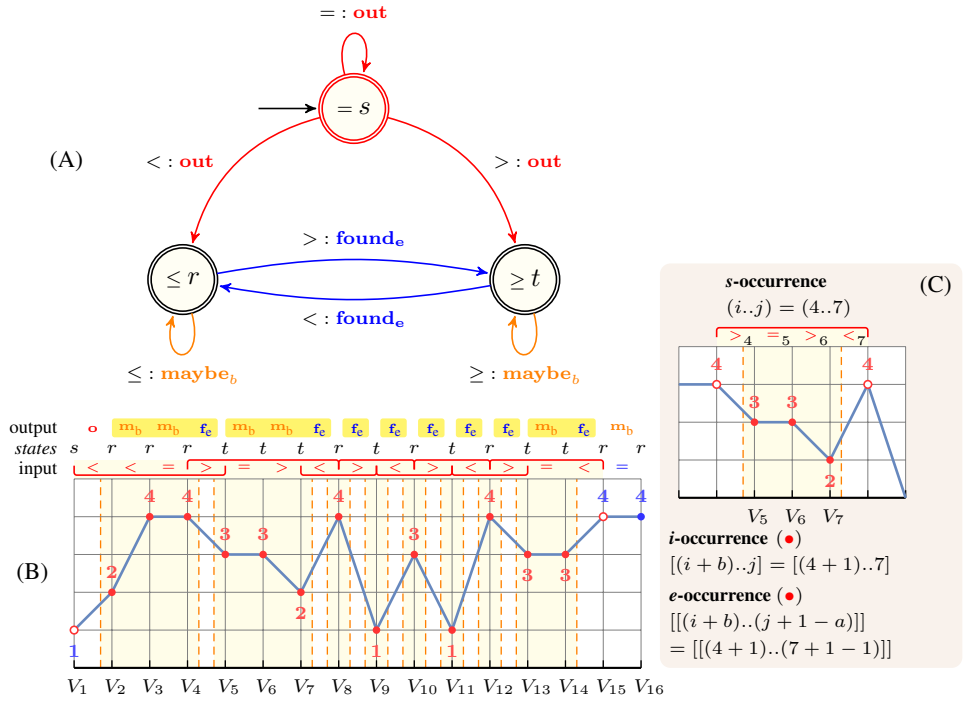


Figure 3.29: (A) Seed transducer for the INFLEXION pattern: '$< (< \mid =)^* > \mid > (> \mid =)^* <$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{f}_\mathrm{e}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{found}_\mathrm{e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

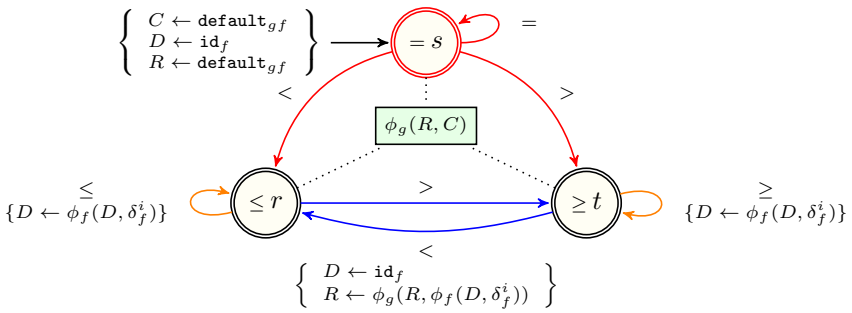


Figure 3.30: Parametrised automaton for any functional time-series constraints of the INFLEXION pattern obtained by applying the decoration table 3.37 to the corresponding transducer (transition $r \to t$ has the same register updates as transition $t \to r$)

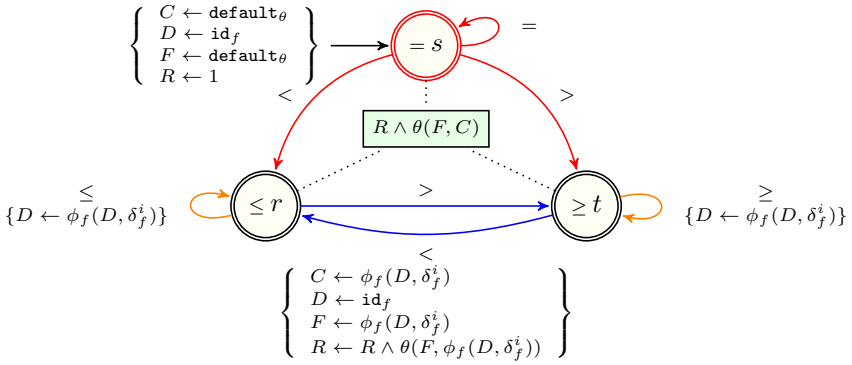
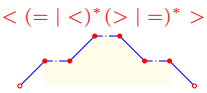


Figure 3.31: Parametrised automaton for any predicate time-series constraints of the INFLEXION pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values; transition $r \to t$ has the same register updates as transition $t \to r$.

3.1.11 PEAK



Given a sequence S over the alphabet $\{\text{`}<\text{'}, \text{`}=\text{'}, \text{`}>\text{'}\}$, an occurrence of the pattern PEAK is the *maximal* subsequence of $S$ which matches the regular expression `$< (= \mid <)^* (> \mid =)^* >$'. Part (A) and parts (B,C) of Figure 3.32 respectively depict the seed transducer associated with the PEAK pattern as well as one example of its execution on a time series.

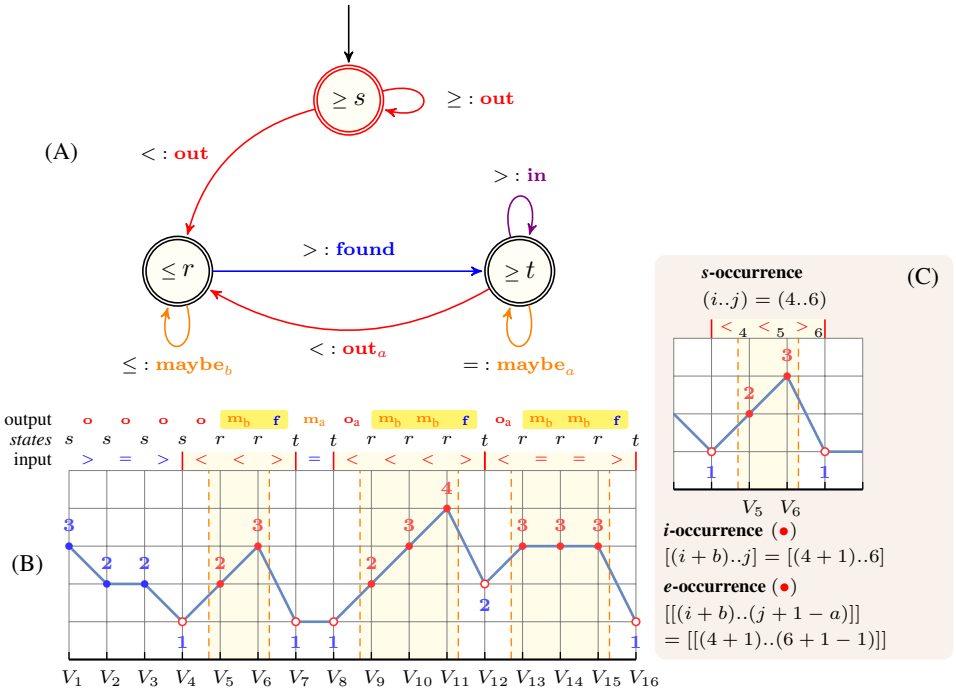


Figure 3.32: (A) Seed transducer for the PEAK pattern: `$< (= \mid <)^* (> \mid =)^* >$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, $\mathbf{m}_\mathrm{b}$, **f**, $\mathbf{m}_\mathrm{a}$, $\mathbf{o}_\mathrm{a}$ are shortcut for **out**, $\mathbf{maybe}_\mathrm{b}$, **found**, $\mathbf{maybe}_\mathrm{a}$, $\mathbf{out}_\mathrm{a}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

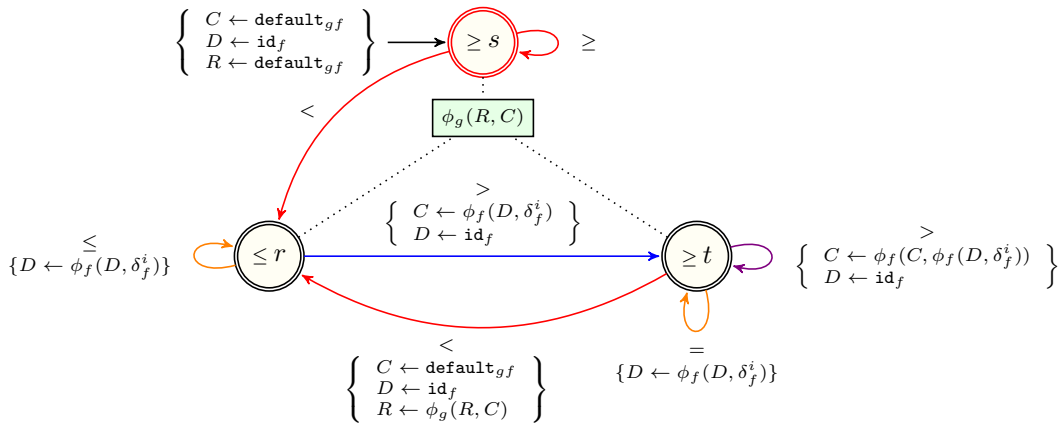


Figure 3.33: Parametrised automaton for any functional time-series constraints of the PEAK pattern obtained by applying the decoration table 3.37 to the corresponding transducer

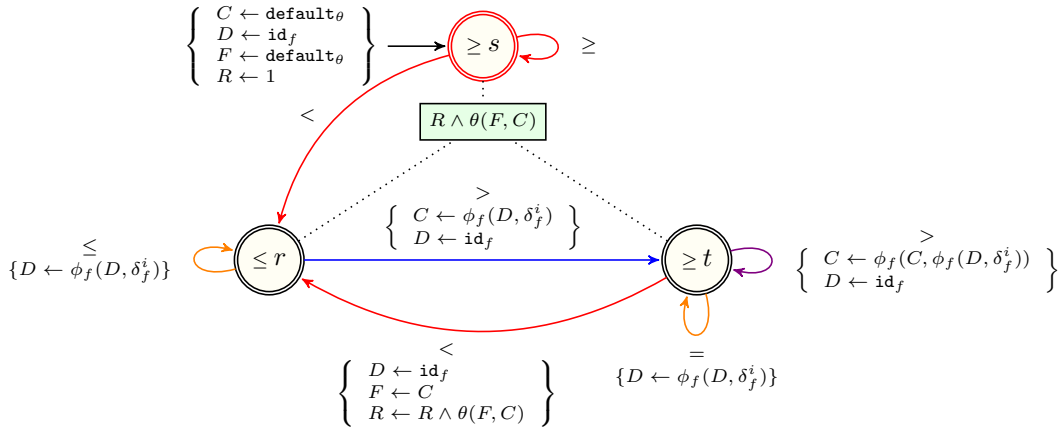


Figure 3.34: Parametrised automaton for any predicate time-series constraints of the PEAK pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.11: Parametrised glue matrix for any g_f_PEAK constraint, where cell annotations have the following meaning: C stands for creating a new peak occurrence from no existing peak, R stands for extending to the right an existing peak, L stands for extending to the left an existing peak; the state associated with each row corresponds to a state of the PEAK transducer – see Figure 3.32, while the state associated with each column corresponds to a state of the reverse of the PEAK transducer, i.e. the PEAK transducer – see Figure 3.32.

3.1.12 PLAIN

Given a sequence S over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, an occurrence of the pattern PLAIN is the *maximal* subsequence of $S$ which matches the regular expression `>=*<'. Part (A) and parts (B,C) of Figure 3.35 respectively depict the seed transducer associated with the PLAIN pattern as well as one example of its execution on a time series.

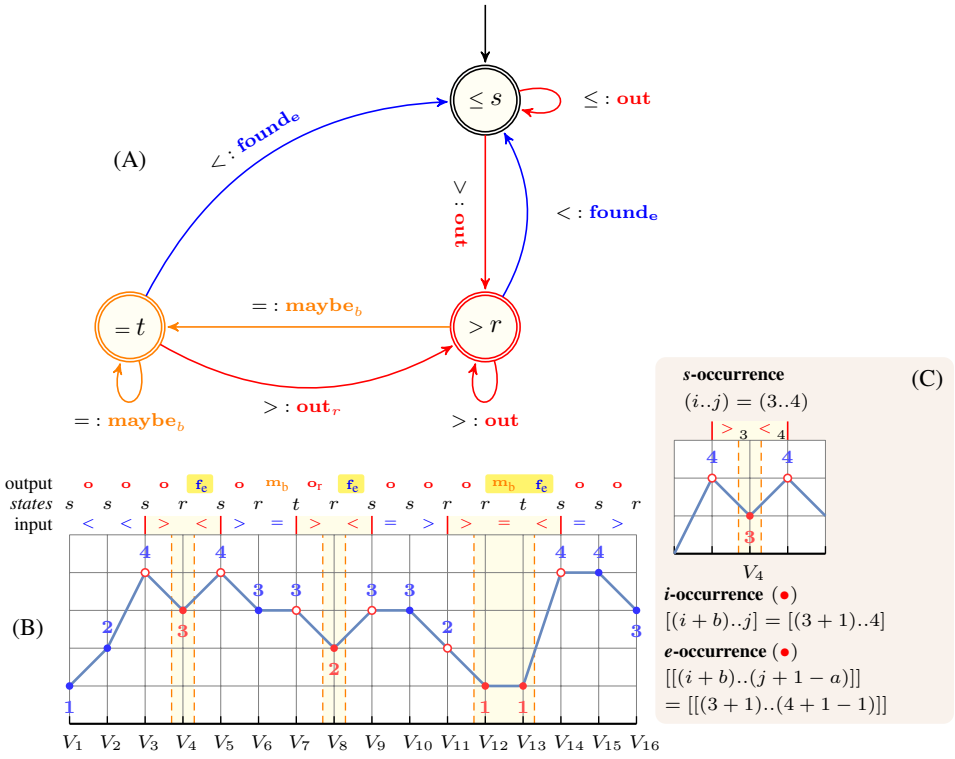


Figure 3.35: (A) Seed transducer for the PLAIN pattern: `>=*<' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{f}_e$, $\mathbf{m}_b$, $\mathbf{o}_r$ are shortcut for $\mathbf{out}$, $\mathbf{found}_e$, $\mathbf{maybe}_b$, $\mathbf{out}_r$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

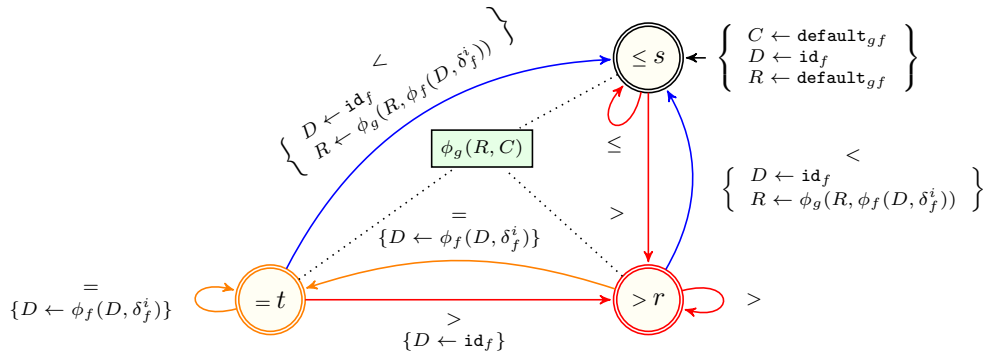


Figure 3.36: Parametrised automaton for any functional time-series constraints of the PLAIN pattern obtained by applying the decoration table 3.37 to the corresponding transducer

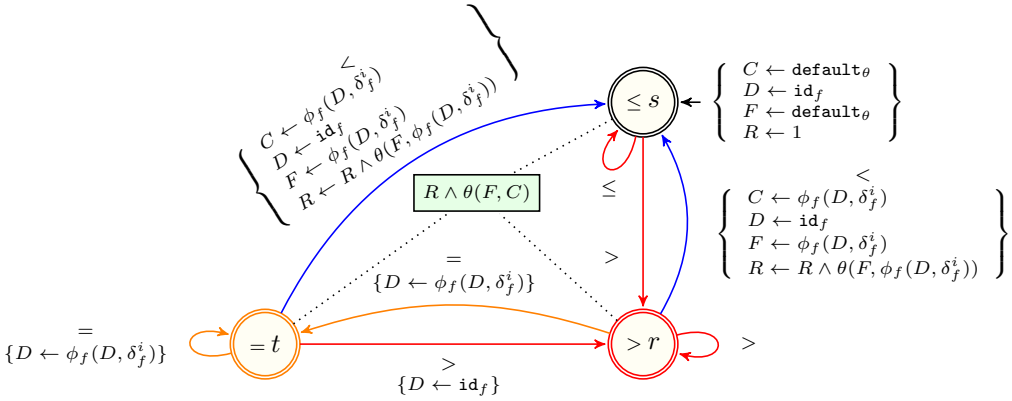


Figure 3.37: Parametrised automaton for any predicate time-series constraints of the PLAIN pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.12: Parametrised glue matrix for any $g_f_$PLAIN constraint, where the cell annotation C stands for creating a new plain occurrence from no existing plain; the state associated with each row corresponds to a state of the PLAIN transducer – see Figure 3.35, while the state associated with each column corresponds to a state of the reverse of the PLAIN transducer, i.e. the PLAIN transducer – see Figure 3.35.

3.1.13 PLATEAU

Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern PLATEAU is the *maximal* subsequence of S which matches the regular expression '$<=^*>$'. Part (A) and parts (B,C) of Figure 3.38 respectively depict the seed transducer associated with the PLATEAU pattern as well as one example of its execution on a time series.

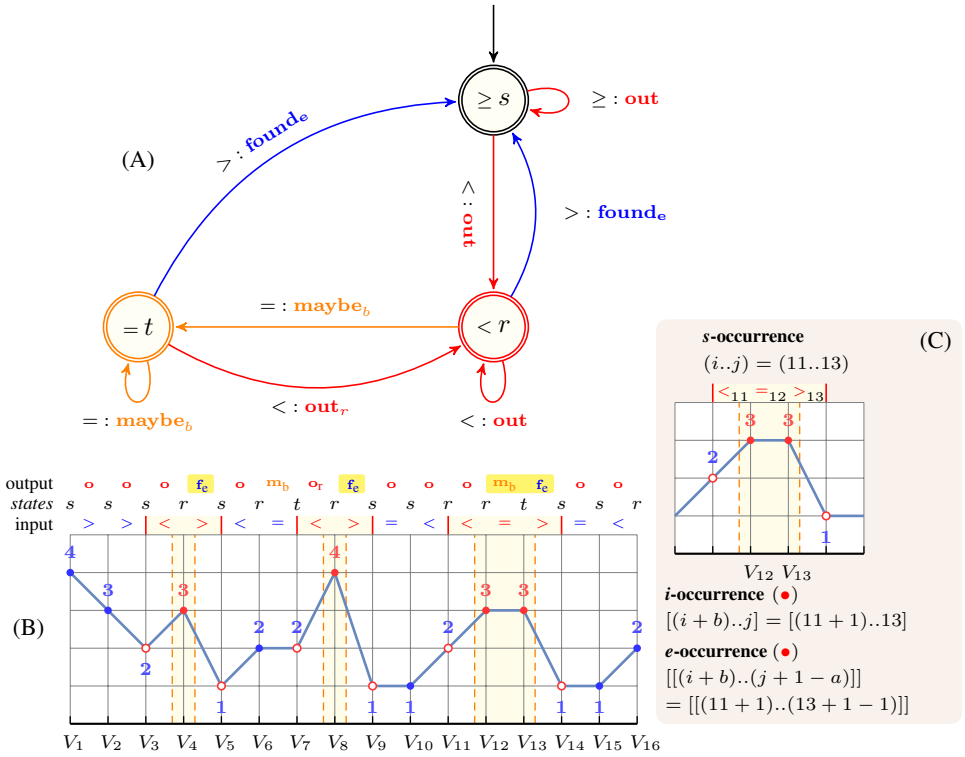


Figure 3.38: (A) Seed transducer for the PLATEAU pattern: '$<=^*>$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{f}_e$, $\mathbf{m}_b$, $\mathbf{o}_r$ are shortcut for $\mathbf{out}$, $\mathbf{found}_e$, $\mathbf{maybe}_b$, $\mathbf{out}_r$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

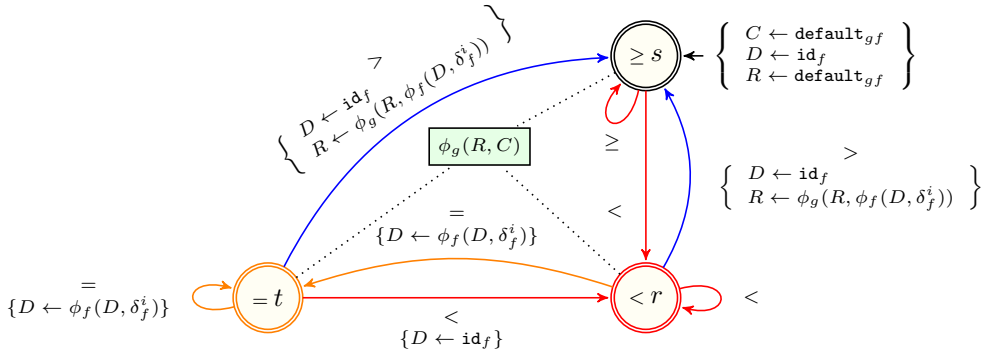


Figure 3.39: Parametrised automaton for any functional time-series constraints of the PLATEAU pattern obtained by applying the decoration table 3.37 to the corresponding transducer

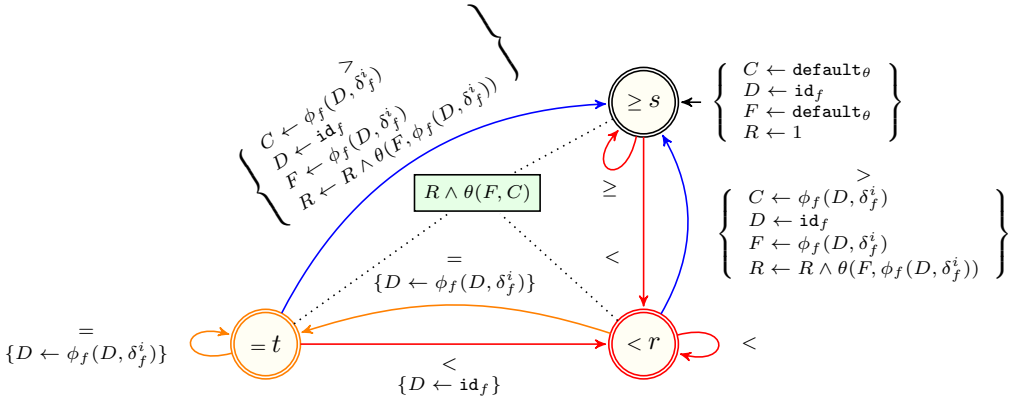
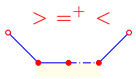


Figure 3.40: Parametrised automaton for any predicate time-series constraints of the PLATEAU pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.13: Parametrised glue matrix for any $g_f_$PLATEAU constraint, where the cell annotation C stands for creating a new plateau occurrence from no existing plateau; the state associated with each row corresponds to a state of the PLATEAU transducer – see Figure 3.38, while the state associated with each column corresponds to a state of the reverse of the PLATEAU transducer, i.e. the PLATEAU transducer – see Figure 3.38.

3.1.14 PROPER_PLAIN



Given a sequence S over the alphabet $\{\text{`<'}, \text{`='}, \text{`>'}\}$, an occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence of $S$ which matches the regular expression `>=+<`. Part (A) and parts (B,C) of Figure 3.41 respectively depict the seed transducer associated with the PROPER_PLAIN pattern as well as one example of its execution on a time series.

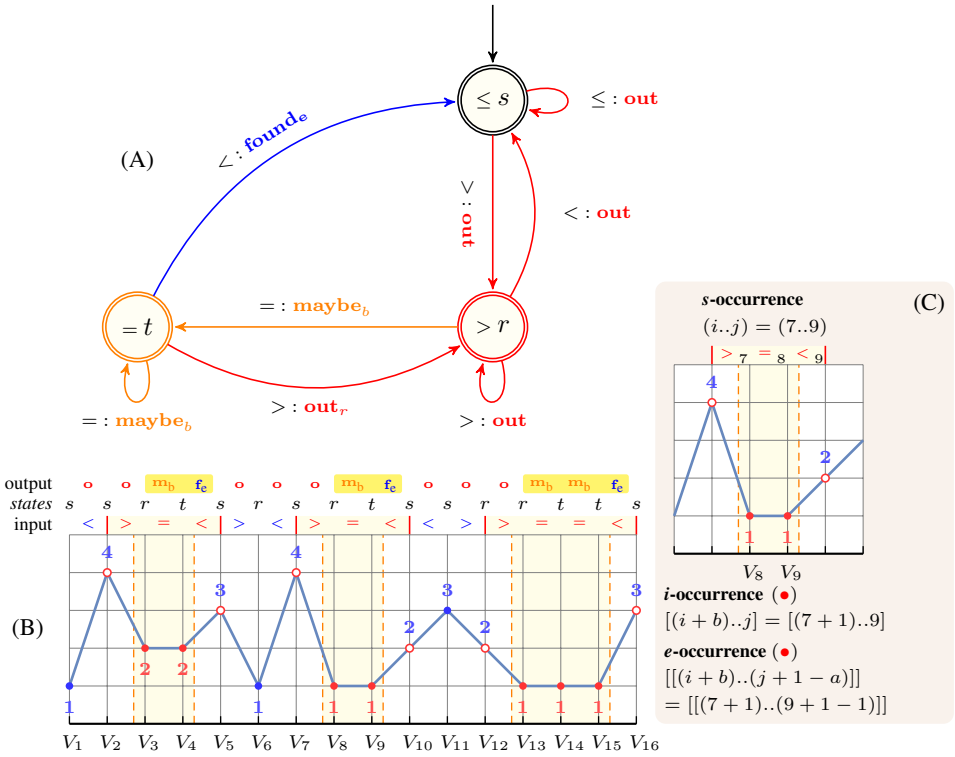


Figure 3.41: (A) Seed transducer for the PROPER_PLAIN pattern: `>=+<` with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, $\mathbf{m}_\text{b}$, $\mathbf{f}_\text{e}$ are shortcut for **out**, $\mathbf{maybe}_\text{b}$, $\mathbf{found}_\text{e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

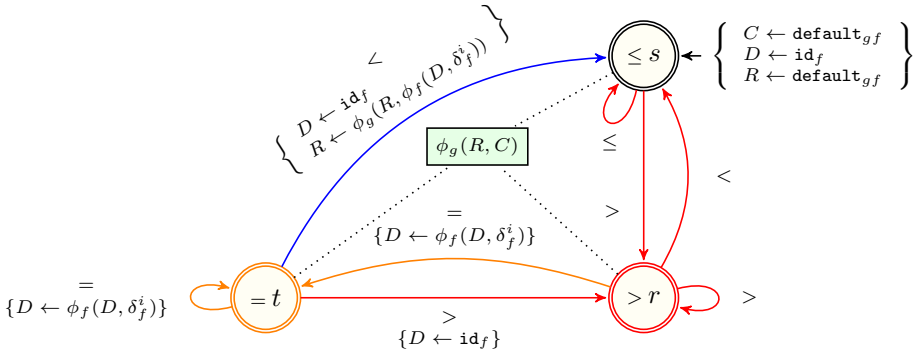


Figure 3.42: Parametrised automaton for any functional time-series constraints of the PROPER_PLAIN pattern obtained by applying the decoration table 3.37 to the corresponding transducer

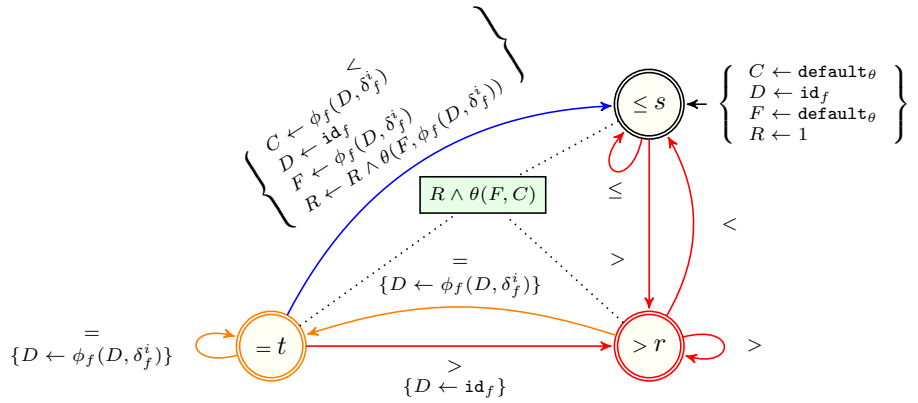


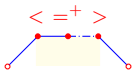
Figure 3.43: Parametrised automaton for any predicate time-series constraints of the PROPER_PLAIN pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.14: Parametrised glue matrix for any g_f_PROPER_PLAIN constraint, where the cell annotation C stands for creating a new proper plain occurrence from no existing proper plain; the state associated with each row corresponds to a state of the PROPER_PLAIN transducer – see Figure 3.41, while the state associated with each column corresponds to a state of the reverse of the PROPER_PLAIN transducer, i.e. the PROPER_PLAIN transducer – see Figure 3.41.

3.1.15 PROPER_PLATEAU

Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence of S which matches the regular expression '$<=^{+}>$'. Part (A) and parts (B,C) of Figure 3.44 respectively depict the seed transducer associated with the PROPER_PLATEAU pattern as well as one example of its execution on a time series.



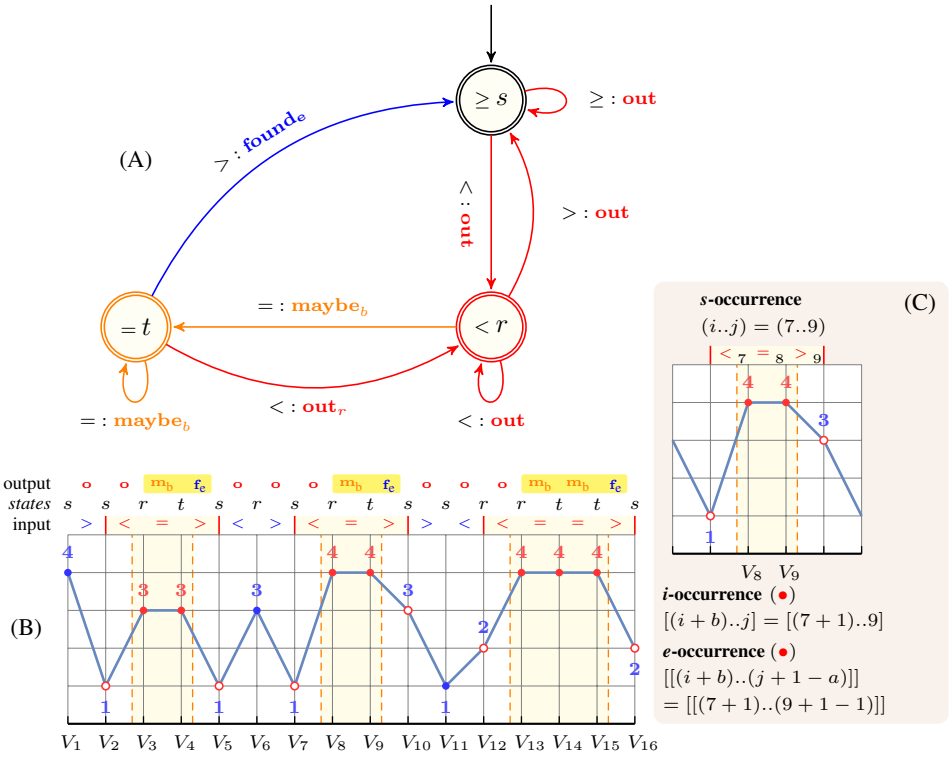


Figure 3.44: (A) Seed transducer for the PROPER_PLATEAU pattern: '$<=^{+}>$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{f}_\mathrm{e}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{found}_\mathrm{e}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

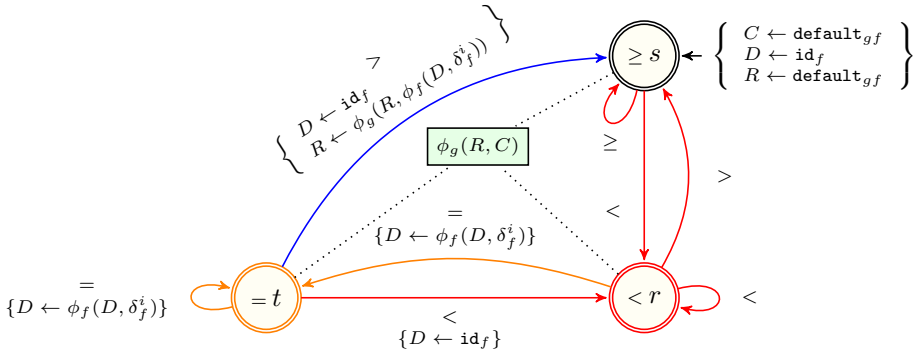


Figure 3.45: Parametrised automaton for any functional time-series constraints of the PROPER_PLATEAU pattern obtained by applying the decoration table 3.37 to the corresponding transducer

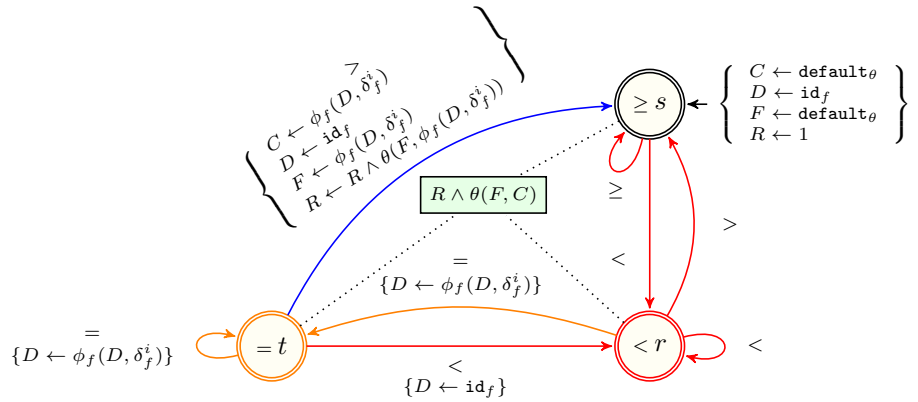


Figure 3.46: Parametrised automaton for any predicate time-series constraints of the PROPER_PLATEAU pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C

Table 3.15: Parametrised glue matrix for any g_f_PROPER_PLATEAU constraint, where the cell annotation C stands for creating a new proper plateau occurrence from no existing proper plateau; the state associated with each row corresponds to a state of the PROPER_PLATEAU transducer – see Figure 3.44, while the state associated with each column corresponds to a state of the reverse of the PROPER_PLATEAU transducer, i.e. the PROPER_PLATEAU transducer – see Figure 3.44.

3.1.16 STEADY

Given a sequence S over the alphabet $\{\text{`}<\text{'}, \text{`}=\text{'}, \text{`}>\text{'}\}$, an occurrence of the pattern STEADY is the subsequence of $S$ which matches the regular expression `=`. Part (A) and parts (B,C) of Figure 3.47 respectively depict the seed transducer associated with the STEADY pattern as well as one example of its execution on a time series.

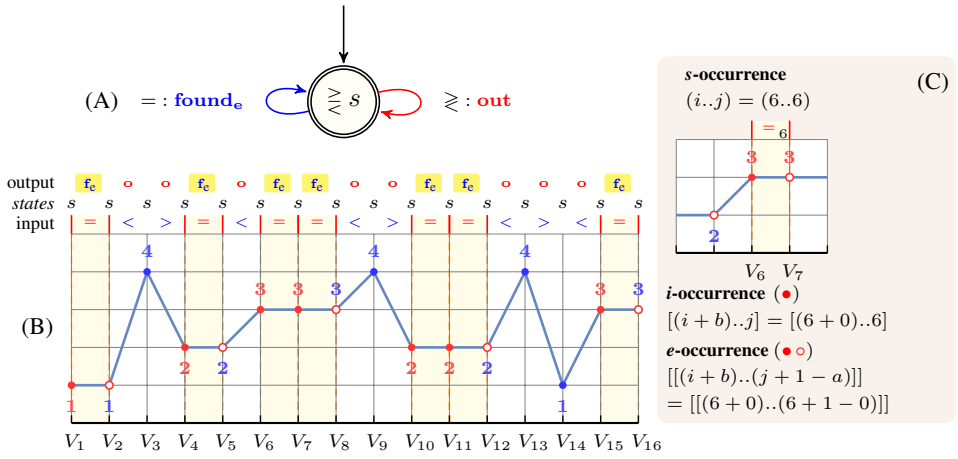


Figure 3.47: (A) Seed transducer for the STEADY pattern: `=` with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{f_e}$, $\mathbf{o}$ are shortcut for $\mathbf{found_e}$, $\mathbf{out}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

	s
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.16: Parametrised glue matrix for any $g_f_$STEADY constraint; the state associated with each row corresponds to a state of the STEADY transducer – see Figure 3.47, while the state associated with each column corresponds to a state of the reverse of the STEADY transducer, i.e. the STEADY transducer – see Figure 3.47.

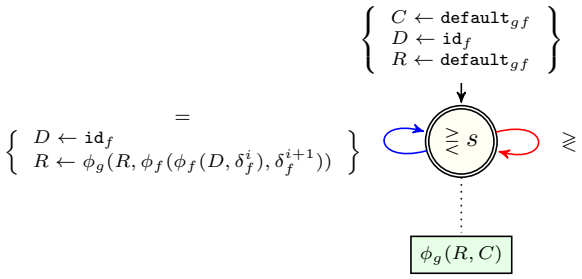


Figure 3.48: Parametrised automaton for any functional time-series constraints of the STEADY pattern obtained by applying the decoration table 3.37 to the corresponding transducer

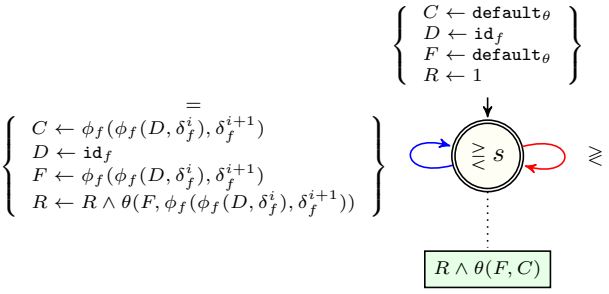
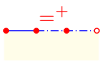


Figure 3.49: Parametrised automaton for any predicate time-series constraints of the STEADY pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

3.1.17 STEADY_SEQUENCE



Given a sequence S over the alphabet $\{\text{'}<\text{'}, \text{'}=\text{'}, \text{'}>\text{'}\}$, an occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence of S which matches the regular expression '$=^+$'. Part (A) and parts (B,C) of Figure 3.50 respectively depict the seed transducer associated with the STEADY_SEQUENCE pattern as well as one example of its execution on a time series.

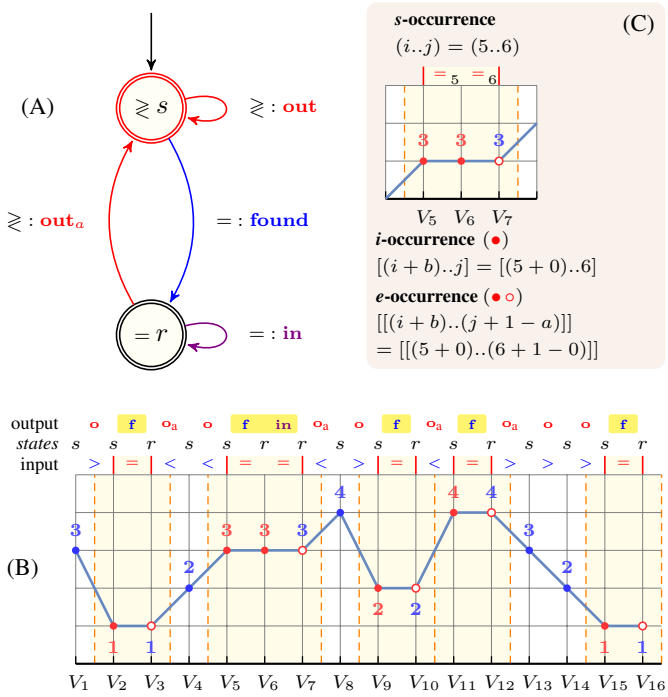


Figure 3.50: (A) Seed transducer for the STEADY_SEQUENCE pattern: '$=^+$' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{o}_a$ are shortcut for **out**, **found**, $\mathbf{out}_a$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

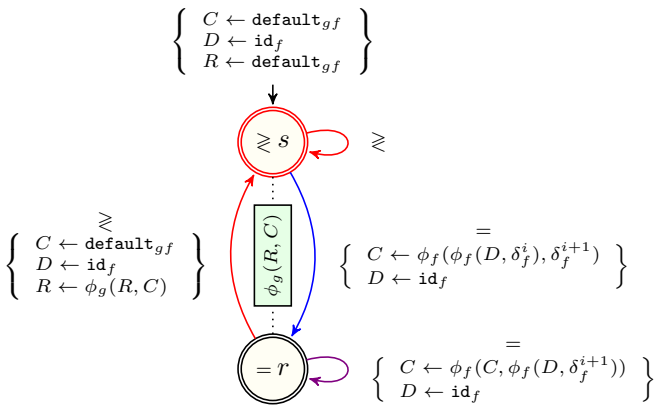


Figure 3.51: Parametrised automaton for any functional time-series constraints of the STEADY_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

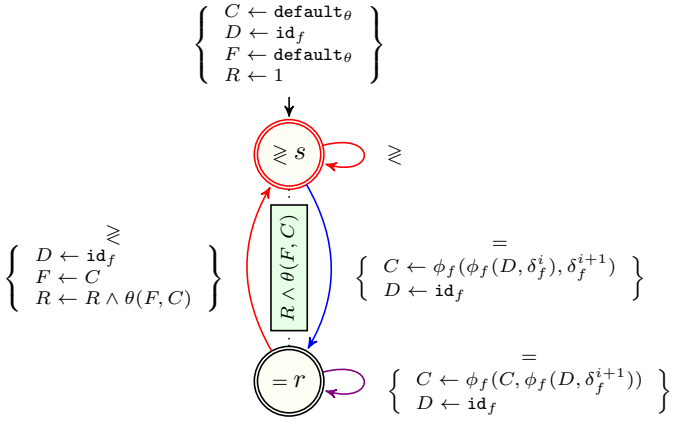


Figure 3.52: Parametrised automaton for any predicate time-series constraints of the STEADY_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \hat{\delta}_f^{i+1})$ M

Table 3.17: Parametrised glue matrix for any g_f_STEADY_SEQUENCE constraint, where the cell annotation M stands for merging two existing steady sequences sharing an element; the state associated with each row corresponds to a state of the STEADY_SEQUENCE transducer – see Figure 3.50, while the state associated with each column corresponds to a state of the reverse of the STEADY_SEQUENCE transducer, i.e. the STEADY_SEQUENCE transducer – see Figure 3.50.

3.1.18 STRICTLY_DECREASING_SEQUENCE

Given a sequence S over the alphabet $\{\text{'}<\text{'}, \text{'}=\text{'}, \text{'}>\text{'}\}$, an occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence of S which matches the regular expression '$>^+$'. Part (A) and parts (B,C) of Figure 3.53 respectively depict the seed transducer associated with the STRICTLY_DECREASING_SEQUENCE pattern as well as one example of its execution on a time series.

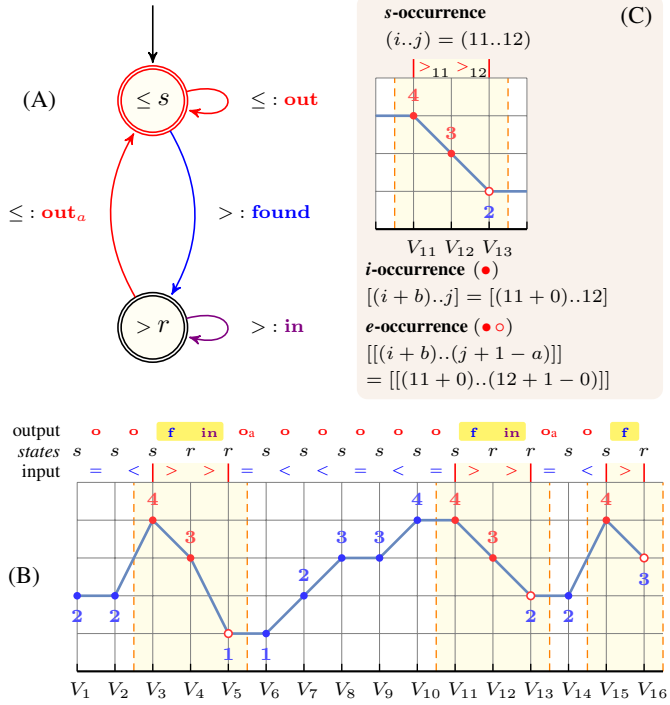


Figure 3.53: (A) Seed transducer for the STRICTLY_DECREASING_SEQUENCE pattern: '$>^+$' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{o}_a$ are shortcut for **out**, **found**, $\mathbf{out}_a$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

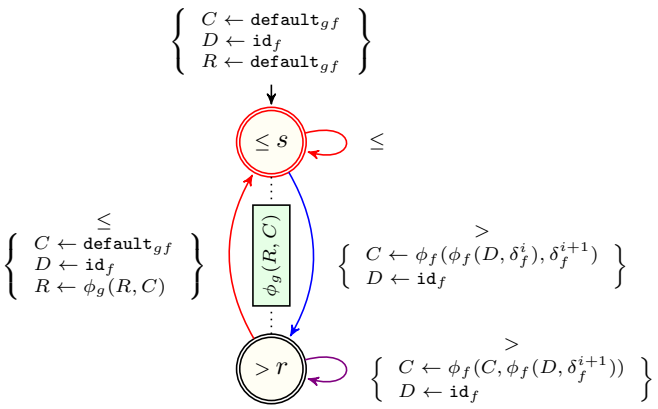


Figure 3.54: Parametrised automaton for any functional time-series constraints of the STRICTLY_DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

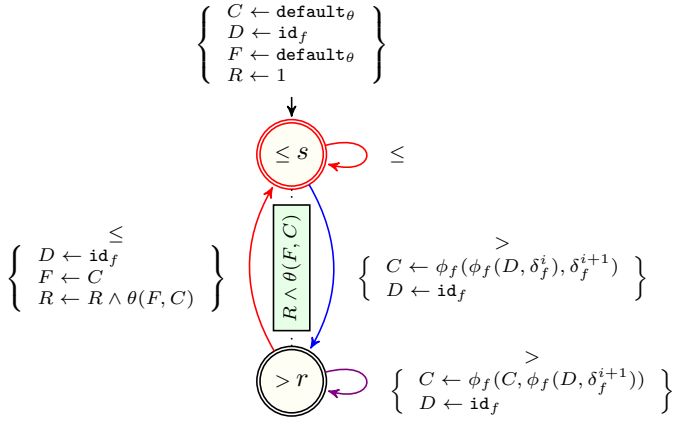


Figure 3.55: Parametrised automaton for any predicate time-series constraints of the STRICTLY_DECREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \hat{\delta}_f^{i+1})$ M

Table 3.18: Parametrised glue matrix for any g_f_STRICTLY_DECREASING_SEQUENCE constraint, where the cell annotation M stands for merging two existing strictly decreasing sequences sharing an element; the state associated with each row corresponds to a state of the STRICTLY_DECREASING_SEQUENCE transducer – see Figure 3.53, while the state associated with each column corresponds to a state of the reverse of the STRICTLY_DECREASING_SEQUENCE transducer, i.e. the STRICTLY_INCREASING_SEQUENCE transducer – see Figure 3.56.

3.1.19 STRICTLY_INCREASING_SEQUENCE

Given a sequence S over the alphabet $\{\text{`}<\text{'}, \text{`}=\text{'}, \text{`}>\text{'}\}$, an occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence of $S$ which matches the regular expression `$<^+$'. Part (A) and parts (B,C) of Figure 3.56 respectively depict the seed transducer associated with the STRICTLY_INCREASING_SEQUENCE pattern as well as one example of its execution on a time series.

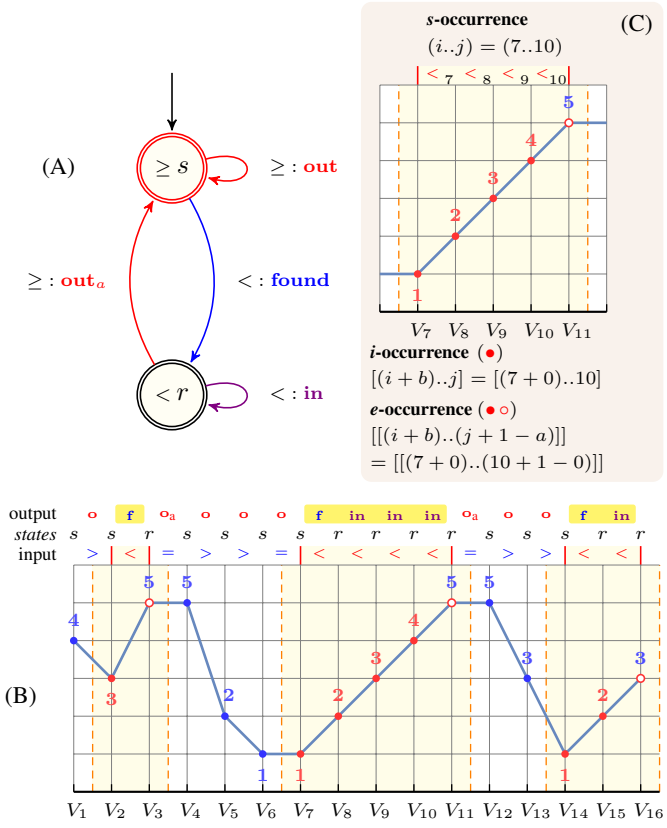


Figure 3.56: (A) Seed transducer for the STRICTLY_INCREASING_SEQUENCE pattern: `$<^+$' with $b = 0$ and $a = 0$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{o}_a$ are shortcut for **out**, **found**, $\mathbf{out}_a$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

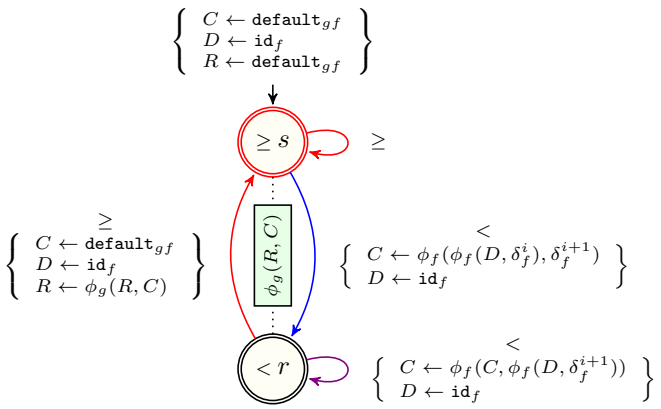


Figure 3.57: Parametrised automaton for any functional time-series constraints of the STRICTLY_INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.37 to the corresponding transducer

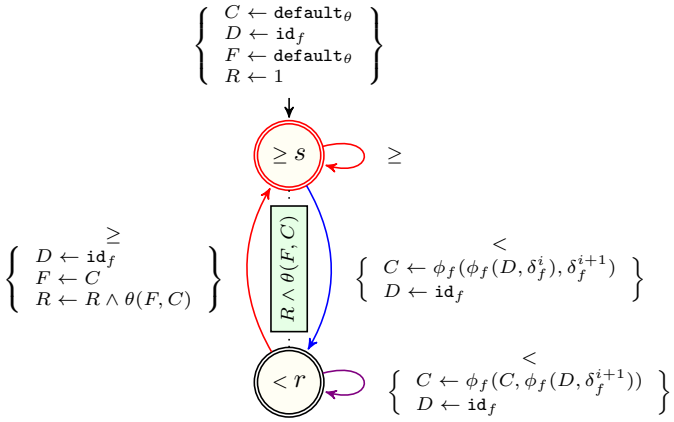
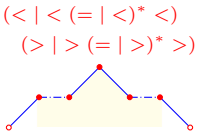


Figure 3.58: Parametrised automaton for any predicate time-series constraints of the STRICTLY_INCREASING_SEQUENCE pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \hat{\delta}_f^{i+1})$ M

Table 3.19: Parametrised glue matrix for any g_f_STRICTLY_INCREASING_SEQUENCE constraint, where the cell annotation M stands for merging two existing strictly increasing sequences sharing an element; the state associated with each row corresponds to a state of the STRICTLY_INCREASING_SEQUENCE transducer – see Figure 3.56, while the state associated with each column corresponds to a state of the reverse of the STRICTLY_INCREASING_SEQUENCE transducer, i.e. the STRICTLY_DECREASING_SEQUENCE transducer – see Figure 3.53.

3.1.20 SUMMIT



Given a sequence S over the alphabet {'<', '=', '>'}, an occurrence of the pattern SUMMIT is the *maximal* subsequence of S which matches the regular expression '(< |(< (= | <)* <))(> |(> (= | >)* >))'. Part (A) and parts (B,C) of Figure 3.59 respectively depict the seed transducer associated with the SUMMIT pattern as well as one example of its execution on a time series.

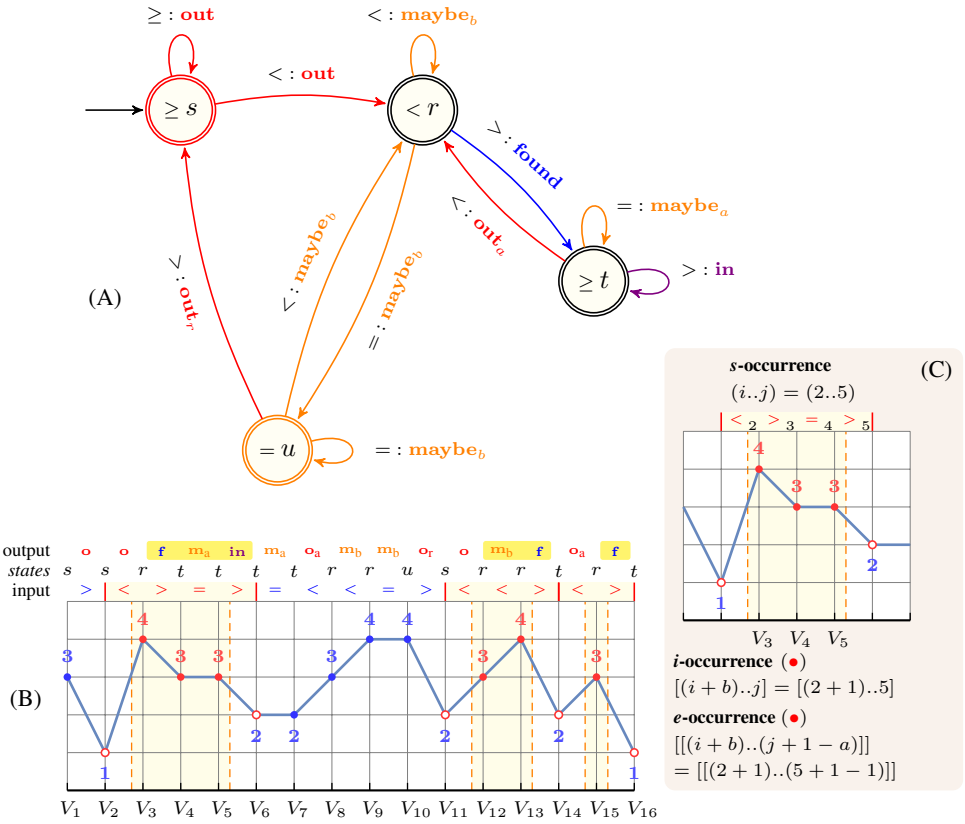


Figure 3.59: (A) Seed transducer for the SUMMIT pattern: '(< |(< (= | <)* <))(> |(> (= | >)* >))' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output **o**, **f**, $\mathbf{m_a}$, $\mathbf{o_a}$, $\mathbf{m_b}$, $\mathbf{o_r}$ are shortcut for **out**, **found**, $\mathbf{maybe_a}$, $\mathbf{out_a}$, $\mathbf{maybe_b}$, $\mathbf{out_r}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

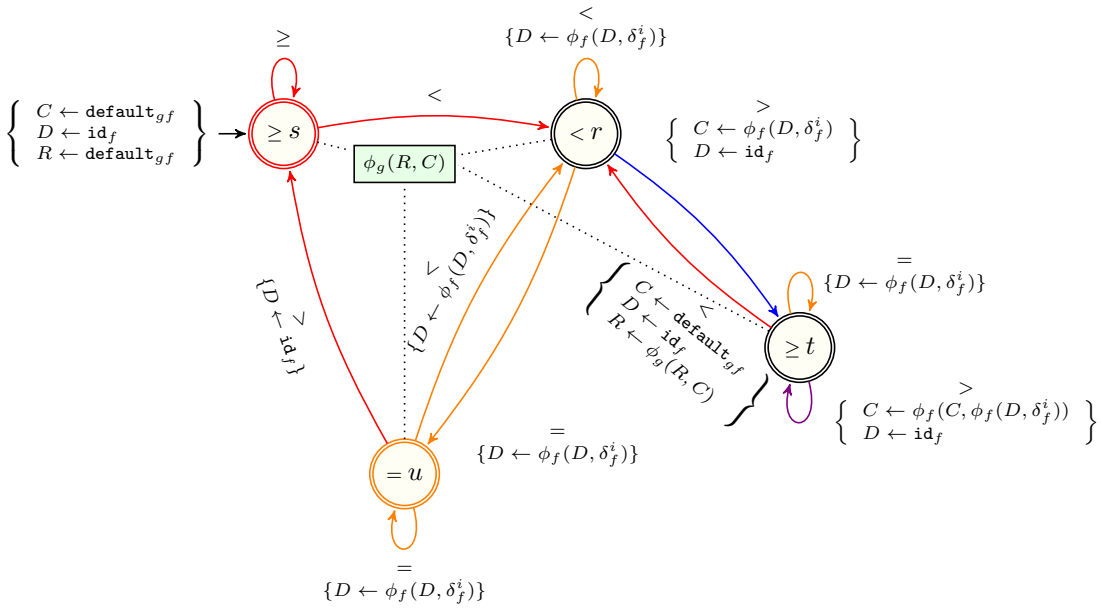


Figure 3.60: Parametrised automaton for any functional time-series constraints of the SUMMIT pattern obtained by applying the decoration table 3.37 to the corresponding transducer (transition $u \to r$ has the same register update as transition $r \to u$)

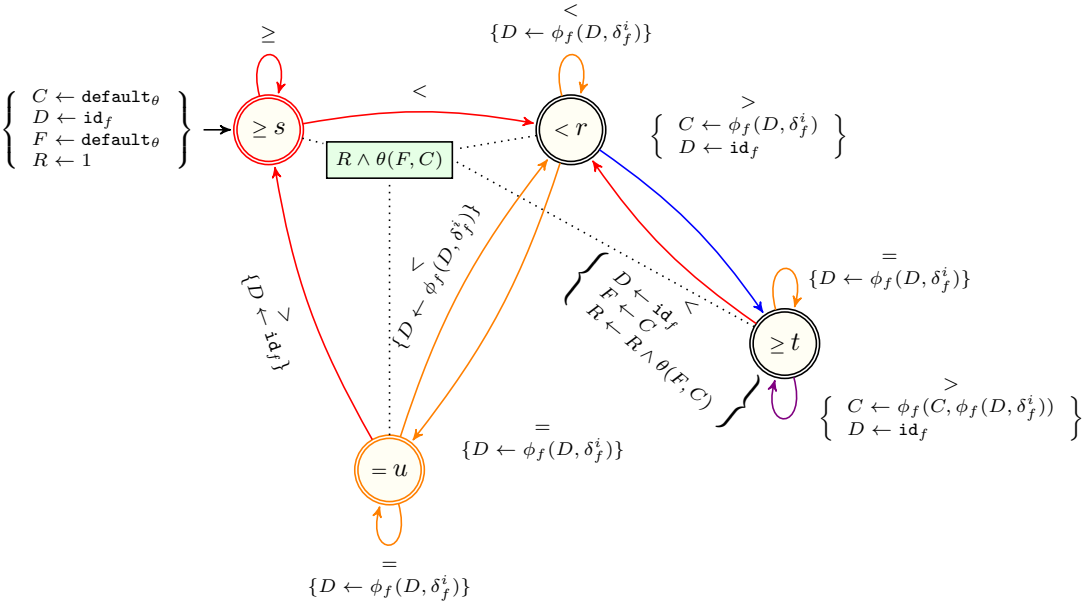
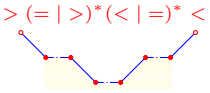


Figure 3.61: Parametrised automaton for any predicate time-series constraints of the SUMMIT pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values; transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$.

	s	r	t	u
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L
u	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.20: Parametrised glue matrix for any $g_f_$SUMMIT constraint, where cell annotations have the following meaning: C stands for creating a new summit occurrence from no existing summit, R stands for extending to the right an existing summit, L stands for extending to the left an existing summit; the state associated with each row corresponds to a state of the SUMMIT transducer – see Figure 3.59, while the state associated with each column corresponds to a state of the reverse of the SUMMIT transducer, i.e. the SUMMIT transducer – see Figure 3.59.

3.1.21 VALLEY



Given a sequence S over the alphabet $\{\text{'<'}, \text{'='}, \text{'>'}\}$, an occurrence of the pattern VALLEY is the *maximal* subsequence of S which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'. Part (A) and parts (B,C) Figure 3.62 respectively depict the seed transducer associated with the VALLEY pattern as well as one example of its execution on a time series.

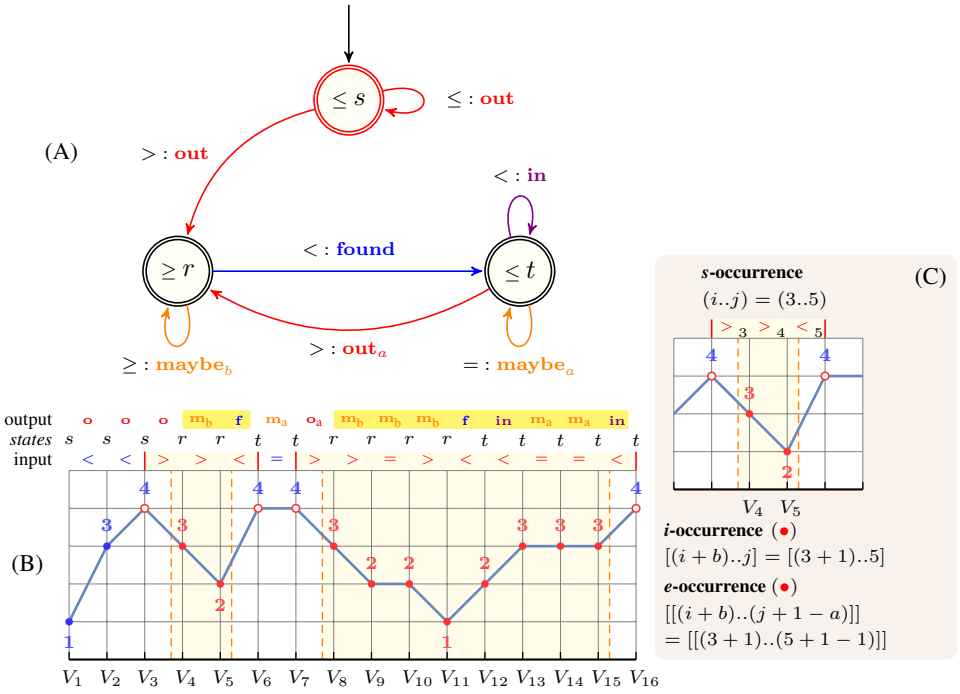


Figure 3.62: (A) Seed transducer for the VALLEY pattern: '$> (= \mid >)^* (< \mid =)^* <$' with $b = 1$ and $a = 1$; (B) Illustrating the execution of the seed transducer on a time series (within the output $\mathbf{o}$, $\mathbf{m_b}$, $\mathbf{f}$, $\mathbf{m_a}$, $\mathbf{o_a}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe_b}$, $\mathbf{found}$, $\mathbf{maybe_a}$, $\mathbf{out_a}$); (C) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

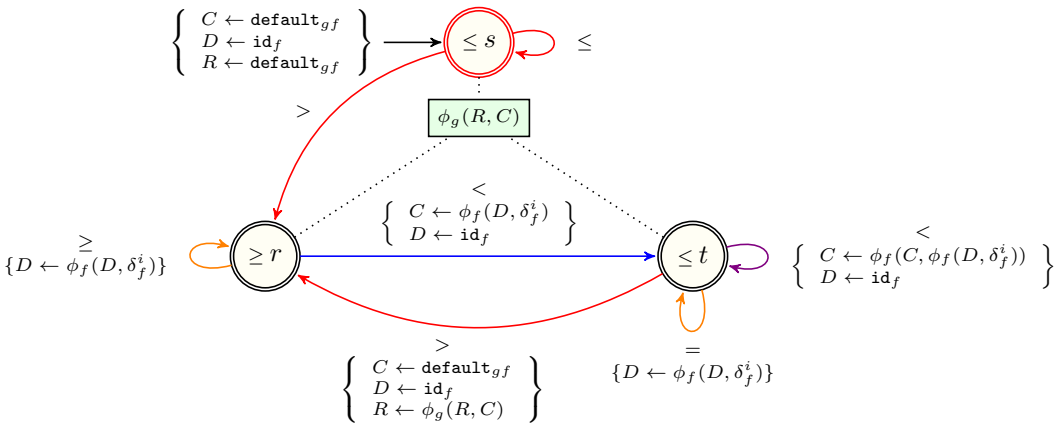


Figure 3.63: Parametrised automaton for any functional time-series constraints of the VALLEY pattern obtained by applying the decoration table 3.37 to the corresponding transducer

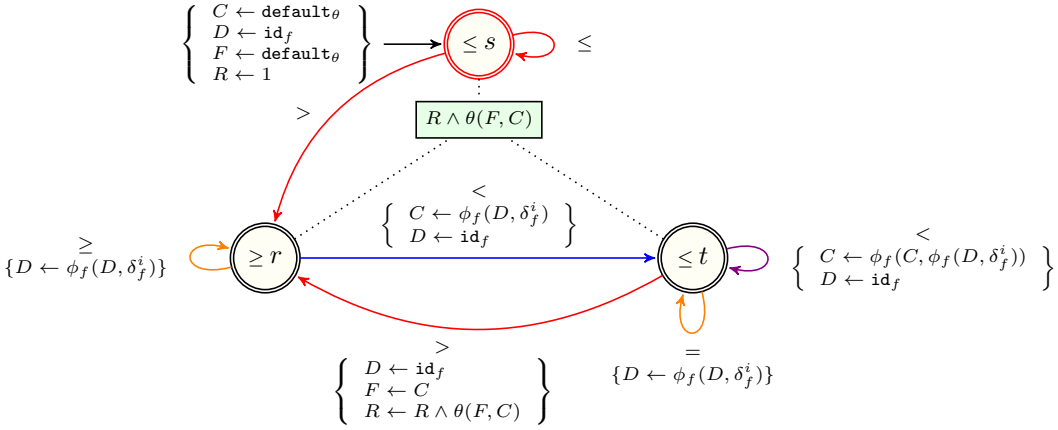
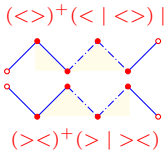


Figure 3.64: Parametrised automaton for any predicate time-series constraints of the VALLEY pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq$, $=$, $\geq$) used for comparing two consecutive feature values

	s	r	t
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
r	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
t	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

Table 3.21: Parametrised glue matrix for any $g_f_$VALLEY constraint, where cell annotations have the following meaning: C stands for creating a new valley occurrence from no existing valley, R stands for extending to the right an existing valley, L stands for extending to the left an existing valley; the state associated with each row corresponds to a state of the VALLEY transducer – see Figure 3.62, while the state associated with each column corresponds to a state of the reverse of the VALLEY transducer, i.e. the VALLEY transducer – see Figure 3.62.

3.1.22 ZIGZAG



Given a sequence S over the alphabet $\{\text{`}<\text{'}, \text{`}=\text{'}, \text{`}>\text{'}\}$, an occurrence of the pattern ZIGZAG is the *maximal* subsequence of S which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'. Figures 3.65 and 3.66 respectively depict the seed transducer associated with the ZIGZAG pattern as well as one example of its execution on a time series.

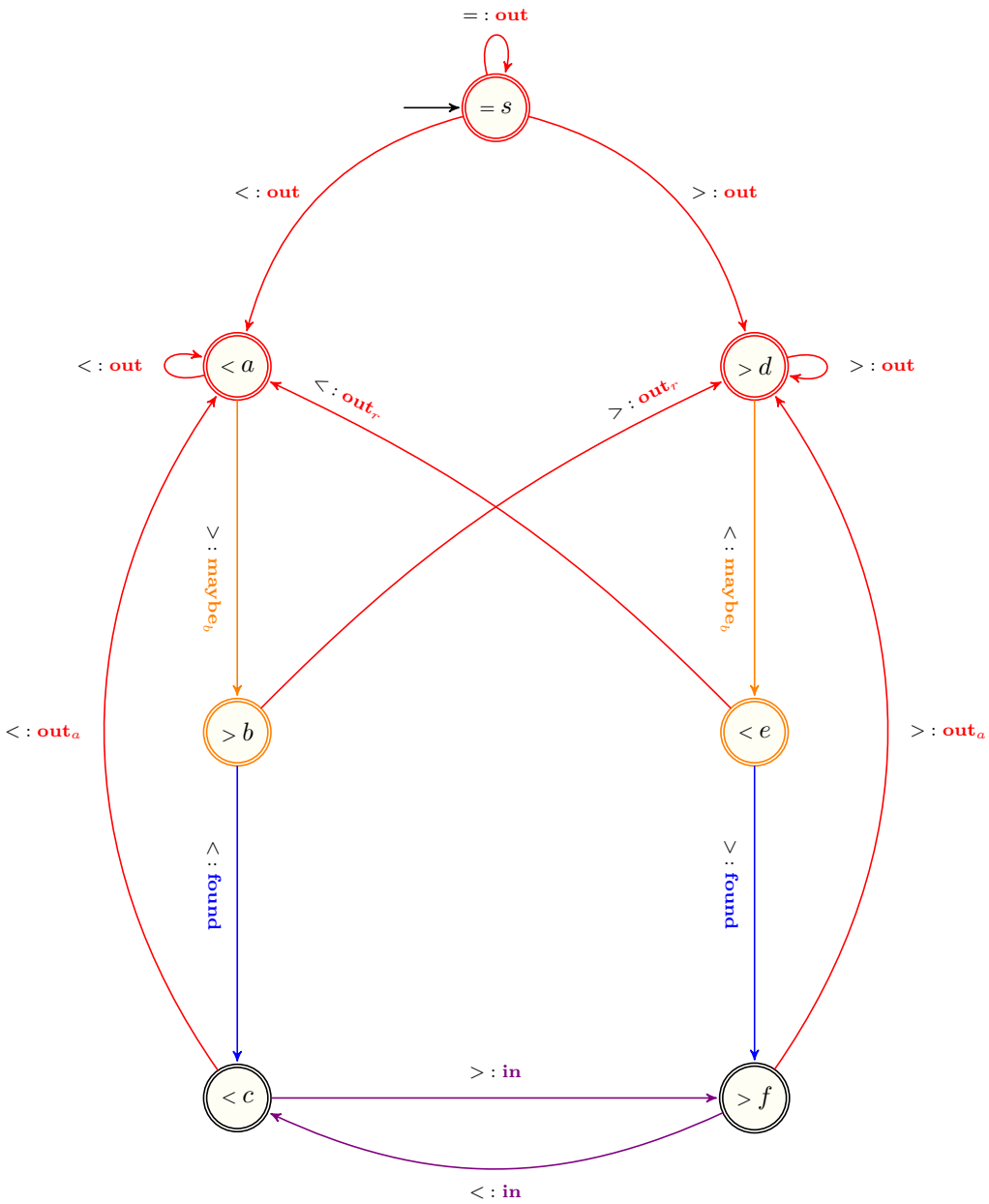


Figure 3.65: Seed transducer for the ZIGZAG pattern: '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$' with $b = 1$ and $a = 1$; missing transitions from a, d to s are labelled by = : **out**, missing transitions from b, e to s are labelled by = : $\mathbf{out}_{\mathrm{r}}$, and missing transitions from c, f to s are labelled by = : $\mathbf{out}_{\mathrm{a}}$.

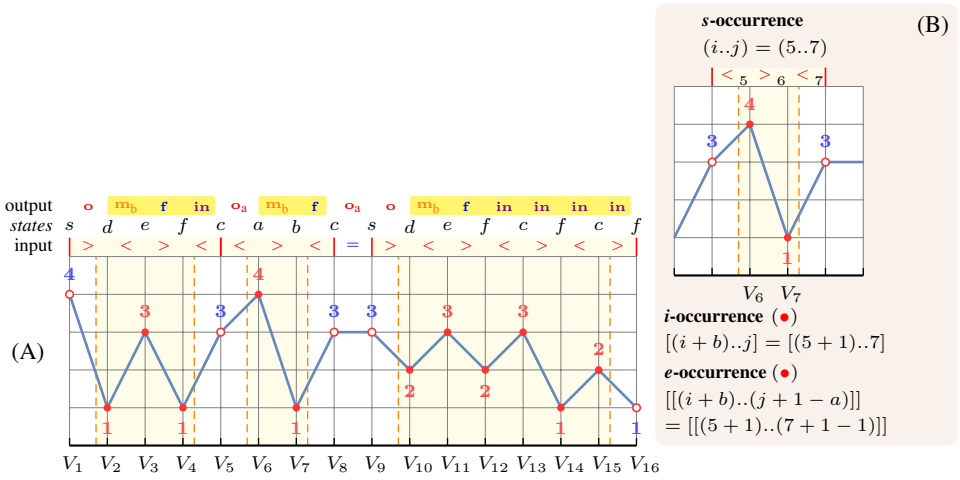


Figure 3.66: (A) Illustrating the execution of the seed transducer associated with the ZIGZAG pattern on a time series (within the output $\mathbf{o}$, $\mathbf{m}_\mathrm{b}$, $\mathbf{f}$, $\mathbf{o}_\mathrm{a}$ are shortcut for $\mathbf{out}$, $\mathbf{maybe}_\mathrm{b}$, $\mathbf{found}$, $\mathbf{out}_\mathrm{a}$); (B) Illustrating the relation between the *s*-occurrence, the *i*-occurrence and the *e*-occurrence.

	s	a	b	c
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
a	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
b	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
c	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ M
d	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
e	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
f	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$

	d	e	f
s	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
a	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
b	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
c	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
d	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ R
e	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ C	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$
f	$\phi_f(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ L	$\phi_g(\overrightarrow{C}, \overleftarrow{C})$	$\phi_f(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \delta_f^{i+1})$ M

Table 3.22: Parametrised glue matrix for any $g_f_$ZIGZAG constraint, where cell annotations have the following meaning: C stands for creating a new zigzag occurrence from no existing zigzag, R stands for extending to the right an existing zigzag, L stands for extending to the left an existing zigzag, M stands for merging two existing zigzags; the state associated with each row corresponds to a state of the ZIGZAG transducer – see Figure 3.65, while the state associated with each column corresponds to a state of the reverse of the ZIGZAG transducer, i.e. the ZIGZAG transducer – see Figure 3.65.

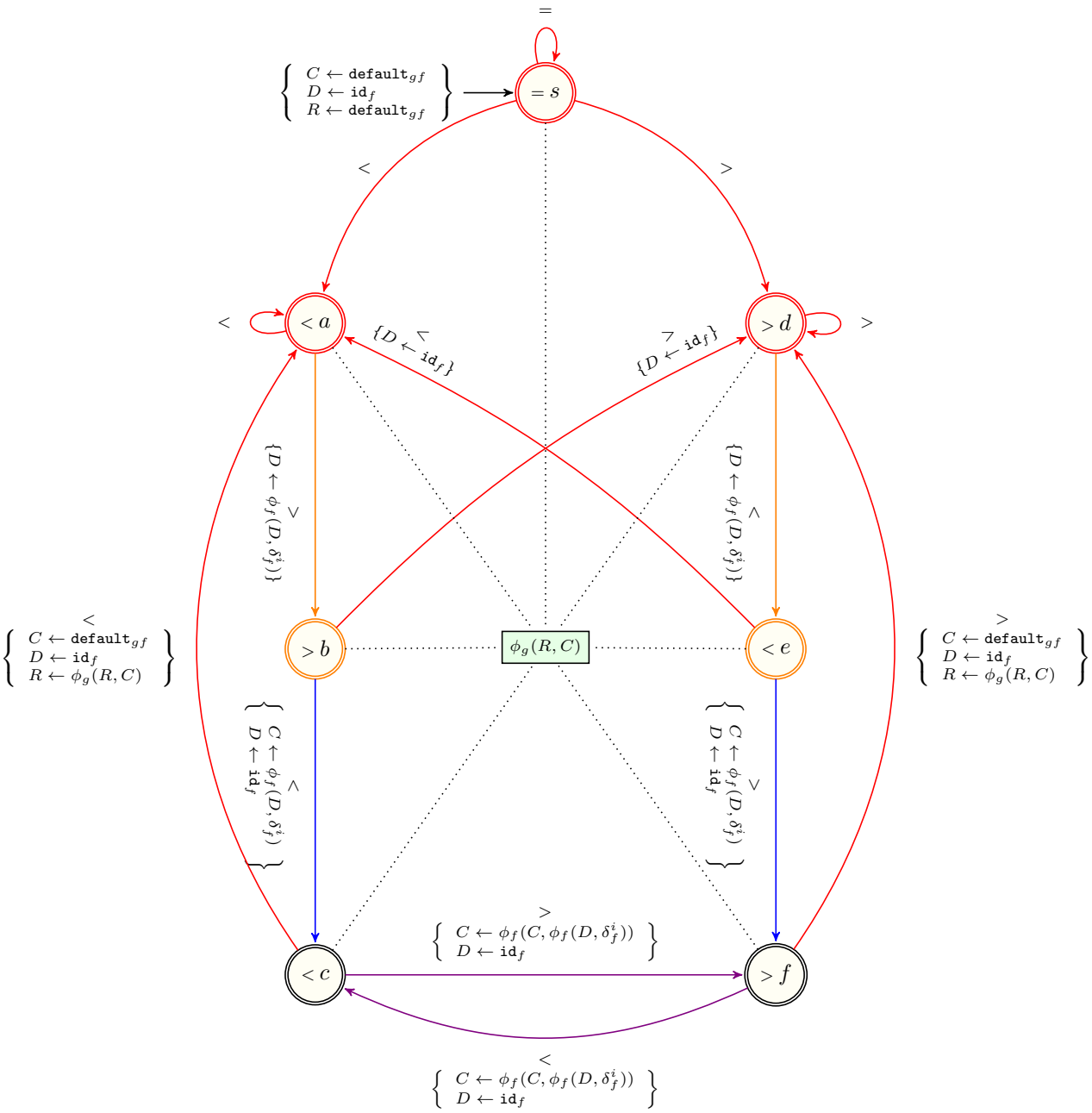


Figure 3.67: Parametrised automaton for any functional time-series constraints of the ZIGZAG pattern obtained by applying the decoration table 3.37 to the corresponding transducer; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.

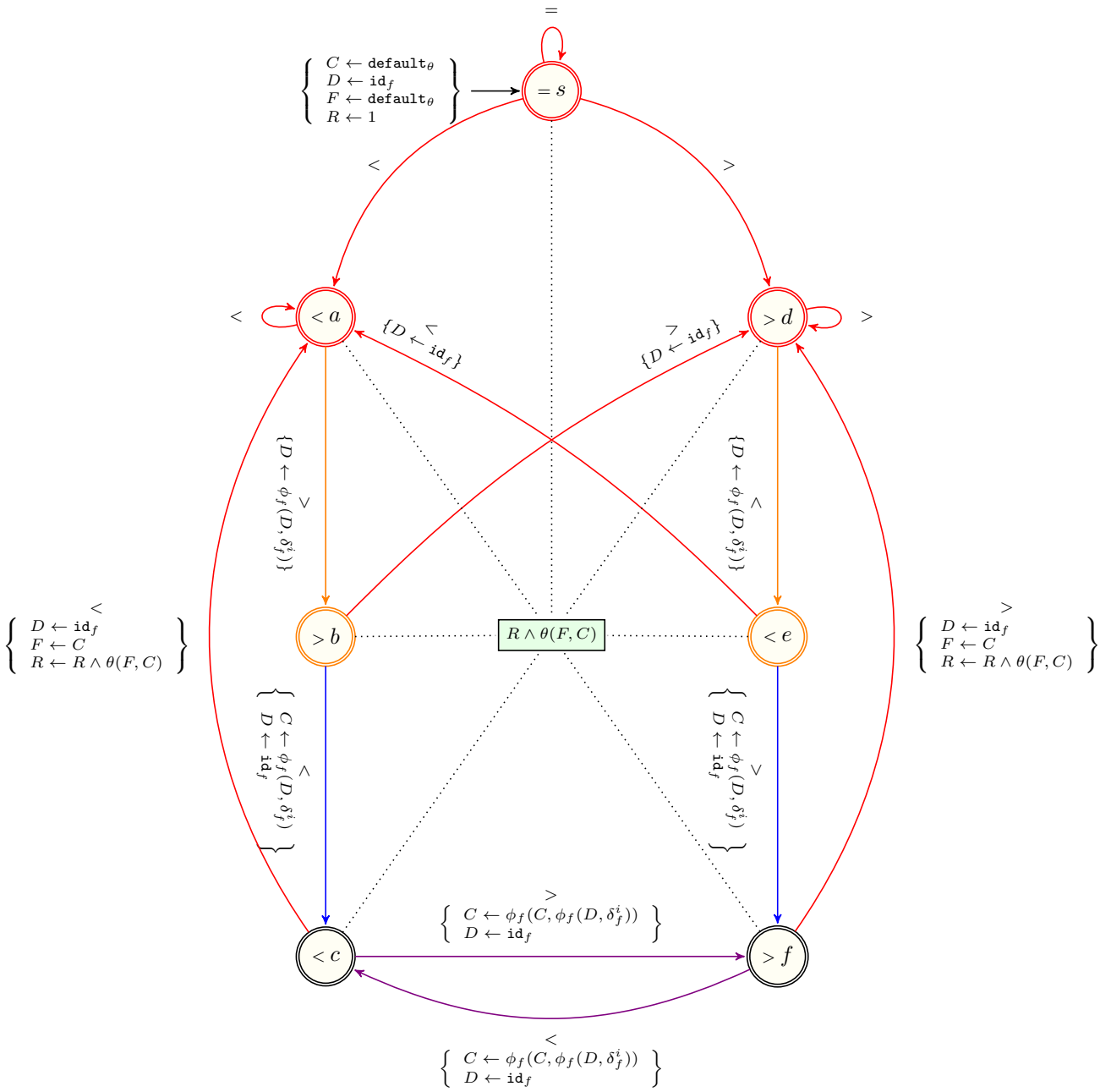


Figure 3.68: Parametrised automaton for any predicate time-series constraints of the ZIGZAG pattern obtained by applying the decoration table 3.38 to the corresponding transducer, where θ is the predicate (i.e. $\leq, =, \geq$) used for comparing two consecutive feature values; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

3.2 Decoration Tables

This section provides different decoration tables used for synthesising the automata with registers of this catalogue. While no automaton of this catalogue relies on tables 3.34 and 3.36, these two decoration tables may be used for generating distance constraints between consecutive occurrences of a pattern.

Table 3.23: Table for Type: cDRFoundEOut

initialization		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \texttt{default}_{gf}$
return		$\phi_g(R, C)$		
Semantic Letter		Decoration		
	After	Update of C	Update of D	Update of R
out			$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_r$			$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_a$		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0		$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1		$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	
found	1	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	
in	0	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^{i+1}))$	$D \leftarrow \texttt{id}_f$	
in	1	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^i))$	$D \leftarrow \texttt{id}_f$	
$\mathbf{found}_\mathbf{e}$	0		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.24: Table for Type: cDRFoundEOutNoMA

initialization		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \texttt{default}_{gf}$
return		$\phi_g(R, C)$		
Semantic Letter		Decoration		
	After	Update of C	Update of D	Update of R
out				
$\mathbf{out}_r$			$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_a$		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0		$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1		$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$C \leftarrow \phi_f(\phi_r(D, \delta_r^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	
found	1	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	
in	0	$C \leftarrow \phi_f(C, \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	
in	1	$C \leftarrow \phi_f(C, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	
$\mathbf{found}_\mathbf{e}$	0		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.25: Table for Type: cRFoundEOutA

initialization		$C \leftarrow \texttt{default}_{gf}$	$R \leftarrow \texttt{default}_{gf}$
return		$\phi_g(R, C)$	
Semantic Letter		Decoration	
	After	Update of C	Update of R
out			
$\mathbf{out}_r$		$C \leftarrow \texttt{default}_{gf}$	
$\mathbf{out}_a$		$C \leftarrow \texttt{default}_{gf}$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$	0		
$\mathbf{maybe}_a$	1		
found	0	$C \leftarrow \phi_f(\delta_f^i, \delta_f^{i+1})$	
found	1	$C \leftarrow \delta_f^i$	
in	0	$C \leftarrow \phi_f(C, \delta_f^{i+1})$	
in	1	$C \leftarrow \phi_f(C, \delta_f^i)$	
$\mathbf{found}_\mathbf{e}$	0	$C \leftarrow \texttt{default}_{gf}$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1	$C \leftarrow \texttt{default}_{gf}$	$R \leftarrow \phi_g(R, \delta_f^i)$

Table 3.26: Table for Type: dRFoundFoundEInA

initialization		$D \leftarrow \mathtt{id}_f$	$R \leftarrow \mathtt{default}_{gf}$
return		R	
Semantic Letter		Decoration	
	After	Update of D	Update of R
out			
$\mathbf{out}_r$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{out}_a$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{maybe}_b$		$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0	$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
found	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
in	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^{i+1}))$
in	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found}_\mathbf{e}$	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.27: Table for Type: dRFoundFoundEInB

initialization		$D \leftarrow \mathtt{id}_f$	$R \leftarrow \mathtt{default}_{gf}$
return		R	
Semantic Letter		Decoration	
	After	Update of D	Update of R
out			
$\mathbf{out}_r$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{out}_a$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{maybe}_b$		$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0	$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$D \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
found	1	$D \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
in	0	$D \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^{i+1}))$
in	1	$D \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found_e}$	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found_e}$	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.28: Table for Type: dRFoundFoundEInMonMB

initialization	$D \leftarrow \texttt{id}_f$	$R \leftarrow \texttt{default}_{gf}$
return	R	
Semantic Letter		Decoration
	Update of D	Update of R
out		
$\mathbf{out}_r$		
$\mathbf{out}_a$		
$\mathbf{maybe}_b$	$D \leftarrow \delta_f^i$	
$\mathbf{maybe}_a$		
found		
in		
$\mathbf{found}_\mathbf{e}$		$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.29: Table for Type: dRFoundFoundEInNoMA

initialization		$D \leftarrow \mathtt{id}_f$	$R \leftarrow \mathtt{default}_{gf}$
return		R	
Semantic Letter		Decoration	
	After	Update of D	Update of R
out			
$\mathbf{out}_r$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{out}_a$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{maybe}_b$		$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0	$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
found	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
in	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \delta_f^{i+1})$
in	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \delta_f^i)$
$\mathbf{found}_\mathbf{e}$	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.30: Table for Type: dRFoundFoundEInNoMBA

initialization		$D \leftarrow \mathtt{id}_f$	$R \leftarrow \mathtt{default}_{gf}$
return		R	
Semantic Letter		Decoration	
	After	Update of D	Update of R
out			
$\mathbf{out}_r$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{out}_a$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$	0	$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
found	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \delta_f^i)$
in	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^{i+1}))$
in	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found}_\mathbf{e}$	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \delta_f^i)$

Table 3.31: Table for Type: dRFoundFoundEInNoMBB

initialization		$D \leftarrow \mathtt{id}_f$	$R \leftarrow \mathtt{default}_{gf}$
return		R	
Semantic Letter		Decoration	
	After	Update of D	Update of R
out			
$\mathbf{out}_r$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{out}_a$		$D \leftarrow \mathtt{id}_f$	
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$	0	$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$D \leftarrow \phi_f(\delta_f^i, \delta_f^{i+1})$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
found	1	$D \leftarrow \delta_f^i$	$R \leftarrow \phi_g(R, \delta_f^i)$
in	0	$D \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^{i+1}))$
in	1	$D \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found_e}$	0	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
$\mathbf{found_e}$	1	$D \leftarrow \mathtt{id}_f$	$R \leftarrow \phi_g(R, \delta_f^i)$

Table 3.32: Table for Type: features

initialization			$C \leftarrow \texttt{id}_f$	$D \leftarrow \texttt{id}_f$
return		$f_n = \texttt{default}$ $e_n = C$		
Semantic Letter		Decoration		
	After	Guard	Update of C	Update of D
out		$f_i = \texttt{default}, e_i = \texttt{default}$		
$\mathbf{out}_r$		$f_i = \texttt{default}, e_i = \texttt{default}$		$D \leftarrow \texttt{id}_f$
$\mathbf{out}_a$		$f_i = \texttt{default}, e_i = c$	$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$
$\mathbf{maybe}_b$		$f_i = \texttt{default}, e_i = e_{i+1}$		$D \leftarrow \phi_f(D, \delta_f^i)$
$\mathbf{maybe}_a$	0	$f_i = \texttt{default}, e_i = e_{i+1}$		$D \leftarrow \phi_f(D, \delta_f^{i+1})$
$\mathbf{maybe}_a$	1	$f_i = \texttt{default}, e_i = e_{i+1}$		$D \leftarrow \phi_f(D, \delta_f^i)$
found	0	$f_i = e_i, e_i = e_{i+1}$	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$
found	1	$f_i = e_i, e_i = e_{i+1}$	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$
in	0	$f_i = \texttt{default}, e_i = e_{i+1}$	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^{i+1}))$	$D \leftarrow \texttt{id}_f$
in	1	$f_i = \texttt{default}, e_i = e_{i+1}$	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^i))$	$D \leftarrow \texttt{id}_f$
$\mathbf{found}_\mathbf{e}$	0	$f_i = \phi_f(\phi_f(d, \delta_f^i), \delta_f^{i+1}), e_i = \texttt{default}$		$D \leftarrow \texttt{id}_f$
$\mathbf{found}_\mathbf{e}$	1	$f_i = \phi_f(d, \delta_f^i), e_i = \texttt{default}$		$D \leftarrow \texttt{id}_f$

Table 3.33: Table for Type: footprint

initialization		$C \leftarrow 0$
return	$p_n = 0$	
Semantic Letter	Decoration	
	Guard	Update of C
out	$p_i = 0$	
$\mathbf{out}_r$	$p_i = 0$	
$\mathbf{out}_a$	$p_i = 0$	
$\mathbf{maybe}_b$	$p_i = p_{i+1}$	
$\mathbf{maybe}_a$	$p_i = p_{i+1}$	
$\mathbf{found_e}$	$p_i = C + 1$	$C \leftarrow C + 1$
found	$p_i = C + 1$	$C \leftarrow C + 1$
in	$p_i = C$	

Table 3.34: Table for Type: footprint_distance

initialization		$C \leftarrow \mathtt{id}_g$	$D \leftarrow 0$	$F \leftarrow 0$	$M \leftarrow 0$
return	C				
Semantic Letter	Decoration				
	Guard	Update of C	Update of D	Update of F	Update of M
out			$D \leftarrow D + 1$		
$\mathbf{out}_r$			$D \leftarrow D + M$		$M \leftarrow 0$
$\mathbf{out}_a$			$D \leftarrow D + 1$		
$\mathbf{maybe}_b$					$M \leftarrow M + 1$
$\mathbf{maybe}_a$			$D \leftarrow D + 1$		
$\mathbf{found_e}$	$F = 1$	$C \leftarrow \phi_g(C, D)$	$D \leftarrow 0$		
$\mathbf{found_e}$	$F = 0$		$D \leftarrow 0$	$F \leftarrow 1$	
found	$F = 1$	$C \leftarrow \phi_g(C, D)$	$D \leftarrow 0$		
found	$F = 0$		$D \leftarrow 0$	$F \leftarrow 1$	
in			$D \leftarrow 0$		

Table 3.35: Table for Type: found ($<_g$ stands for $<$ when $g = \min$ and $>$ when $g = \max$)

initialization			$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \texttt{default}_{gf}$
return		$f_n = 0$ $C <_g R \Rightarrow ct_n = 1, at_n = 0$ $C = R, R = \texttt{default}_{gf} \Rightarrow ct_n = 0, at_n = 0$ $C = R, R \neq \texttt{default}_{gf} \Rightarrow ct_n = 1, at_n = 1$ $R <_g C \Rightarrow ct_n = 0, at_n = 1$ **return** $\phi_g(R, C)$			
Semantic Letter	After	Guard	Decoration Update of C	Update of D	Update of R
out		$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_r$		$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{out}_a$		$c <_g r, f_i = 0, at_i = 0, at_{i+1} = ct_i$	$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{out}_a$		$c = r, f_i = 0, at_{i+1} = ct_i, at_{i+1} = at_i$	$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{out}_a$		$r <_g c, f_i = 0, ct_i = 0, at_{i+1} = at_i$	$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$		$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0	$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1	$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$ct_i = 0, ct_{i+1} = f_i, at_{i+1} = at_i$	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	
found	1	$ct_i = 0, ct_{i+1} = f_i, at_{i+1} = at_i$	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	
in	0	$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^{i+1}))$	$D \leftarrow \texttt{id}_f$	
in	1	$f_i = 0, ct_{i+1} = ct_i, at_{i+1} = at_i$	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^i))$	$D \leftarrow \texttt{id}_f$	
$\mathbf{found_e}$	0	$\phi_f(\phi_f(d, \delta_f^i), \delta_f^{i+1}) <_g r, ct_i = f_i, at_i = 0, at_{i+1} = ct_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found_e}$	0	$\phi_f(\phi_f(d, \delta_f^i), \delta_f^{i+1}) = r, at_{i+1} = f_i, at_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found_e}$	0	$r <_g \phi_f(\phi_f(d, \delta_f^i), \delta_f^{i+1}), f_i = 0, ct_i = 0, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found_e}$	1	$\phi_f(d, \delta_f^i) <_g r, ct_i = f_i, at_i = 0, at_{i+1} = ct_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found_e}$	1	$\phi_f(d, \delta_f^i) = r, at_{i+1} = f_i, at_{i+1} = ct_i, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$
$\mathbf{found_e}$	1	$r <_g \phi_f(d, \delta_f^i), f_i = 0, ct_i = 0, at_{i+1} = at_i$		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.36: Table for Type: found_distance

initialization		$C \leftarrow \texttt{id}_g$	$F \leftarrow 0$
return	C		
Semantic Letter	Decoration		
	Guard	Update of C	Update of F
out			
$\mathbf{out}_r$			
$\mathbf{out}_a$			
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$			
$\mathbf{found_e}$	$F > 0$	$C \leftarrow \phi_g(C, i - F)$	$F \leftarrow i$
$\mathbf{found_e}$	$F = 0$		$F \leftarrow i$
found	$F > 0$	$C \leftarrow \phi_g(C, i - F)$	$F \leftarrow i$
found	$F = 0$		$F \leftarrow i$
in			

Table 3.37: Table for Type: function

initialization		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \texttt{default}_{gf}$
return		$\phi_g(R, C)$		
Semantic Letter		Decoration		
	After	Update of C	Update of D	Update of R
out			$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_r$			$D \leftarrow \texttt{id}_f$	
$\mathbf{out}_a$		$C \leftarrow \texttt{default}_{gf}$	$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			$D \leftarrow \phi_f(D, \delta_f^i)$	
$\mathbf{maybe}_a$	0		$D \leftarrow \phi_f(D, \delta_f^{i+1})$	
$\mathbf{maybe}_a$	1		$D \leftarrow \phi_f(D, \delta_f^i)$	
found	0	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	
found	1	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	
in	0	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^{i+1}))$	$D \leftarrow \texttt{id}_f$	
in	1	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^i))$	$D \leftarrow \texttt{id}_f$	
$\mathbf{found}_\mathbf{e}$	0		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found}_\mathbf{e}$	1		$D \leftarrow \texttt{id}_f$	$R \leftarrow \phi_g(R, \phi_f(D, \delta_f^i))$

Table 3.38: Table for Type: predicate

initialization		$C \leftarrow \texttt{default}_\theta$	$D \leftarrow \texttt{id}_f$	$F \leftarrow \texttt{default}_\theta$	$R \leftarrow 1$
return		$R \wedge \theta(F, C)$			
Semantic Letter			Decoration		
	After	Update of C	Update of D	Update of F	Update of R
out					
$\mathbf{out}_r$			$D \leftarrow \texttt{id}_f$		
$\mathbf{out}_a$			$D \leftarrow \texttt{id}_f$	$F \leftarrow C$	$R \leftarrow R \wedge \theta(F, C)$
$\mathbf{maybe}_b$			$D \leftarrow \phi_f(D, \delta_f^i)$		
$\mathbf{maybe}_a$	0		$D \leftarrow \phi_f(D, \delta_f^{i+1})$		
$\mathbf{maybe}_a$	1		$D \leftarrow \phi_f(D, \delta_f^i)$		
found	0	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$		
found	1	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$		
in	0	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^{i+1}))$	$D \leftarrow \texttt{id}_f$		
in	1	$C \leftarrow \phi_f(C, \phi_f(D, \delta_f^i))$	$D \leftarrow \texttt{id}_f$		
$\mathbf{found_e}$	0	$C \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$D \leftarrow \texttt{id}_f$	$F \leftarrow \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1})$	$R \leftarrow R \wedge \theta(F, \phi_f(\phi_f(D, \delta_f^i), \delta_f^{i+1}))$
$\mathbf{found_e}$	1	$C \leftarrow \phi_f(D, \delta_f^i)$	$D \leftarrow \texttt{id}_f$	$F \leftarrow \phi_f(D, \delta_f^i)$	$R \leftarrow R \wedge \theta(F, \phi_f(D, \delta_f^i))$

Table 3.39: Table for Type: rFoundFoundE

initialization	$R \leftarrow \texttt{default}_{gf}$
return	R
Semantic Letter	Decoration
	Update of R
out	
$\mathbf{out}_r$	
$\mathbf{out}_a$	
$\mathbf{maybe}_b$	
$\mathbf{maybe}_a$	
found	$R \leftarrow \phi_g(R, \delta_f^i)$
in	
$\mathbf{found_e}$	$R \leftarrow \phi_g(R, \delta_f^i)$

Table 3.40: Table for Type: rFoundFoundEIn

initialization	$R \leftarrow \texttt{default}_{gf}$
return	R
Semantic Letter	Decoration
	Update of R
out	
$\mathbf{out}_r$	
$\mathbf{out}_a$	
$\mathbf{maybe}_b$	
$\mathbf{maybe}_a$	
found	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$
in	$R \leftarrow \phi_g(R, \delta_f^{i+1})$
$\mathbf{found}_\mathbf{e}$	$R \leftarrow \phi_g(R, \phi_f(\delta_f^i, \delta_f^{i+1}))$

Table 3.41: Table for Type: rFoundInSWZigzag

initialization	$R \leftarrow \texttt{default}_{gf}$
return	R
Semantic Letter	Decoration
	Update of R
out	
$\mathbf{out}_r$	
$\mathbf{out}_a$	
$\mathbf{maybe}_b$	
$\mathbf{maybe}_a$	
found	$R \leftarrow \phi_g(\phi_g(R, \delta_f^i), \delta_f^{i+1})$
in	$R \leftarrow \phi_g(R, \delta_f^i)$
$\mathbf{found}_\mathbf{e}$	

Table 3.42: Table for Type: rangeCHRFoundEOutA

initialization	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow \texttt{default}_{gf}$
return	$\phi_g(R, C)$		
Semantic Letter	Decoration		
	Update of C	Update of H	Update of R
out		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$			
found	$C \leftarrow H - \delta_f^{i+1}$		
in	$C \leftarrow H - \delta_f^{i+1}$		
$\mathbf{found_e}$		$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, H - \delta_f^{i+1})$

Table 3.43: Table for Type: rangeCHRFoundEOutB

initialization	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow \texttt{default}_{gf}$
return	$\phi_g(R, C)$		
Semantic Letter	Decoration		
	Update of C	Update of H	Update of R
out		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$			
found	$C \leftarrow \delta_f^{i+1} - H$		
in	$C \leftarrow \delta_f^{i+1} - H$		
$\mathbf{found_e}$		$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, \delta_f^{i+1} - H)$

Table 3.44: Table for Type: rangeHRFoundFoundEInA

initialization	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow \texttt{default}_{gf}$
return	R	
Semantic Letter		Decoration
	Update of H	Update of R
out	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{maybe}_b$		
$\mathbf{maybe}_a$		
found		$R \leftarrow \phi_g(R, H - \delta_f^{i+1})$
in		$R \leftarrow \phi_g(R, H - \delta_f^{i+1})$
$\mathbf{found_e}$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, H - \delta_f^{i+1})$

Table 3.45: Table for Type: rangeHRFoundFoundEInB

initialization	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow \texttt{default}_{gf}$
return	R	
Semantic Letter		Decoration
	Update of H	Update of R
out	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$	$H \leftarrow \delta_f^{i+1}$	
$\mathbf{maybe}_b$		
$\mathbf{maybe}_a$		
found		$R \leftarrow \phi_g(R, \delta_f^{i+1} - H)$
in		$R \leftarrow \phi_g(R, \delta_f^{i+1} - H)$
$\mathbf{found_e}$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, \delta_f^{i+1} - H)$

Table 3.46: Table for Type: rangeRFoundFoundEInA

initialization	$R \leftarrow \texttt{default}_{gf}$
return	R
Semantic Letter	Decoration
	Update of R
out	
$\mathbf{out}_r$	
$\mathbf{out}_a$	
$\mathbf{maybe}_b$	
$\mathbf{maybe}_a$	
found	$R \leftarrow \phi_g(R, \max(0, \delta_f^i - \delta_f^{i+1}))$
in	$R \leftarrow \phi_g(R, \max(0, \delta_f^i - \delta_f^{i+1}))$
$\mathbf{found_e}$	$R \leftarrow \phi_g(R, \max(0, \delta_f^i - \delta_f^{i+1}))$

Table 3.47: Table for Type: rangeRFoundFoundEInB

initialization	$R \leftarrow \texttt{default}_{gf}$
return	R
Semantic Letter	Decoration
	Update of R
out	
$\mathbf{out}_r$	
$\mathbf{out}_a$	
$\mathbf{maybe}_b$	
$\mathbf{maybe}_a$	
found	$R \leftarrow \phi_g(R, \max(0, \delta_f^{i+1} - \delta_f^i))$
in	$R \leftarrow \phi_g(R, \max(0, \delta_f^{i+1} - \delta_f^i))$
$\mathbf{found}_\mathbf{e}$	$R \leftarrow \phi_g(R, \max(0, \delta_f^{i+1} - \delta_f^i))$

Table 3.48: Table for Type: range_function

initialization	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow \texttt{default}_{gf}$
return	$\phi_g(R, C)$		
Semantic Letter	Decoration		
	Update of C	Update of H	Update of R
out		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$		$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$	$C \leftarrow \texttt{default}_{gf}$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, C)$
$\mathbf{maybe}_b$			
$\mathbf{maybe}_a$			
$\mathbf{found}_\mathbf{e}$		$H \leftarrow \delta_f^{i+1}$	$R \leftarrow \phi_g(R, \lvert H - \delta_f^{i+1} \rvert)$
found	$C \leftarrow \lvert H - \delta_f^{i+1} \rvert$		
in	$C \leftarrow \lvert H - \delta_f^{i+1} \rvert$		

Table 3.49: Table for Type: range_predicate

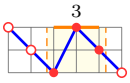
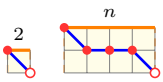
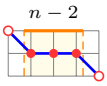
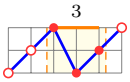
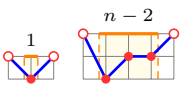
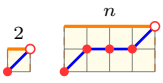
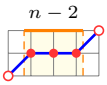
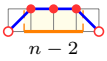
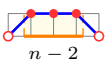
initialization	$C \leftarrow \texttt{default}_\theta$	$F \leftarrow \texttt{default}_\theta$	$H \leftarrow \texttt{VAR}_1$	$R \leftarrow 1$
return	$R \wedge \theta(F, C)$			
Semantic Letter	Decoration			
	Update of C	Update of F	Update of H	Update of R
out			$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_r$			$H \leftarrow \delta_f^{i+1}$	
$\mathbf{out}_a$		$F \leftarrow \lvert H - \delta_f^i \rvert$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow R \wedge \theta(F, C)$
$\mathbf{maybe}_b$				
$\mathbf{maybe}_a$				
$\mathbf{found}_\mathbf{e}$	$C \leftarrow \lvert H - \delta_f^{i+1} \rvert$	$F \leftarrow \lvert H - \delta_f^{i+1} \rvert$	$H \leftarrow \delta_f^{i+1}$	$R \leftarrow R \wedge \theta(F, \lvert H - \delta_f^{i+1} \rvert)$
found	$C \leftarrow \lvert H - \delta_f^{i+1} \rvert$			
in	$C \leftarrow \lvert H - \delta_f^{i+1} \rvert$			

3.3 Tables of Regular-Expression Characteristics

This section provides the different tables of regular-expression characteristics defined in [6, 4] used to come up with concrete bounds and concrete AMONG implied constraints. The notation system used for these characteristics was described in [6]. Within these tables `Bump`, `Dec`, `DecSeq`, `DecTer`, `Dip`, `Inc`, `IncSeq`, `IncTer`, `PropPlain`, `PropPlateau`, `SteadySeq`, `SDecSeq`, `SIncSeq` are respectively shortcuts for `BumpOnDecreasingSequence`, `Decreasing`, `DecreasingSequence`, `DecreasingTerrace`, `DipOnIncreasingSequence`, `Increasing`, `IncreasingSequence`, `IncreasingTerrace`, `ProperPlain`, `ProperPlateau`, `SteadySequence`, `StrictlyDecreasingSequence`, `StrictlyIncreasingSequence`.

3.3.1 Big-width

Definition 5. *Consider a regular expression σ, and a time series X of length n over an integer interval domain $[\ell, u]$. The* big width *of σ wrt $\langle \ell, u, n \rangle$, denoted by $\beta_\sigma^{\langle \ell,u,n \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}^+$ to $\mathbb{N}$. It equals the maximum width of a σ-pattern in X. If X cannot have any σ-patterns, then $\beta_\sigma^{\langle \ell,u,n \rangle}$ is 0.*

name σ	illustration	$\beta_\sigma^{\langle \ell,u \rangle}$
`Bump`		$\begin{cases} 3, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Dec`		$\begin{cases} 2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`DecSeq`		$\begin{cases} 2, \text{ if } u-\ell = \eta_\sigma \\ n, \text{ if } u-\ell > \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`DecTer`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Dip`		$\begin{cases} 3, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Gorge`		$\begin{cases} 1, \text{ if } u-\ell = \eta_\sigma \\ n-2, \text{ if } u-\ell > \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Inc`		$\begin{cases} 2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`IncSeq`		$\begin{cases} 2, \text{ if } u-\ell = \eta_\sigma \\ n, \text{ if } u-\ell > \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`IncTer`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Inflexion`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Peak`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$

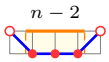
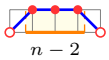
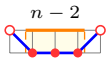
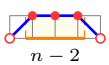
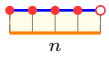
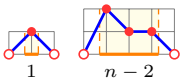
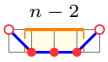
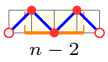
`Plain`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Plateau`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`PropPlain`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`PropPlateau`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Steady`		2
`SteadySeq`		n
`SDecSeq`		$\begin{cases} u-\ell+1, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`SIncSeq`		$\begin{cases} u-\ell+1, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Summit`		$\begin{cases} 1, \text{ if } u-\ell = \eta_\sigma \\ n-2, \text{ if } u-\ell > \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Valley`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$
`Zigzag`		$\begin{cases} n-2, \text{ if } u-\ell \geq \eta_\sigma \\ 0, \text{ otherwise} \end{cases}$

Table 3.50: Regular-expression short names σ and corresponding *big-width* (see Definition 5) shown as thick orange horizontal line segments, where η_σ stands for the *height* characteristics of the corresponding σ (see Definition 8)

3.3.2 Height

Definition 6. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* set of supporting time series *of a word v in $\mathcal{L}_\sigma$ wrt $\langle \ell, u \rangle$, denoted by $\Omega_\sigma^{\langle \ell, u \rangle}(v)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \Sigma^*$ to $\mathbb{Z}^*$. Each element of $\Omega_\sigma^{\langle \ell, u \rangle}(v)$ is a time series over $[\ell, u]$ whose signature is v, and is called a* supporting time series of v wrt $\langle \ell, u \rangle$.

Definition 7. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* height *of a word v in $\mathcal{L}_\sigma$, denoted by $\eta_\sigma(v)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \Sigma^*$ to $\mathbb{N}$. It is defined by $\eta_\sigma(v) = \min_{\Omega_\sigma^{\langle \ell, u \rangle}(v) \neq \emptyset} (u - \ell)$.*

Definition 8. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* height *of σ, denoted by η_σ, is a function that maps an element of $\mathcal{R}_\Sigma$ to $\mathbb{N}$. It is defined by $\eta_\sigma = \min_{v \in \mathcal{L}_\sigma} \eta_\sigma(v)$.*

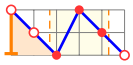
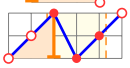
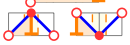
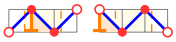
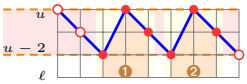
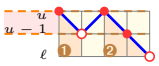
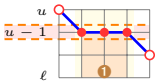
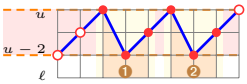
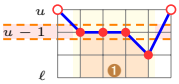
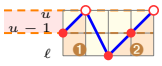
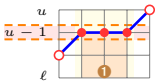
name σ	illustration	η_σ
`Bump`		2
`Dec`		1
`DecSeq`		1
`DecTer`		2
`Dip`		2
`Gorge`		1
`Inc`		1
`IncSeq`		1
`IncTer`		2
`Inflexion`		1
`Peak`		1
`Plain`		1
`Plateau`		1
`PropPlain`		1
`PropPlateau`		1
`Steady`		0
`SteadySeq`		0
`SDecSeq`		1
`SIncSeq`		1
`Summit`		1
`Valley`		1
`Zigzag`		1

Table 3.51: Regular-expression short names σ and corresponding *height* shown as thick orange vertical line segments (see Definition 8)

3.3.3 Interval of interest

Definition 9. *Given a $g\_f\_\sigma(X, N)$ time-series constraint with X being a time series over an integer interval domain $[\ell, u]$. The* interval of interest *of $\langle g, f, \sigma\rangle$ wrt $\langle \ell, u\rangle$, denoted by $\mathcal{I}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$, is a function that maps an element of $\mathcal{T}\times\mathbb{Z}\times\mathbb{Z}$ to $\mathbb{Z}\times\mathbb{Z}$, where $\mathcal{T}$ denotes the set of all time-series constraints, and the result pair of integers is considered as an interval. If $u-\ell<\eta_\sigma$, then the interval of interest of $\langle g, f, \sigma\rangle$ wrt $\langle \ell, u\rangle$ is undefined. If $u-\ell\geq\eta_\sigma$, then*

- *The upper limit of $\mathcal{I}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$, denoted by $\overline{\mathcal{I}}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$, is the largest value in $[\ell, u]$ that can occur in a σ-pattern of a time series over $[\ell, u]$.*
- *The lower limit of $\mathcal{I}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$, denoted by $\underline{\mathcal{I}}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$, is the smallest value v in $[\max(\ell, u-\eta_\sigma-1), u]$ such that for any n in $\mathbb{N}$, the number of occurrences of v in the union of the σ-patterns of any maximal time series for g_f_σ of length n over $[\ell, u]$, is a non-constant function of n. If such v does not exist, then $\underline{\mathcal{I}}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$ equals $\overline{\mathcal{I}}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}-\eta_\sigma$.*

name σ	illustration	$\overline{\mathcal{I}}^{\langle \ell,u\rangle}_{\langle g,f,\sigma\rangle}$
`Bump`		$\begin{cases}[u-2, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`Dec`		$\begin{cases}[u-1, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`DecSeq`		$\begin{cases}[u-1, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`DecTer`		$\begin{cases}[u-1, u-1], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`Dip`		$\begin{cases}[u-2, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`Gorge`		$\begin{cases}[u-1, u-1], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`Inc`		$\begin{cases}[u-1, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`IncSeq`		$\begin{cases}[u-1, u], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$
`IncTer`		$\begin{cases}[u-1, u-1], \text{ if } u-\ell\geq\eta_\sigma\\ \text{undefined, otherwise}\end{cases}$

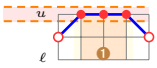
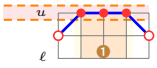
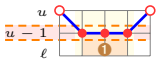
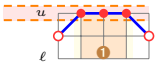
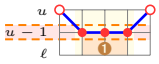
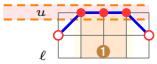
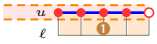
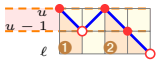
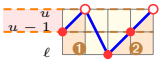
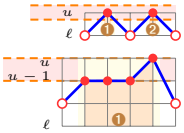
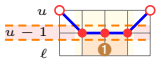
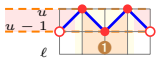
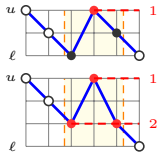
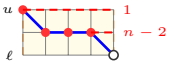
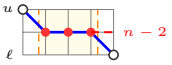
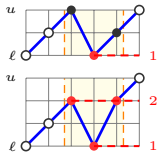
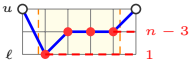
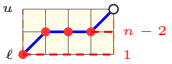
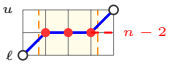
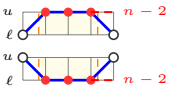
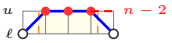
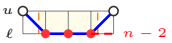
`Inflexion`		$\begin{cases}[u,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Peak`		$\begin{cases}[u,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Plain`		$\begin{cases}[u-1,u-1], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Plateau`		$\begin{cases}[u,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`PropPlain`		$\begin{cases}[u-1,u-1], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`PropPlateau`		$\begin{cases}[u,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Steady`		$\begin{cases}[u,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`SteadySeq`		$\begin{cases}[u-1,u], \text{ if } u-\ell \geq \eta_\sigma \wedge g = \texttt{Sum} \\ [u,u], \text{ if } u-\ell \geq \eta_\sigma \wedge (g = \texttt{Max} \vee g = \texttt{Min}) \\ \text{undefined, otherwise}\end{cases}$
`SDecSeq`		$\begin{cases}[u-1,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`SIncSeq`		$\begin{cases}[u-1,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Summit`		$\begin{cases}[u-1,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Valley`		$\begin{cases}[u-1,u-1], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$
`Zigzag`		$\begin{cases}[u-1,u], \text{ if } u-\ell \geq \eta_\sigma \\ \text{undefined, otherwise}\end{cases}$

Table 3.52: Regular-expression short names σ and corresponding *interval of interests* (see Definition 9) of $\langle g, f, \sigma \rangle$ wrt $\langle \ell, u \rangle$, where η_σ stands for the *height* characteristics of the corresponding σ (see Definition 8); ℓ and u resp. stands for the minimum and maximum value of the variables of the time series.

3.3.4 Maximum-value-occurrence-number

Definition 10. *Consider a regular expression σ, and a time series X of length n over an integer interval domain $[\ell, u]$. The* maximum value occurrence number *of v in $\mathbb{Z}$ wrt $\langle \ell, u, n \rangle$, denoted by $\mu_\sigma^{\langle \ell,u,n \rangle}(v)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}^+ \times \mathbb{Z}$ to $\mathbb{N}$. It equals the maximum number of occurrences of the value v in one σ-pattern of X.*

name σ	illustration	$\mu_\sigma^{\langle \ell,u,n \rangle}(v)$
`Bump`		$\begin{cases} 1, \text{ if } u-\ell = \eta_\sigma \\ 1, \text{ if } u-\ell > \eta_\sigma \wedge v \in \{\ell, u-1, u\} \\ 2, \text{ if } u-\ell > \eta_\sigma \wedge v \in [\ell+1, u-2] \end{cases}$
`Dec`		$1, \forall v \in [\ell, u]$
`DecSeq`		$\begin{cases} 1, \text{ if } v \in \{\ell, u\} \\ n-2, \text{ if } v \in [\ell+1, u-1] \end{cases}$
`DecTer`		$\begin{cases} 0, \text{ if } v \in \{\ell, u\} \\ n-2, \text{ if } v \in [\ell+1, u-1] \end{cases}$
`Dip`		$\begin{cases} 1, \text{ if } u-\ell = \eta_\sigma \\ 1, \text{ if } u-\ell > \eta_\sigma \wedge v \in \{\ell, \ell+1, u\} \\ 2, \text{ if } u-\ell > \eta_\sigma \wedge v \in [\ell+2, u-1] \end{cases}$
`Gorge`		$\begin{cases} 0, \text{ if } v = u \\ n-3, \text{ if } v \in [\ell+1, u-1] \\ 1, \text{ if } v = \ell \end{cases}$
`Inc`		$1, \forall v \in [\ell, u]$
`IncSeq`		$\begin{cases} 1, \text{ if } v \in \{\ell, u\} \\ n-2, \text{ if } v \in [\ell+1, u-1] \end{cases}$
`IncTer`		$\begin{cases} 0, \text{ if } v \in \{\ell, u\} \\ n-2, \text{ if } v \in [\ell+1, u-1] \end{cases}$
`Inflexion`		$n-2, \ \forall v \in [\ell, u]$
`Peak`		$\begin{cases} 0, \text{ if } v = \ell \\ n-2, \text{ if } v \in [\ell+1, u] \end{cases}$
`Plain`		$\begin{cases} 0, \text{ if } v = u \\ n-2, \text{ if } v \in [\ell, u-1] \end{cases}$

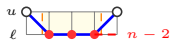
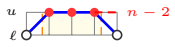
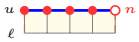
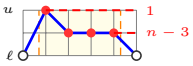
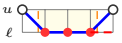
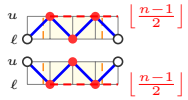
`Plateau`		$\begin{cases} 0, \text{ if } v = \ell \\ n-2, \text{ if } v \in [\ell+1, u] \end{cases}$
`PropPlain`		$\begin{cases} 0, \text{ if } v = u \\ n-2, \text{ if } v \in [\ell, u-1] \end{cases}$
`PropPlateau`		$\begin{cases} 0, \text{ if } v = \ell \\ n-2, \forall v \in [\ell+1, u] \end{cases}$
`Steady`		$2, \ \forall v \in [\ell, u]$
`SteadySeq`		$n, \ \forall v \in [\ell, u]$
`SDecSeq`		$1, \ \forall v \in [\ell, u]$
`SIncSeq`		$1, \ \forall v \in [\ell, u]$
`Summit`		$\begin{cases} 1, \text{ if } v = u \\ n-3, \text{ if } v \in [\ell+1, u-1] \\ 0, \text{ if } v = \ell \end{cases}$
`Valley`		$\begin{cases} 0, \text{ if } v = u \\ n-2, \text{ if } v \in [\ell, u-1] \end{cases}$
`Zigzag`		$\left\lfloor \frac{n-1}{2} \right\rfloor, \ \forall v \in [\ell, u]$

Table 3.53: Regular-expression short names σ and corresponding *maximum value occurrence number* of a value v (see Definition 10); for each pattern σ it assumes that the range of possible values is big enough to have at least one occurrence of pattern, i.e. $u - \ell \geq \eta_\sigma$ where u and ℓ are the largest and smallest value that can be used.

3.3.5 Overlap

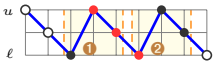
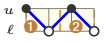
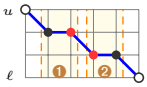
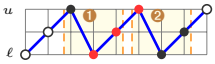
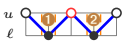
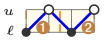
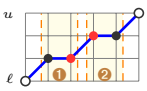
Definition 11. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* set of superpositions *of two words, v and w in $\mathcal{L}_\sigma$, wrt $\langle \ell, u \rangle$, denoted by $\Gamma_\sigma^{\langle \ell,u \rangle}(v, w)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \Sigma^* \times \Sigma^*$ to $\mathcal{P}(\Sigma^*)$, where $\mathcal{P}(\Sigma^*)$ is the power set of Σ^*. Each element z in $\Gamma_\sigma^{\langle \ell,u \rangle}(v, w)$ is a word over Σ, called a superposition of v and w wrt $\langle \ell, u \rangle$ and satisfying all the following conditions*

(i) $z \notin \mathcal{L}_\sigma$, (ii) $\Omega_\sigma^{\langle \ell,u \rangle}(z) \neq \emptyset$, (iii) v is a prefix of z, (iv) w is a suffix of z, (v) $|z| \leq |vw|$.

Definition 12. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* overlap *of two words, v and w in $\mathcal{L}_\sigma$, wrt $\langle \ell, u \rangle$, denoted by $o_\sigma^{\langle \ell,u \rangle}(v, w)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \Sigma^* \times \Sigma^*$ to $\mathbb{N}$. It is defined by*

$$o_\sigma^{\langle \ell,u \rangle}(v, w) = \begin{cases} \left(|vw| - \min_{z \in \Gamma_\sigma^{\langle \ell,u \rangle}(v,w)} |z| \right) + 1 & \text{if } \Gamma_\sigma^{\langle \ell,u \rangle}(v, w) \neq \emptyset \quad (3.1) \\ 0, & \text{otherwise.} \quad (3.2) \end{cases}$$

Definition 13. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* overlap *of σ wrt $\langle \ell, u \rangle$, denoted by $o_\sigma^{\langle \ell,u \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z}$ to $\mathbb{N}$. If there exists a constant c in $\mathbb{N}$ such that for any pair words v, w in $\mathcal{L}_\sigma$, the value of $o_\sigma^{\langle \ell,u \rangle}(v, w)$ is bounded by c, then the overlap of σ wrt $\langle \ell, u \rangle$ is defined by $o_\sigma^{\langle \ell,u \rangle} = \max_{v,w \in \mathcal{L}_\sigma} o_\sigma^{\langle \ell,u \rangle}(v, w)$. Otherwise, $o_\sigma^{\langle \ell,u \rangle}$ is not defined.*

name σ	illustration	$o_\sigma^{\langle \ell,u \rangle}$
`Bump`		3
`Dec`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 1 & \text{otherwise} \end{cases}$
`DecSeq`		0
`DecTer`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 2 & \text{otherwise} \end{cases}$
`Dip`		3
`Gorge`		1
`Inc`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 1 & \text{otherwise} \end{cases}$
`IncSeq`		0
`IncTer`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 2 & \text{otherwise} \end{cases}$

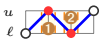
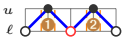
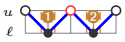
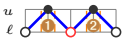
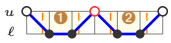
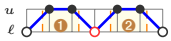
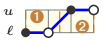
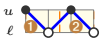
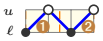
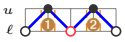
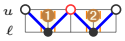
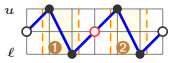
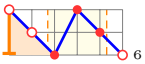
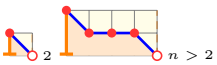
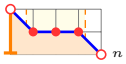
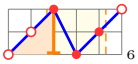
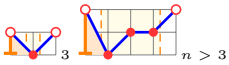
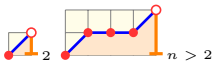
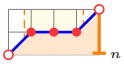
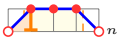
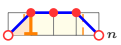
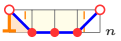
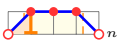
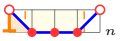
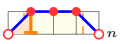
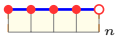
`Inflexion`		2
`Peak`		1
`Plain`		1
`Plateau`		1
`PropPlain`		1
`PropPlateau`		1
`Steady`		1
`SteadySeq`		0
`SDecSeq`		0
`SIncSeq`		0
`Summit`		1
`Valley`		1
`Zigzag`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 1 & \text{otherwise} \end{cases}$

Table 3.54: Regular-expression short names σ and corresponding *overlap* between two consecutive pattern occurrences ❶ and ❷ illustrated in red, i.e., ● or ○ (see Definition 13), where η_σ stands for the *height* characteristics of the corresponding σ (see Definition 8)

3.3.6 Range

Definition 14. *Consider a regular expression σ and a time series length n. The* range *of σ wrt $\langle n \rangle$, denoted by $\phi_\sigma^{\langle n \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{N}^+$ to $\mathbb{N}$. It is defined by $\phi_\sigma^{\langle n \rangle} = \min_{v \in \mathcal{L}_\sigma,\ |v|=n-1} \eta_\sigma(v)$, where $\eta_\sigma(v)$ is the height of the word v.*

If $\mathcal{L}_\sigma$ does not contain any word of length $n-1$, then the value of $\phi_\sigma^{\langle n \rangle}$ is undefined.

name σ	$\langle e_\sigma, c_\sigma \rangle$	illustration	$\phi_\sigma^{\langle n \rangle}$
`Bump`	undefined		$\begin{cases} 2 & \text{if } n = \omega_\sigma + 1 \\ \text{undefined} & \text{otherwise} \end{cases}$
`Dec`	undefined		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ \text{undefined} & \text{otherwise} \end{cases}$
`DecSeq`	$\langle 0, 1 \rangle$		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ 2 & \text{if } n > \omega_\sigma + 1 \end{cases}$
`DecTer`	$\langle 0, 0 \rangle$		2
`Dip`	undefined		$\begin{cases} 2 & \text{if } n = \omega_\sigma + 1 \\ \text{undefined} & \text{otherwise} \end{cases}$
`Gorge`	$\langle 0, 1 \rangle$		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ 2 & \text{if } n > \omega_\sigma + 1 \end{cases}$
`Inc`	undefined		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ \text{undefined} & \text{otherwise} \end{cases}$
`IncSeq`	$\langle 0, 1 \rangle$		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ 2 & \text{if } n > \omega_\sigma + 1 \end{cases}$
`IncTer`	$\langle 0, 0 \rangle$		2
`Inflexion`	$\langle 0, 0 \rangle$		1
`Peak`	$\langle 0, 0 \rangle$		1
`Plain`	$\langle 0, 0 \rangle$		1
`Plateau`	$\langle 0, 0 \rangle$		1
`PropPlain`	$\langle 0, 0 \rangle$		1
`PropPlateau`	$\langle 0, 0 \rangle$		1
`Steady`	undefined		$\begin{cases} 0 & \text{if } n = \omega_\sigma + 1 \\ \text{undefined} & \text{otherwise} \end{cases}$
`SteadySeq`	$\langle 0, 0 \rangle$		0
`SDecSeq`	$\langle 1, 0 \rangle$		$n-1$

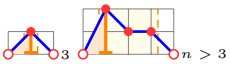
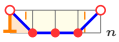
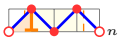
`SIncSeq`	$\langle 1, 0\rangle$		$n-1$
`Summit`	$\langle 0, 1\rangle$		$\begin{cases} 1 & \text{if } n = \omega_\sigma + 1 \\ 2 & \text{if } n > \omega_\sigma + 1 \end{cases}$
`Valley`	$\langle 0, 0\rangle$		1
`Zigzag`	$\langle 0, 0\rangle$		1

Table 3.55: Regular-expression short names σ and corresponding *range* shown as thick orange vertical line segments (see Definition 14); for a non-fixed-length regular expression σ and for any $n > \omega_\sigma+1$, $\phi_\sigma^{\langle n\rangle} = e_\sigma\cdot(n-1-\eta_\sigma)+c_\sigma+\eta_\sigma$, where ω_σ and η_σ respectively correspond to the *size* (see Definition 18) and the *height* (see Definition 8) of the corresponding σ.

3.3.7 Set-of-inducing-words

Definition 15. *Consider a disjunction-capsuled regular expression σ. The (unique) non-empty shortest word of $\mathcal{L}_\sigma$ is the* inducing word *of $\mathcal{L}_\sigma$.*

Definition 16. *Consider a regular expression σ* that is in the form of $\sigma = \sigma_1|\ldots|\sigma_t$ with $t \geq 1$, where every σ_i (with $i \in [1,t]$) is a disjunction-capsuled regular expression. *The* set of inducing words *of σ, denoted by Θ_σ, is a function that maps an element of $\mathcal{R}_\Sigma$ to $\mathcal{P}(\Sigma^*)$, where $\mathcal{P}(\Sigma^*)$ is the power set of Σ^*. Each element v of Θ_σ is a word, called an* inducing word *of σ such that all the following conditions are satisfied*

(i) v is in $\{w_1,\ldots,w_t\}$, where w_i is the inducing word of σ_i.

(ii) No word in $\{w_1,w_2,\ldots,w_t\}$ is a proper factor of v.

name σ	regular expression	Θ_σ
Bump	‘ >><>> ’	{‘ >><>> ’}
Dec	‘ > ’	{‘ > ’}
DecSeq	‘(> (> | =)*)* > ’	{‘ > ’}
DecTer	‘ >= =* > ’	{‘ >=> ’}
Dip	‘ <<>>< ’	{‘ <<>>< ’}
Gorge	‘(> (> | =)*)* >< ((< | =)* <)*’	{‘ >< ’}
Inc	‘ < ’	{‘ < ’}
IncSeq	‘(< (< | =)*)* < ’	{‘ < ’}
IncTer	‘ <= =* < ’	{‘ <=< ’}
Inflexion	‘ < (< | =)* > | > (> | =)* < ’	{‘ <> ’, ‘ >< ’}
Peak	‘ < (< | =)*(> | =)* > ’	{‘ <> ’}
Plain	‘ > =* < ’	{‘ >< ’}
Plateau	‘ < =* > ’	{‘ <> ’}
PropPlain	‘ >= =* < ’	{‘ >=< ’}
PropPlateau	‘ <= =* > ’	{‘ <=> ’}
Steady	‘ = ’	{‘ = ’}
SteadySeq	‘ = =* ’	{‘ = ’}
SDecSeq	‘ > >* ’	{‘ > ’}
SIncSeq	‘ < <* ’	{‘ < ’}
Summit	‘(< (< | =)*)* <> ((> | =)* >)*’	{‘ <> ’}
Valley	‘ > (> | =)*(< | =)* < ’	{‘ >< ’}
Zigzag	‘(<>)* <>< (> |ε) | (><)* ><> (< |ε)’	{‘ <>< ’, ‘ ><> ’}

Table 3.56: Regular-expression short names σ and corresponding *inducing words* (see Definition 16)

3.3.8 Shift

Definition 17. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* shift *of σ, denoted by $\overline{\delta}_\sigma$, is a function that maps an element of $\mathcal{R}_\Sigma$ to $\mathbb{N}$. It is defined by $\overline{\delta}_\sigma = \min_{v \in \mathcal{L}_\sigma} \min_{t \in \Omega_\sigma^{\langle 0, |v| \rangle}(v)} \max_{i \in [1+b_\sigma, |t|-a_\sigma]} |v| - t_i$, where $\Omega_\sigma^{\langle 0, |v| \rangle}(v)$ is the set of supporting time series of v wrt $[0, |v|]$, and $|t|$ is the length of the time series t.*

name σ	$\underline{\nu}_\sigma$
Bump	2
Dec	1
DecSeq	1
DecTer	1
Dip	2
Gorge	1
Inc	1
IncSeq	1
IncTer	1
Inflexion	0
Peak	0
Plain	1
Plateau	0
PropPlain	1
PropPlateau	0
Steady	0
SteadySeq	0
SDecSeq	1
SIncSeq	1
Summit	0
Valley	1
Zigzag	1

Table 3.57: Regular-expression short names σ and corresponding *shift* of σ (see Definitions 17)

3.3.9 Size

Definition 18. *Consider a regular expression σ. The* size *of σ, denoted by ω_σ, is a function that maps an element of $\mathcal{R}_\Sigma$ to $\mathbb{N}^+$. It is defined by $\omega_\sigma = \min_{v \in \mathcal{L}_\sigma} |v|$.*

name σ	regular expression	ω_σ
Bump	' >><>> '	5
Dec	' > '	1
DecSeq	'(> (> | =)*)* > '	1
DecTer	' >= =* > '	3
Dip	' <<><< '	5
Gorge	'(> | > (> | =)* >)(< | < (< | =)* <)'	2
Inc	' < '	1
IncSeq	'(< (< | =)*)* < '	1
IncTer	' <= =* < '	3
Inflexion	' < (< | =)* > | > (> | =)* < '	2
Peak	' < (< | =)*(> | =)* > '	2
Plain	' > =* < '	2
Plateau	' < =* > '	2
PropPlain	' >= =* < '	3
PropPlateau	' <= =* > '	3
Steady	' = '	1
SteadySeq	' = =* '	1
SDecSeq	' > >* '	1
SIncSeq	' < <* '	1
Summit	'(< | < (< | =)* <)(> | > (> | =)* >)'	2
Valley	' > (> | =)*(< | =)* < '	2
Zigzag	'(<>)* <>< (> |ε) | (><)* ><> (< |ε)'	3

Table 3.58: Regular-expression short names σ and corresponding *size* (see Definition 18); within each regular expression subparts corresponding to a smallest length word are highlighted in yellow.

3.3.10 Smallest-variation-of-maxima

Definition 19. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* shift *of a proper factor w in a word v in $\mathcal{L}_\sigma$ wrt $\langle \ell, u \rangle$, denoted by $\overline{\nu}_\sigma^{\langle \ell,u \rangle}(v, w, i)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \Sigma^* \times \Sigma^* \times \mathbb{N}$ to $\mathbb{N}$. It is defined by*

$$\overline{\nu}_\sigma^{\langle \ell,u \rangle}(v, w, i) = \min_{t \in \Omega_\sigma^{\langle \ell,u \rangle}(v)} \min_{x \in t_{w_i}} (\max(t) - x),$$

where $\max(t)$ is the maximum value of a time series t, a supporting time series of v wrt $\langle \ell, u \rangle$, and t_{w_i} is a subseries of t corresponding to the i^{th} extended σ-pattern whose signature is w. If w is not a proper factor of v, or i is strictly greater than the number of occurrences of w in v, then $\overline{\nu}_\sigma^{\langle \ell,u \rangle}(v, w, i)$ is undefined.

Definition 20. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* smallest variation of maxima *of superpositions of two words w and v in $\mathcal{L}_\sigma$ wrt $\langle \ell, u \rangle$, denoted by $\delta_\sigma^{\langle \ell,u \rangle}(v, w)$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z} \times \Sigma^* \times \Sigma^*$ to $\mathbb{N}$. It is defined by*

$$\delta_\sigma^{\langle \ell,u \rangle}(v, w) = \begin{cases} \overline{\nu}_\sigma(z_*, v, 1) - \overline{\nu}_\sigma(z_*, w, 1), & \text{if } v \neq w \text{ and } \Gamma_\sigma^{\langle \ell,u \rangle}(v, w) \neq \emptyset \\ \overline{\nu}_\sigma(z_{**}, v, 1) - \overline{\nu}_\sigma(z_{**}, w, 2), & \text{if } v = w \text{ and } \Gamma_\sigma^{\langle \ell,u \rangle}(v, w) \neq \emptyset \\ 0, & \text{if } \Gamma_\sigma^{\langle \ell,u \rangle}(v, w) = \emptyset \end{cases}$$

where the words z_* and z_{**} both belongs to $\Gamma_\sigma^{\langle \ell,u \rangle}(v, w)$, and the value $\min_{z \in \Gamma_\sigma^{\langle \ell,u \rangle}(v,w)} |\overline{\nu}_\sigma(z, v, 1) - \overline{\nu}_\sigma(z, w, 1)|$ (respectively $\min_{z \in \Gamma_\sigma^{\langle \ell,u \rangle}(v,w)} |\overline{\nu}_\sigma(z, v, 1) - \overline{\nu}_\sigma(z, w, 2)|$) is reached when z is z_* (respectively z_{**}).

Definition 21. *Consider a regular expression σ and an integer interval domain $[\ell, u]$. The* smallest variation of maxima *of σ wrt $\langle \ell, u \rangle$, denoted by $\delta_\sigma^{\langle \ell,u \rangle}$, is a function that maps an element of $\mathcal{R}_\Sigma \times \mathbb{Z} \times \mathbb{Z}$ to $\mathbb{N}$. It is defined by*

$$\delta_\sigma^{\langle \ell,u \rangle} = \begin{cases} \delta_\sigma^{\langle \ell,u \rangle}(v_*, w_*), & \text{if } \exists v, w \in \mathcal{L}_\sigma \text{ such that } \delta_\sigma^{\langle \ell,u \rangle}(v, w) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

where the words v_* and w_* both belong to $\mathcal{L}_\sigma$ and the value $\min_{v,w \in \mathcal{L}_\sigma} |\delta_\sigma^{\langle \ell,u \rangle}(v, w)|$ is reached when v is v_* and w is w_*.

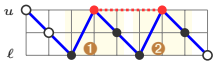
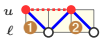
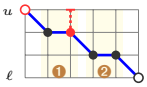
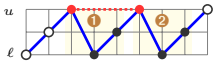
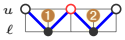
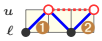
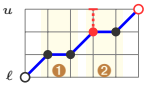
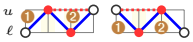
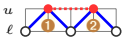
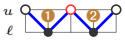
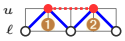
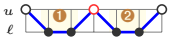
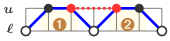
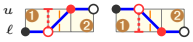
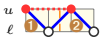
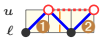
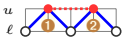
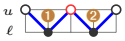
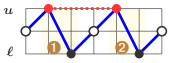
name σ	illustration	$\delta_\sigma^{\langle \ell, u \rangle}$
`Bump`		0
`Dec`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ -1 & \text{otherwise} \end{cases}$
`DecSeq`		0
`DecTer`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ -1 & \text{otherwise} \end{cases}$
`Dip`		0
`Gorge`		0
`Inc`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 1 & \text{otherwise} \end{cases}$
`IncSeq`		0
`IncTer`		$\begin{cases} 0 & \text{if } u - \ell \leq \eta_\sigma \\ 1 & \text{otherwise} \end{cases}$
`Inflexion`		0
`Peak`		0
`Plain`		0
`Plateau`		0
`PropPlain`		0
`PropPlateau`		0
`Steady`		0
`SteadySeq`		0
`SDecSeq`		0
`SIncSeq`		0
`Summit`		0
`Valley`		0
`Zigzag`		0

Table 3.59: Regular-expression short names σ and corresponding *smallest variation of maxima* (see Definition 21), where η_σ stands for the *height* characteristics of the corresponding σ (see Definition 8); maxima of two consecutive pattern occurrences ❶ and ❷ are shown in red, i.e., ● or ○.

4

Global Constraint Catalogue

«Mon problème, avec les classements, c'est qu'ils ne durent pas ; à peine ai-je fini de mettre de l'ordre que cet ordre est déjà caduc.»

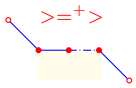
– Georges Perec, *Penser/Classer*

Contents

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_DECREASING_TERRACE

► ▷ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint ALL_EQUAL_HEIGHT_DECREASING_TERRACE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 6,3,3,1,5,4,3,3,2,4,4,5,3,3,1,1\rangle)$

Figure 4.1 provides an example where the ALL_EQUAL_HEIGHT_DECREASING_TERRACE $([6,3,3,1,5,4,3,3,2,4,4,5,3,3,1,1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

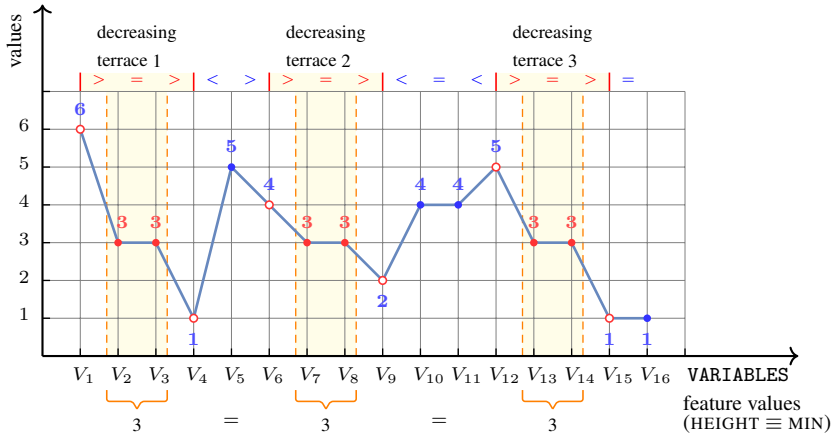


Figure 4.1: Illustrating the ALL_EQUAL_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.2 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_DECREASING_TERRACE.

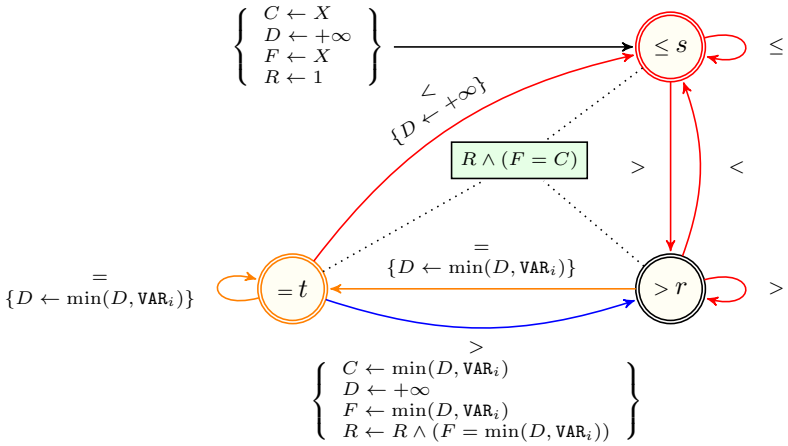
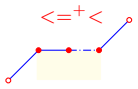


Figure 4.2: Automaton for the ALL_EQUAL_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_INCREASING_TERRACE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint ALL_EQUAL_HEIGHT_INCREASING_TERRACE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 1,4,4,6,2,3,4,4,5,3,3,2,4,4,6,6\rangle)$

Figure 4.3 provides an example where the ALL_EQUAL_HEIGHT_INCREASING_TERRACE $([1,4,4,6,2,3,4,4,5,3,3,2,4,4,6,6])$ constraint holds.

Typical
$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

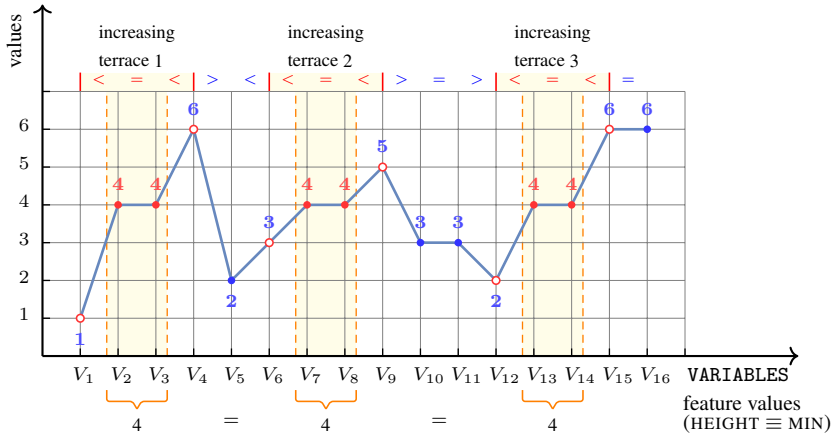


Figure 4.3: Illustrating the ALL_EQUAL_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.4 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_INCREASING_TERRACE.

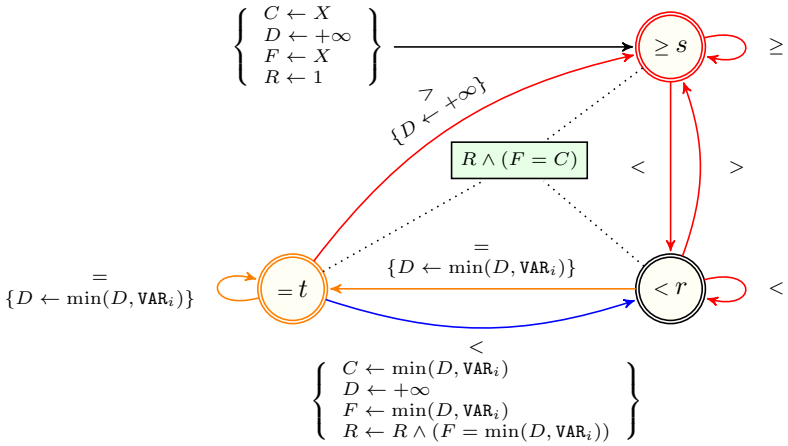


Figure 4.4: Automaton for the ALL_EQUAL_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_PLAIN

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint ALL_EQUAL_HEIGHT_PLAIN(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 1,6,3,3,7,6,6,3,3,5,5,4,3,3,6,3\rangle)$

Figure 4.5 provides an example where the ALL_EQUAL_HEIGHT_PLAIN $([1,6,3,3,7,6,6,3,3,5,5,4,3,3,6,3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

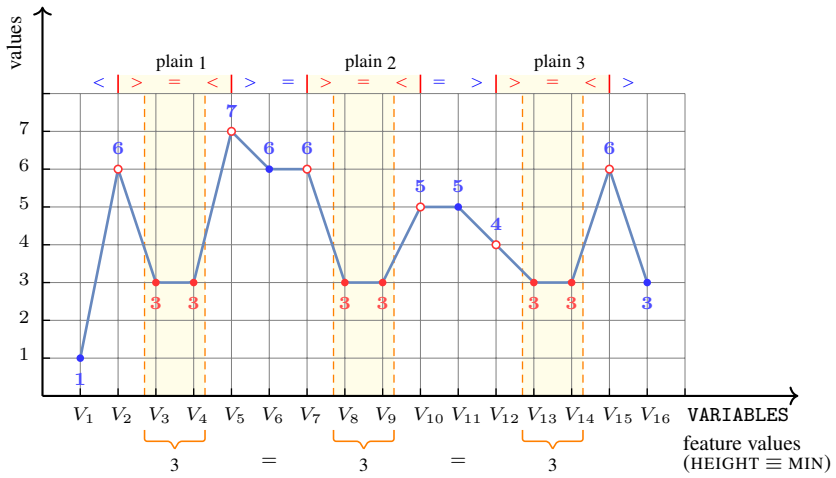


Figure 4.5: Illustrating the ALL_EQUAL_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figure 4.6 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_PLAIN.

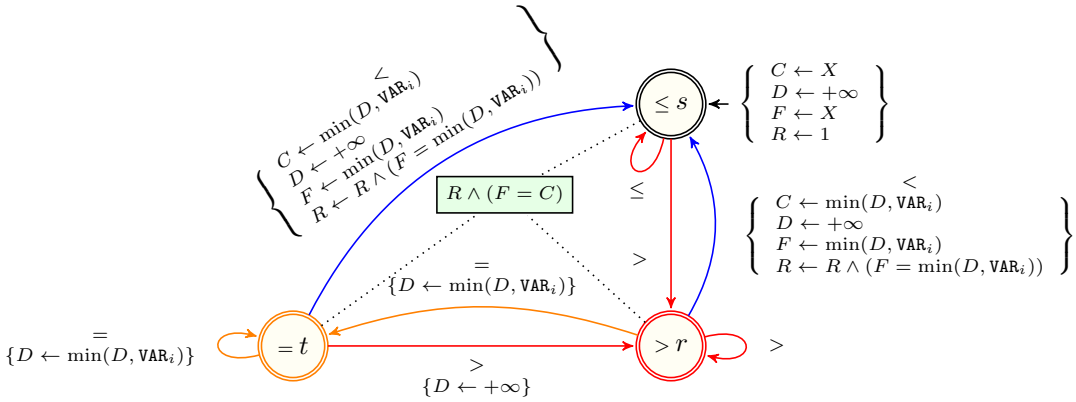


Figure 4.6: Automaton for the ALL_EQUAL_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_PLATEAU

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint ALL_EQUAL_HEIGHT_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,2,5,5,1,2,2,5,5,3,3,4,5,5,2,5\rangle)$

Figure 4.7 provides an example where the ALL_EQUAL_HEIGHT_PLATEAU $([7,2,5,5,1,2,2,5,5,3,3,4,5,5,2,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

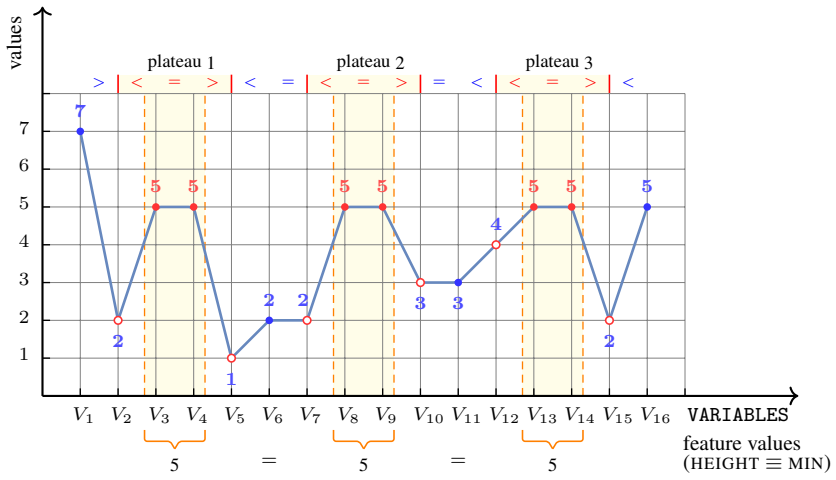


Figure 4.7: Illustrating the ALL_EQUAL_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figure 4.8 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_PLATEAU.

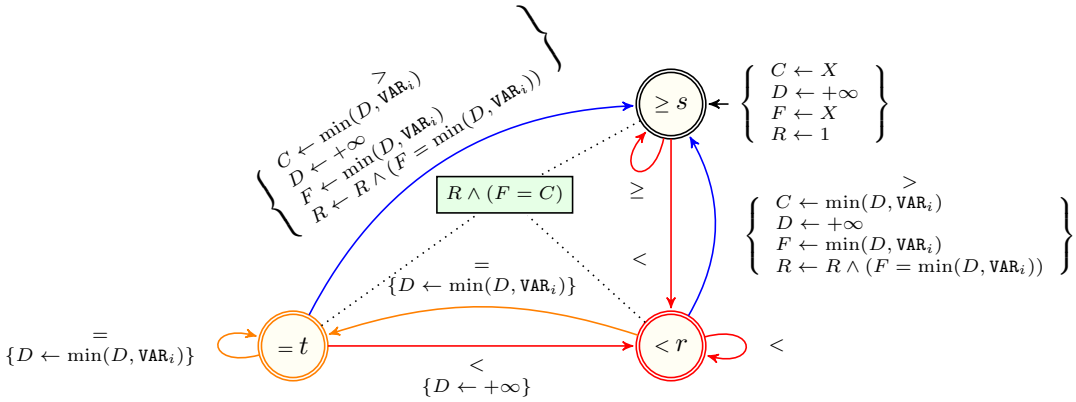
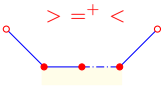


Figure 4.8: Automaton for the ALL_EQUAL_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_PROPER_PLAIN

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLAIN pattern.

Constraint ALL_EQUAL_HEIGHT_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 2,7,3,3,6,6,3,7,3,3,5,6,5,3,3,5\rangle)$

Figure 4.9 provides an example where the ALL_EQUAL_HEIGHT_PROPER_PLAIN $([2,7,3,3,6,6,3,7,3,3,5,6,5,3,3,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

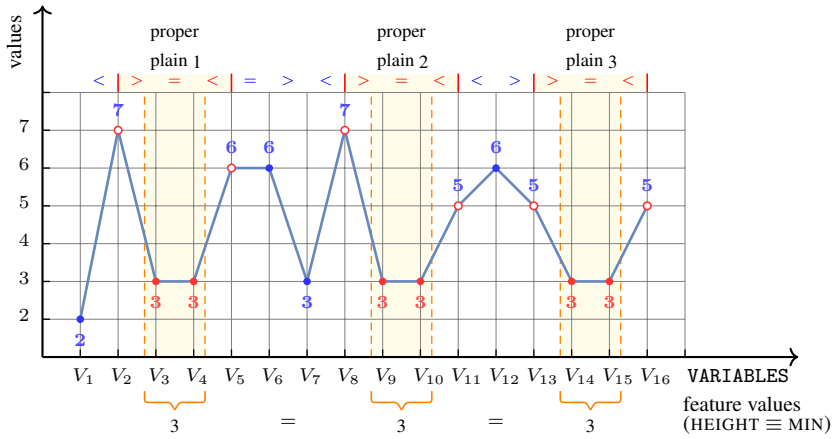


Figure 4.9: Illustrating the ALL_EQUAL_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.10 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_PROPER_PLAIN.

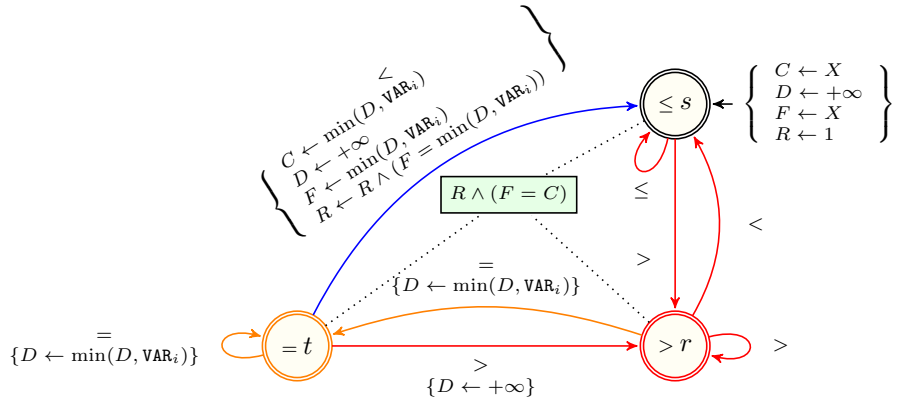
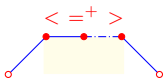


Figure 4.10: Automaton for the ALL_EQUAL_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_PROPER_PLATEAU

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint ALL_EQUAL_HEIGHT_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,1,5,5,2,2,5,1,5,5,3,2,3,5,5,3\rangle)$

Figure 4.11 provides an example where the ALL_EQUAL_HEIGHT_PROPER_PLATEAU $([7,1,5,5,2,2,5,1,5,5,3,2,3,5,5,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

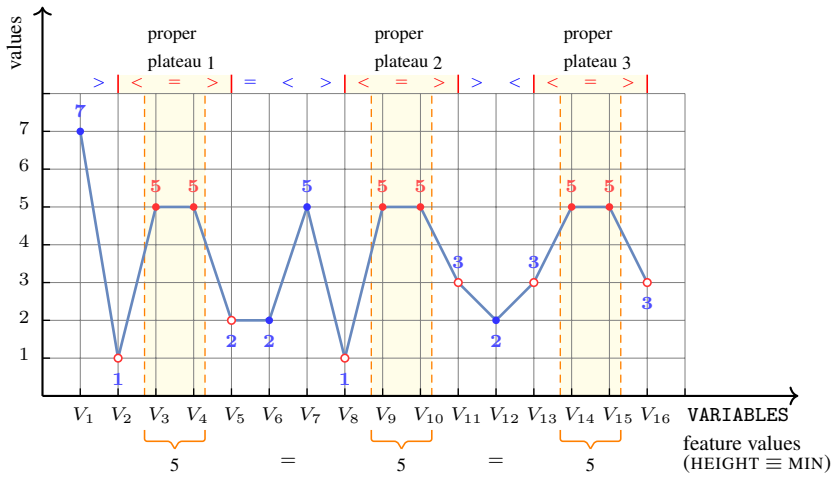


Figure 4.11: Illustrating the ALL_EQUAL_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton Figure 4.12 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_PROPER_PLATEAU.

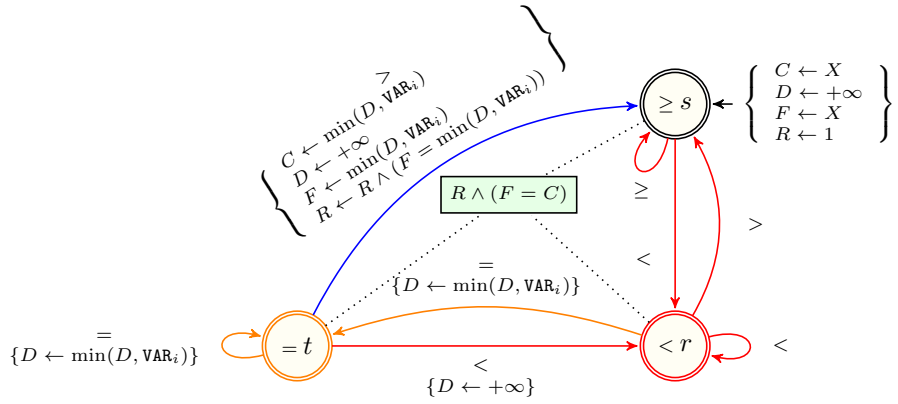


Figure 4.12: Automaton for the ALL_EQUAL_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_STEADY

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint ALL_EQUAL_HEIGHT_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 4, 4, 4, 2, 6, 4, 4, 4, 4, 3, 2, 1, 4, 4, 3, 4\rangle)$

Figure 4.13 provides an example where the ALL_EQUAL_HEIGHT_STEADY $([4, 4, 4, 2, 6, 4, 4, 4, 4, 3, 2, 1, 4, 4, 3, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

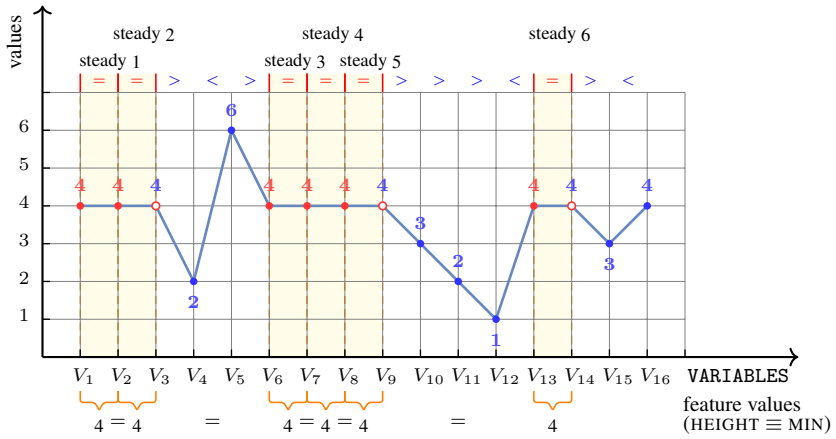


Figure 4.13: Illustrating the ALL_EQUAL_HEIGHT_STEADY constraint of the **Example** slot

Automaton Figure 4.14 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_STEADY.

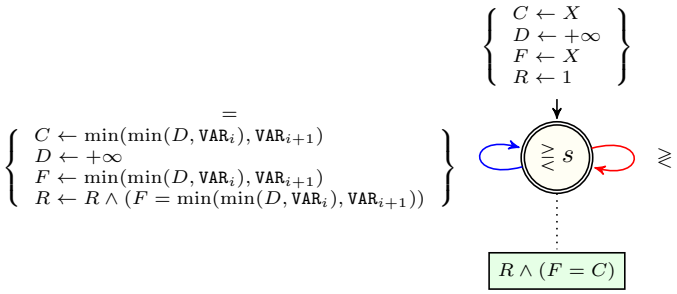
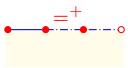


Figure 4.14: Automaton for the ALL_EQUAL_HEIGHT_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_HEIGHT_STEADY_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint ALL_EQUAL_HEIGHT_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,3,5,5,5,3,1,5,5,6,5,5,5,5,3,5\rangle)$

Figure 4.15 provides an example where the ALL_EQUAL_HEIGHT_STEADY_SEQUENCE $([4,3,5,5,5,3,1,5,5,6,5,5,5,5,3,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

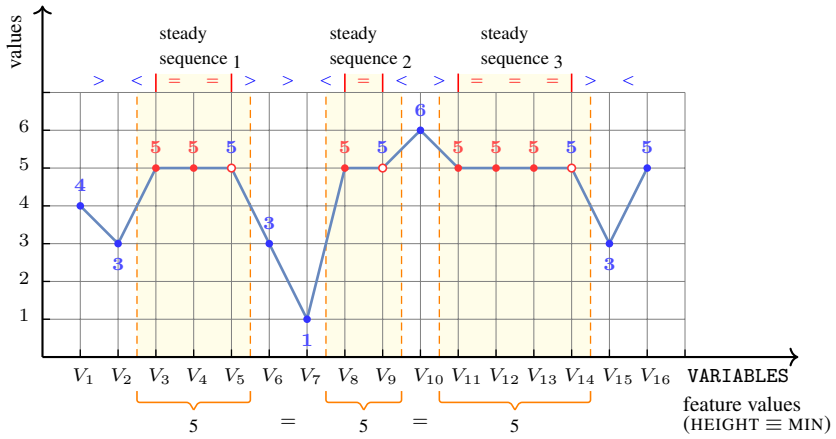


Figure 4.15: Illustrating the ALL_EQUAL_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.16 depicts the automaton associated with the constraint ALL_EQUAL_HEIGHT_STEADY_SEQUENCE.

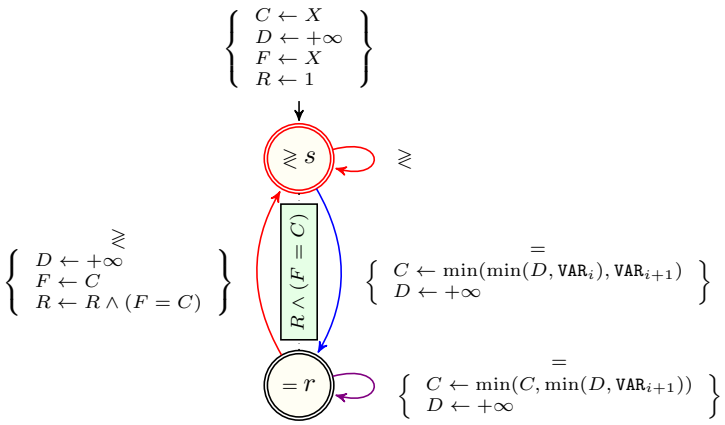
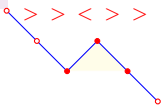


Figure 4.16: Automaton for the ALL_EQUAL_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(\langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 6, 5, 4, 4\rangle)$

Figure 4.17 provides an example where the ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 6, 5, 4, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

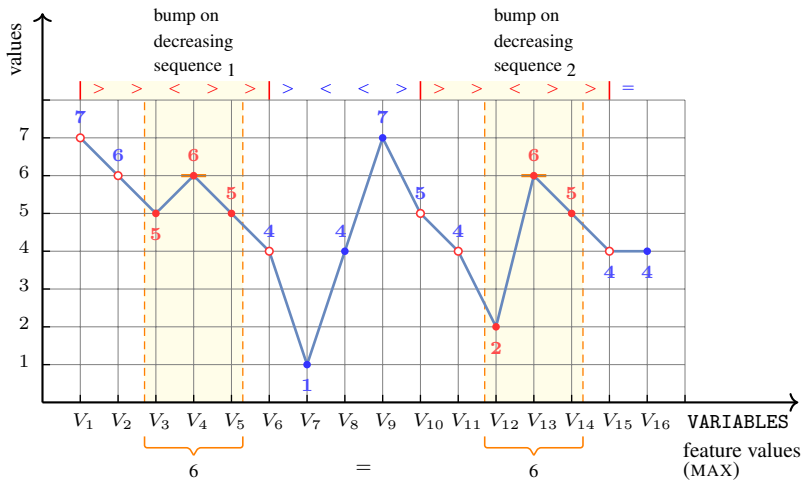


Figure 4.17: Illustrating the ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.18 depicts the automaton associated with the constraint ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE.

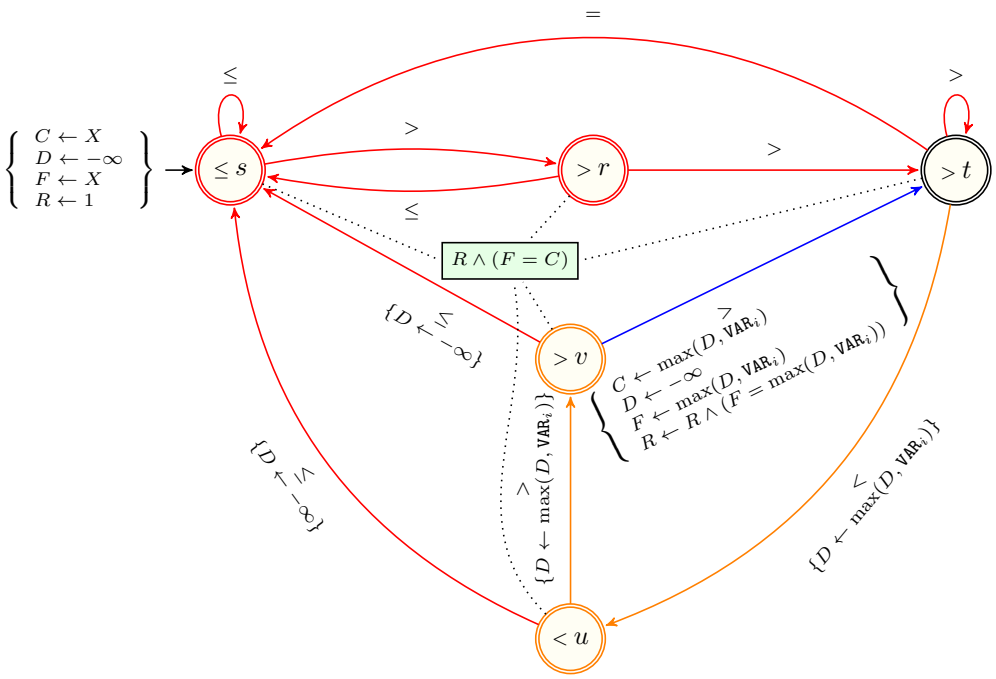


Figure 4.18: Automaton for the ALL_EQUAL_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_DECREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint ALL_EQUAL_MAX_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 1,6,6,1,1,4,6,6,4,4,4,5,5,6,1,5\rangle)$

Figure 4.19 provides an example where the ALL_EQUAL_MAX_DECREASING $([1,6,6,1,1,4,6,6,4,4,4,5,5,6,1,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

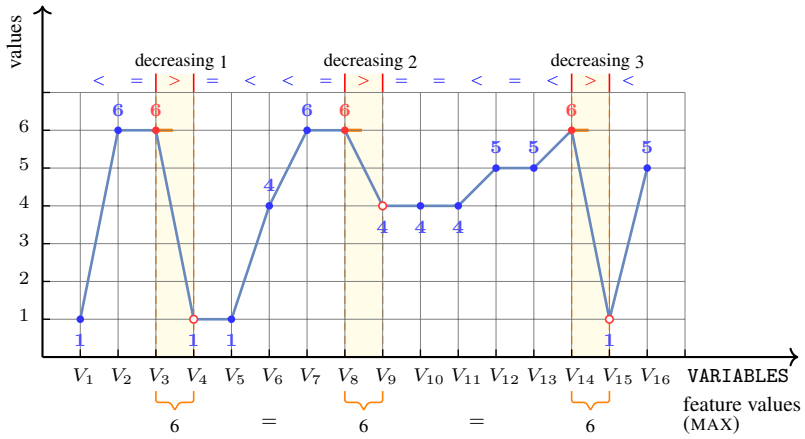


Figure 4.19: Illustrating the ALL_EQUAL_MAX_DECREASING constraint of the **Example** slot

Automaton Figure 4.20 depicts the automaton associated with the constraint ALL_EQUAL_MAX_DECREASING.

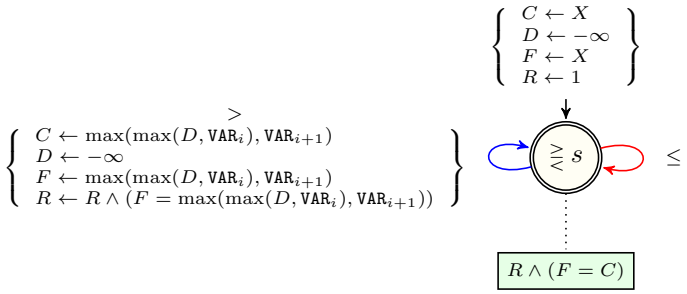
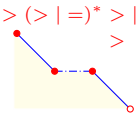


Figure 4.20: Automaton for the ALL_EQUAL_MAX_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_DECREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 1, 6, 6, 1, 1, 4, 6, 6, 4, 4, 2, 1, 3, 6, 1, 5\rangle)$

Figure 4.21 provides an example where the ALL_EQUAL_MAX_DECREASING_SEQUENCE $([1, 6, 6, 1, 1, 4, 6, 6, 4, 4, 2, 1, 3, 6, 1, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

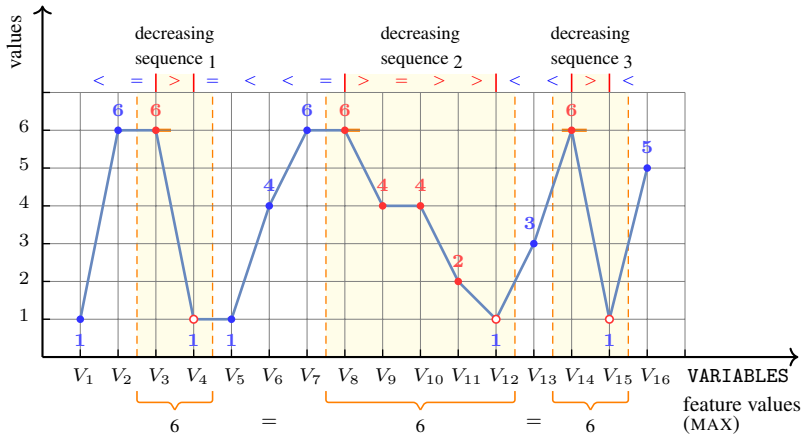


Figure 4.21: Illustrating the ALL_EQUAL_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.22 depicts the automaton associated with the constraint ALL_EQUAL_MAX_DECREASING_SEQUENCE.

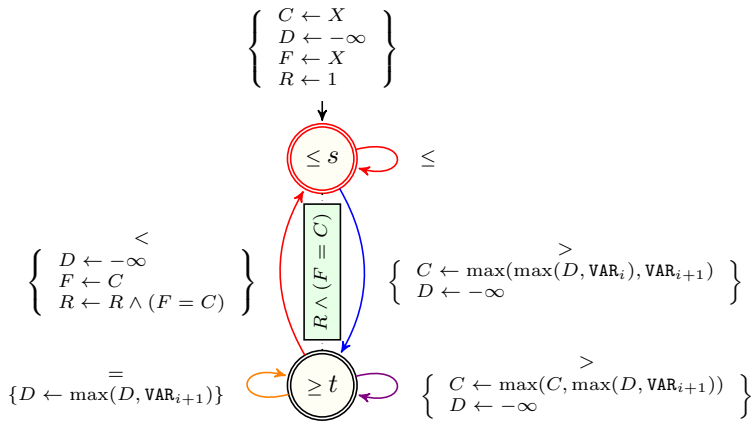
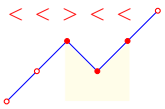


Figure 4.22: Automaton for the ALL_EQUAL_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(\langle 0, 1, 2, 1, 5, 6, 7, 3, 0, 2, 3, 5, 0, 2, 3, 3\rangle)$

Figure 4.23 provides an example where the ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE $([0, 1, 2, 1, 5, 6, 7, 3, 0, 2, 3, 5, 0, 2, 3, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

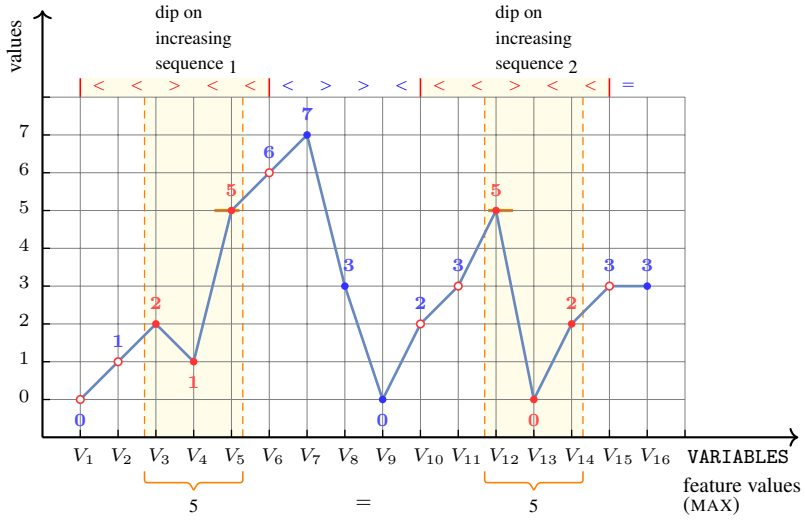


Figure 4.23: Illustrating the ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.24 depicts the automaton associated with the constraint ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE.

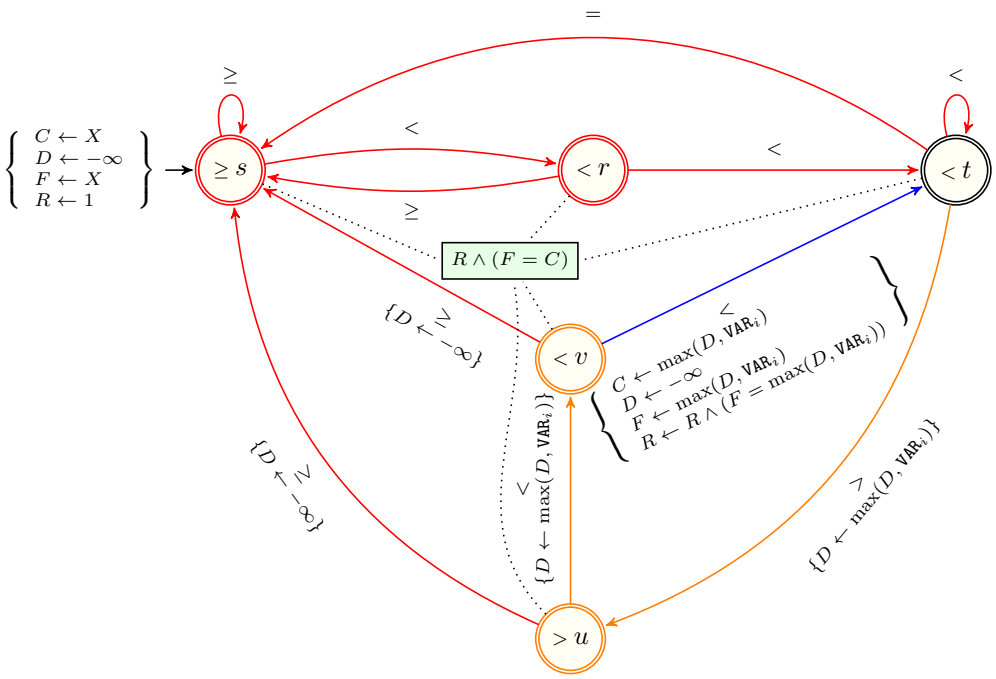


Figure 4.24: Automaton for the ALL_EQUAL_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_INCREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint ALL_EQUAL_MAX_INCREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 6,1,1,6,6,6,5,6,5,3,3,6,4,1,6,2\rangle)$

Figure 4.25 provides an example where the ALL_EQUAL_MAX_INCREASING $([6,1,1,6,6,6,5,6,5,3,3,6,4,1,6,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

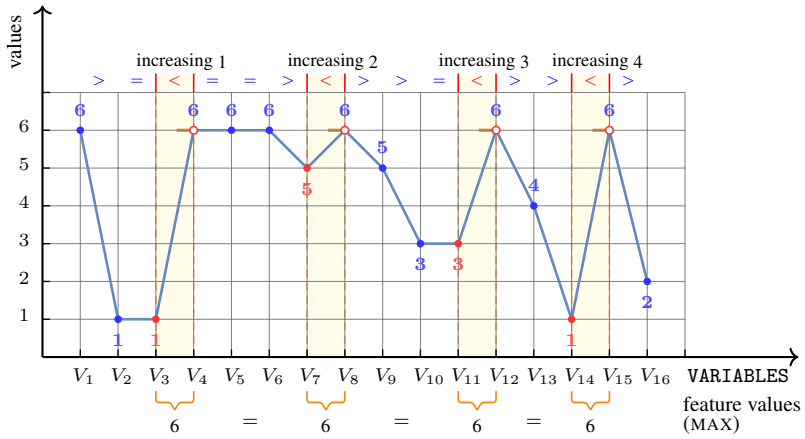


Figure 4.25: Illustrating the ALL_EQUAL_MAX_INCREASING constraint of the **Example** slot

Automaton

Figure 4.26 depicts the automaton associated with the constraint ALL_EQUAL_MAX_INCREASING.

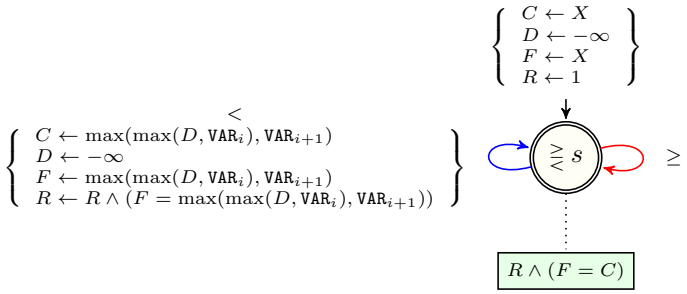
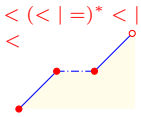


Figure 4.26: Automaton for the ALL_EQUAL_MAX_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 6, 1, 1, 6, 6, 3, 1, 1, 3, 3, 5, 6, 4, 1, 6, 2\rangle)$

Figure 4.27 provides an example where the ALL_EQUAL_MAX_INCREASING_SEQUENCE $([6, 1, 1, 6, 6, 3, 1, 1, 3, 3, 5, 6, 4, 1, 6, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

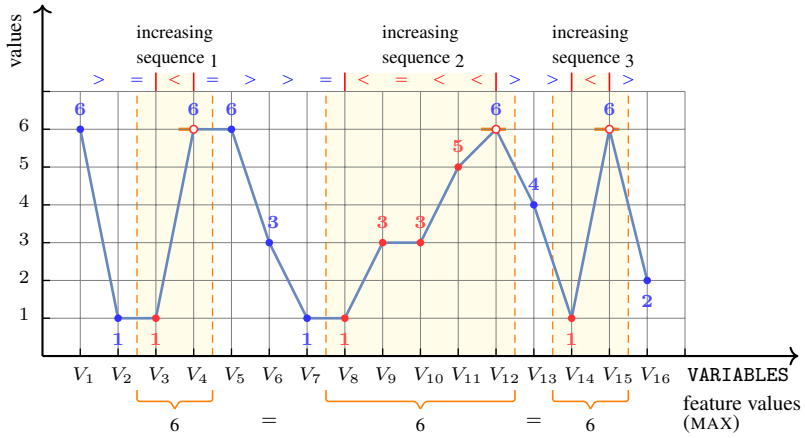


Figure 4.27: Illustrating the ALL_EQUAL_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.28 depicts the automaton associated with the constraint ALL_EQUAL_MAX_INCREASING_SEQUENCE.

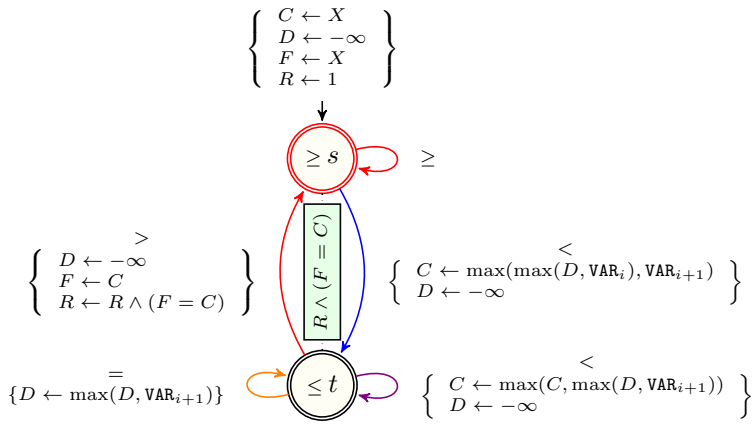
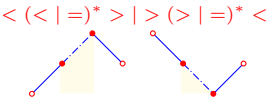


Figure 4.28: Automaton for the ALL_EQUAL_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_INFLEXION



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INFLEXION pattern.

Constraint ALL_EQUAL_MAX_INFLEXION(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 5, 5, 4, 4, 3, 2, 2, 3, 3, 3, 4, 4, 2, 2\rangle)$

Figure 4.29 provides an example where the ALL_EQUAL_MAX_INFLEXION $([5, 5, 4, 4, 3, 2, 2, 3, 3, 3, 4, 4, 2, 2])$ constraint holds.

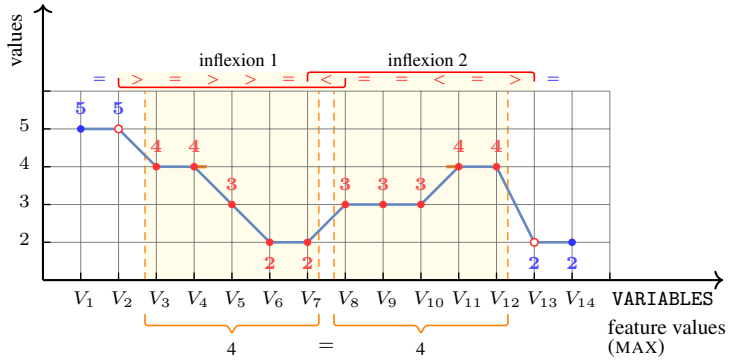


Figure 4.29: Illustrating the ALL_EQUAL_MAX_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Automaton Figure 4.30 depicts the automaton associated with the constraint ALL_EQUAL_MAX_INFLEXION.

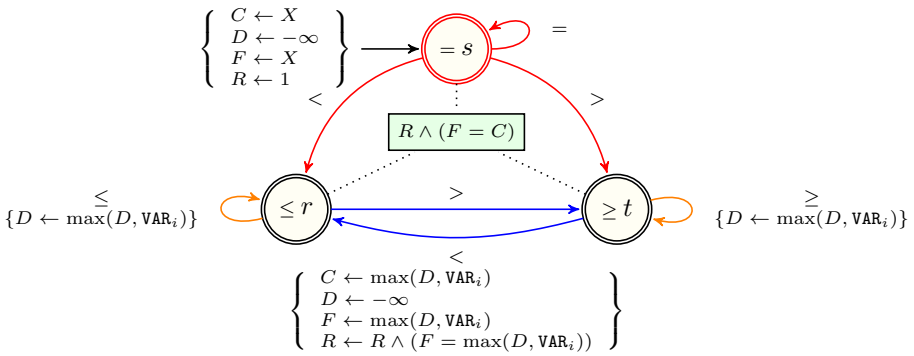
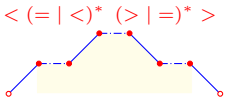


Figure 4.30: Automaton for the ALL_EQUAL_MAX_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_PEAK

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint ALL_EQUAL_MAX_PEAK(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1 \rangle)$

Figure 4.31 provides an example where the ALL_EQUAL_MAX_PEAK $([1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

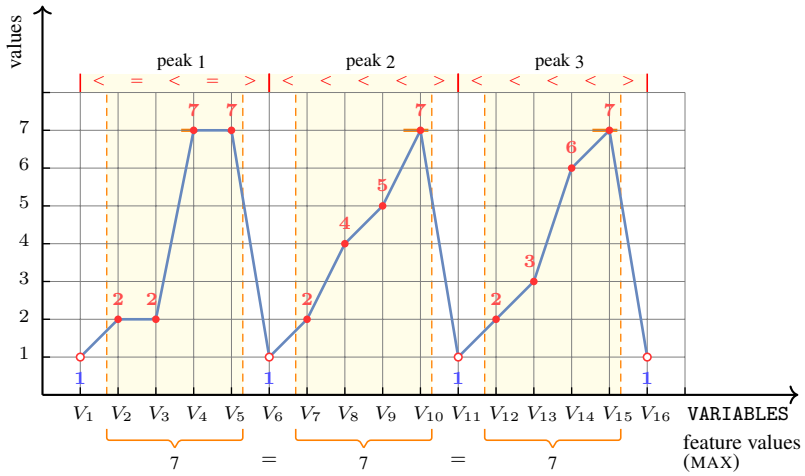


Figure 4.31: Illustrating the ALL_EQUAL_MAX_PEAK constraint of the **Example** slot

Automaton Figure 4.32 depicts the automaton associated with the constraint ALL_EQUAL_MAX_PEAK.

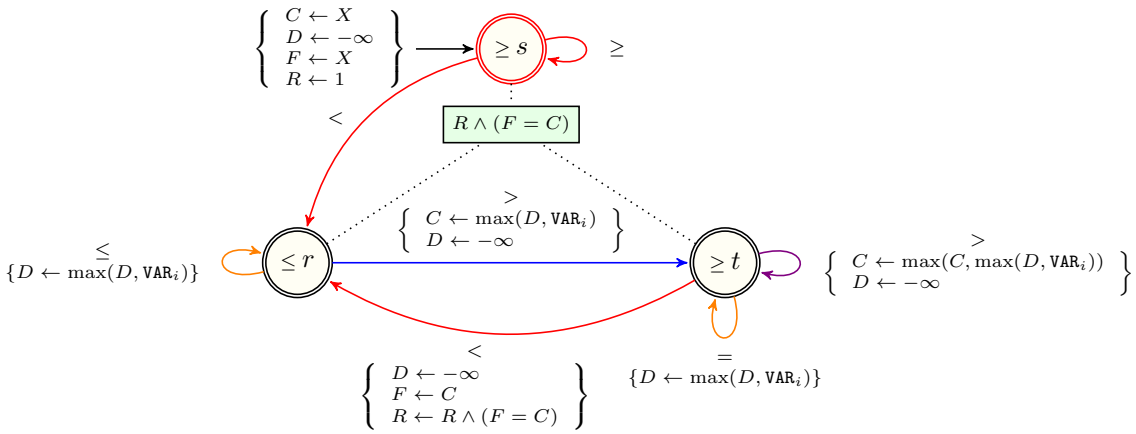


Figure 4.32: Automaton for the ALL_EQUAL_MAX_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 5,5,6,3,1,2,4,4,6,6,4,1,1,6,6,1\rangle)$

Figure 4.33 provides an example where the ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE $([5,5,6,3,1,2,4,4,6,6,4,1,1,6,6,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

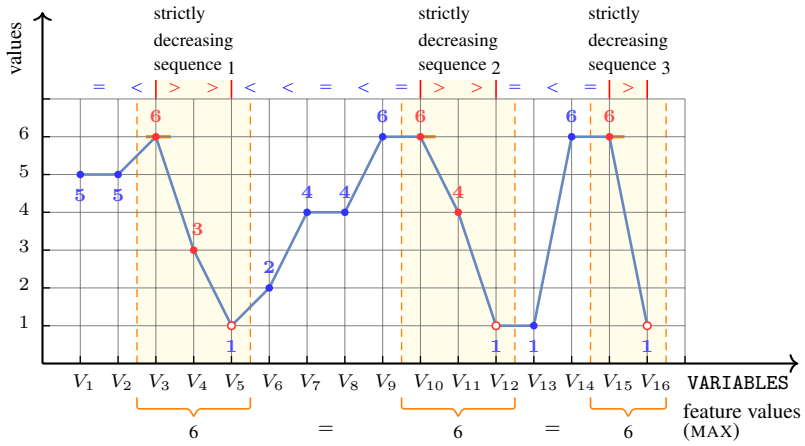


Figure 4.33: Illustrating the ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.34 depicts the automaton associated with the constraint ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE.

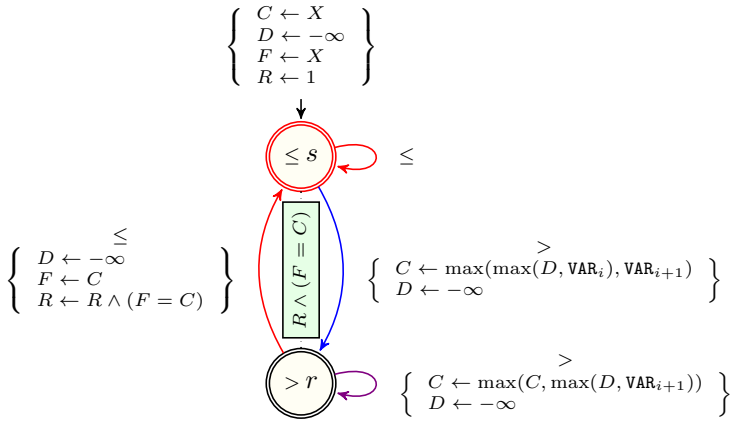


Figure 4.34: Automaton for the ALL_EQUAL_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 6, 1, 1, 6, 6, 3, 1, 1, 3, 4, 5, 6, 4, 1, 6, 2\rangle)$

Figure 4.35 provides an example where the ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE $([6, 1, 1, 6, 6, 3, 1, 1, 3, 4, 5, 6, 4, 1, 6, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

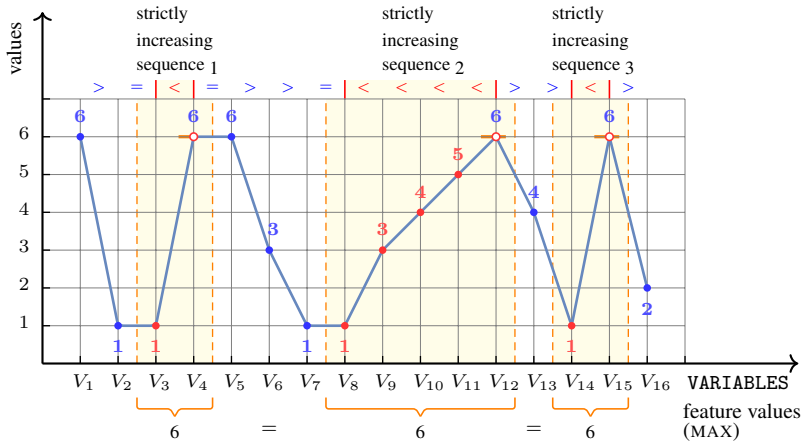


Figure 4.35: Illustrating the ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.36 depicts the automaton associated with the constraint ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE.

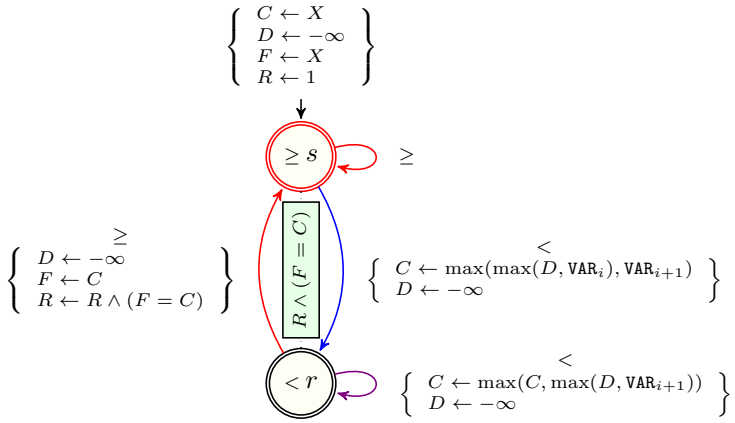


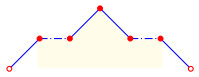
Figure 4.36: Automaton for the ALL_EQUAL_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_SUMMIT

► ▷ ◄ DESCRIPTION AUTOMATON

(< | < (= | <)* <)(> | > (= | >)* >)



Origin Based on the SUMMIT pattern.

Constraint ALL_EQUAL_MAX_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2\rangle)$

Figure 4.37 provides an example where the ALL_EQUAL_MAX_SUMMIT $([1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

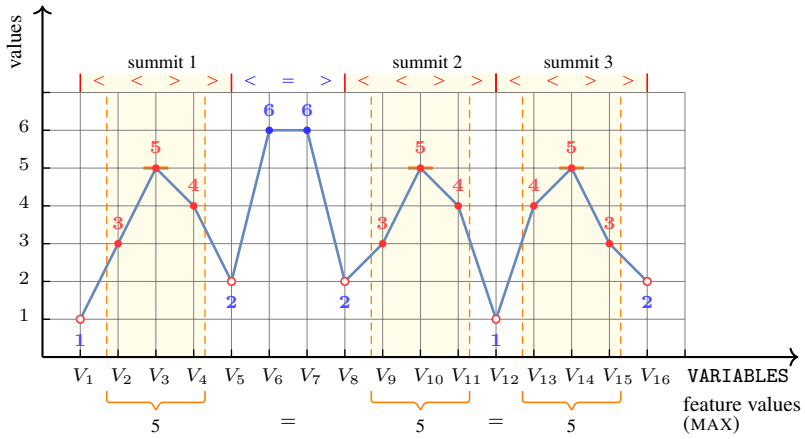


Figure 4.37: Illustrating the ALL_EQUAL_MAX_SUMMIT constraint of the **Example** slot

Automaton Figure 4.38 depicts the automaton associated with the constraint ALL_EQUAL_MAX_SUMMIT.

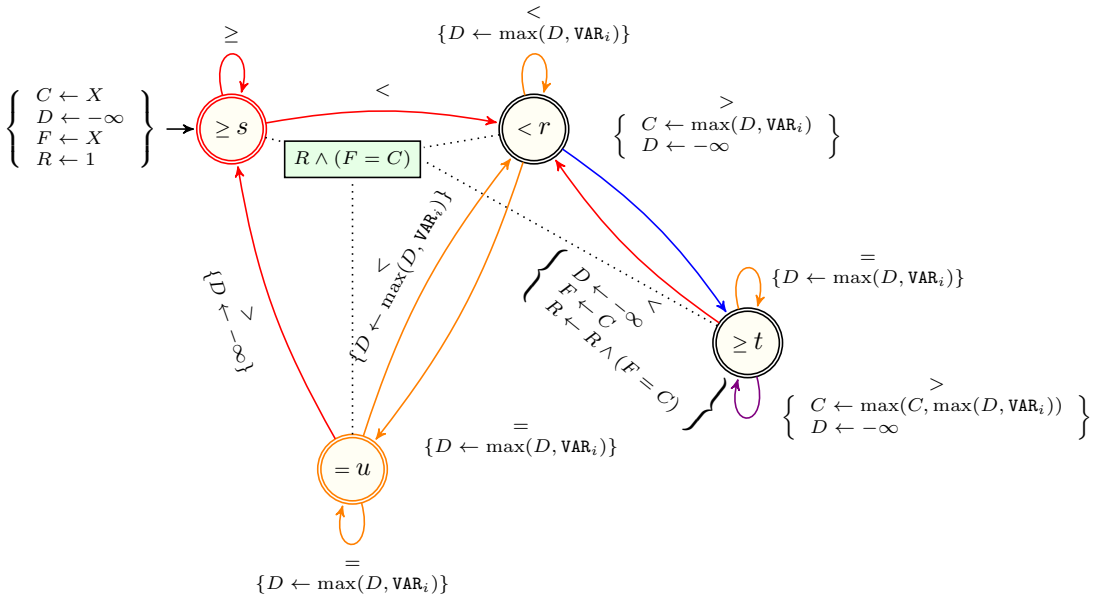
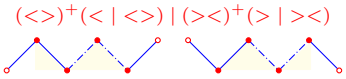


Figure 4.38: Automaton for the ALL_EQUAL_MAX_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_MAX_ZIGZAG

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint ALL_EQUAL_MAX_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 6, 1, 2, 7, 5, 1, 6, 3, 4, 2, 5, 6, 3, 4, 7\rangle)$

Figure 4.39 provides an example where the ALL_EQUAL_MAX_ZIGZAG $([1, 6, 1, 2, 7, 5, 1, 6, 3, 4, 2, 5, 6, 3, 4, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

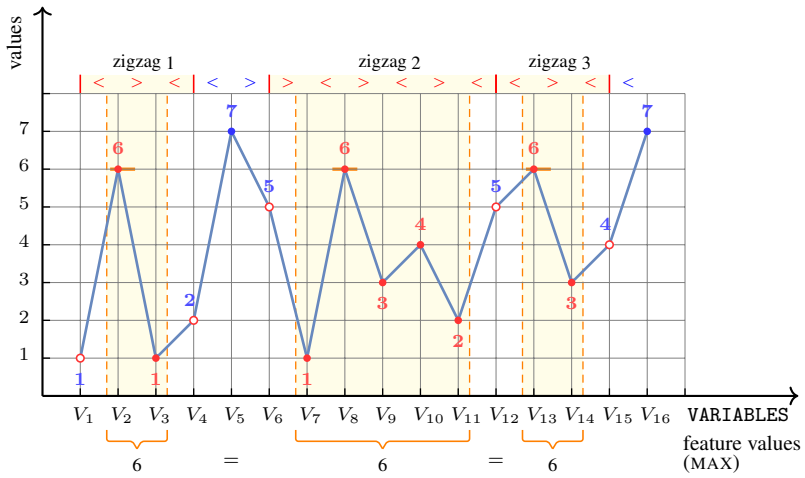


Figure 4.39: Illustrating the ALL_EQUAL_MAX_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.40 depicts the automaton associated with the constraint ALL_EQUAL_MAX_ZIGZAG.

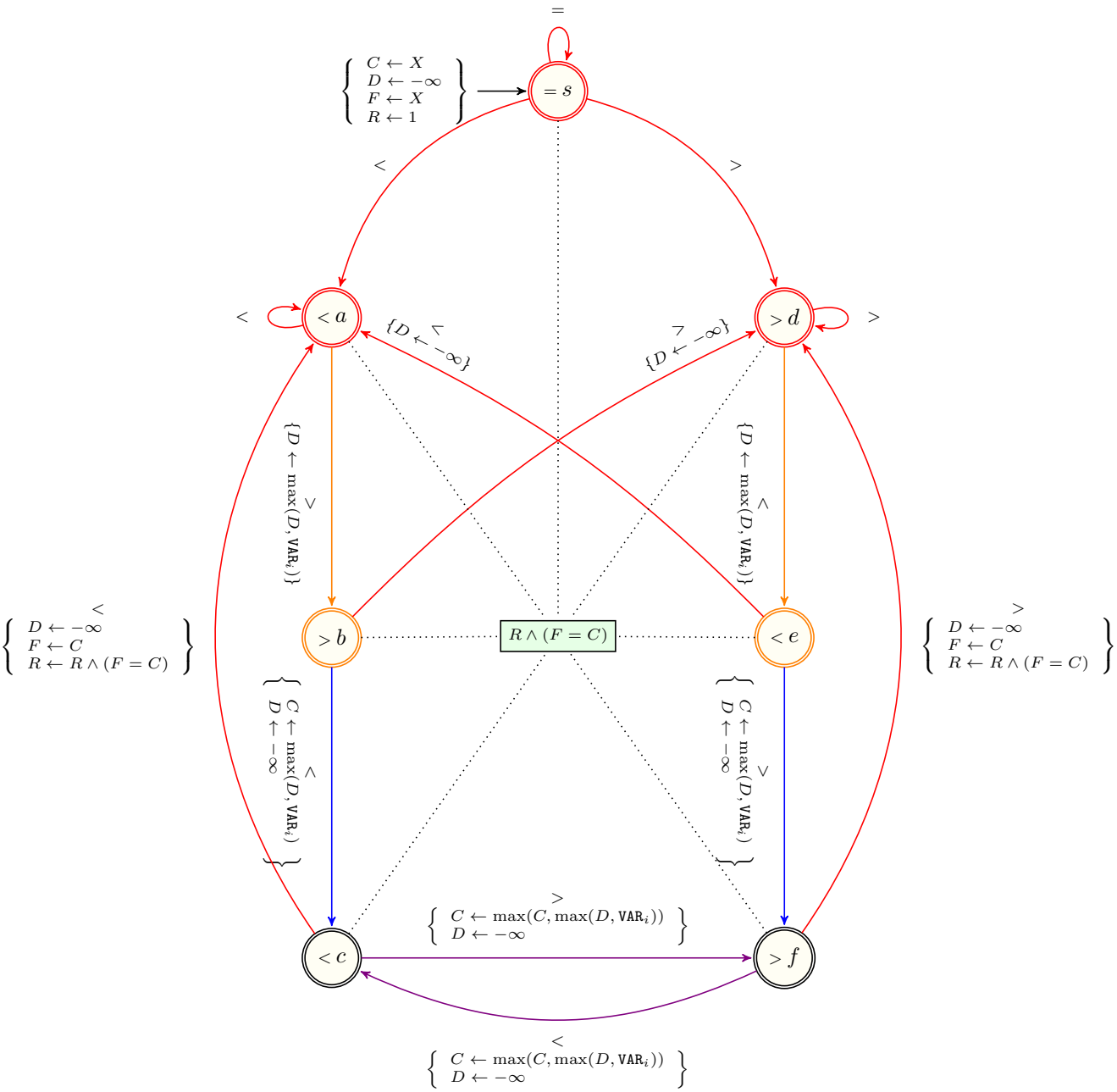
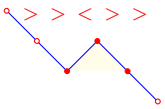


Figure 4.40: Automaton for the ALL_EQUAL_MAX_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 7, 6, 5, 6, 2, 1, 1, 5, 4, 3, 2, 6, 5, 4, 4, 5\rangle)$

Figure 4.41 provides an example where the ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 2, 1, 1, 5, 4, 3, 2, 6, 5, 4, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

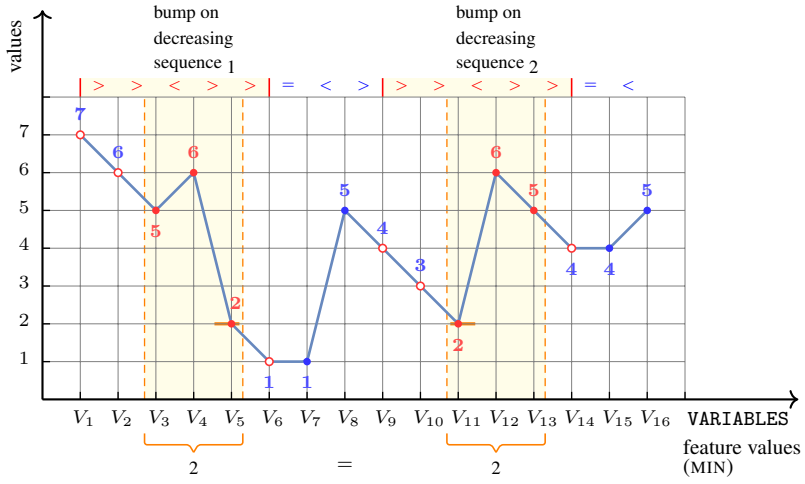


Figure 4.41: Illustrating the ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.42 depicts the automaton associated with the constraint ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE.

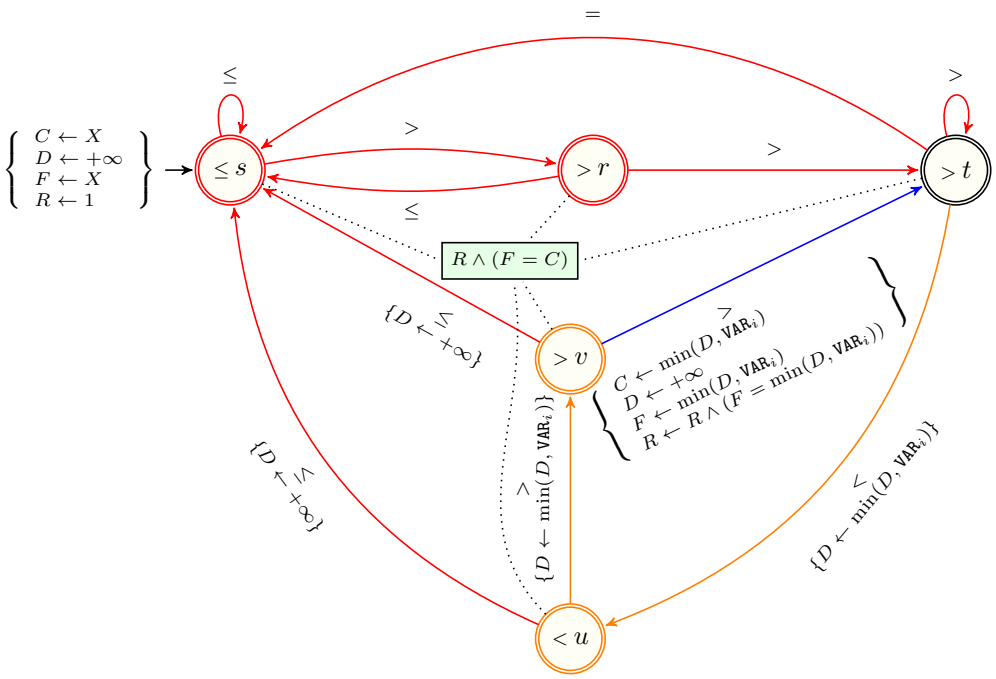


Figure 4.42: Automaton for the ALL_EQUAL_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_DECREASING

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint ALL_EQUAL_MIN_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 1,6,6,1,1,4,4,5,5,5,5,1,3,6,1,5\rangle)$

Figure 4.43 provides an example where the ALL_EQUAL_MIN_DECREASING $([1,6,6,1,1,4,4,5,5,5,5,1,3,6,1,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

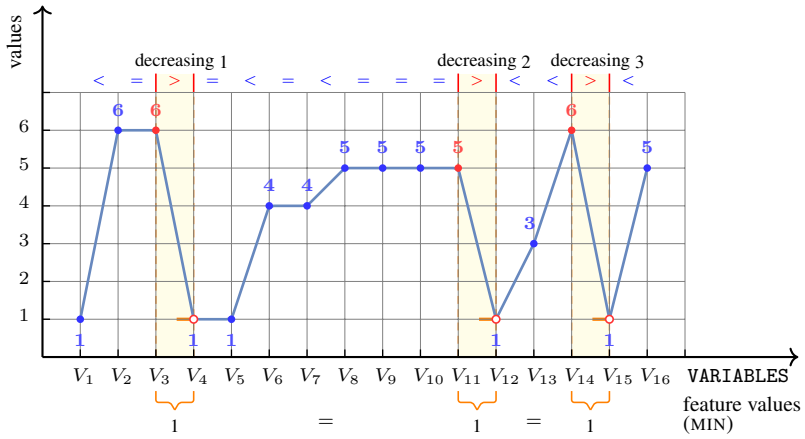


Figure 4.43: Illustrating the ALL_EQUAL_MIN_DECREASING constraint of the **Example** slot

Automaton Figure 4.44 depicts the automaton associated with the constraint ALL_EQUAL_MIN_DECREASING.

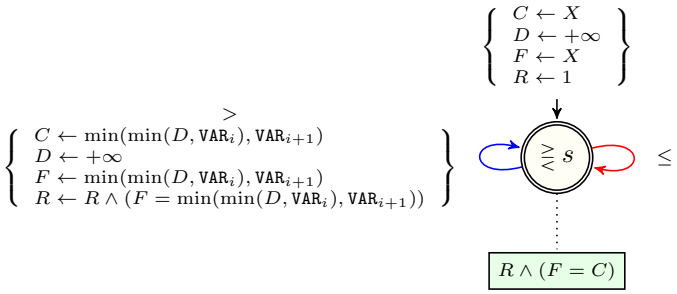
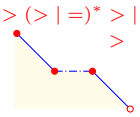


Figure 4.44: Automaton for the ALL_EQUAL_MIN_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_DECREASING_SEQUENCE



► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_DECREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 1, 6, 6, 1, 1, 4, 5, 5, 4, 4, 2, 1, 3, 4, 1, 5 \rangle)$

Figure 4.45 provides an example where the ALL_EQUAL_MIN_DECREASING_SEQUENCE $([1, 6, 6, 1, 1, 4, 5, 5, 4, 4, 2, 1, 3, 4, 1, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

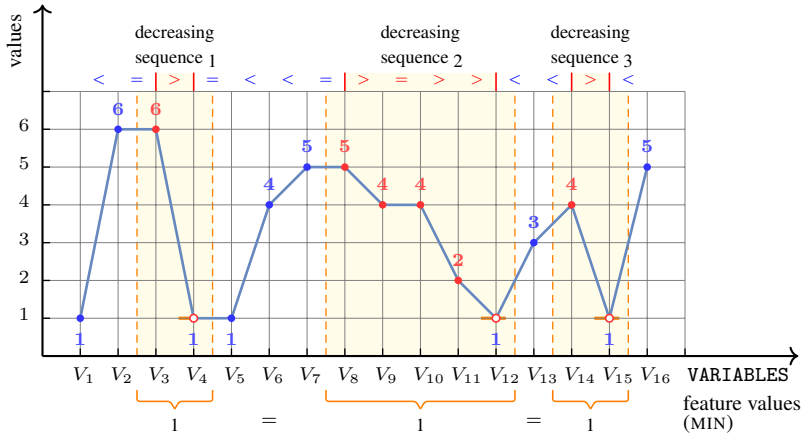


Figure 4.45: Illustrating the ALL_EQUAL_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.46 depicts the automaton associated with the constraint ALL_EQUAL_MIN_DECREASING_SEQUENCE.

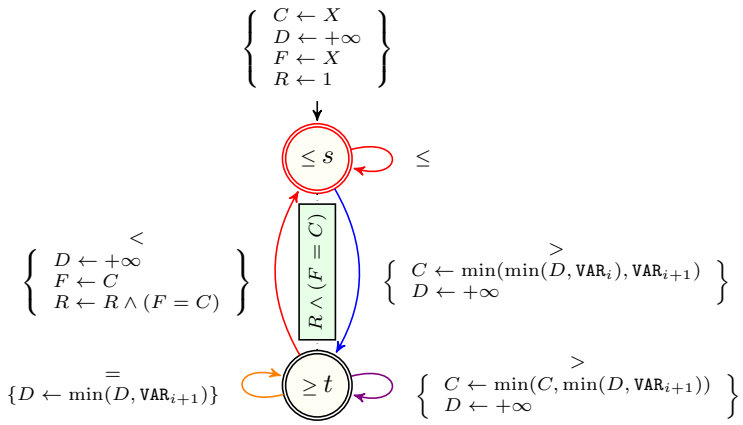
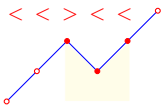


Figure 4.46: Automaton for the ALL_EQUAL_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 1,2,3,2,4,7,7,3,4,5,6,2,3,4,4,3\rangle)$

Figure 4.47 provides an example where the ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE $([1,2,3,2,4,7,7,3,4,5,6,2,3,4,4,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

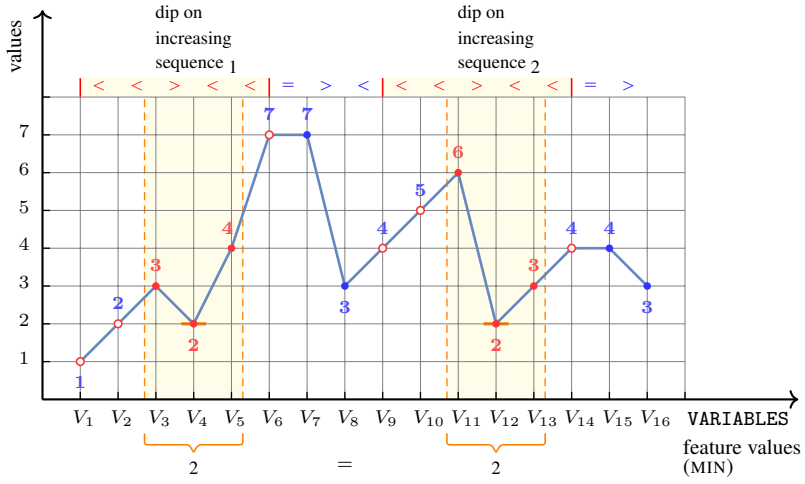


Figure 4.47: Illustrating the ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.48 depicts the automaton associated with the constraint ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE.

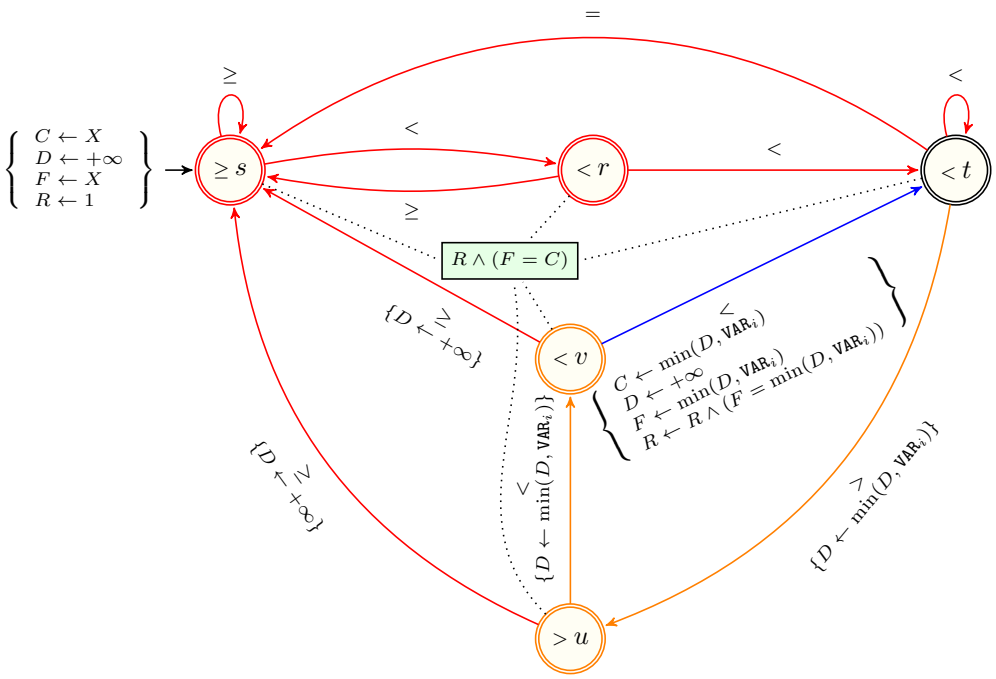
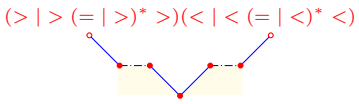


Figure 4.48: Automaton for the ALL_EQUAL_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_GORGE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint ALL_EQUAL_MIN_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 6,4,2,3,5,1,1,5,4,2,3,6,3,2,4,5\rangle)$

Figure 4.49 provides an example where the ALL_EQUAL_MIN_GORGE $([6,4,2,3,5,1,1,5,4,2,3,6,3,2,4,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

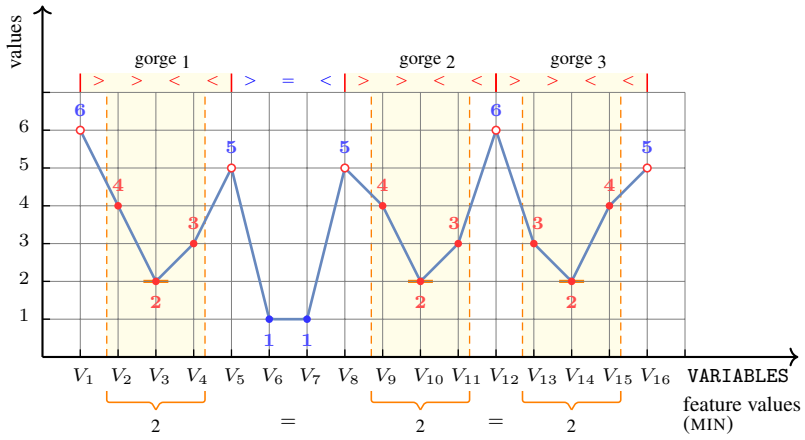


Figure 4.49: Illustrating the ALL_EQUAL_MIN_GORGE constraint of the **Example** slot

Automaton Figure 4.50 depicts the automaton associated with the constraint ALL_EQUAL_MIN_GORGE.

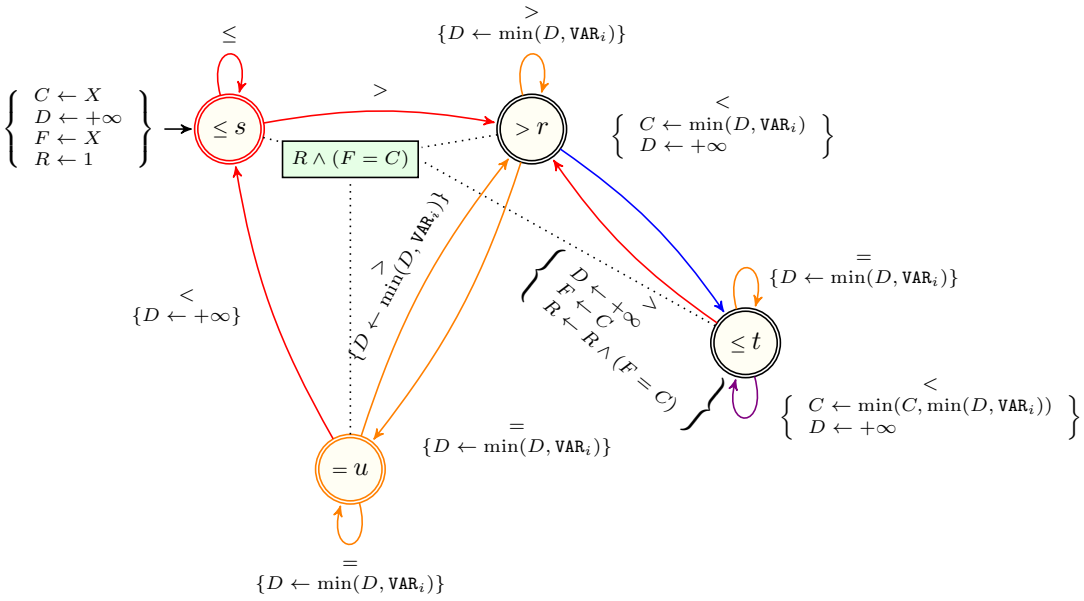


Figure 4.50: Automaton for the ALL_EQUAL_MIN_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_INCREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint ALL_EQUAL_MIN_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '$<$'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 7,2,2,6,6,3,2,2,7,4,4,3,3,2,5,1\rangle)$

Figure 4.51 provides an example where the ALL_EQUAL_MIN_INCREASING $([7,2,2,6,6,3,2,2,7,4,4,3,3,2,5,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

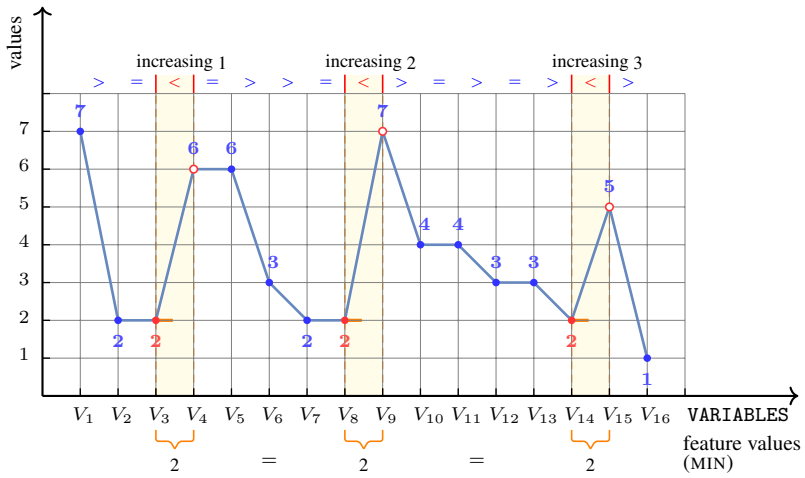


Figure 4.51: Illustrating the ALL_EQUAL_MIN_INCREASING constraint of the **Example** slot

Automaton

Figure 4.52 depicts the automaton associated with the constraint ALL_EQUAL_MIN_INCREASING.

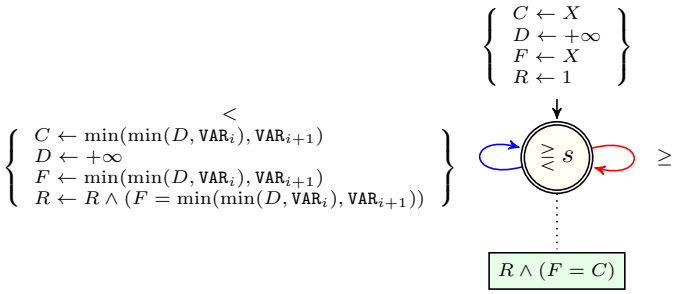
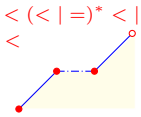


Figure 4.52: Automaton for the ALL_EQUAL_MIN_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 7, 2, 2, 6, 6, 3, 2, 2, 3, 3, 5, 6, 4, 2, 5, 1\rangle)$

Figure 4.53 provides an example where the ALL_EQUAL_MIN_INCREASING_SEQUENCE $([7, 2, 2, 6, 6, 3, 2, 2, 3, 3, 5, 6, 4, 2, 5, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

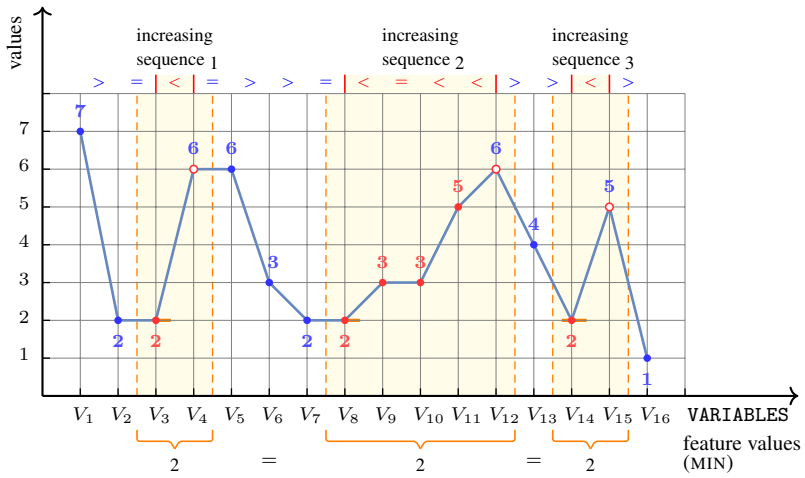


Figure 4.53: Illustrating the ALL_EQUAL_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.54 depicts the automaton associated with the constraint ALL_EQUAL_MIN_INCREASING_SEQUENCE.

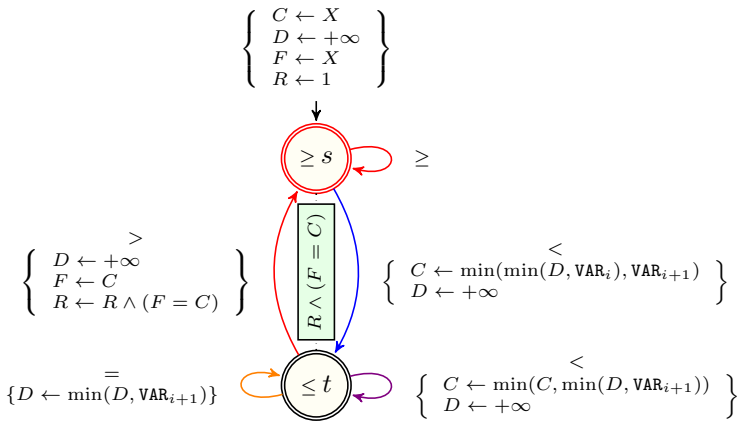
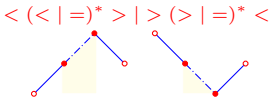


Figure 4.54: Automaton for the ALL_EQUAL_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_INFLEXION

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint ALL_EQUAL_MIN_INFLEXION(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 2, 2, 3, 6, 6, 5, 4, 3, 4, 4, 5\rangle)$

Figure 4.55 provides an example where the ALL_EQUAL_MIN_INFLEXION $([2, 2, 3, 6, 6, 5, 4, 3, 4, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

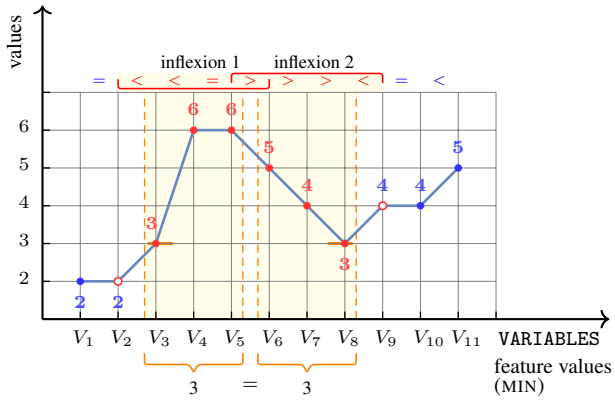


Figure 4.55: Illustrating the ALL_EQUAL_MIN_INFLEXION constraint of the **Example** slot

Automaton Figure 4.56 depicts the automaton associated with the constraint ALL_EQUAL_MIN_INFLEXION.

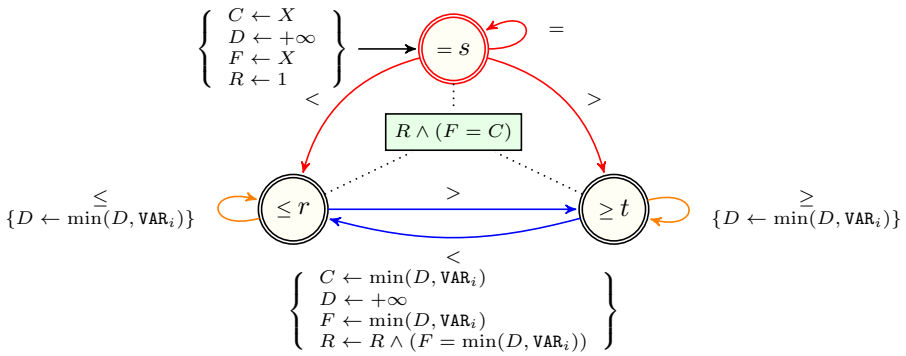


Figure 4.56: Automaton for the ALL_EQUAL_MIN_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 5,1,4,3,1,2,4,4,5,5,4,1,1,6,6,1\rangle)$

Figure 4.57 provides an example where the ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE $([5,1,4,3,1,2,4,4,5,5,4,1,1,6,6,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

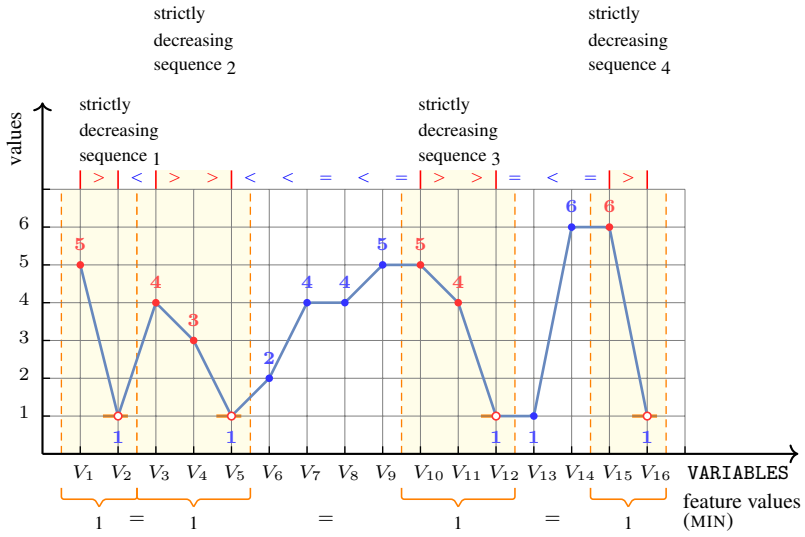


Figure 4.57: Illustrating the ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.58 depicts the automaton associated with the constraint ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE.

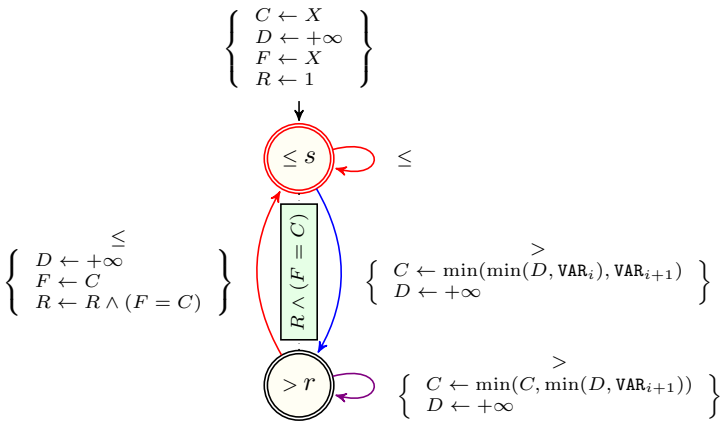


Figure 4.58: Automaton for the ALL_EQUAL_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 7, 2, 2, 6, 6, 3, 2, 2, 3, 4, 5, 6, 4, 2, 5, 1\rangle)$

Figure 4.59 provides an example where the ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE $([7, 2, 2, 6, 6, 3, 2, 2, 3, 4, 5, 6, 4, 2, 5, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

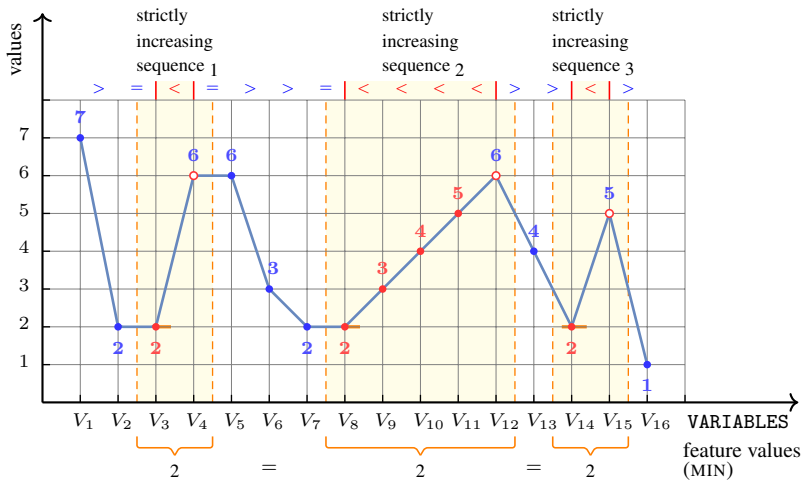


Figure 4.59: Illustrating the ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.60 depicts the automaton associated with the constraint ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE.

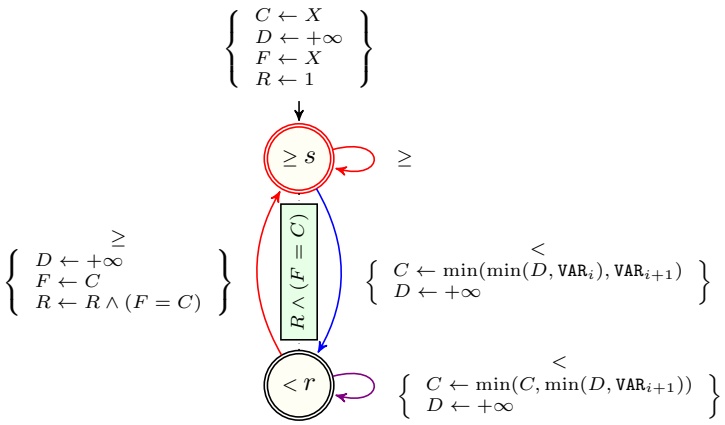
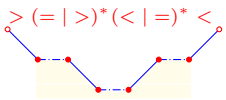


Figure 4.60: Automaton for the ALL_EQUAL_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_VALLEY

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint ALL_EQUAL_MIN_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7, 6, 6, 1, 1, 7, 6, 4, 3, 1, 7, 6, 5, 2, 1, 7\rangle)$

Figure 4.61 provides an example where the ALL_EQUAL_MIN_VALLEY $([7, 6, 6, 1, 1, 7, 6, 4, 3, 1, 7, 6, 5, 2, 1, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

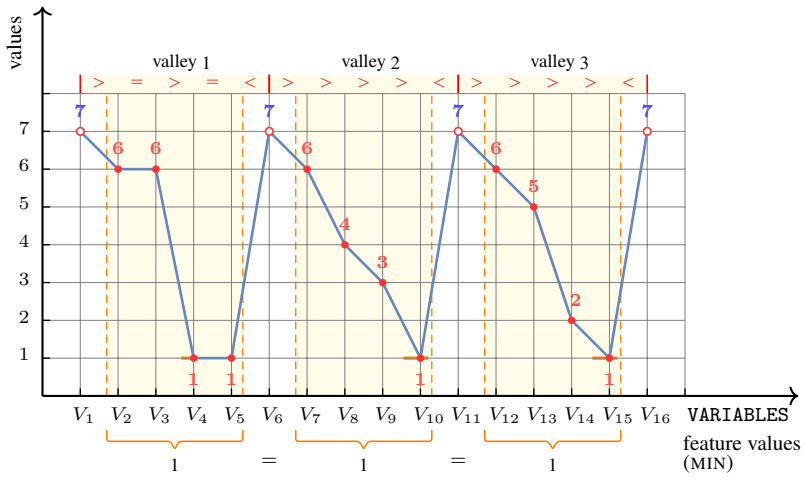


Figure 4.61: Illustrating the ALL_EQUAL_MIN_VALLEY constraint of the **Example** slot

Automaton Figure 4.62 depicts the automaton associated with the constraint ALL_EQUAL_MIN_VALLEY.

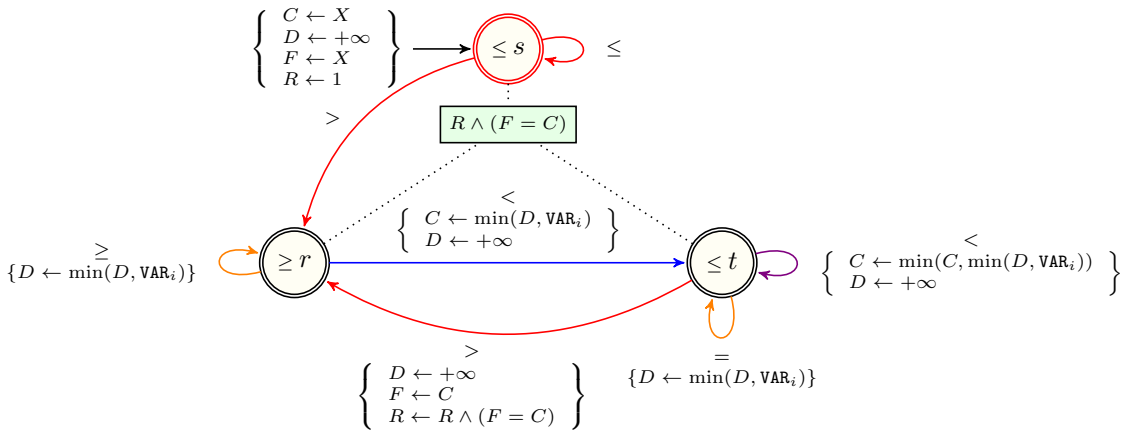
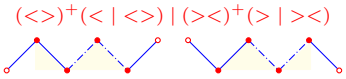


Figure 4.62: Automaton for the ALL_EQUAL_MIN_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_MIN_ZIGZAG

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint ALL_EQUAL_MIN_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are all the same.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 1, 3, 1, 2, 6, 5, 1, 4, 3, 4, 3, 5, 6, 1, 4, 7\rangle)$

Figure 4.63 provides an example where the ALL_EQUAL_MIN_ZIGZAG $([1, 3, 1, 2, 6, 5, 1, 4, 3, 4, 3, 5, 6, 1, 4, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

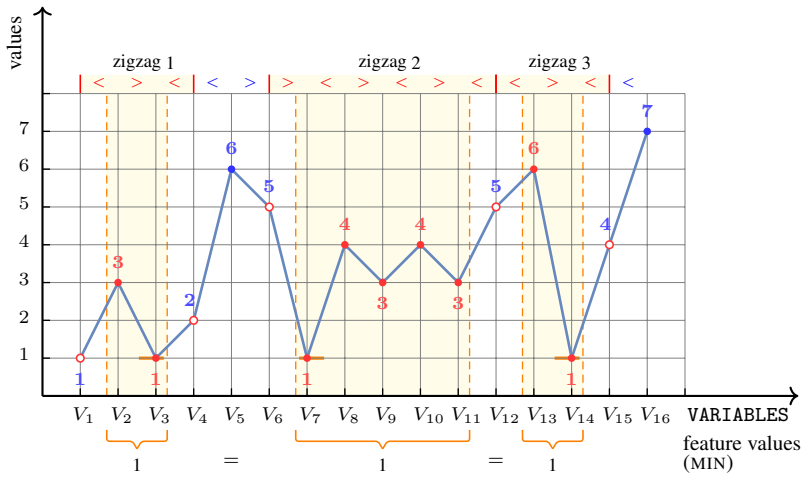


Figure 4.63: Illustrating the ALL_EQUAL_MIN_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.64 depicts the automaton associated with the constraint ALL_EQUAL_MIN_ZIGZAG.

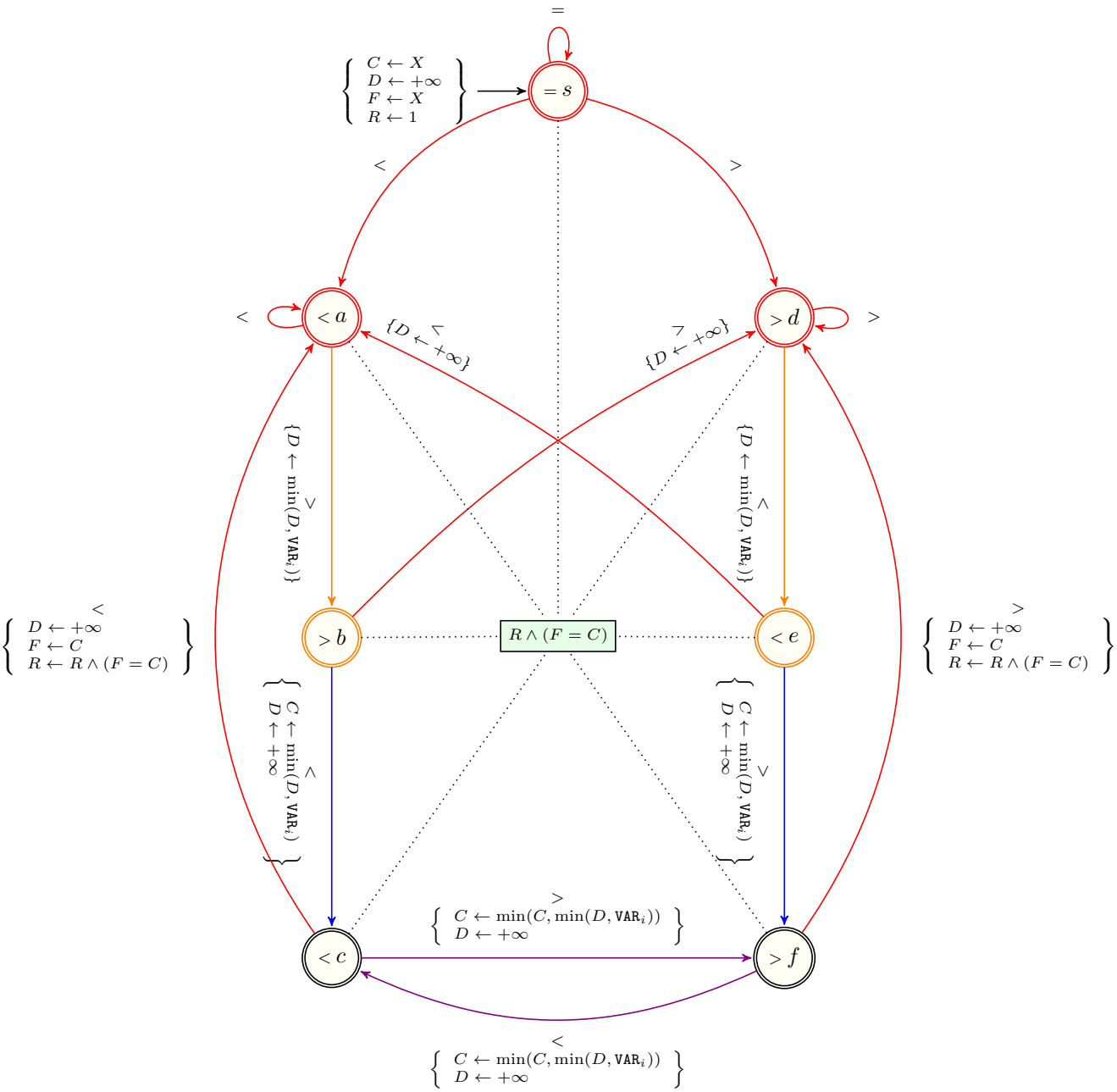


Figure 4.64: Automaton for the ALL_EQUAL_MIN_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_DECREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint ALL_EQUAL_RANGE_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 1,4,6,2,2,4,4,5,5,5,5,1,3,6,2,5\rangle)$

Figure 4.65 provides an example where the ALL_EQUAL_RANGE_DECREASING $([1,4,6,2,2,4,4,5,5,5,5,1,3,6,2,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

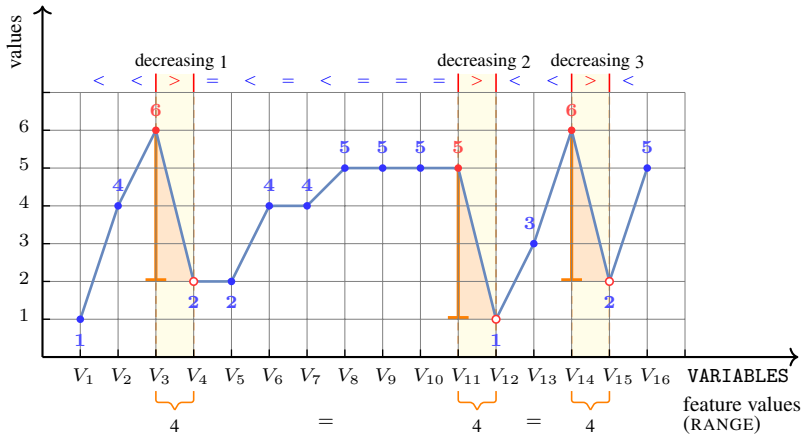


Figure 4.65: Illustrating the ALL_EQUAL_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figure 4.66 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_DECREASING.

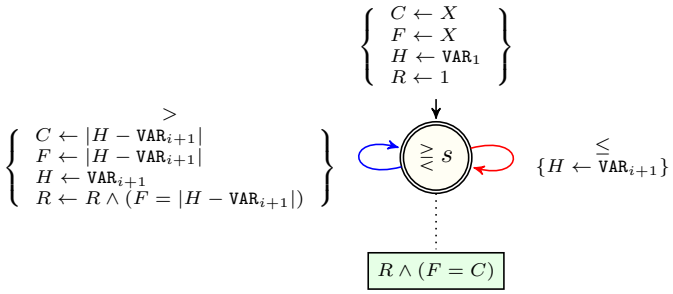
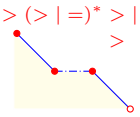


Figure 4.66: Automaton for the ALL_EQUAL_RANGE_DECREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_DECREASING_SEQUENCE



► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_RANGE_DECREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 1, 6, 7, 2, 2, 4, 5, 5, 4, 4, 2, 0, 3, 6, 1, 5\rangle)$

Figure 4.67 provides an example where the ALL_EQUAL_RANGE_DECREASING_SEQUENCE $([1, 6, 7, 2, 2, 4, 5, 5, 4, 4, 2, 0, 3, 6, 1, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

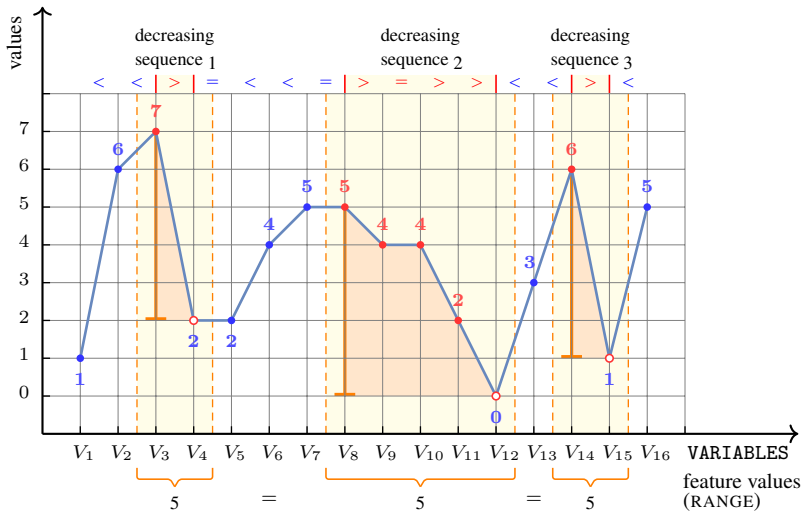


Figure 4.67: Illustrating the ALL_EQUAL_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.68 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_DECREASING_SEQUENCE.

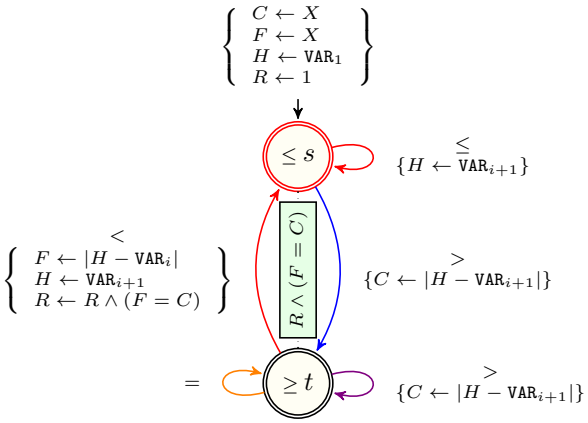


Figure 4.68: Automaton for the ALL_EQUAL_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_INCREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint ALL_EQUAL_RANGE_INCREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 7, 3, 3, 6, 6, 3, 2, 2, 5, 4, 4, 3, 3, 1, 4, 1 \rangle)$

Figure 4.69 provides an example where the ALL_EQUAL_RANGE_INCREASING $([7, 3, 3, 6, 6, 3, 2, 2, 5, 4, 4, 3, 3, 1, 4, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

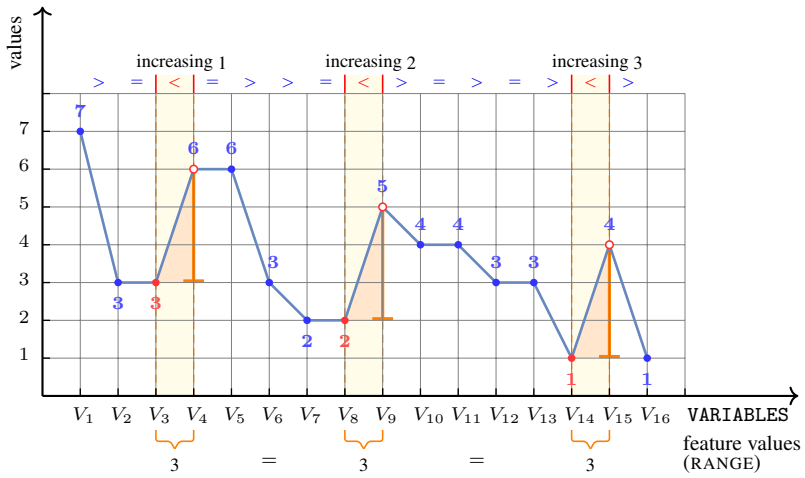


Figure 4.69: Illustrating the ALL_EQUAL_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figure 4.70 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_INCREASING.

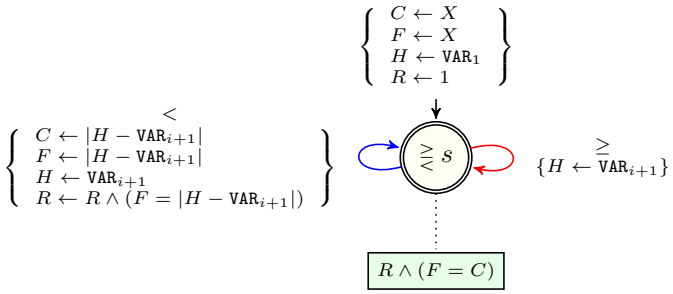
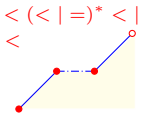


Figure 4.70: Automaton for the ALL_EQUAL_RANGE_INCREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_RANGE_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 7, 2, 2, 6, 6, 3, 1, 1, 2, 3, 3, 5, 4, 2, 6, 1 \rangle)$

Figure 4.71 provides an example where the ALL_EQUAL_RANGE_INCREASING_SEQUENCE $([7, 2, 2, 6, 6, 3, 1, 1, 2, 3, 3, 5, 4, 2, 6, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

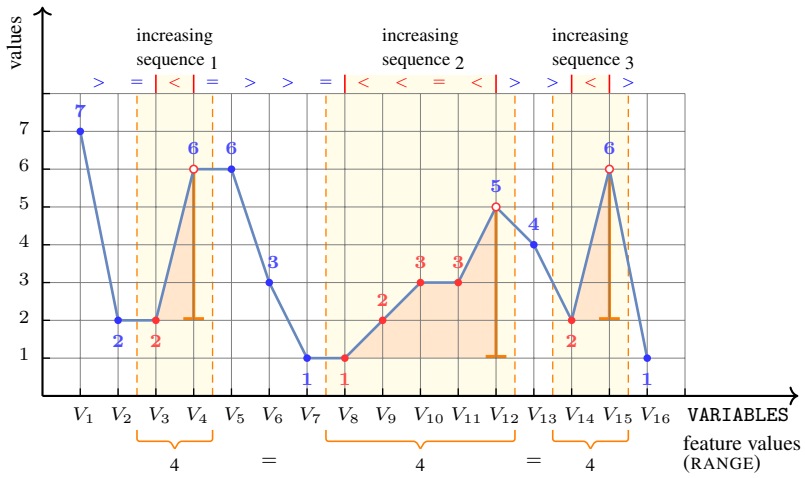


Figure 4.71: Illustrating the ALL_EQUAL_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.72 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_INCREASING_SEQUENCE.

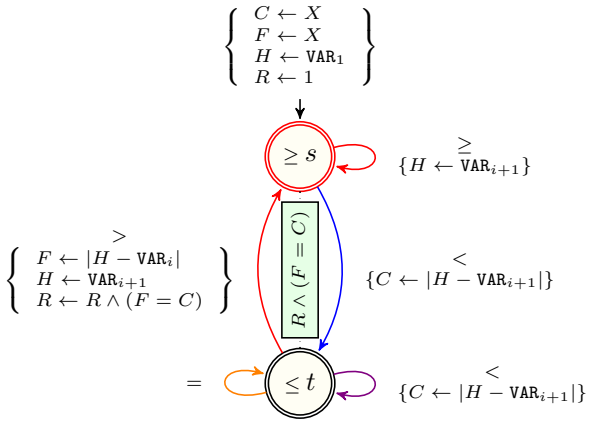


Figure 4.72: Automaton for the ALL_EQUAL_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 4, 1, 4, 3, 1, 2, 4, 4, 5, 5, 4, 2, 2, 6, 6, 3\rangle)$

Figure 4.73 provides an example where the ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE $([4, 1, 4, 3, 1, 2, 4, 4, 5, 5, 4, 2, 2, 6, 6, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

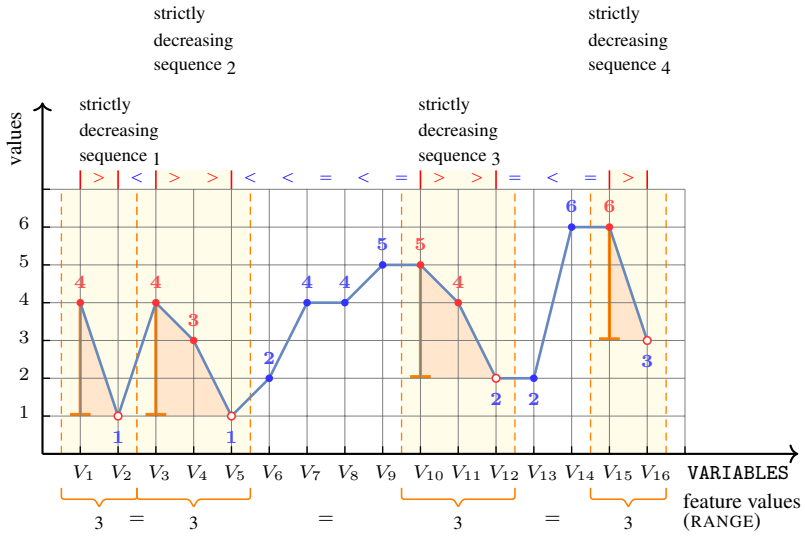


Figure 4.73: Illustrating the ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.74 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE.

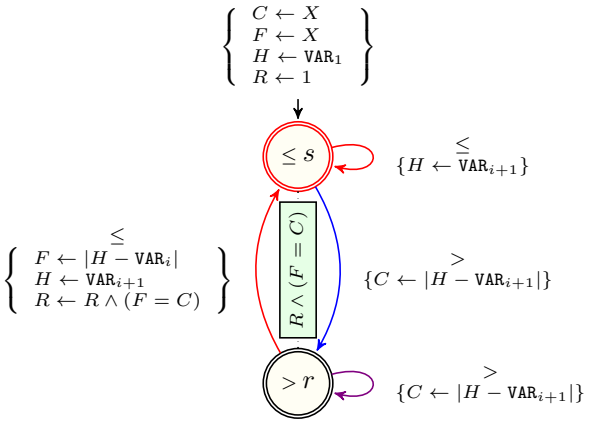


Figure 4.74: Automaton for the ALL_EQUAL_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 7, 2, 2, 6, 6, 3, 1, 1, 2, 3, 4, 5, 4, 0, 4, 1\rangle)$

Figure 4.75 provides an example where the ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE $([7, 2, 2, 6, 6, 3, 1, 1, 2, 3, 4, 5, 4, 0, 4, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

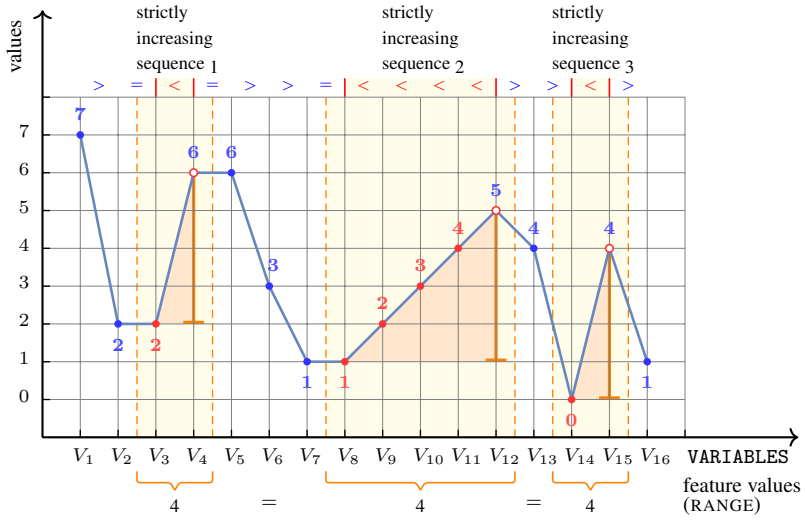


Figure 4.75: Illustrating the ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.76 depicts the automaton associated with the constraint ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE.

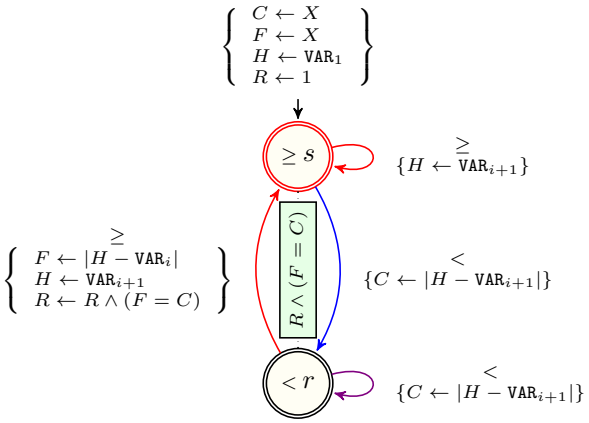
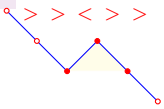


Figure 4.76: Automaton for the ALL_EQUAL_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(\langle 7, 6, 5, 6, 5, 4, 2, 3, 7, 4, 3, 2, 8, 6, 4, 4\rangle)$

Figure 4.77 provides an example where the ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 5, 4, 2, 3, 7, 4, 3, 2, 8, 6, 4, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

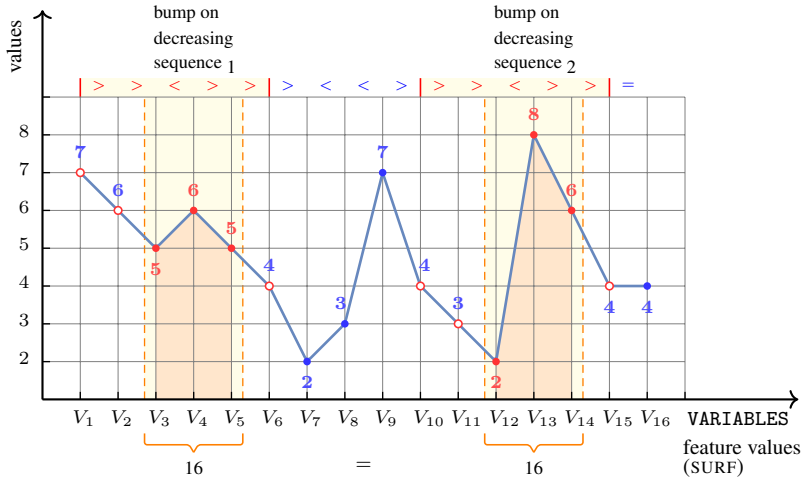


Figure 4.77: Illustrating the ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.78 depicts the automaton associated with the constraint ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE.

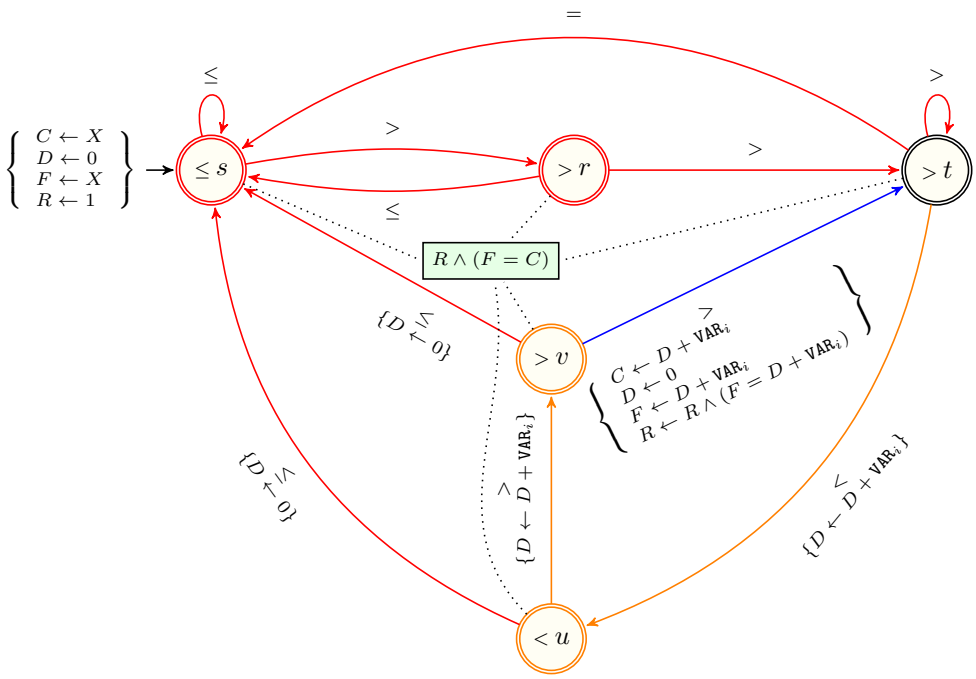


Figure 4.78: Automaton for the ALL_EQUAL_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_DECREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint ALL_EQUAL_SURF_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the DECREASING pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 1, 5, 5, 2, 2, 3, 4, 6, 1, 2, 3, 4, 5, 2, 2, 3 \rangle)$

Figure 4.79 provides an example where the ALL_EQUAL_SURF_DECREASING $([1, 5, 5, 2, 2, 3, 4, 6, 1, 2, 3, 4, 5, 2, 2, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

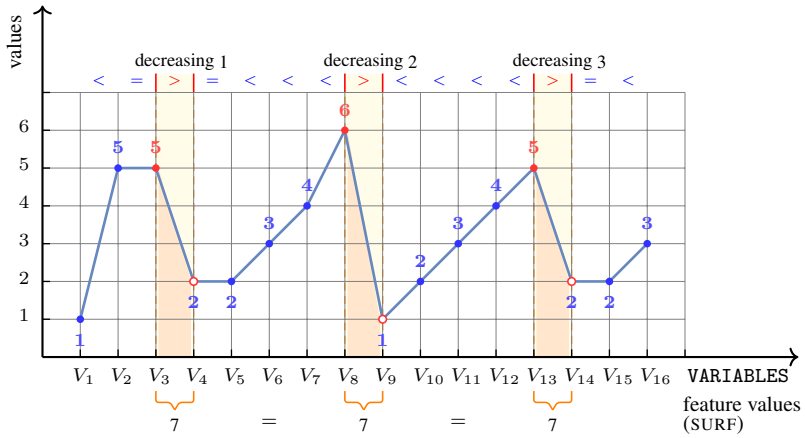


Figure 4.79: Illustrating the ALL_EQUAL_SURF_DECREASING constraint of the **Example** slot

Automaton Figure 4.80 depicts the automaton associated with the constraint ALL_EQUAL_SURF_DECREASING.

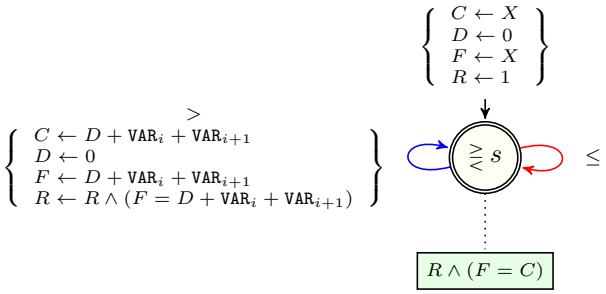
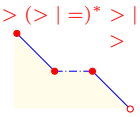


Figure 4.80: Automaton for the ALL_EQUAL_SURF_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_DECREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 1, 6, 6, 4, 3, 3, 2, 2, 5, 6, 6, 5, 5, 2, 3, 3\rangle)$

Figure 4.81 provides an example where the ALL_EQUAL_SURF_DECREASING_SEQUENCE $([1, 6, 6, 4, 3, 3, 2, 2, 5, 6, 6, 5, 5, 2, 3, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

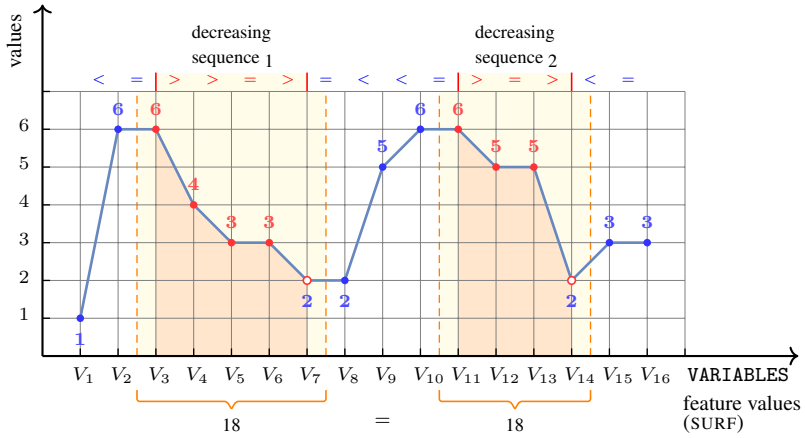


Figure 4.81: Illustrating the ALL_EQUAL_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.82 depicts the automaton associated with the constraint ALL_EQUAL_SURF_DECREASING_SEQUENCE.

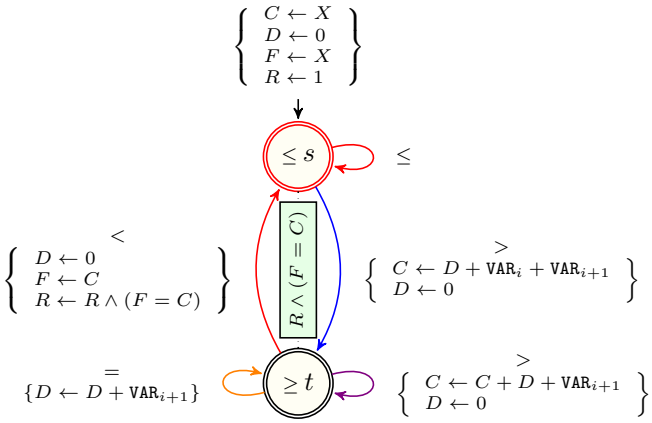
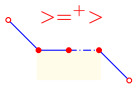


Figure 4.82: Automaton for the ALL_EQUAL_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_DECREASING_TERRACE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint ALL_EQUAL_SURF_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 5,4,4,1,6,6,2,2,2,2,1,6,4,4,2,2\rangle)$

Figure 4.83 provides an example where the ALL_EQUAL_SURF_DECREASING_TERRACE $([5,4,4,1,6,6,2,2,2,2,1,6,4,4,2,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

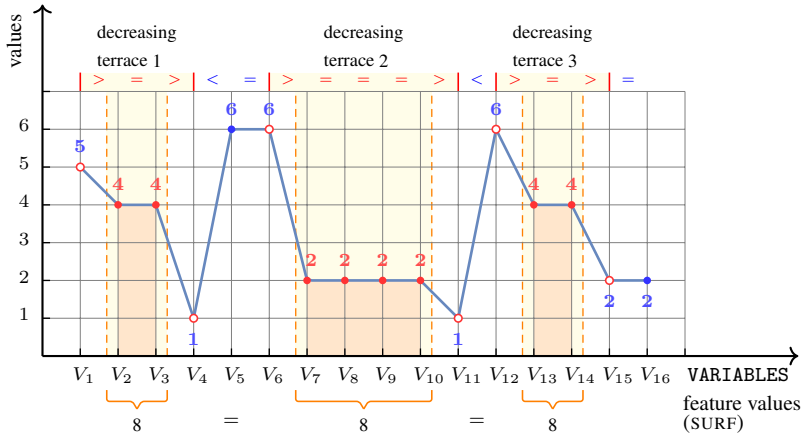


Figure 4.83: Illustrating the ALL_EQUAL_SURF_DECREASING_TERRACE constraint of the **Example** slot

Automaton Figure 4.84 depicts the automaton associated with the constraint ALL_EQUAL_SURF_DECREASING_TERRACE.

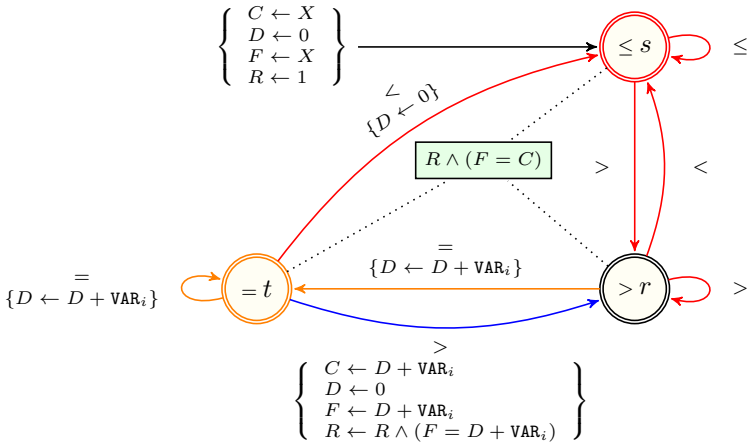
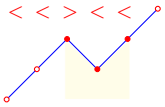


Figure 4.84: Automaton for the ALL_EQUAL_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+2$ to index j.

Example $(\langle 1,2,3,2,3,4,6,5,1,4,5,6,0,2,4,4\rangle)$

Figure 4.85 provides an example where the ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE $([1,2,3,2,3,4,6,5,1,4,5,6,0,2,4,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

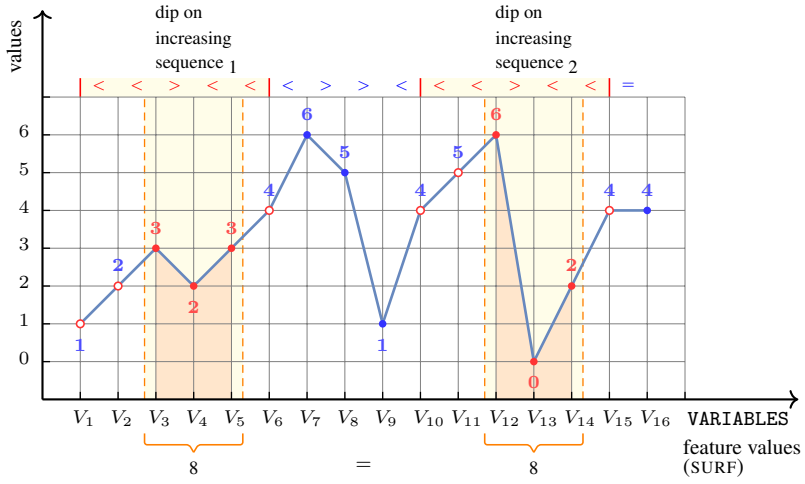


Figure 4.85: Illustrating the ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.86 depicts the automaton associated with the constraint ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE.

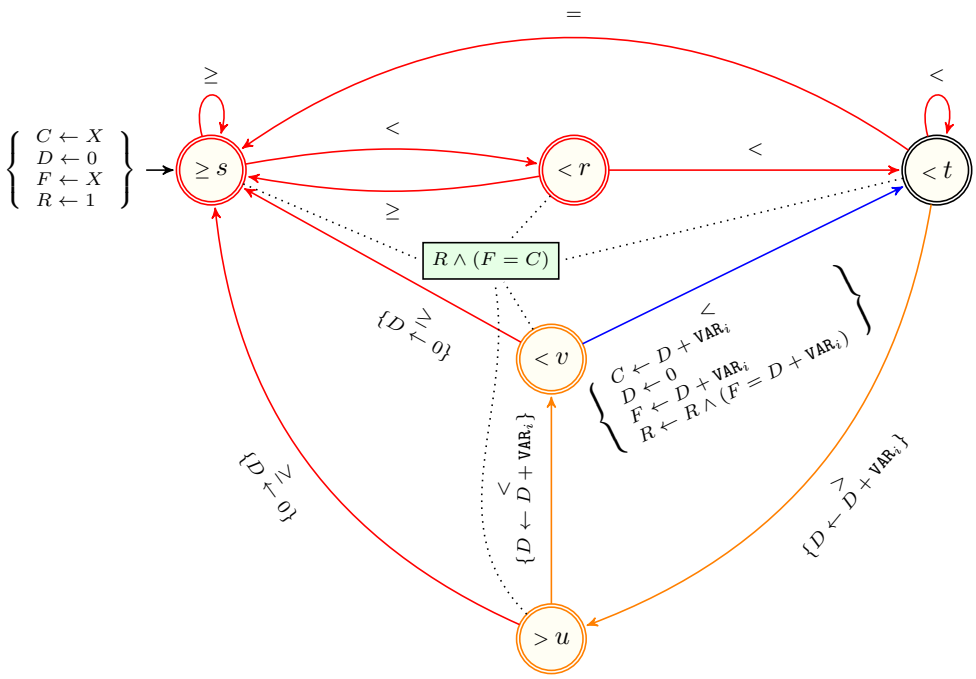
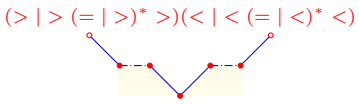


Figure 4.86: Automaton for the ALL_EQUAL_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_GORGE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint ALL_EQUAL_SURF_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the GORGE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 6,4,2,3,5,1,1,5,4,2,3,6,3,2,4,5\rangle)$

Figure 4.87 provides an example where the ALL_EQUAL_SURF_GORGE $([6,4,2,3,5,1,1,5,4,2,3,6,3,2,4,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

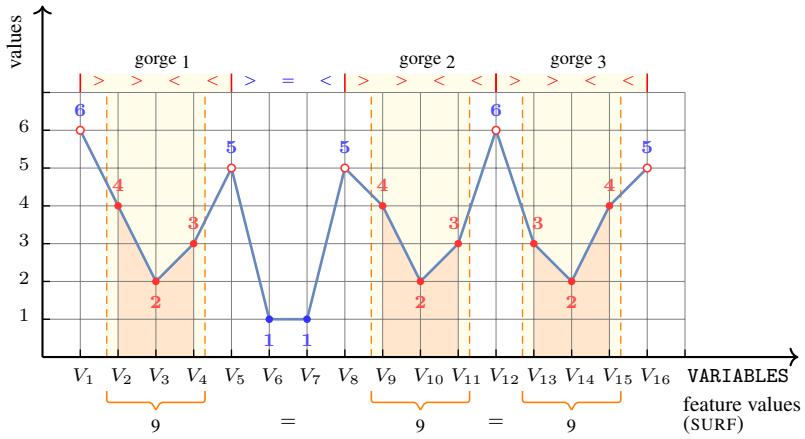


Figure 4.87: Illustrating the ALL_EQUAL_SURF_GORGE constraint of the **Example** slot

Automaton Figure 4.88 depicts the automaton associated with the constraint ALL_EQUAL_SURF_GORGE.

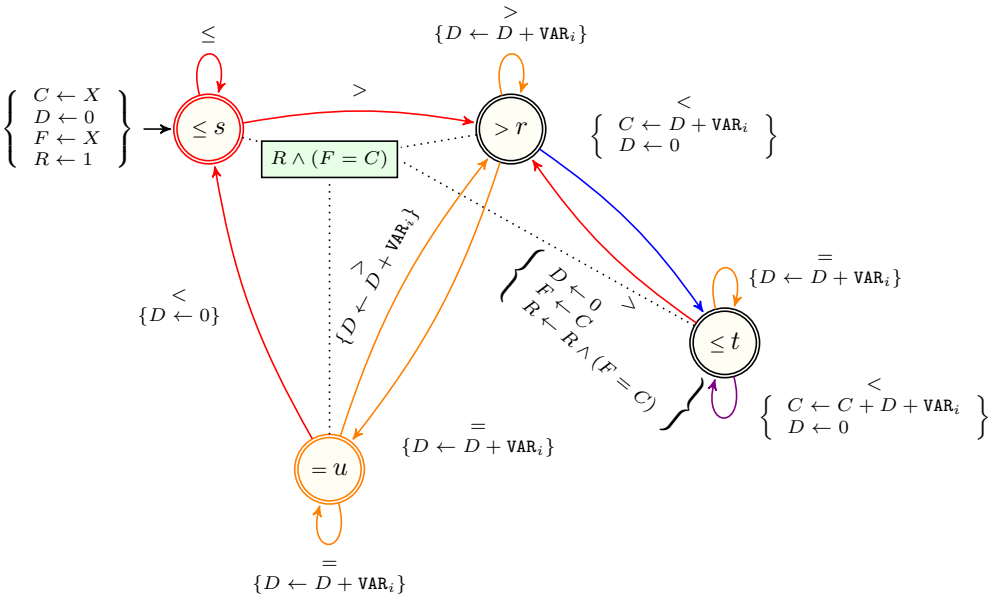


Figure 4.88: Automaton for the ALL_EQUAL_SURF_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_INCREASING

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint ALL_EQUAL_SURF_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the INCREASING pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 6,1,1,7,4,4,2,2,6,4,2,6,5,3,5,5\rangle)$

Figure 4.89 provides an example where the ALL_EQUAL_SURF_INCREASING $([6,1,1,7,4,4,2,2,6,4,2,6,5,3,5,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

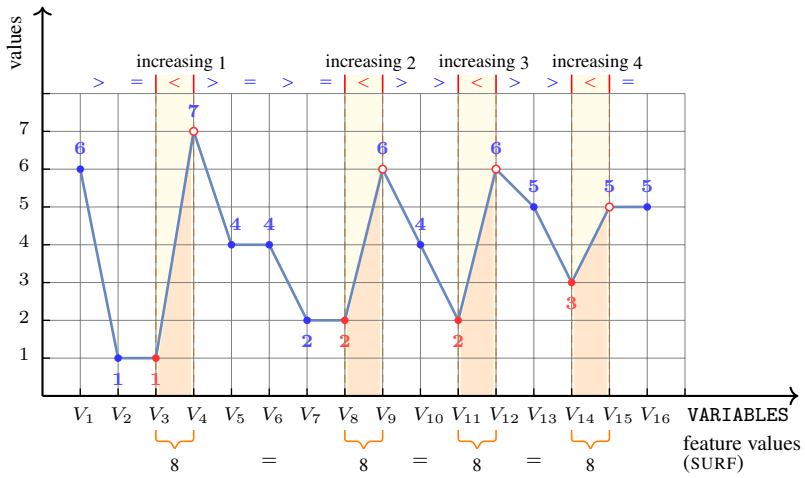


Figure 4.89: Illustrating the ALL_EQUAL_SURF_INCREASING constraint of the **Example** slot

Automaton Figure 4.90 depicts the automaton associated with the constraint ALL_EQUAL_SURF_INCREASING.

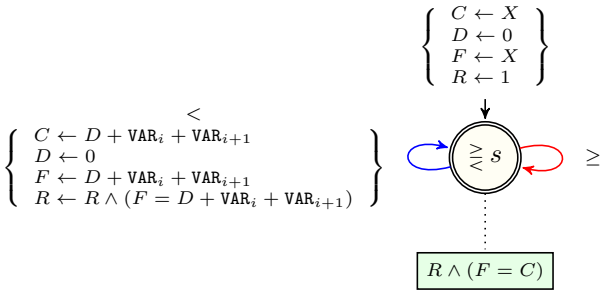
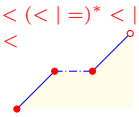


Figure 4.90: Automaton for the ALL_EQUAL_SURF_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 6, 1, 1, 3, 4, 4, 5, 5, 2, 1, 1, 5, 5, 6, 4, 4 \rangle)$

Figure 4.91 provides an example where the ALL_EQUAL_SURF_INCREASING_SEQUENCE $([6, 1, 1, 3, 4, 4, 5, 5, 2, 1, 1, 5, 5, 6, 4, 4])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

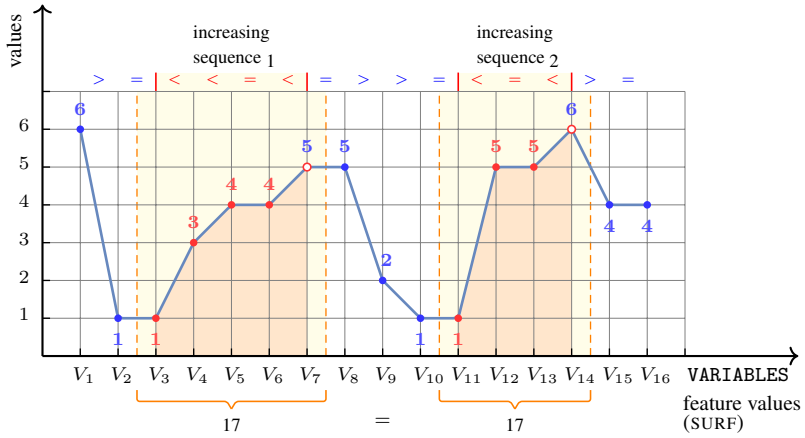


Figure 4.91: Illustrating the ALL_EQUAL_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.92 depicts the automaton associated with the constraint ALL_EQUAL_SURF_INCREASING_SEQUENCE.

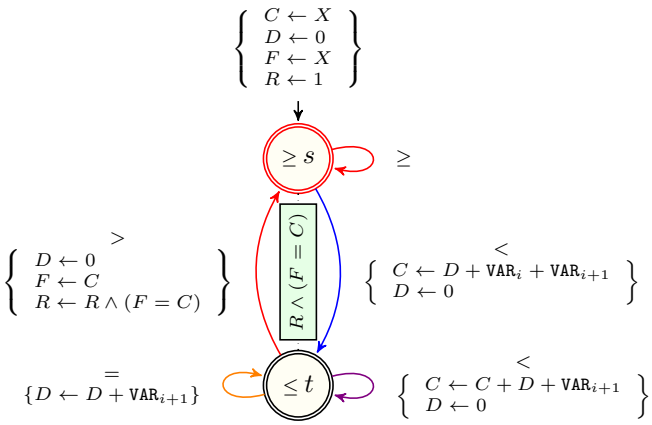
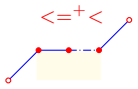


Figure 4.92: Automaton for the ALL_EQUAL_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_INCREASING_TERRACE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INCREASING_TERRACE pattern.

Constraint ALL_EQUAL_SURF_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,3,3,6,1,1,2,2,2,4,6,1,3,3,5,5\rangle)$

Figure 4.93 provides an example where the ALL_EQUAL_SURF_INCREASING_TERRACE $([2,3,3,6,1,1,2,2,2,4,6,1,3,3,5,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

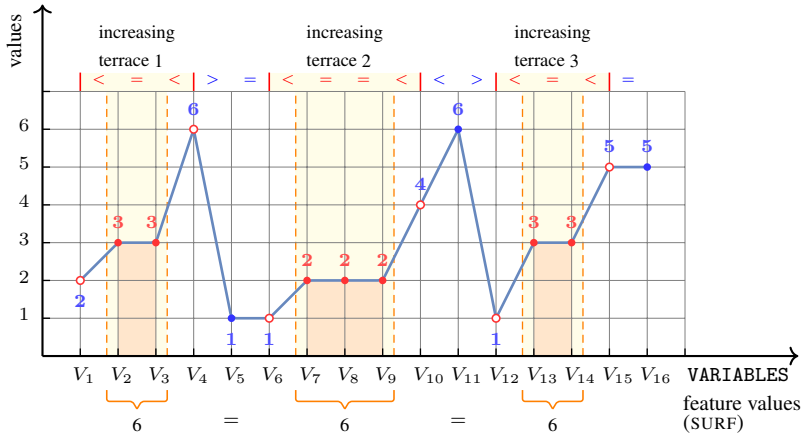


Figure 4.93: Illustrating the ALL_EQUAL_SURF_INCREASING_TERRACE constraint of the **Example** slot

Automaton Figure 4.94 depicts the automaton associated with the constraint ALL_EQUAL_SURF_INCREASING_TERRACE.

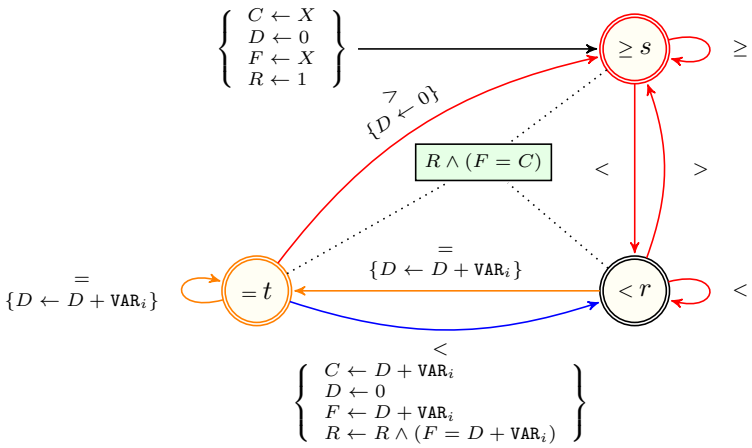
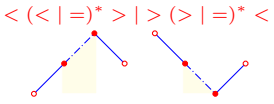


Figure 4.94: Automaton for the ALL_EQUAL_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_INFLEXION

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint ALL_EQUAL_SURF_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,2,3,6,6,5,5,3,2,4,5,6,3,3,3,3\rangle)$

Figure 4.95 provides an example where the ALL_EQUAL_SURF_INFLEXION $([2,2,3,6,6,5,5,3,2,4,5,6,3,3,3,3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

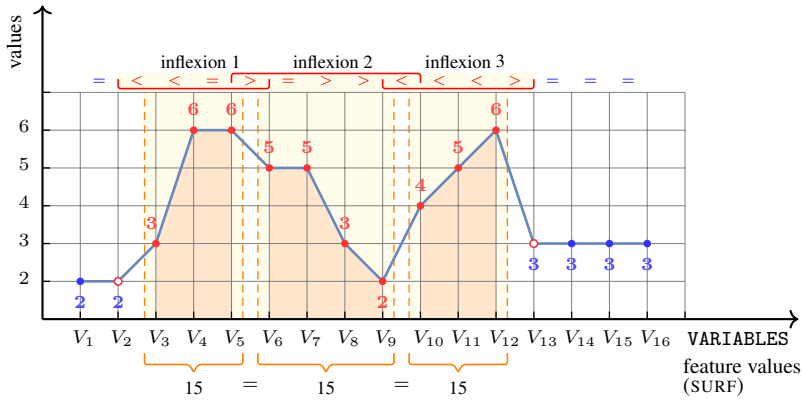


Figure 4.95: Illustrating the ALL_EQUAL_SURF_INFLEXION constraint of the **Example** slot

Automaton Figure 4.96 depicts the automaton associated with the constraint ALL_EQUAL_SURF_INFLEXION.

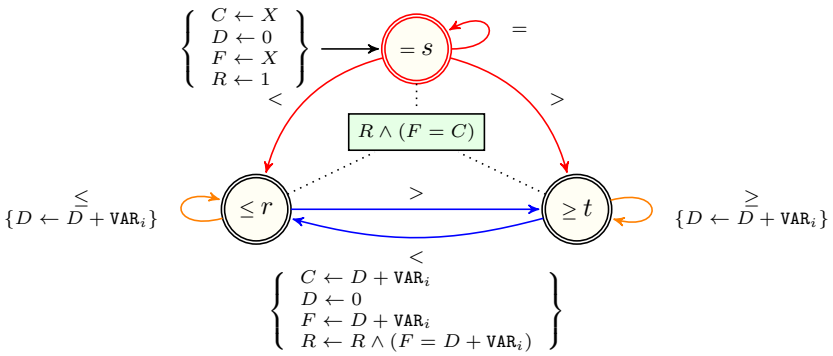
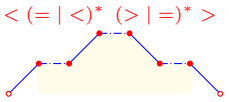


Figure 4.96: Automaton for the ALL_EQUAL_SURF_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_PEAK

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint ALL_EQUAL_SURF_PEAK(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the PEAK pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1 \rangle)$

Figure 4.97 provides an example where the ALL_EQUAL_SURF_PEAK $([1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

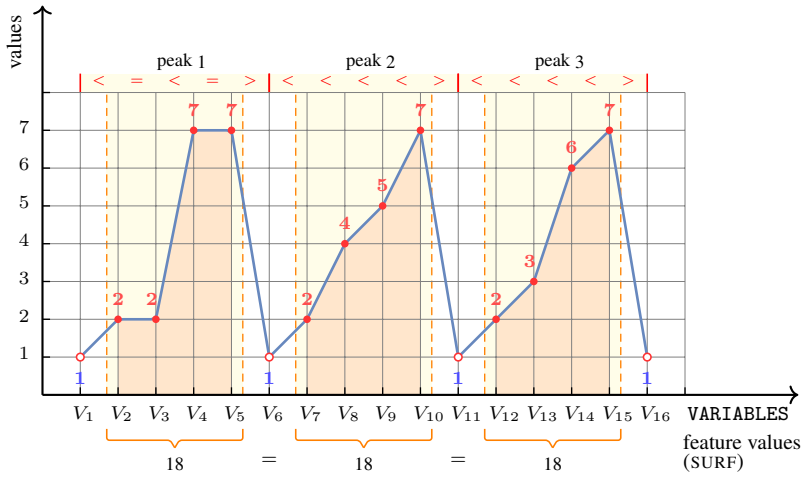


Figure 4.97: Illustrating the ALL_EQUAL_SURF_PEAK constraint of the **Example** slot

Automaton Figure 4.98 depicts the automaton associated with the constraint ALL_EQUAL_SURF_PEAK.

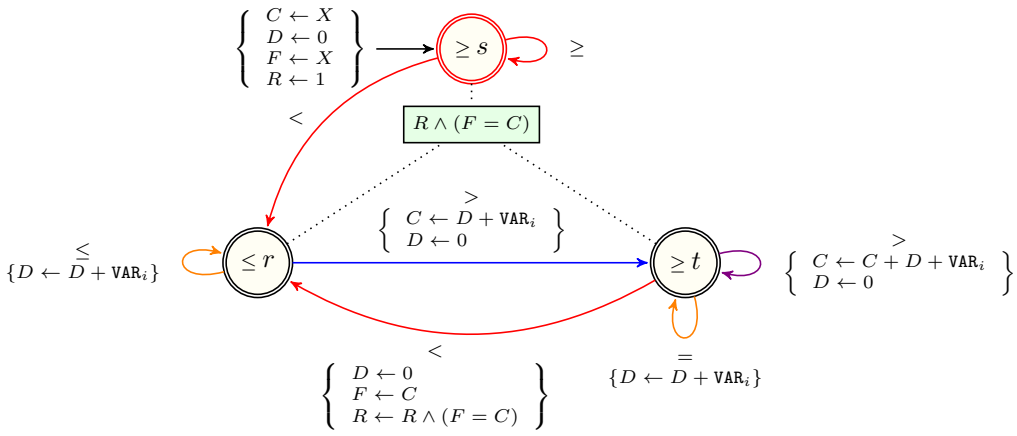


Figure 4.98: Automaton for the ALL_EQUAL_SURF_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_PLAIN

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint ALL_EQUAL_SURF_PLAIN(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1,6,3,3,7,6,6,3,3,5,5,4,3,3,6,3\rangle)$

Figure 4.99 provides an example where the ALL_EQUAL_SURF_PLAIN $([1,6,3,3,7,6,6,3,3,5,5,4,3,3,6,3])$ constraint holds.

Typical
$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

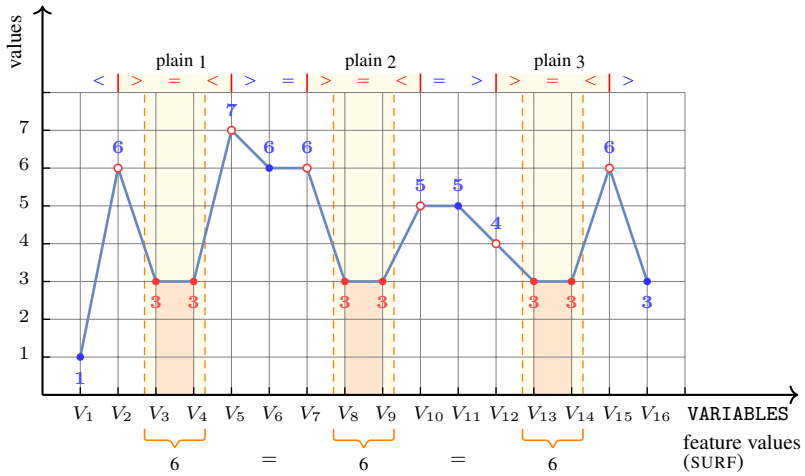


Figure 4.99: Illustrating the ALL_EQUAL_SURF_PLAIN constraint of the **Example** slot

Automaton Figure 4.100 depicts the automaton associated with the constraint ALL_EQUAL_SURF_PLAIN.

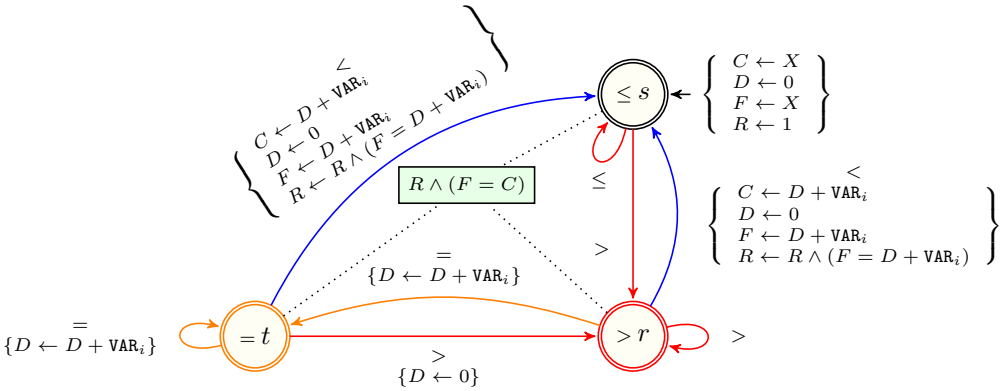


Figure 4.100: Automaton for the ALL_EQUAL_SURF_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_PLATEAU

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint ALL_EQUAL_SURF_PLATEAU(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,2,5,5,1,2,2,5,5,3,3,4,5,5,2,5\rangle)$

Figure 4.101 provides an example where the ALL_EQUAL_SURF_PLATEAU $([7,2,5,5,1,2,2,5,5,3,3,4,5,5,2,5])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

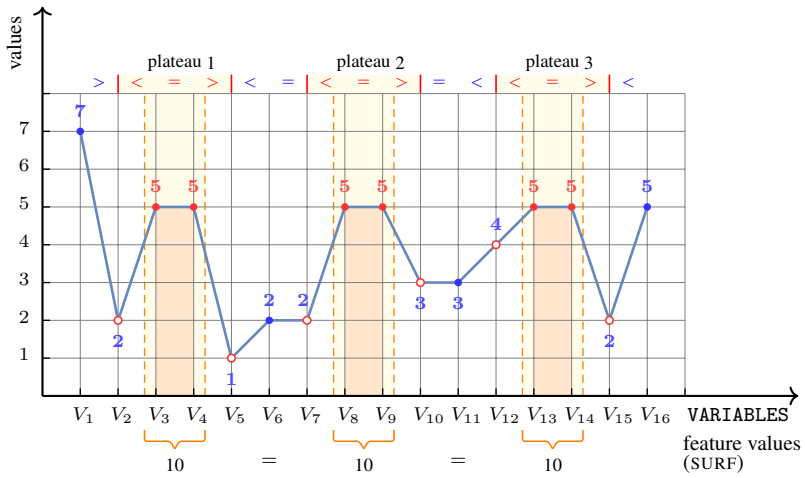


Figure 4.101: Illustrating the ALL_EQUAL_SURF_PLATEAU constraint of the **Example** slot

Automaton Figure 4.102 depicts the automaton associated with the constraint ALL_EQUAL_SURF_PLATEAU.

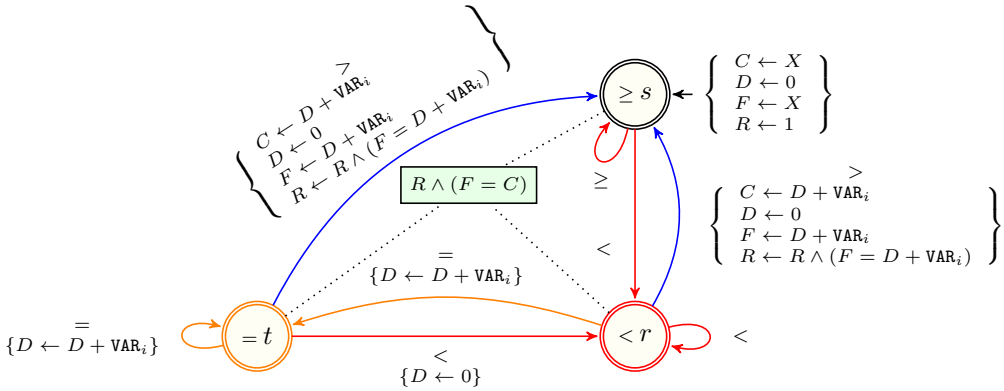
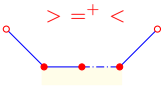


Figure 4.102: Automaton for the ALL_EQUAL_SURF_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_PROPER_PLAIN

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint ALL_EQUAL_SURF_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,7,3,3,6,6,3,7,3,3,5,6,5,3,3,5\rangle)$

Figure 4.103 provides an example where the ALL_EQUAL_SURF_PROPER_PLAIN $([2,7,3,3,6,6,3,7,3,3,5,6,5,3,3,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

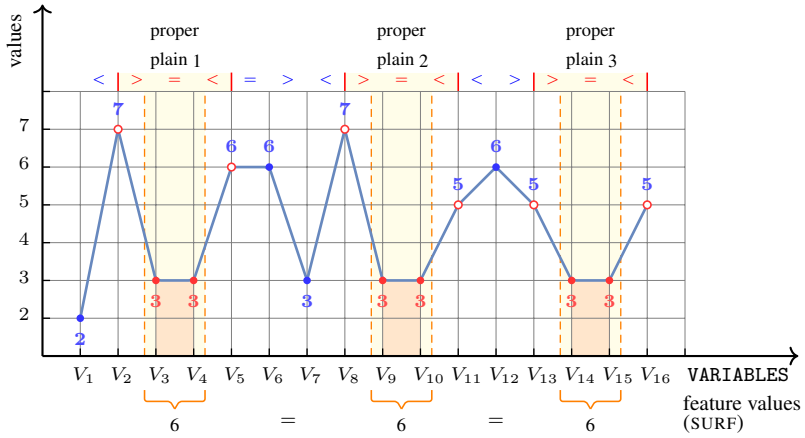


Figure 4.103: Illustrating the ALL_EQUAL_SURF_PROPER_PLAIN constraint of the **Example** slot

Automaton Figure 4.104 depicts the automaton associated with the constraint ALL_EQUAL_SURF_PROPER_PLAIN.

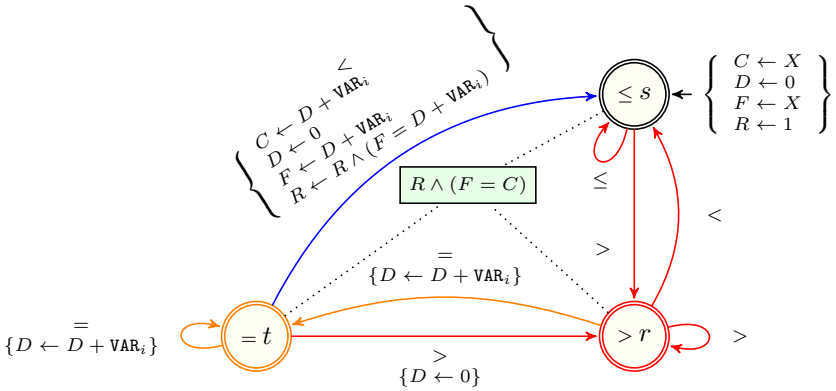
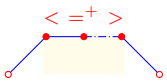


Figure 4.104: Automaton for the ALL_EQUAL_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_PROPER_PLATEAU

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint ALL_EQUAL_SURF_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,1,5,5,2,2,5,1,5,5,3,2,3,5,5,3\rangle)$

Figure 4.105 provides an example where the ALL_EQUAL_SURF_PROPER_PLATEAU $([7,1,5,5,2,2,5,1,5,5,3,2,3,5,5,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

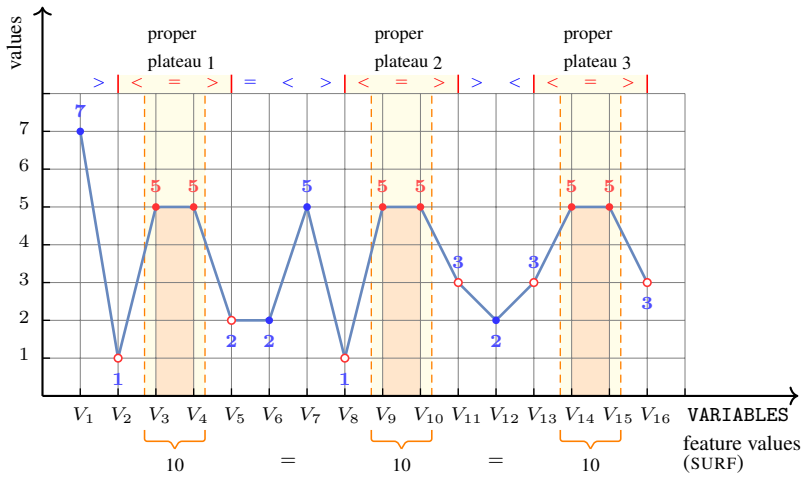


Figure 4.105: Illustrating the ALL_EQUAL_SURF_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.106 depicts the automaton associated with the constraint ALL_EQUAL_SURF_PROPER_PLATEAU.

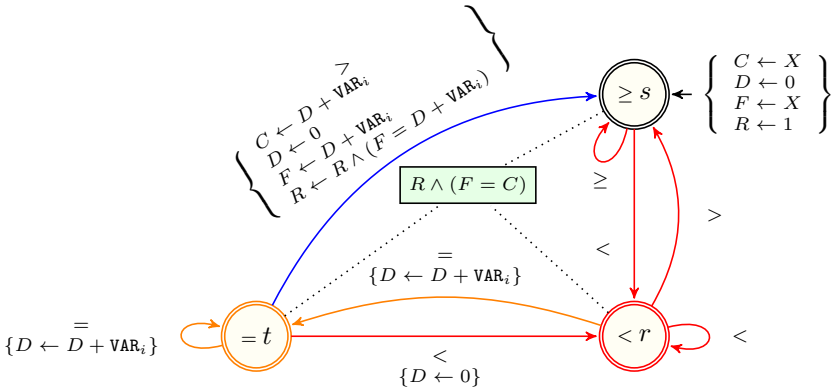


Figure 4.106: Automaton for the ALL_EQUAL_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_STEADY

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint ALL_EQUAL_SURF_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the STEADY pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 4, 4, 4, 2, 6, 4, 4, 4, 4, 3, 2, 1, 4, 4, 3, 4\rangle)$

Figure 4.107 provides an example where the ALL_EQUAL_SURF_STEADY $([4, 4, 4, 2, 6, 4, 4, 4, 4, 3, 2, 1, 4, 4, 3, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

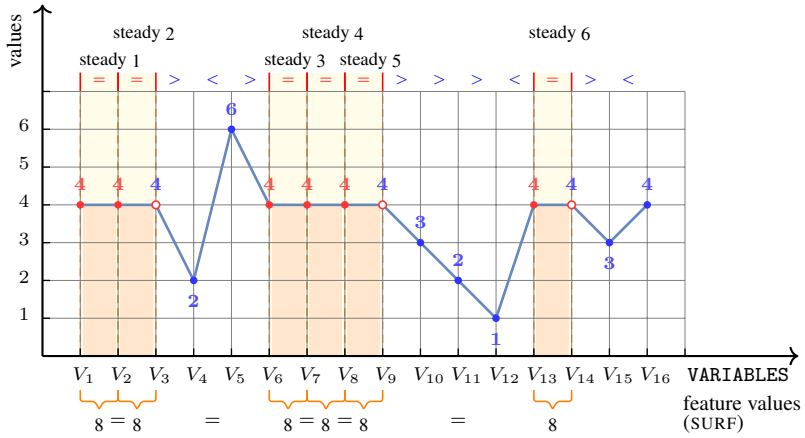


Figure 4.107: Illustrating the ALL_EQUAL_SURF_STEADY constraint of the **Example** slot

Automaton Figure 4.108 depicts the automaton associated with the constraint ALL_EQUAL_SURF_STEADY.

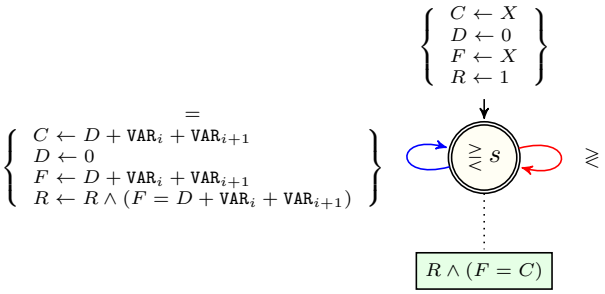
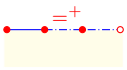


Figure 4.108: Automaton for the ALL_EQUAL_SURF_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_STEADY_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 6,6,5,3,3,3,3,1,4,4,4,5,6,6,7,2\rangle)$

Figure 4.109 provides an example where the ALL_EQUAL_SURF_STEADY_SEQUENCE $([6,6,5,3,3,3,3,1,4,4,4,5,6,6,7,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

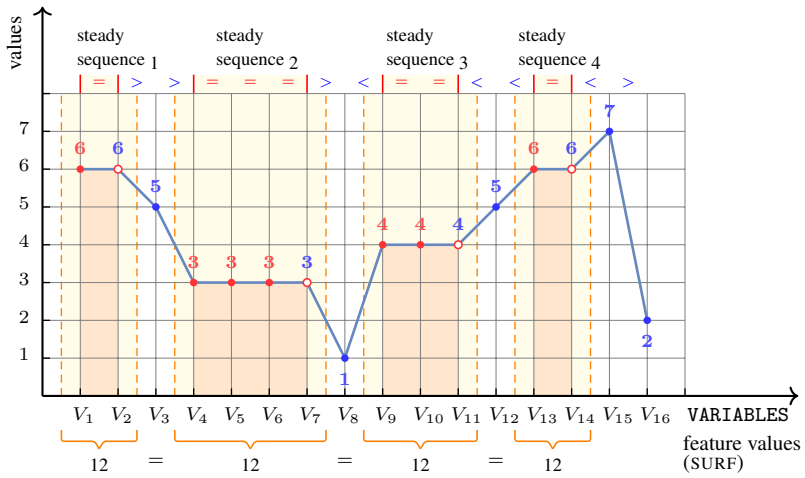


Figure 4.109: Illustrating the ALL_EQUAL_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.110 depicts the automaton associated with the constraint ALL_EQUAL_SURF_STEADY_SEQUENCE.

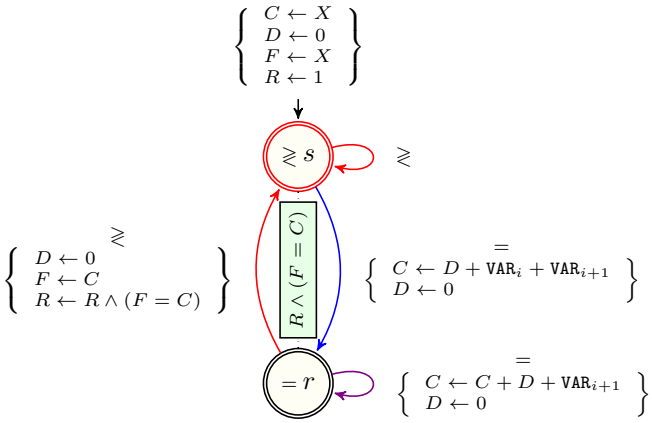


Figure 4.110: Automaton for the ALL_EQUAL_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 3,6,5,4,5,5,6,6,5,3,1,2,6,5,3,1\rangle)$

Figure 4.111 provides an example where the ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE $([3,6,5,4,5,5,6,6,5,3,1,2,6,5,3,1])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

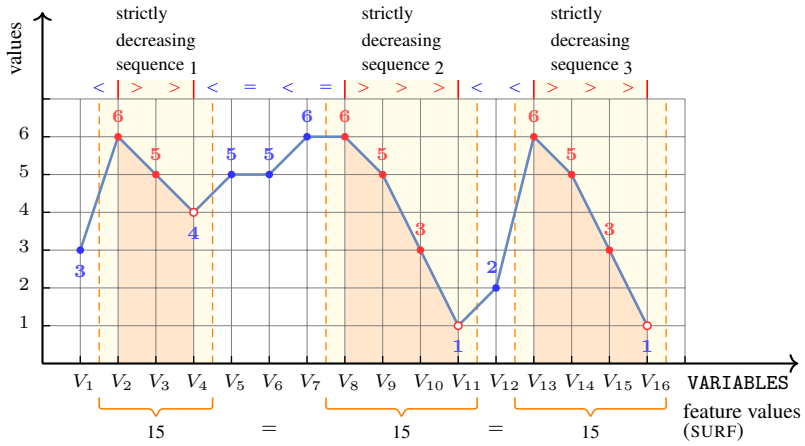


Figure 4.111: Illustrating the ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.112 depicts the automaton associated with the constraint ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE.

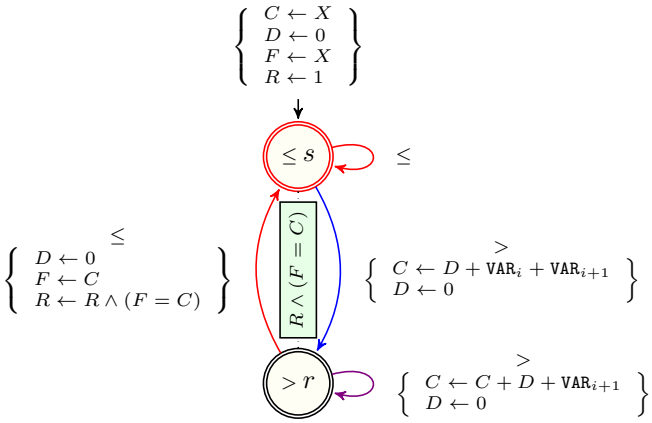


Figure 4.112: Automaton for the ALL_EQUAL_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 6, 1, 1, 2, 3, 4, 5, 5, 2, 1, 1, 3, 5, 6, 4, 4\rangle)$

Figure 4.113 provides an example where the ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE $([6, 1, 1, 2, 3, 4, 5, 5, 2, 1, 1, 3, 5, 6, 4, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

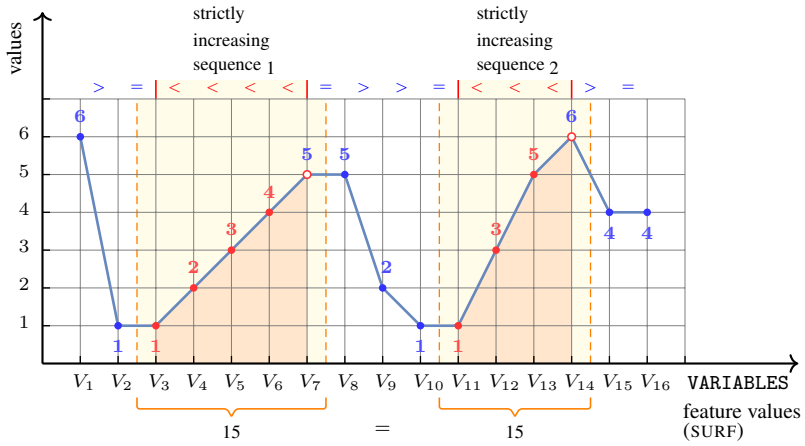


Figure 4.113: Illustrating the ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.114 depicts the automaton associated with the constraint ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE.

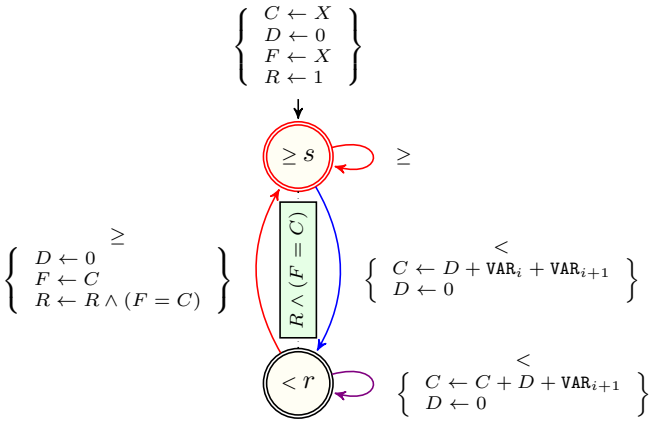


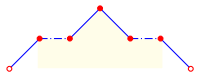
Figure 4.114: Automaton for the ALL_EQUAL_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the SUMMIT pattern.

Constraint ALL_EQUAL_SURF_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2 \rangle)$

Figure 4.115 provides an example where the ALL_EQUAL_SURF_SUMMIT $([1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

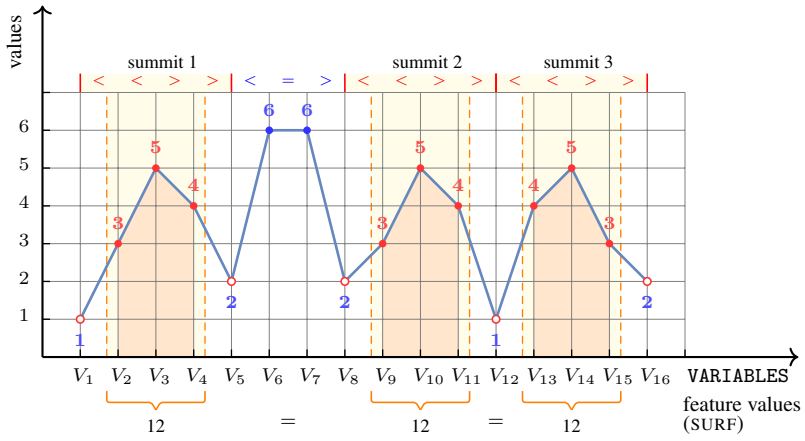


Figure 4.115: Illustrating the ALL_EQUAL_SURF_SUMMIT constraint of the **Example** slot

Automaton Figure 4.116 depicts the automaton associated with the constraint ALL_EQUAL_SURF_SUMMIT.

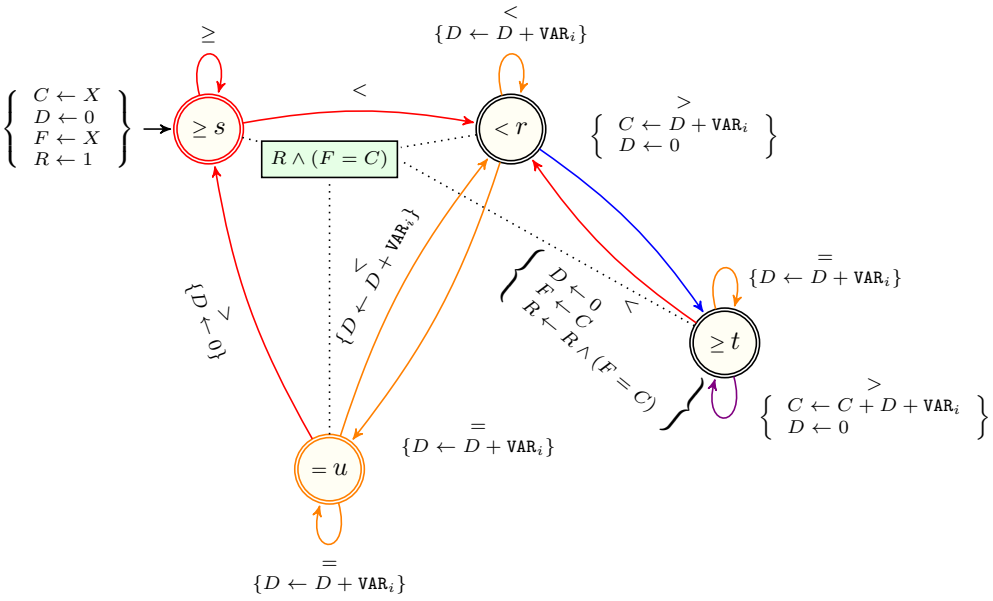
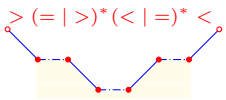


Figure 4.116: Automaton for the ALL_EQUAL_SURF_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_VALLEY

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint ALL_EQUAL_SURF_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,6,6,1,1,7,6,4,3,1,7,6,5,2,1,7\rangle)$

Figure 4.117 provides an example where the ALL_EQUAL_SURF_VALLEY $([7,6,6,1,1,7,6,4,3,1,7,6,5,2,1,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

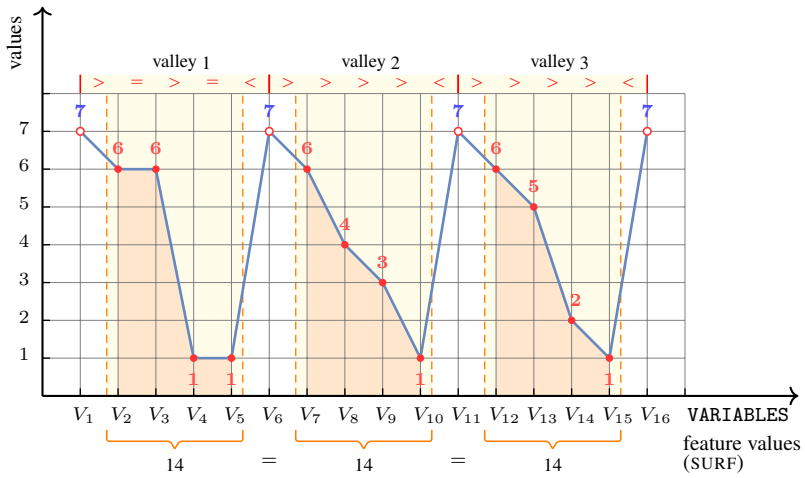


Figure 4.117: Illustrating the ALL_EQUAL_SURF_VALLEY constraint of the **Example** slot

Automaton Figure 4.118 depicts the automaton associated with the constraint ALL_EQUAL_SURF_VALLEY.

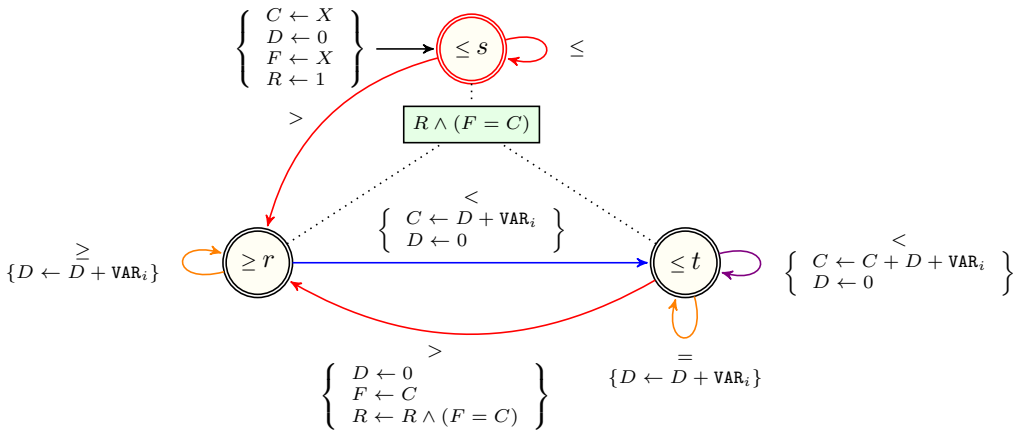
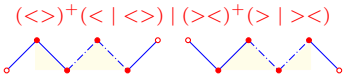


Figure 4.118: Automaton for the ALL_EQUAL_SURF_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_SURF_ZIGZAG

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint ALL_EQUAL_SURF_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the surface of all occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^{+}(< \mid <>) \mid (><)^{+}(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 6, 1, 2, 6, 5, 1, 2, 1, 2, 1, 4, 5, 2, 4, 7\rangle)$

Figure 4.119 provides an example where the ALL_EQUAL_SURF_ZIGZAG $([1, 6, 1, 2, 6, 5, 1, 2, 1, 2, 1, 4, 5, 2, 4, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

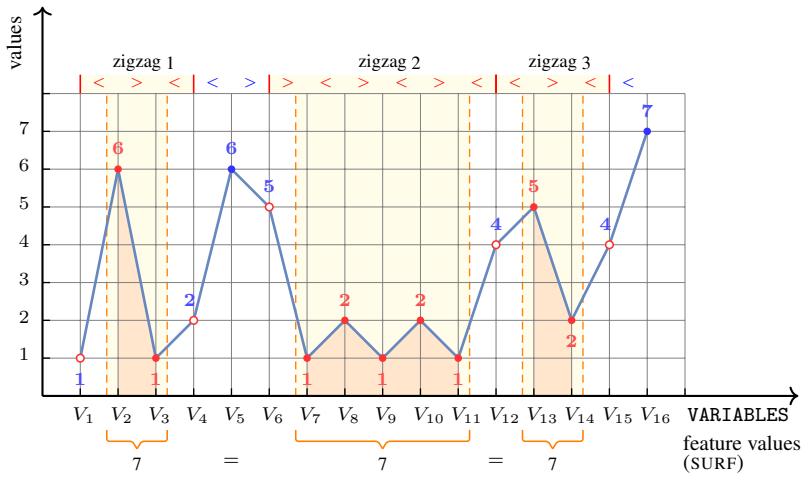


Figure 4.119: Illustrating the ALL_EQUAL_SURF_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.120 depicts the automaton associated with the constraint ALL_EQUAL_SURF_ZIGZAG.

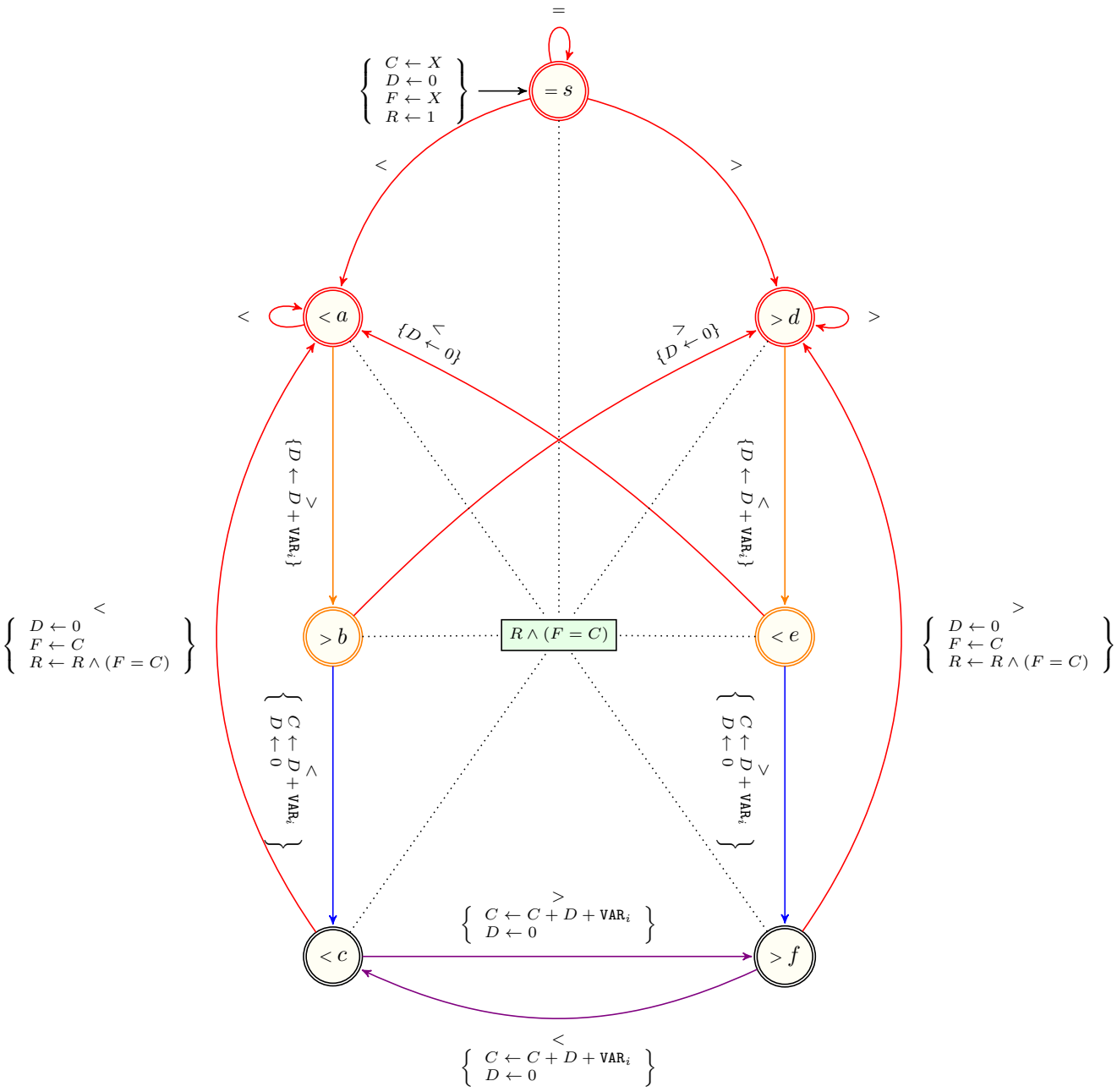
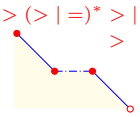


Figure 4.120: Automaton for the ALL_EQUAL_SURF_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_DECREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_WIDTH_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 1, 3, 3, 2, 2, 1, 1, 4, 6, 6, 4, 4, 2, 2, 3, 4 \rangle)$

Figure 4.121 provides an example where the ALL_EQUAL_WIDTH_DECREASING_SEQUENCE $([1, 3, 3, 2, 2, 1, 1, 4, 6, 6, 4, 4, 2, 2, 3, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

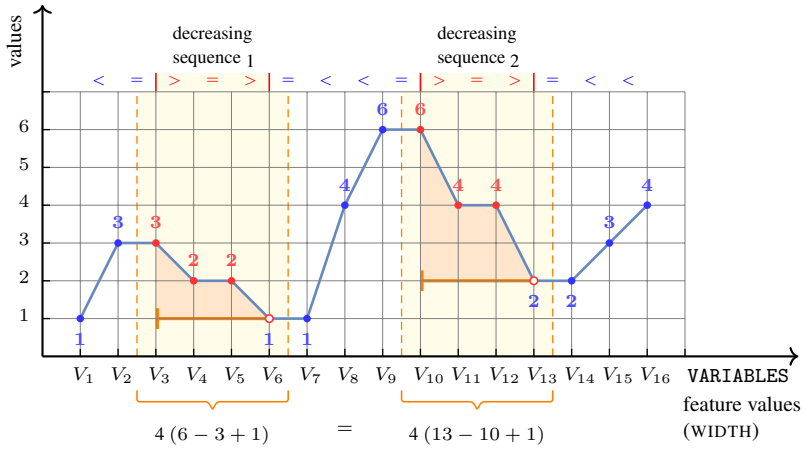


Figure 4.121: Illustrating the ALL_EQUAL_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.122 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_DECREASING_SEQUENCE.

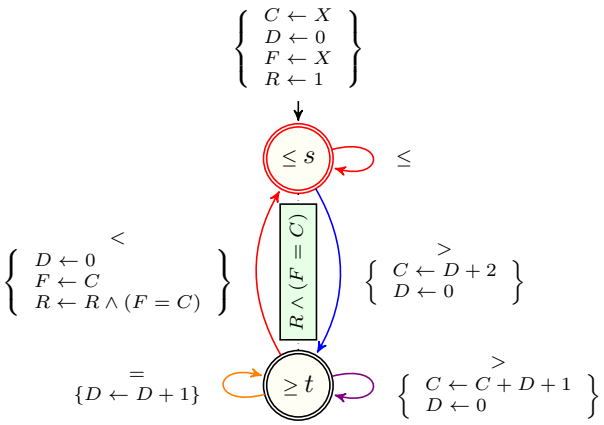
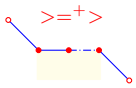


Figure 4.122: Automaton for the ALL_EQUAL_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_DECREASING_TERRACE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint ALL_EQUAL_WIDTH_DECREASING_TERRACE(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 5, 5, 4, 3, 3, 1, 1, 3, 3, 5, 2, 2, 1, 1, 1\rangle)$

Figure 4.123 provides an example where the ALL_EQUAL_WIDTH_DECREASING_TERRACE $([6, 5, 5, 4, 3, 3, 1, 1, 3, 3, 5, 2, 2, 1, 1, 1])$ constraint holds.

Typical
$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

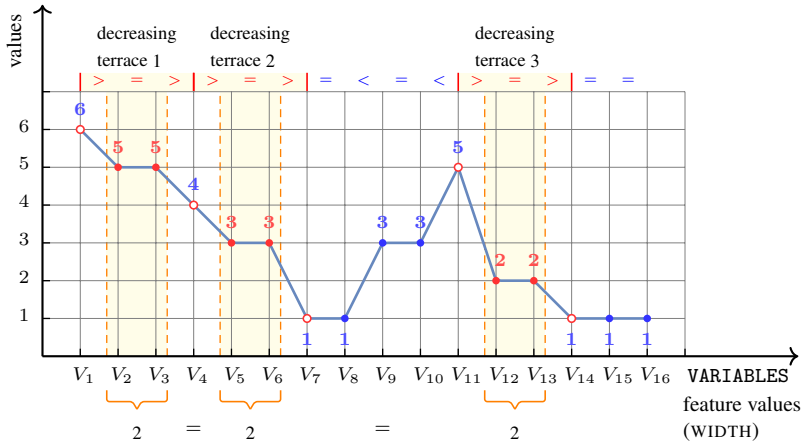


Figure 4.123: Illustrating the ALL_EQUAL_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.124 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_DECREASING_TERRACE.

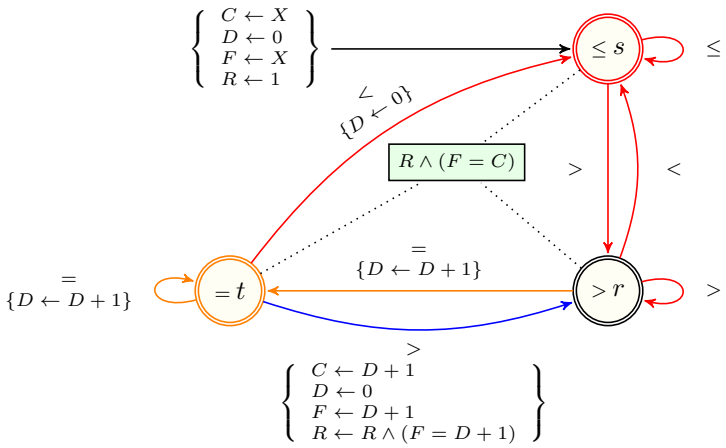


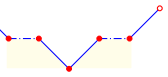
Figure 4.124: Automaton for the ALL_EQUAL_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_GORGE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$



Origin Based on the GORGE pattern.

Constraint ALL_EQUAL_WIDTH_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the GORGE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 4, 2, 3, 5, 1, 1, 5, 4, 2, 3, 6, 3, 2, 4, 5 \rangle)$

Figure 4.125 provides an example where the ALL_EQUAL_WIDTH_GORGE $([6, 4, 2, 3, 5, 1, 1, 5, 4, 2, 3, 6, 3, 2, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

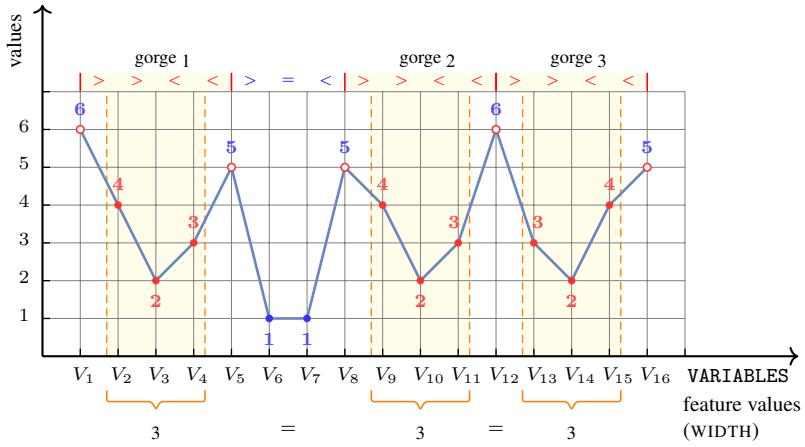


Figure 4.125: Illustrating the ALL_EQUAL_WIDTH_GORGE constraint of the **Example** slot

Automaton Figure 4.126 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_GORGE.

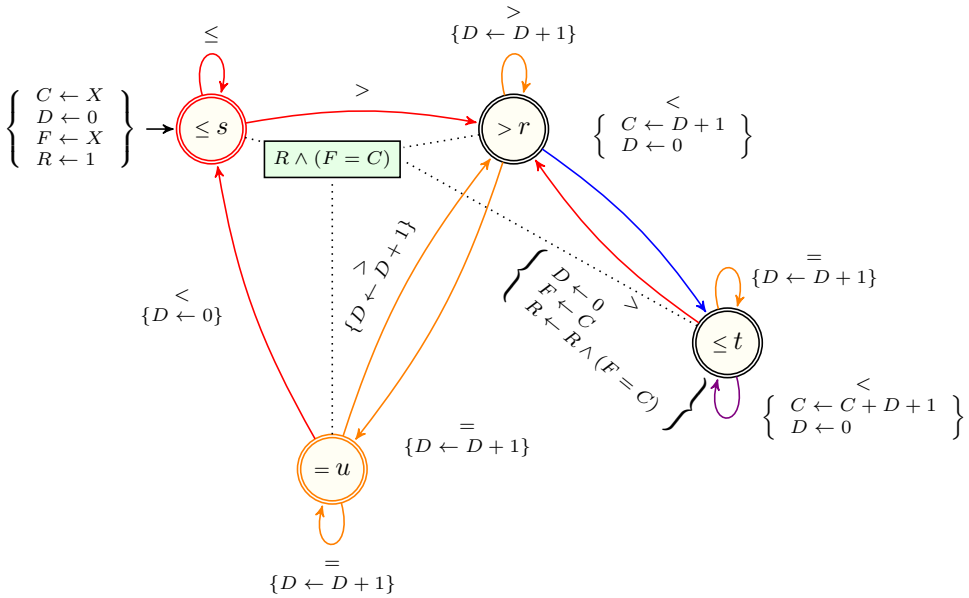
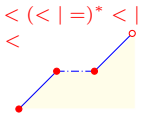


Figure 4.126: Automaton for the ALL_EQUAL_WIDTH_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_INCREASING_SEQUENCE



► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_WIDTH_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 6, 4, 4, 5, 5, 6, 6, 3, 1, 1, 3, 3, 5, 5, 4, 3 \rangle)$

Figure 4.127 provides an example where the ALL_EQUAL_WIDTH_INCREASING_SEQUENCE $([6, 4, 4, 5, 5, 6, 6, 3, 1, 1, 3, 3, 5, 5, 4, 3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

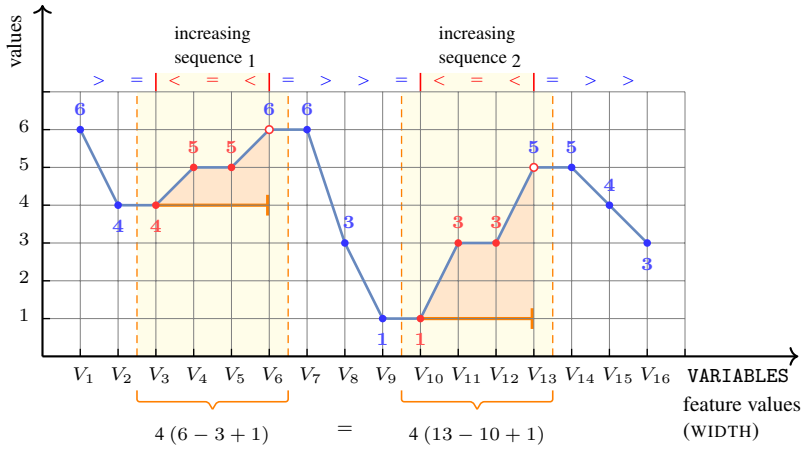


Figure 4.127: Illustrating the ALL_EQUAL_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.128 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_INCREASING_SEQUENCE.

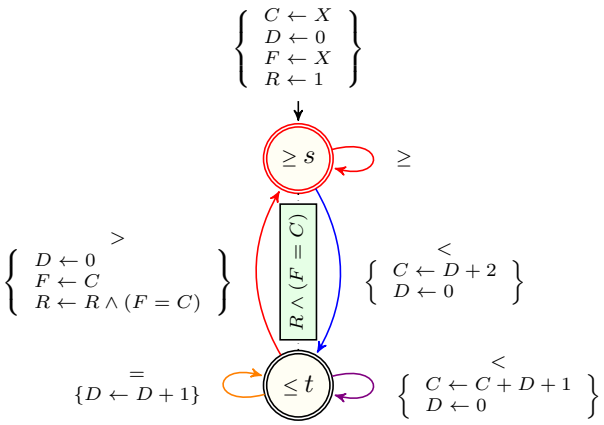
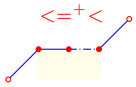


Figure 4.128: Automaton for the ALL_EQUAL_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_INCREASING_TERRACE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint ALL_EQUAL_WIDTH_INCREASING_TERRACE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 2, 2, 3, 4, 4, 6, 6, 4, 4, 2, 5, 5, 6, 6, 6\rangle)$

Figure 4.129 provides an example where the ALL_EQUAL_WIDTH_INCREASING_TERRACE $([1, 2, 2, 3, 4, 4, 6, 6, 4, 4, 2, 5, 5, 6, 6, 6])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

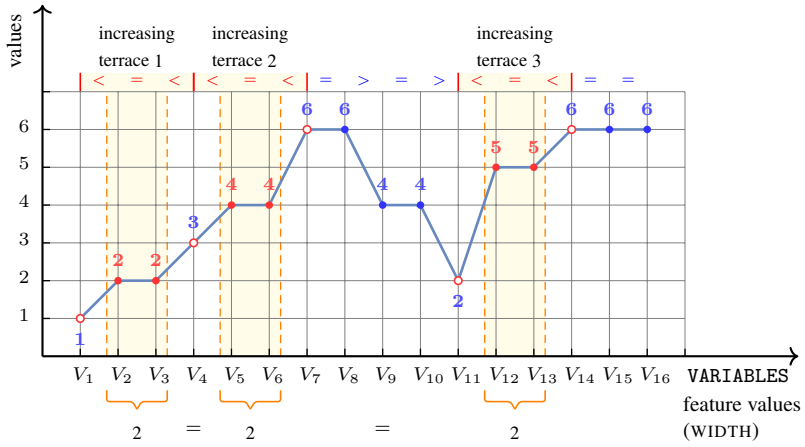


Figure 4.129: Illustrating the ALL_EQUAL_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.130 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_INCREASING_TERRACE.

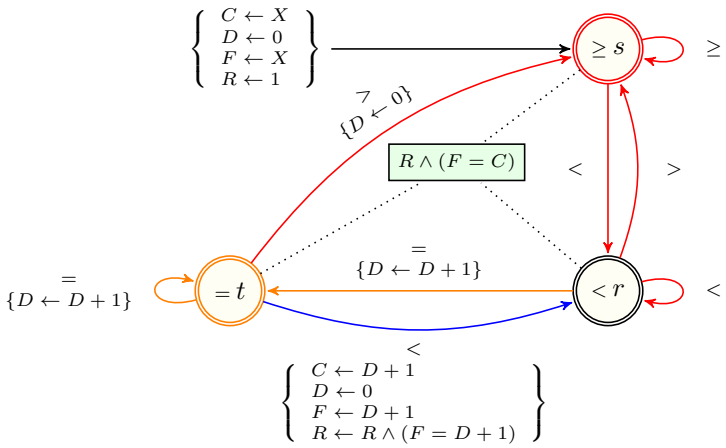
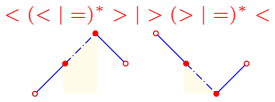


Figure 4.130: Automaton for the ALL_EQUAL_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_INFLEXION

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint ALL_EQUAL_WIDTH_INFLEXION(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 2, 2, 3, 6, 6, 5, 3, 3, 4, 4, 5, 3, 3, 3, 4, 6 \rangle)$

Figure 4.131 provides an example where the ALL_EQUAL_WIDTH_INFLEXION $([2, 2, 3, 6, 6, 5, 3, 3, 4, 4, 5, 3, 3, 3, 4, 6])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

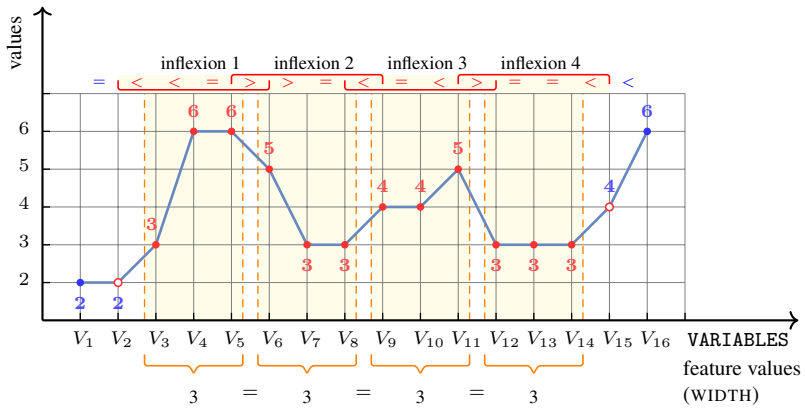


Figure 4.131: Illustrating the ALL_EQUAL_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.132 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_INFLEXION.

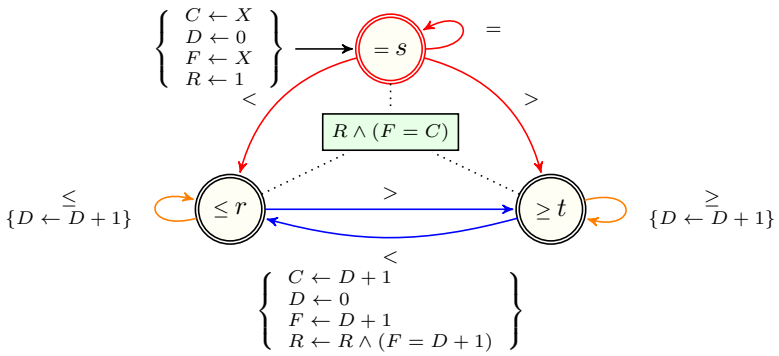
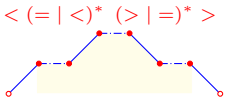


Figure 4.132: Automaton for the ALL_EQUAL_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_PEAK

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint ALL_EQUAL_WIDTH_PEAK(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the PEAK pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1 \rangle)$

Figure 4.133 provides an example where the ALL_EQUAL_WIDTH_PEAK $([1, 2, 2, 7, 7, 1, 2, 4, 5, 7, 1, 2, 3, 6, 7, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

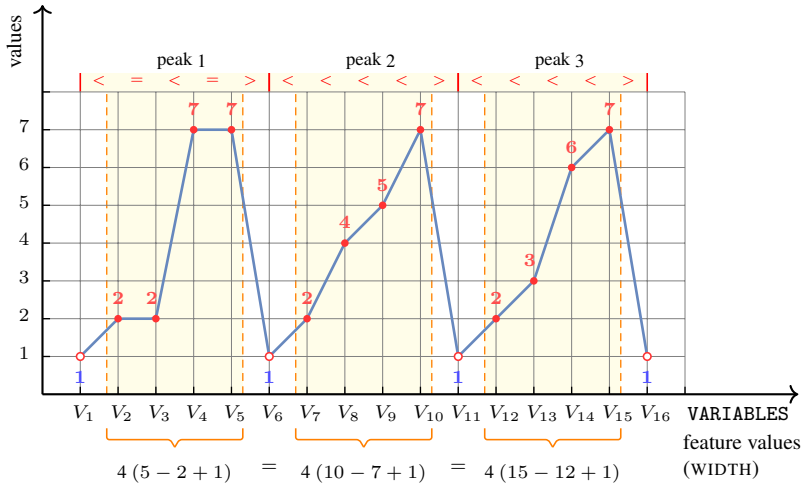


Figure 4.133: Illustrating the ALL_EQUAL_WIDTH_PEAK constraint of the **Example** slot

Automaton Figure 4.134 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_PEAK.

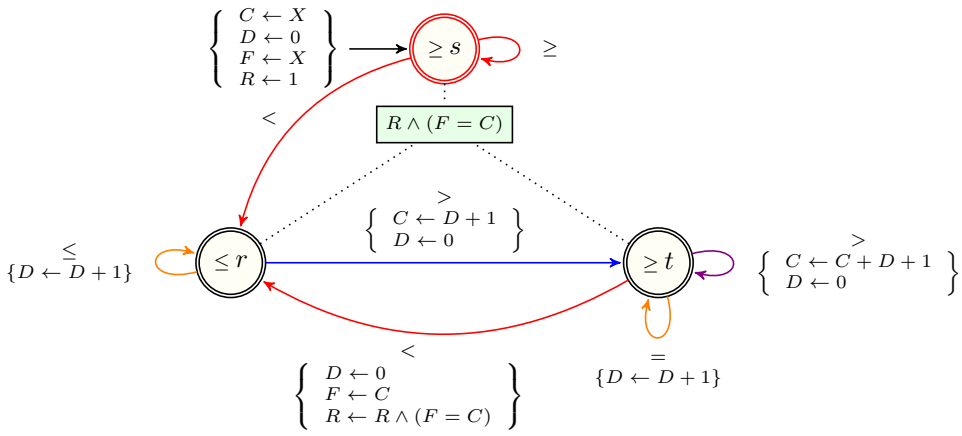


Figure 4.134: Automaton for the ALL_EQUAL_WIDTH_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_PLAIN

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint ALL_EQUAL_WIDTH_PLAIN(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 6, 3, 3, 7, 6, 6, 3, 3, 5, 5, 4, 3, 3, 6, 3 \rangle)$

Figure 4.135 provides an example where the ALL_EQUAL_WIDTH_PLAIN $([1, 6, 3, 3, 7, 6, 6, 3, 3, 5, 5, 4, 3, 3, 6, 3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

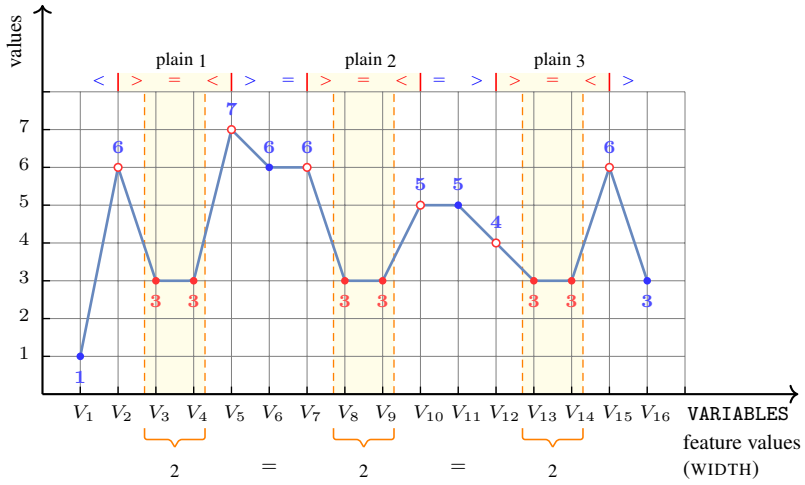


Figure 4.135: Illustrating the ALL_EQUAL_WIDTH_PLAIN constraint of the **Example** slot

Automaton Figure 4.136 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_PLAIN.

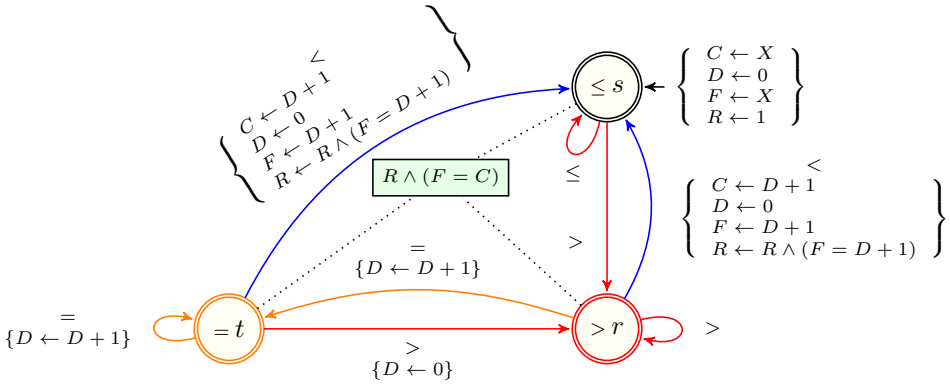


Figure 4.136: Automaton for the ALL_EQUAL_WIDTH_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_PLATEAU

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLATEAU pattern.

Constraint ALL_EQUAL_WIDTH_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 2, 5, 5, 1, 2, 2, 5, 5, 3, 3, 4, 5, 5, 2, 5\rangle)$

Figure 4.137 provides an example where the ALL_EQUAL_WIDTH_PLATEAU $([7, 2, 5, 5, 1, 2, 2, 5, 5, 3, 3, 4, 5, 5, 2, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

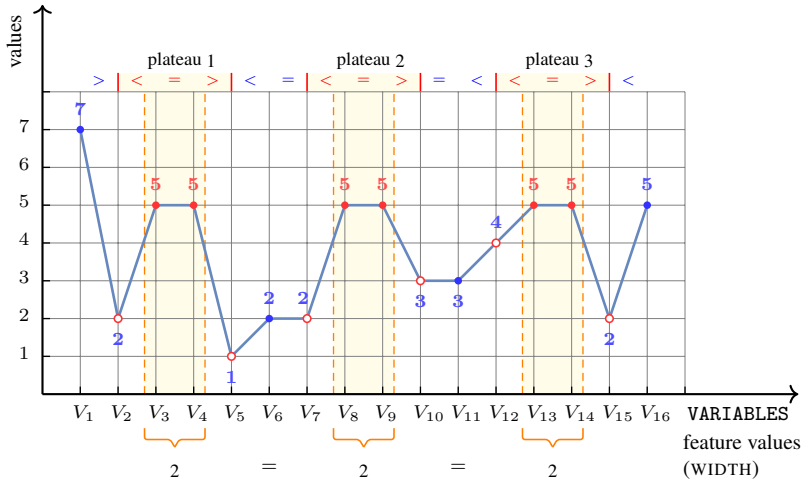


Figure 4.137: Illustrating the ALL_EQUAL_WIDTH_PLATEAU constraint of the **Example** slot

Automaton Figure 4.138 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_PLATEAU.

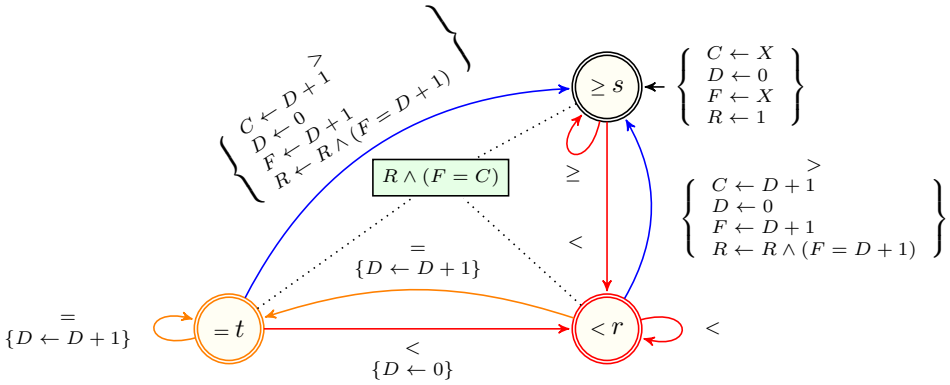
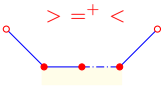


Figure 4.138: Automaton for the ALL_EQUAL_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_PROPER_PLAIN

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint ALL_EQUAL_WIDTH_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 2, 7, 3, 3, 6, 6, 3, 7, 3, 3, 5, 6, 5, 3, 3, 5\rangle)$

Figure 4.139 provides an example where the ALL_EQUAL_WIDTH_PROPER_PLAIN $([2, 7, 3, 3, 6, 6, 3, 7, 3, 3, 5, 6, 5, 3, 3, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

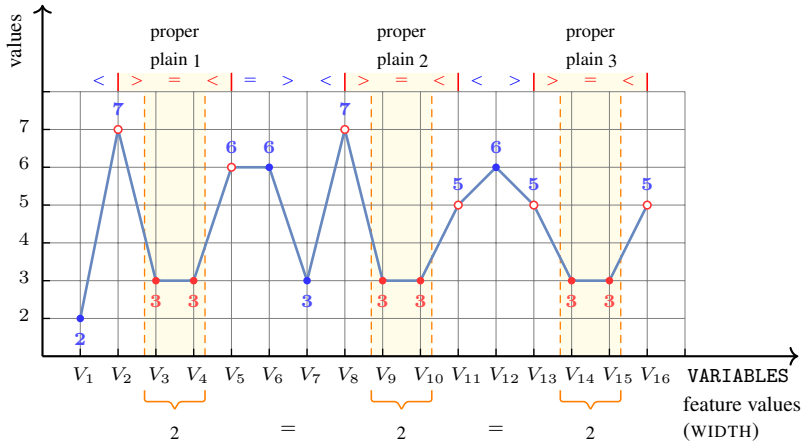


Figure 4.139: Illustrating the ALL_EQUAL_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton Figure 4.140 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_PROPER_PLAIN.

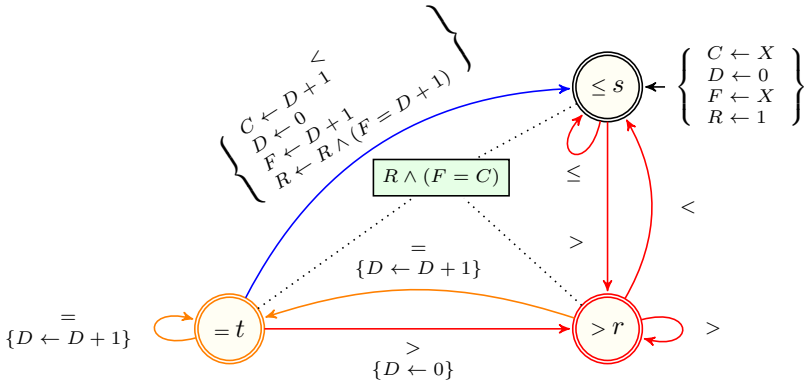
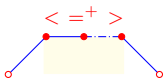


Figure 4.140: Automaton for the ALL_EQUAL_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_PROPER_PLATEAU

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint ALL_EQUAL_WIDTH_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 1, 5, 5, 2, 2, 5, 1, 5, 5, 3, 2, 3, 5, 5, 3\rangle)$

Figure 4.141 provides an example where the ALL_EQUAL_WIDTH_PROPER_PLATEAU $([7, 1, 5, 5, 2, 2, 5, 1, 5, 5, 3, 2, 3, 5, 5, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

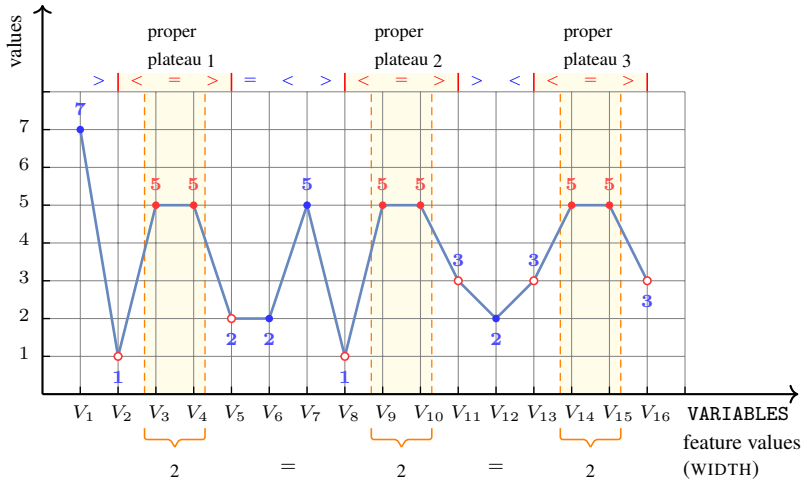


Figure 4.141: Illustrating the ALL_EQUAL_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton Figure 4.142 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_PROPER_PLATEAU.

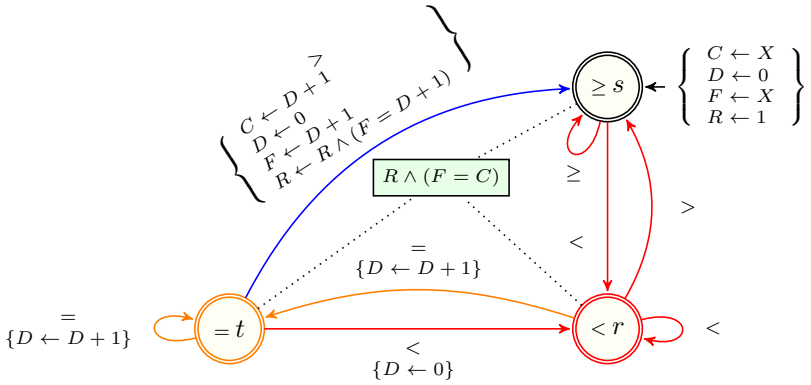
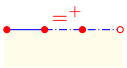


Figure 4.142: Automaton for the ALL_EQUAL_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_STEADY_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint ALL_EQUAL_WIDTH_STEADY_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 3, 5, 5, 5, 3, 1, 4, 4, 4, 6, 5, 3, 2, 2, 2, 5\rangle)$

Figure 4.143 provides an example where the ALL_EQUAL_WIDTH_STEADY_SEQUENCE $([3, 5, 5, 5, 3, 1, 4, 4, 4, 6, 5, 3, 2, 2, 2, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

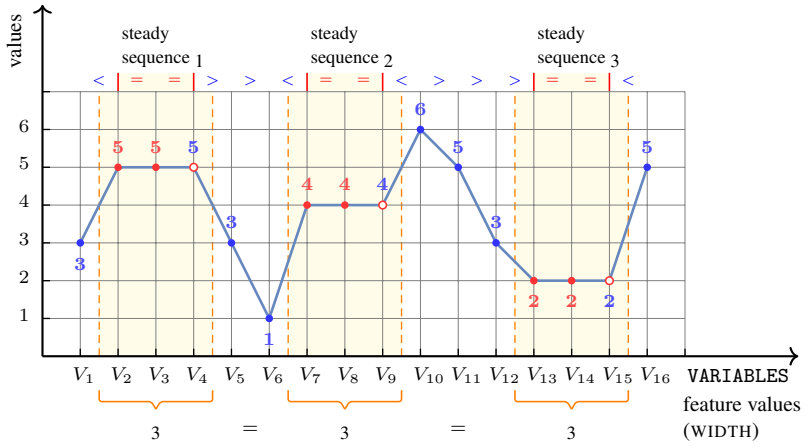


Figure 4.143: Illustrating the ALL_EQUAL_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.144 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_STEADY_SEQUENCE.

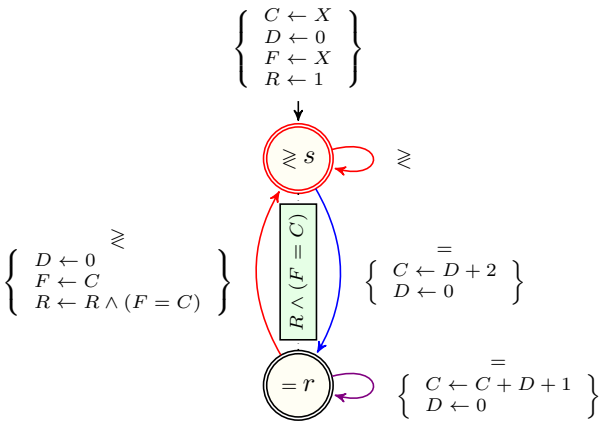


Figure 4.144: Automaton for the ALL_EQUAL_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 4, 3, 2, 2, 4, 4, 6, 6, 4, 1, 1, 2, 2, 4, 3, 1\rangle)$

Figure 4.145 provides an example where the ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE $([4, 3, 2, 2, 4, 4, 6, 6, 4, 1, 1, 2, 2, 4, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

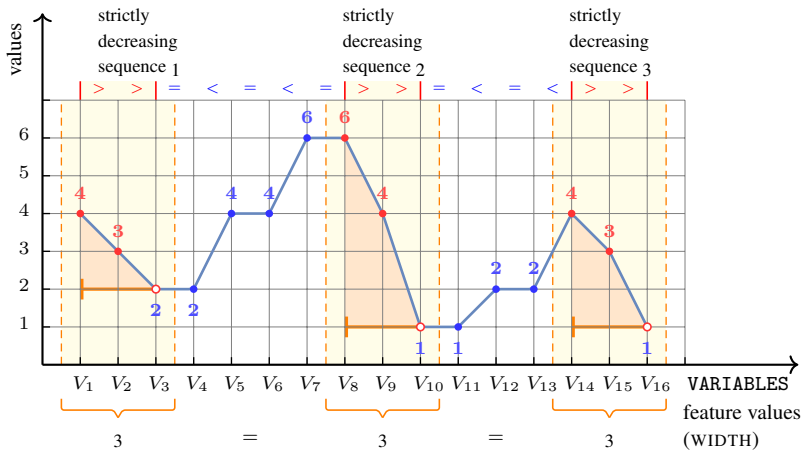


Figure 4.145: Illustrating the ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.146 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE.

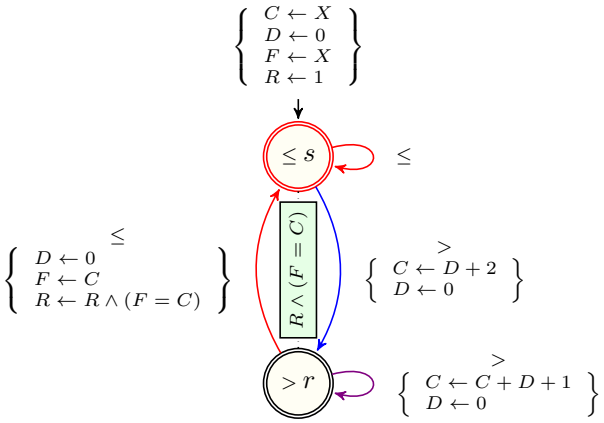


Figure 4.146: Automaton for the ALL_EQUAL_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 6, 3, 3, 4, 5, 6, 6, 3, 1, 1, 2, 3, 5, 5, 4, 3 \rangle)$

Figure 4.147 provides an example where the ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE $([6, 3, 3, 4, 5, 6, 6, 3, 1, 1, 2, 3, 5, 5, 4, 3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

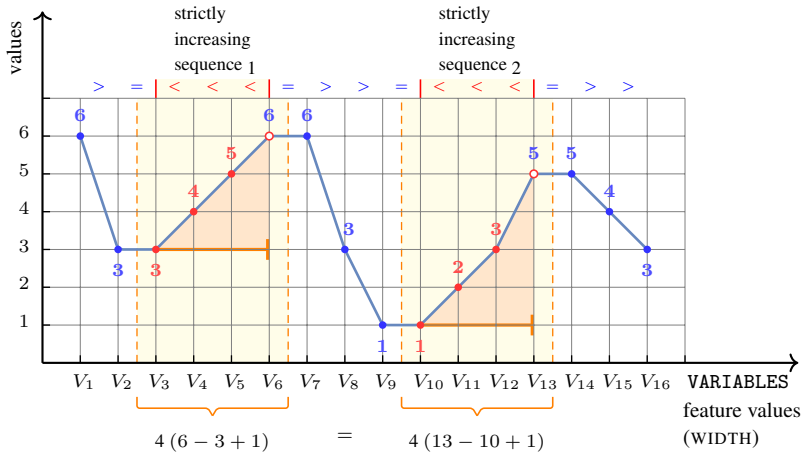


Figure 4.147: Illustrating the ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.148 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE.

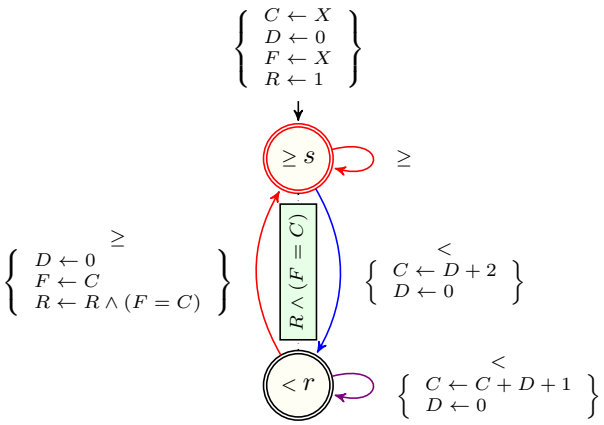


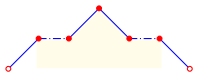
Figure 4.148: Automaton for the ALL_EQUAL_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_SUMMIT

► ▷ ◄ **DESCRIPTION** **AUTOMATON**

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



Origin Based on the SUMMIT pattern.

Constraint ALL_EQUAL_WIDTH_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2 \rangle)$

Figure 4.149 provides an example where the ALL_EQUAL_WIDTH_SUMMIT $([1, 3, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 4, 5, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

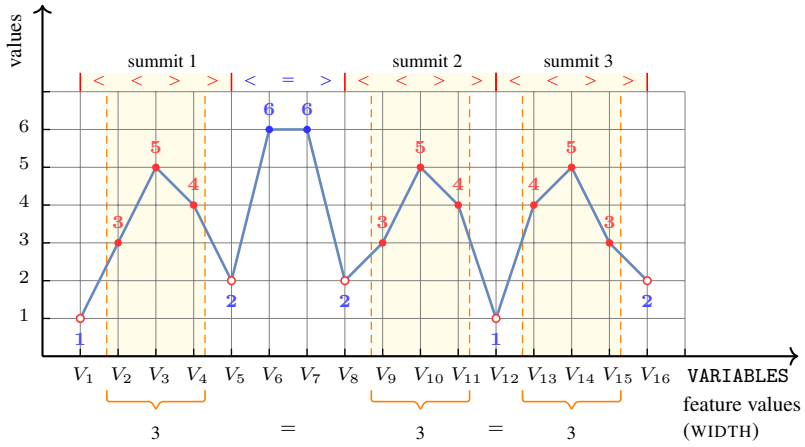


Figure 4.149: Illustrating the ALL_EQUAL_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figure 4.150 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_SUMMIT.

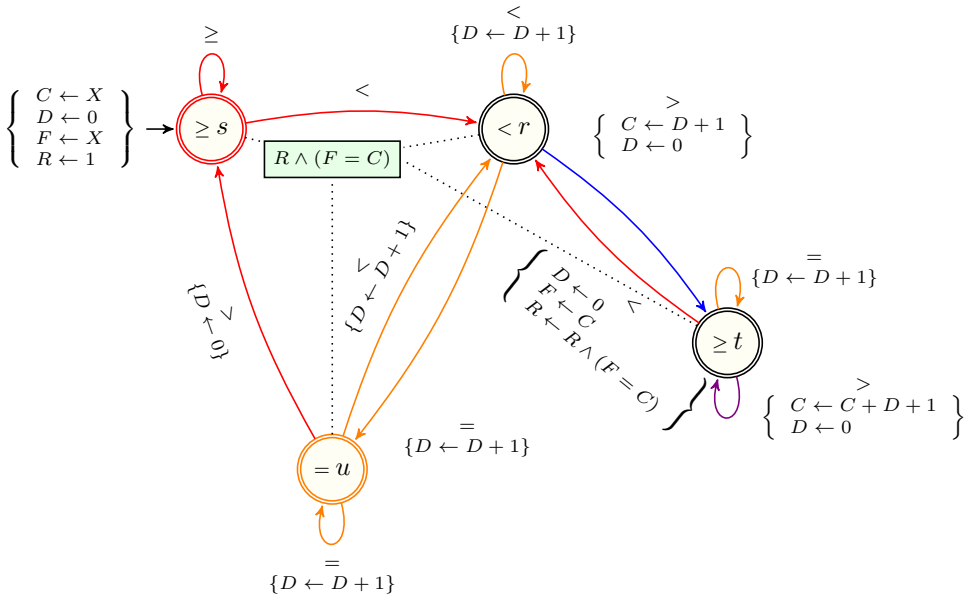
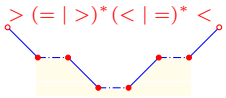


Figure 4.150: Automaton for the ALL_EQUAL_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_VALLEY

► ▷ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint ALL_EQUAL_WIDTH_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 6, 6, 1, 1, 7, 6, 4, 3, 1, 7, 6, 5, 2, 1, 7 \rangle)$

Figure 4.151 provides an example where the ALL_EQUAL_WIDTH_VALLEY $([7, 6, 6, 1, 1, 7, 6, 4, 3, 1, 7, 6, 5, 2, 1, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

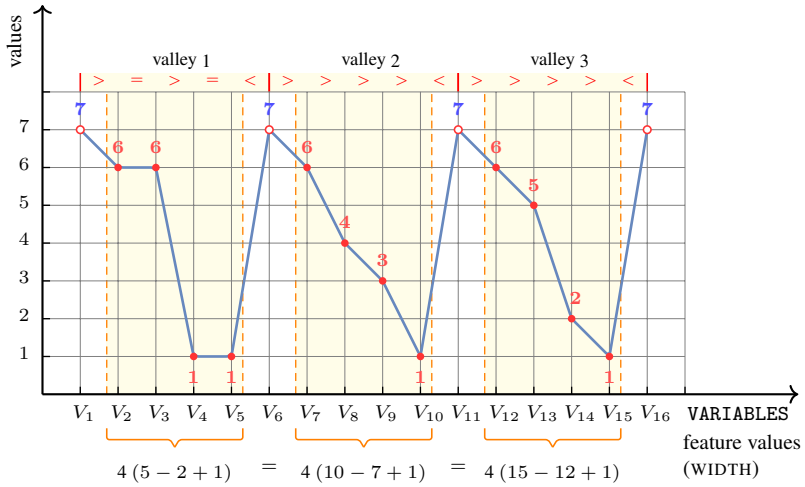


Figure 4.151: Illustrating the ALL_EQUAL_WIDTH_VALLEY constraint of the **Example** slot

Automaton Figure 4.152 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_VALLEY.

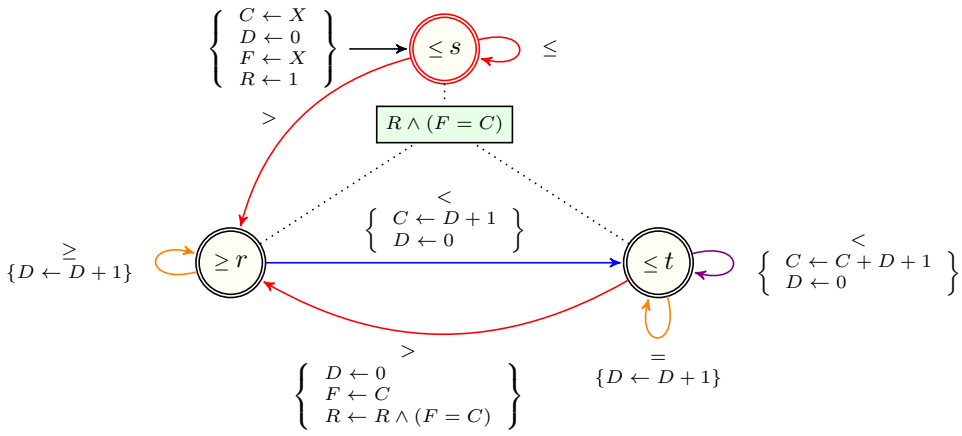
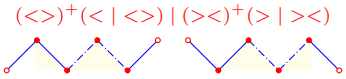


Figure 4.152: Automaton for the ALL_EQUAL_WIDTH_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

ALL_EQUAL_WIDTH_ZIGZAG

► ▷ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint ALL_EQUAL_WIDTH_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the width of all occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection are the same.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 3, 4, 1, 4, 3, 3, 4, 2, 6, 3, 2, 0, 3, 1, 2, 3\rangle)$

Figure 4.153 provides an example where the ALL_EQUAL_WIDTH_ZIGZAG $([3, 4, 1, 4, 3, 3, 4, 2, 6, 3, 2, 0, 3, 1, 2, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

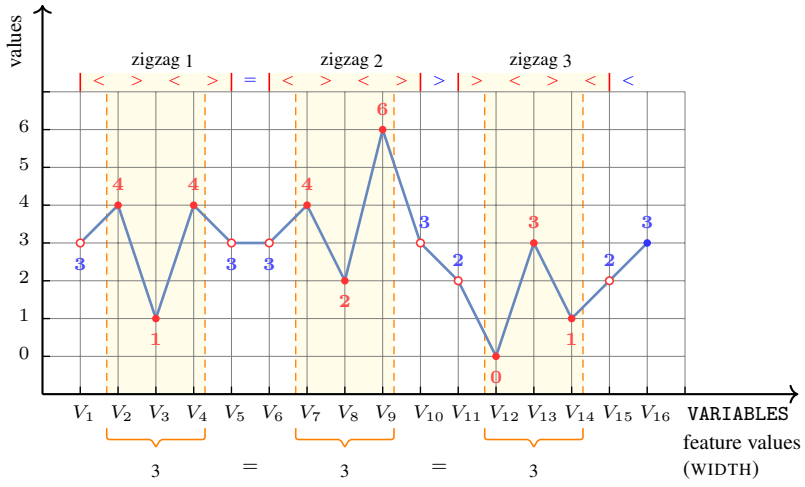


Figure 4.153: Illustrating the ALL_EQUAL_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.154 depicts the automaton associated with the constraint ALL_EQUAL_WIDTH_ZIGZAG.

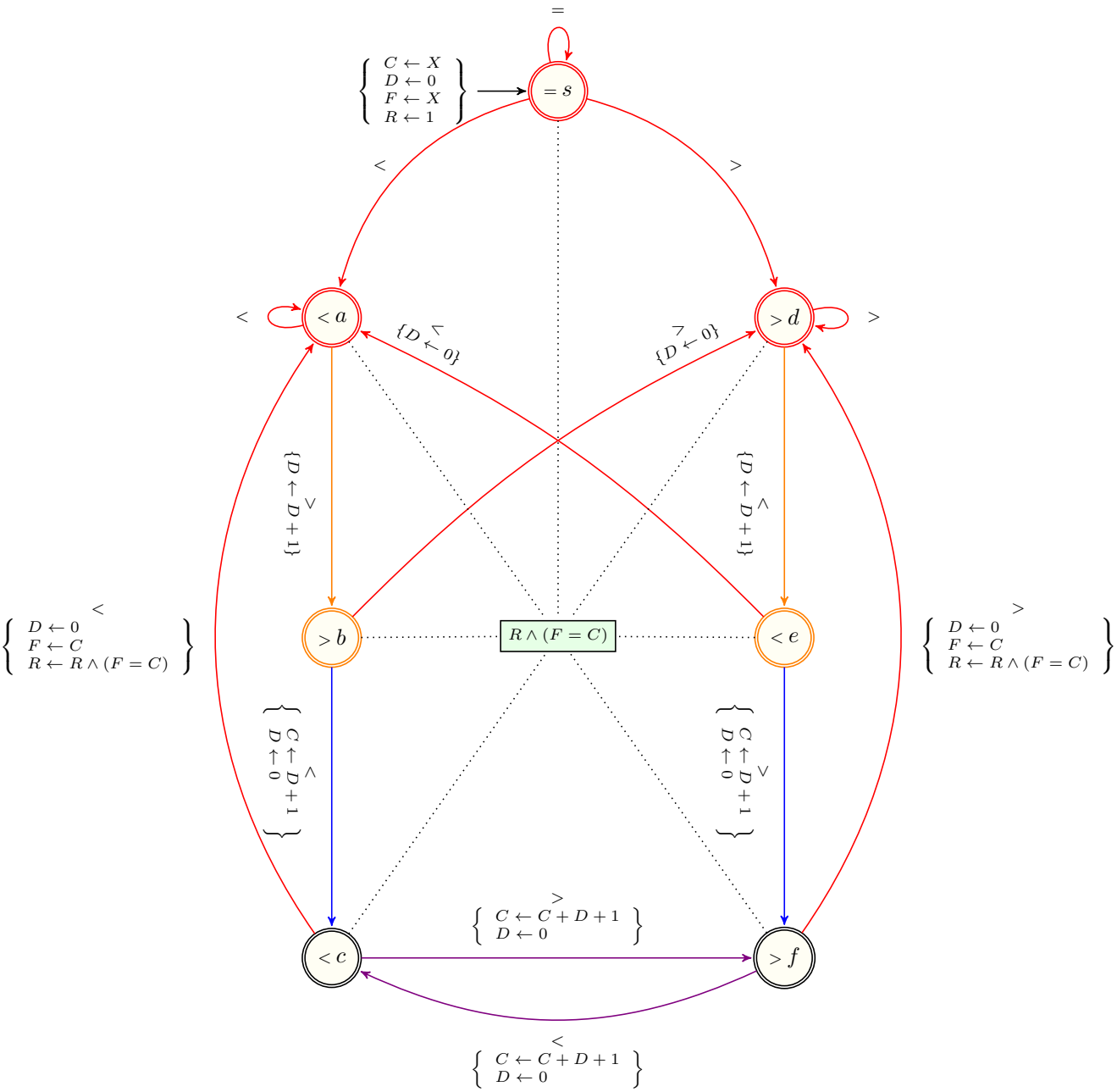
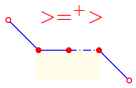


Figure 4.154: Automaton for the ALL_EQUAL_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint DECREASING_HEIGHT_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,6,6,6,4,3,3,2,2,4,4,6,3,3,1,1\rangle)$

Figure 4.155 provides an example where the DECREASING_HEIGHT_DECREASING_TERRACE $([7,6,6,6,4,3,3,2,2,4,4,6,3,3,1,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

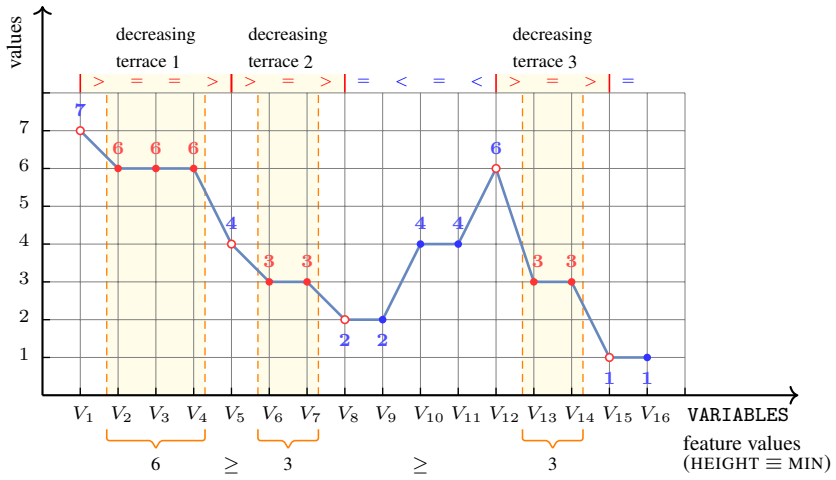


Figure 4.155: Illustrating the DECREASING_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.156 depicts the automaton associated with the constraint DECREASING_HEIGHT_DECREASING_TERRACE.

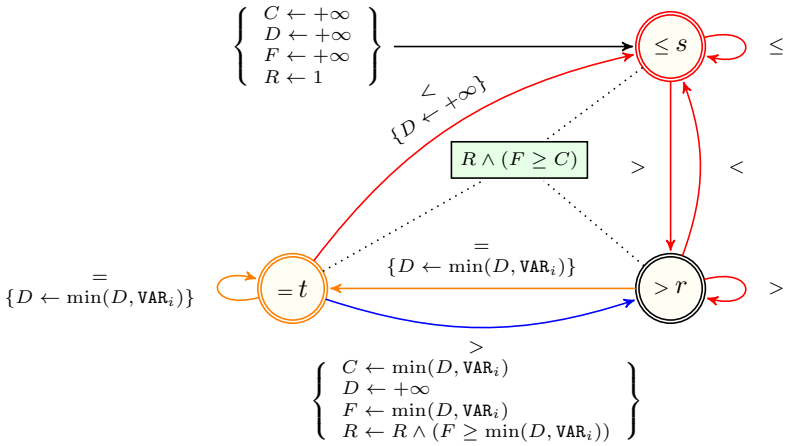
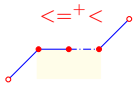


Figure 4.156: Automaton for the DECREASING_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INCREASING_TERRACE pattern.

Constraint DECREASING_HEIGHT_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 2,5,5,5,6,2,3,4,4,5,3,3,1,2,2,3\rangle)$

Figure 4.157 provides an example where the DECREASING_HEIGHT_INCREASING_TERRACE $([2,5,5,5,6,2,3,4,4,5,3,3,1,2,2,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

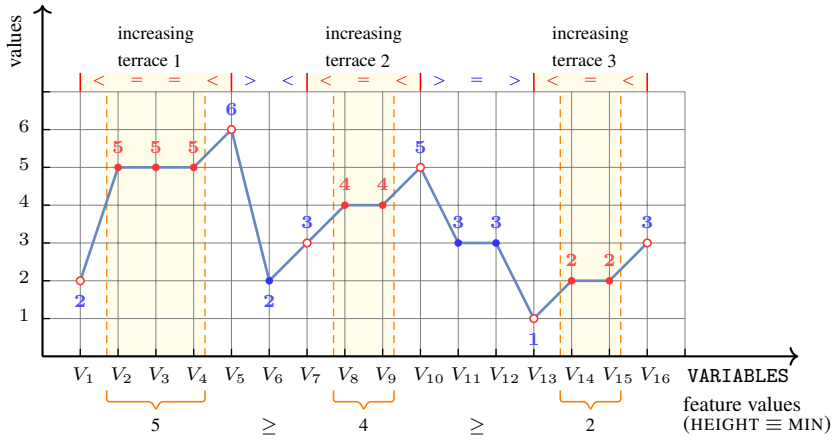


Figure 4.157: Illustrating the DECREASING_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.158 depicts the automaton associated with the constraint DECREASING_HEIGHT_INCREASING_TERRACE.

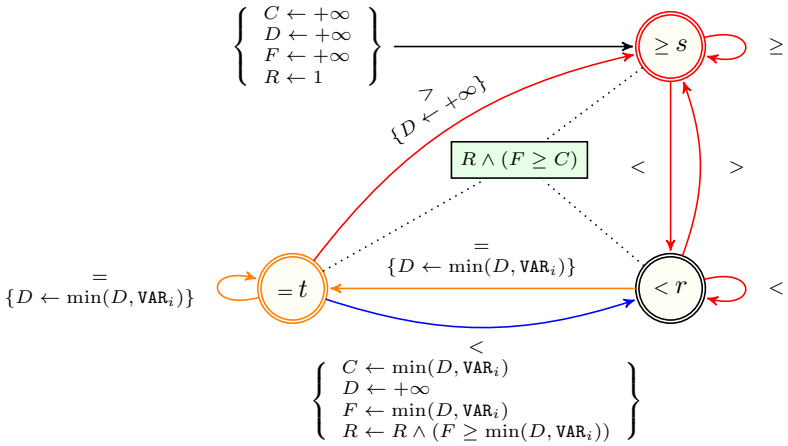


Figure 4.158: Automaton for the DECREASING_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint DECREASING_HEIGHT_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3\rangle)$

Figure 4.159 provides an example where the DECREASING_HEIGHT_PLAIN $([2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

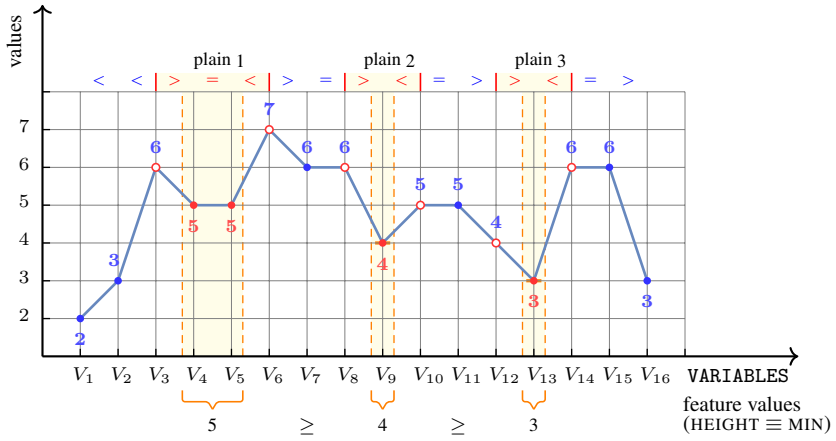


Figure 4.159: Illustrating the DECREASING_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figure 4.160 depicts the automaton associated with the constraint DECREASING_HEIGHT_PLAIN.

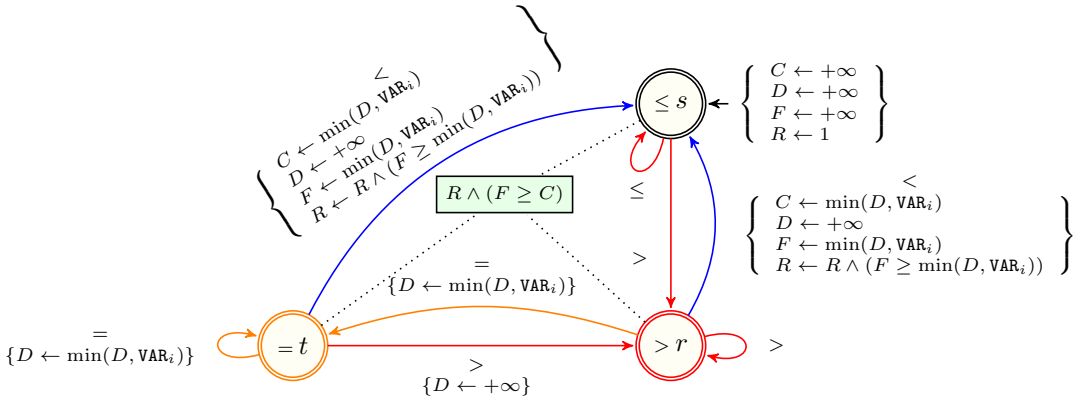


Figure 4.160: Automaton for the DECREASING_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint DECREASING_HEIGHT_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 5,2,2,5,5,4,3,3,4,2,2,1,3,2,5,7\rangle)$

Figure 4.161 provides an example where the DECREASING_HEIGHT_PLATEAU $([5,2,2,5,5,4,3,3,4,2,2,1,3,2,5,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

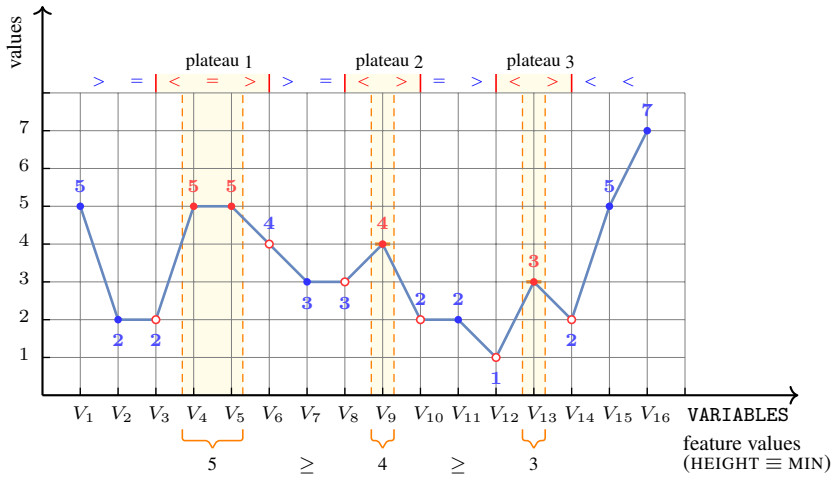


Figure 4.161: Illustrating the DECREASING_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figure 4.162 depicts the automaton associated with the constraint DECREASING_HEIGHT_PLATEAU.

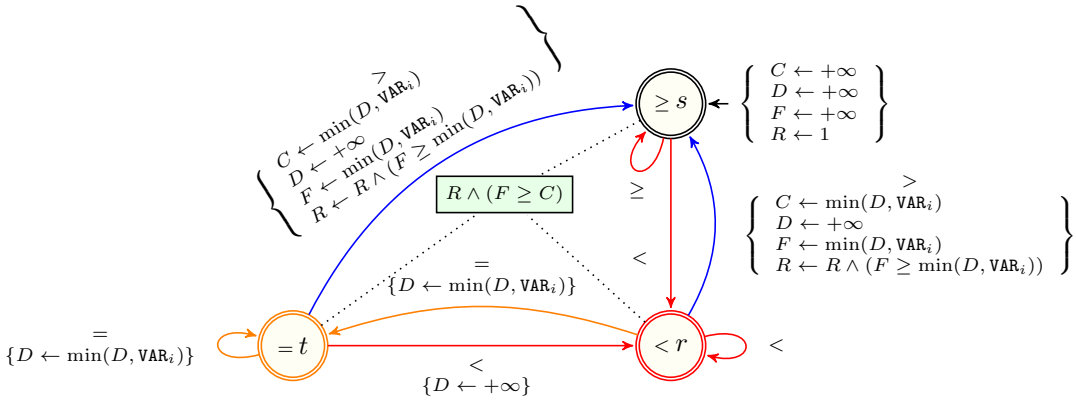
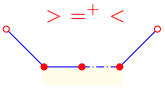


Figure 4.162: Automaton for the DECREASING_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint DECREASING_HEIGHT_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 2,7,5,5,5,6,3,7,4,4,5,6,5,3,3,5\rangle)$

Figure 4.163 provides an example where the DECREASING_HEIGHT_PROPER_PLAIN $([2,7,5,5,5,6,3,7,4,4,5,6,5,3,3,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

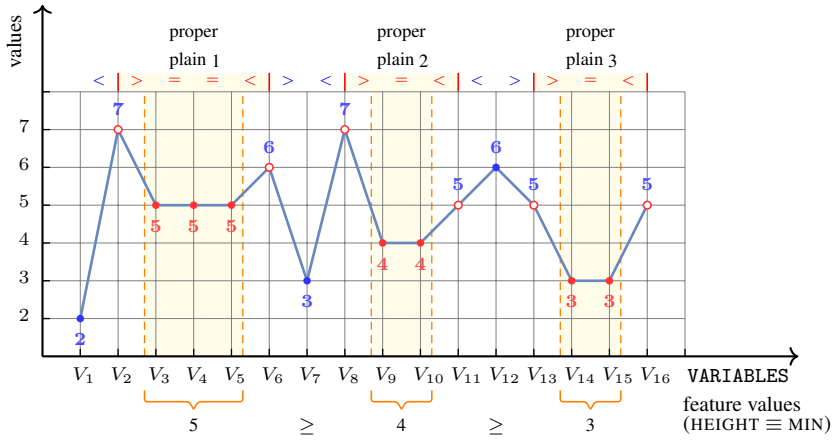


Figure 4.163: Illustrating the DECREASING_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.164 depicts the automaton associated with the constraint DECREASING_HEIGHT_PROPER_PLAIN.

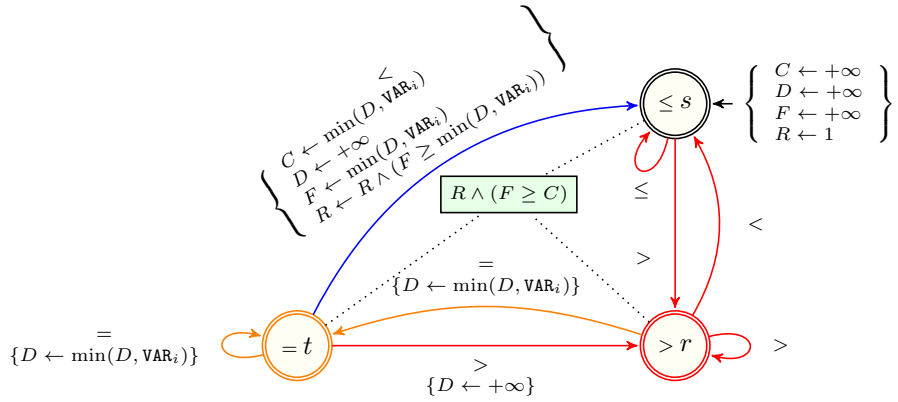
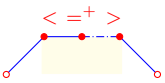


Figure 4.164: Automaton for the DECREASING_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint DECREASING_HEIGHT_PROPER_PLATEAU(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 3,5,5,5,3,2,3,4,4,1,5,2,3,3,1,7\rangle)$

Figure 4.165 provides an example where the DECREASING_HEIGHT_PROPER_PLATEAU $([3,5,5,5,3,2,3,4,4,1,5,2,3,3,1,7])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

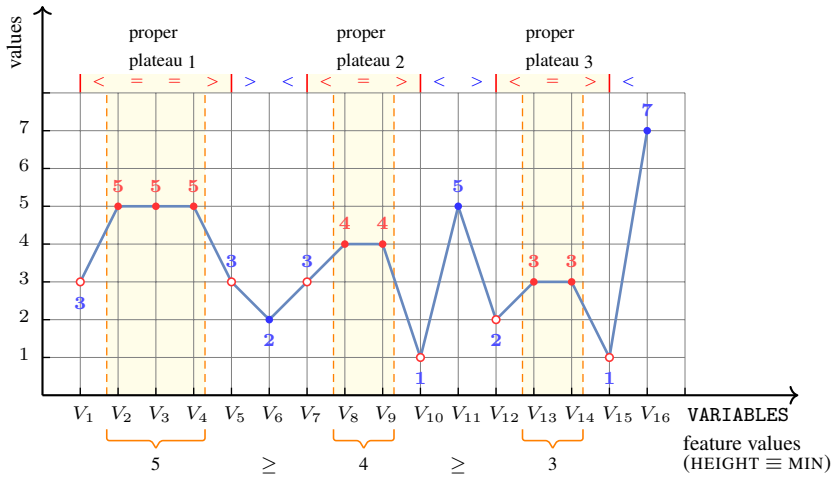


Figure 4.165: Illustrating the DECREASING_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.166 depicts the automaton associated with the constraint DECREASING_HEIGHT_PROPER_PLATEAU.

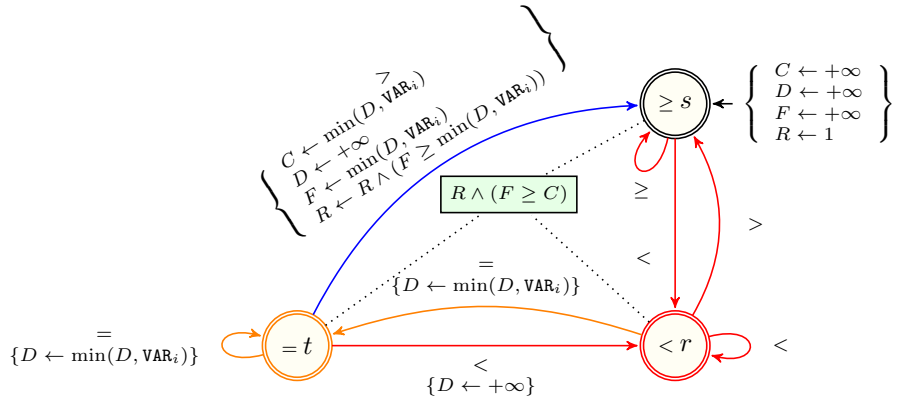


Figure 4.166: Automaton for the DECREASING_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint DECREASING_HEIGHT_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 3, 6, 6, 6, 2, 7, 5, 5, 5, 6, 5, 5, 5, 3, 3, 7\rangle)$

Figure 4.167 provides an example where the DECREASING_HEIGHT_STEADY $([3, 6, 6, 6, 2, 7, 5, 5, 5, 6, 5, 5, 5, 3, 3, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

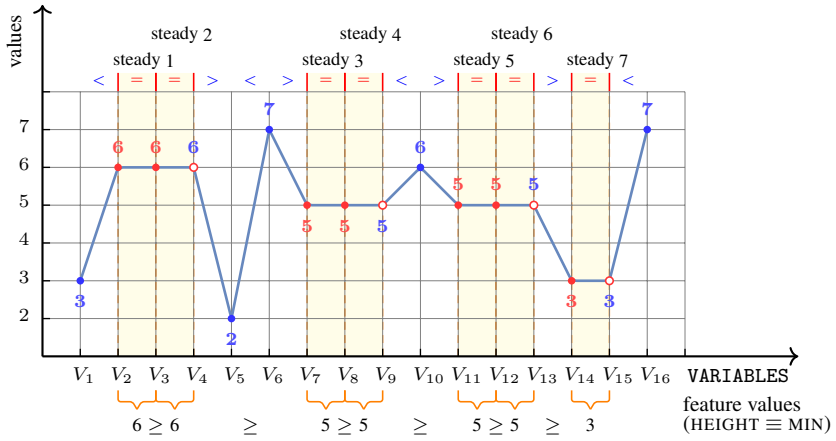


Figure 4.167: Illustrating the DECREASING_HEIGHT_STEADY constraint of the **Example** slot

Automaton

Figure 4.168 depicts the automaton associated with the constraint DECREASING_HEIGHT_STEADY.

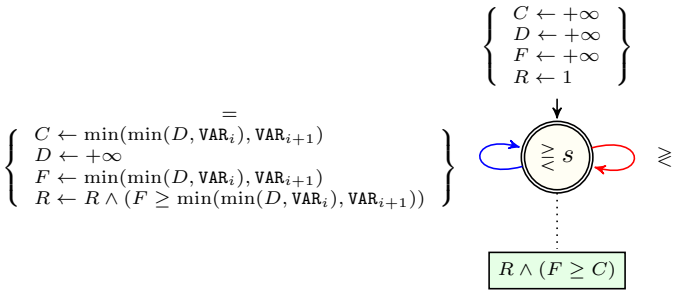
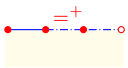


Figure 4.168: Automaton for the DECREASING_HEIGHT_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

DECREASING_HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the STEADY_SEQUENCE pattern.

Constraint DECREASING_HEIGHT_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,3,2,5,5,5,3,2,4,4,6,5,3,2,2,6\rangle)$

Figure 4.169 provides an example where the DECREASING_HEIGHT_STEADY_SEQUENCE $([4,3,2,5,5,5,3,2,4,4,6,5,3,2,2,6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

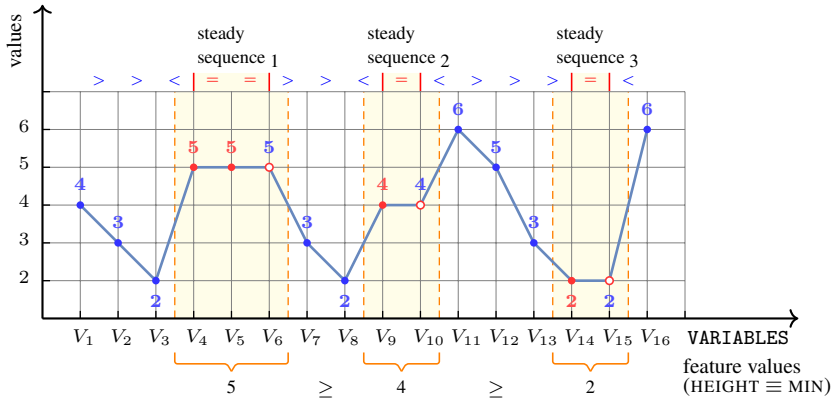


Figure 4.169: Illustrating the DECREASING_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.170 depicts the automaton associated with the constraint DECREASING_HEIGHT_STEADY_SEQUENCE.

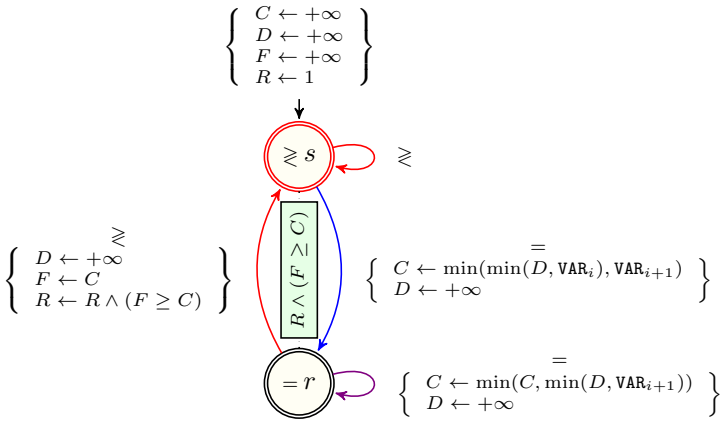
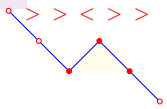


Figure 4.170: Automaton for the DECREASING_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+2$ to index j.

Example $(\langle 7,6,5,6,5,4,1,2,7,5,4,2,5,4,3,3\rangle)$

Figure 4.171 provides an example where the DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE $([7,6,5,6,5,4,1,2,7,5,4,2,5,4,3,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

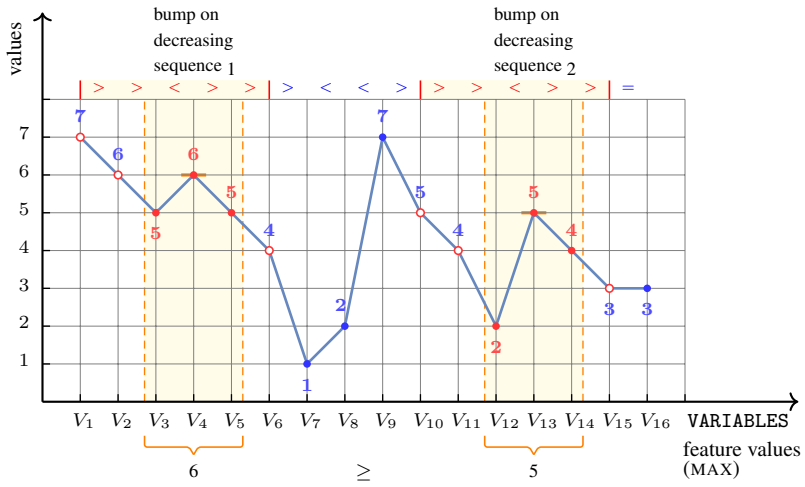


Figure 4.171: Illustrating the DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.172 depicts the automaton associated with the constraint DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE.

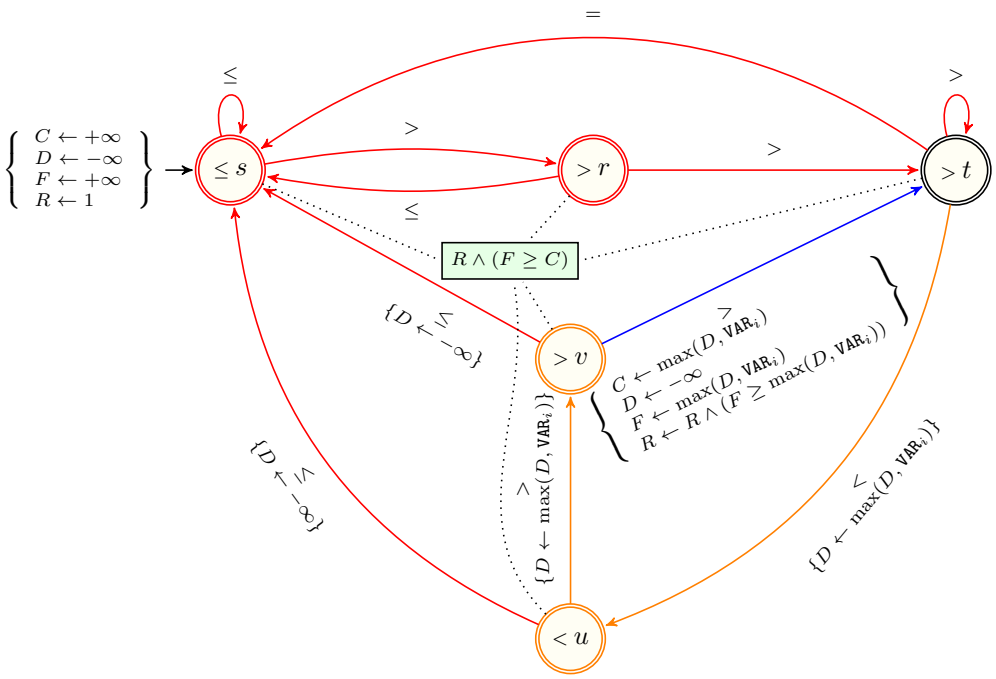


Figure 4.172: Automaton for the DECREASING_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint DECREASING_MAX_DECREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 1,1,3,4,4,5,7,7,5,4,4,2,2,2,5,5\rangle)$

Figure 4.173 provides an example where the DECREASING_MAX_DECREASING $([1,1,3,4,4,5,7,7,5,4,4,2,2,2,5,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

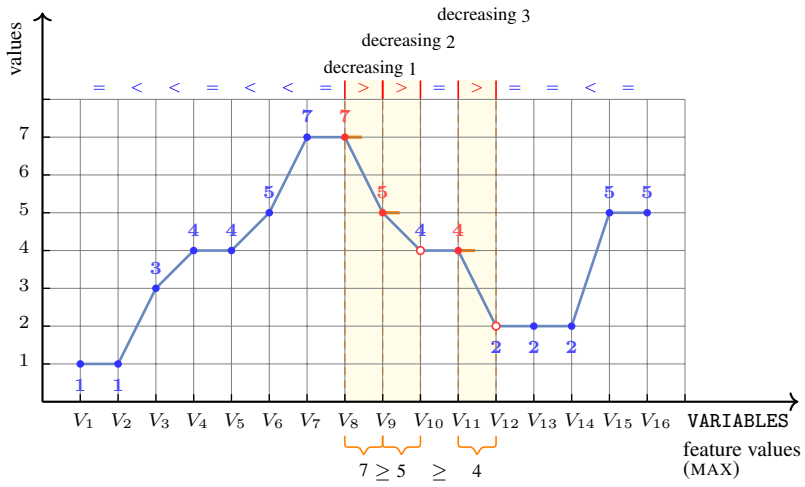


Figure 4.173: Illustrating the DECREASING_MAX_DECREASING constraint of the **Example** slot

Automaton

Figure 4.174 depicts the automaton associated with the constraint DECREASING_MAX_DECREASING.

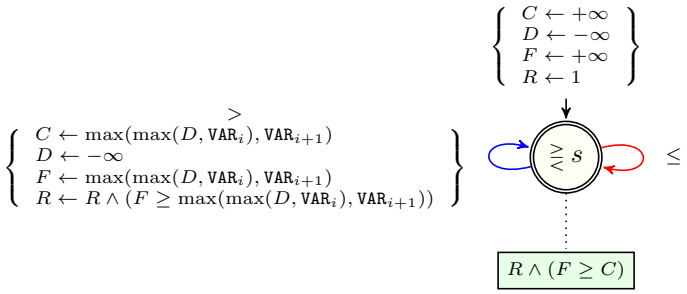
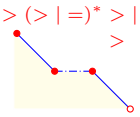


Figure 4.174: Automaton for the DECREASING_MAX_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 4, 7, 4, 2, 3, 4, 4, 6, 5, 5, 4, 3, 2, 2, 3, 2 \rangle)$

Figure 4.175 provides an example where the DECREASING_MAX_DECREASING_SEQUENCE $([4, 7, 4, 2, 3, 4, 4, 6, 5, 5, 4, 3, 2, 2, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

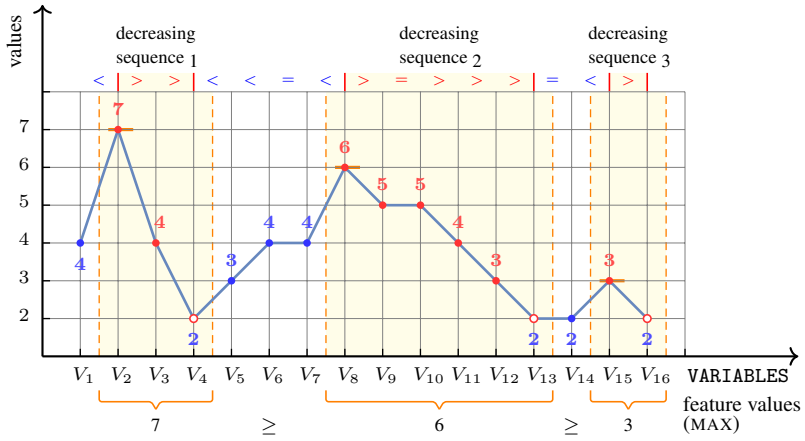


Figure 4.175: Illustrating the DECREASING_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.176 depicts the automaton associated with the constraint DECREASING_MAX_DECREASING_SEQUENCE.

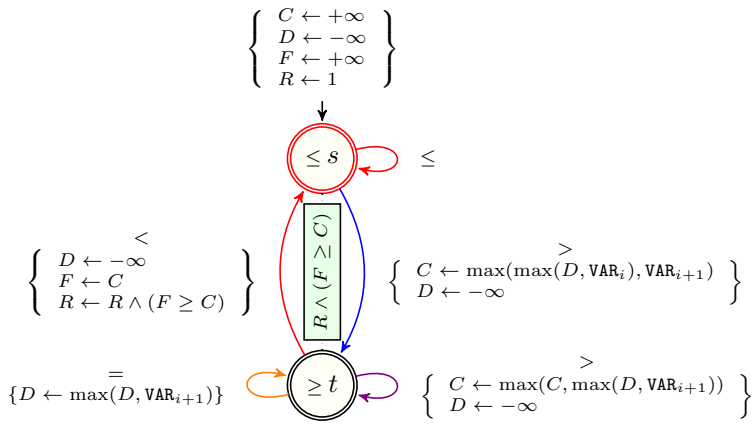
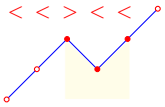


Figure 4.176: Automaton for the DECREASING_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+2$ to index j.

Example $(\langle 4,5,6,0,2,4,4,1,2,3,2,3,4,6,5,1\rangle)$

Figure 4.177 provides an example where the DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE $([4,5,6,0,2,4,4,1,2,3,2,3,4,6,5,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

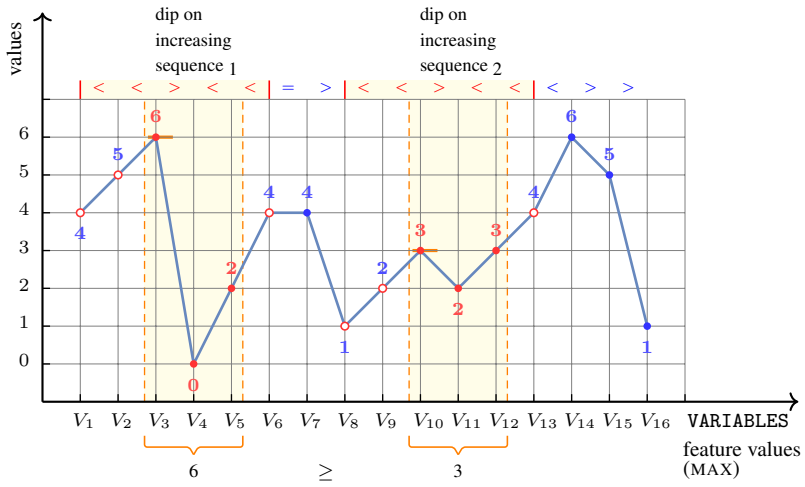


Figure 4.177: Illustrating the DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.178 depicts the automaton associated with the constraint DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE.

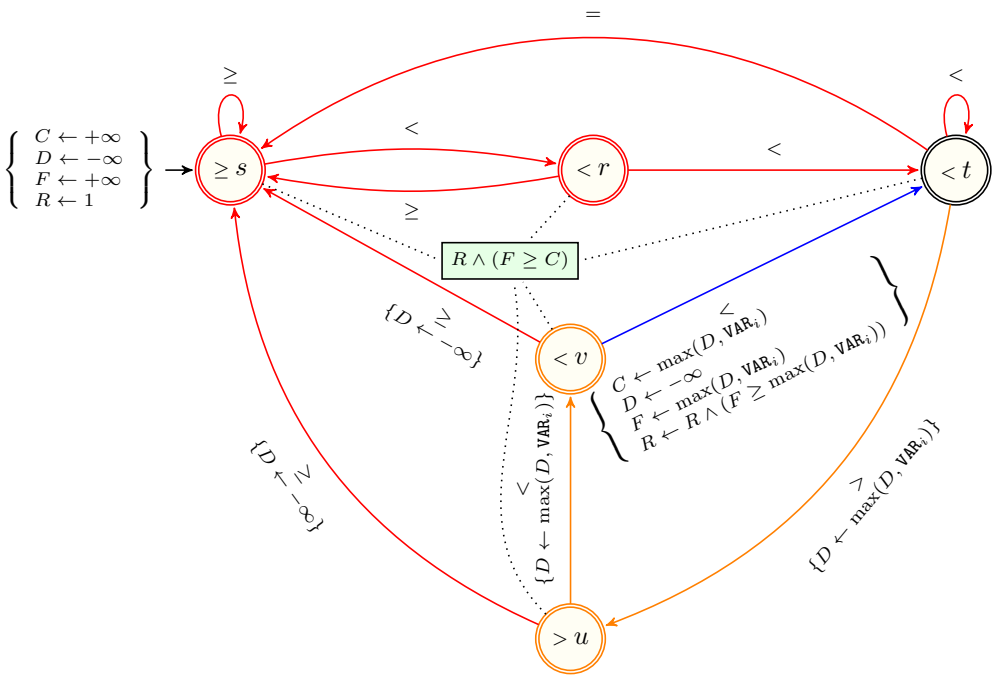


Figure 4.178: Automaton for the DECREASING_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint DECREASING_MAX_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 4,4,5,5,4,3,3,1,4,2,4,3,3,1,2,1\rangle)$

Figure 4.179 provides an example where the DECREASING_MAX_INCREASING $([4,4,5,5,4,3,3,1,4,2,4,3,3,1,2,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

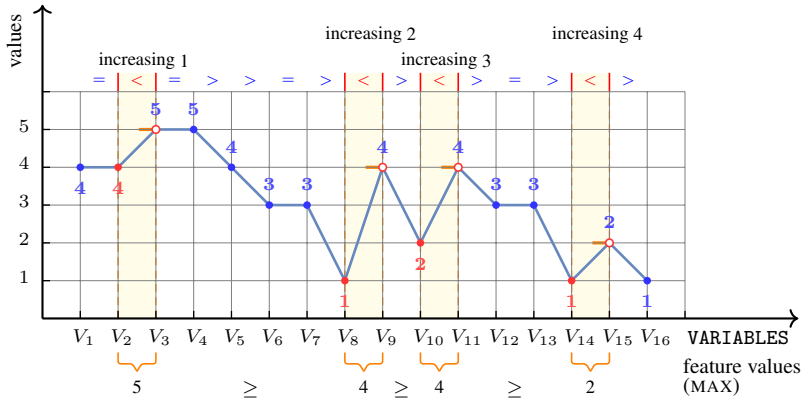


Figure 4.179: Illustrating the DECREASING_MAX_INCREASING constraint of the **Example** slot

Automaton

Figure 4.180 depicts the automaton associated with the constraint DECREASING_MAX_INCREASING.

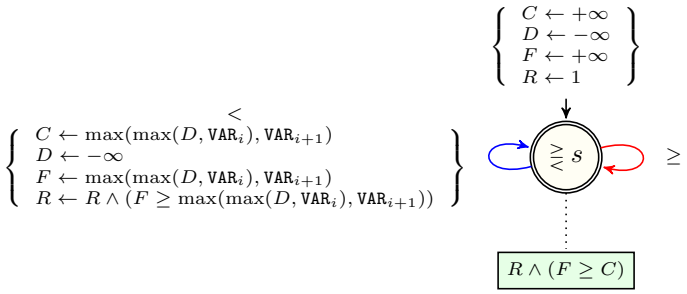
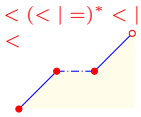


Figure 4.180: Automaton for the DECREASING_MAX_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 3, 4, 5, 5, 4, 3, 3, 1, 2, 2, 4, 3, 3, 1, 3, 1 \rangle)$

Figure 4.181 provides an example where the DECREASING_MAX_INCREASING_SEQUENCE $([3, 4, 5, 5, 4, 3, 3, 1, 2, 2, 4, 3, 3, 1, 3, 1])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

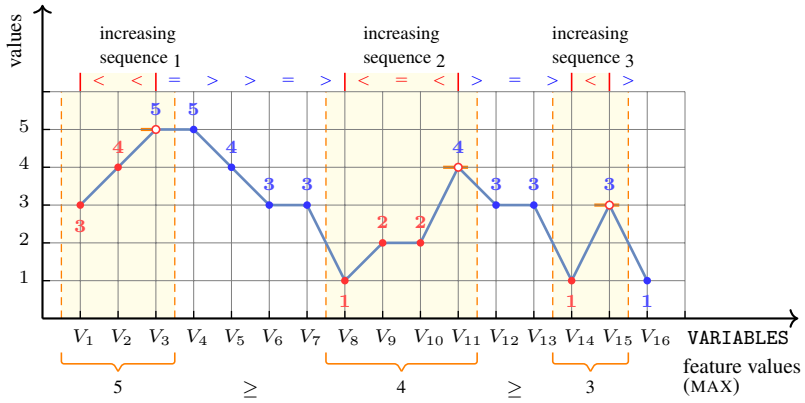


Figure 4.181: Illustrating the DECREASING_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.182 depicts the automaton associated with the constraint DECREASING_MAX_INCREASING_SEQUENCE.

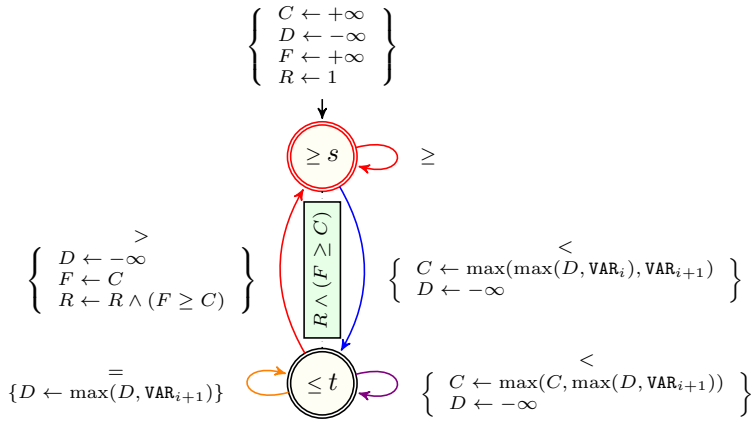
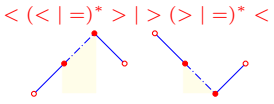


Figure 4.182: Automaton for the DECREASING_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INFLEXION pattern.

Constraint DECREASING_MAX_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 3, 4, 3, 2, 2, 1, 1, 2, 2, 5, 5 \rangle)$

Figure 4.183 provides an example where the DECREASING_MAX_INFLEXION $([3, 4, 3, 2, 2, 1, 1, 2, 2, 5, 5])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

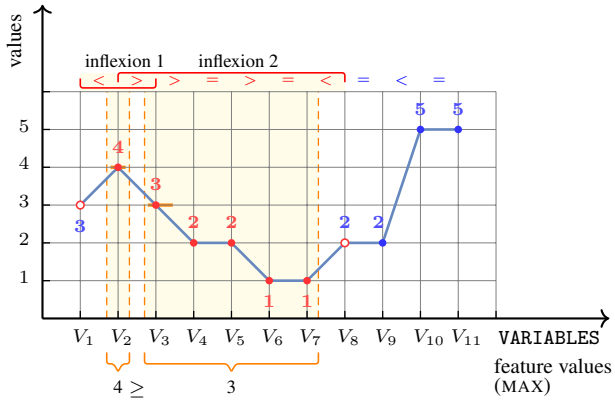


Figure 4.183: Illustrating the DECREASING_MAX_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.184 depicts the automaton associated with the constraint DECREASING_MAX_INFLEXION.

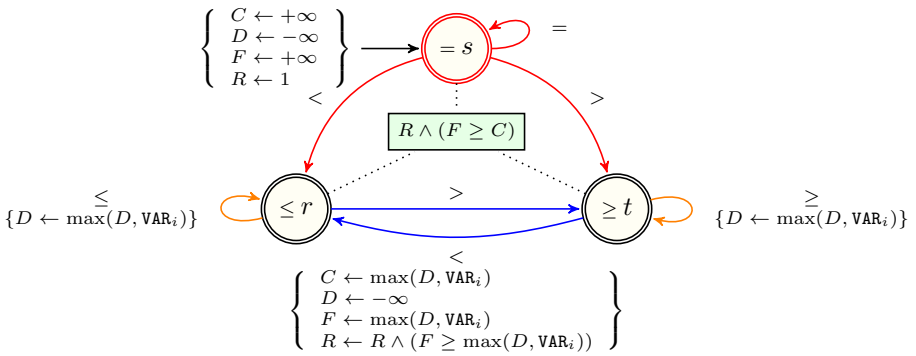
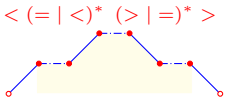


Figure 4.184: Automaton for the DECREASING_MAX_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

DECREASING_MAX_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint DECREASING_MAX_PEAK(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7\rangle)$

Figure 4.185 provides an example where the DECREASING_MAX_PEAK $([1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

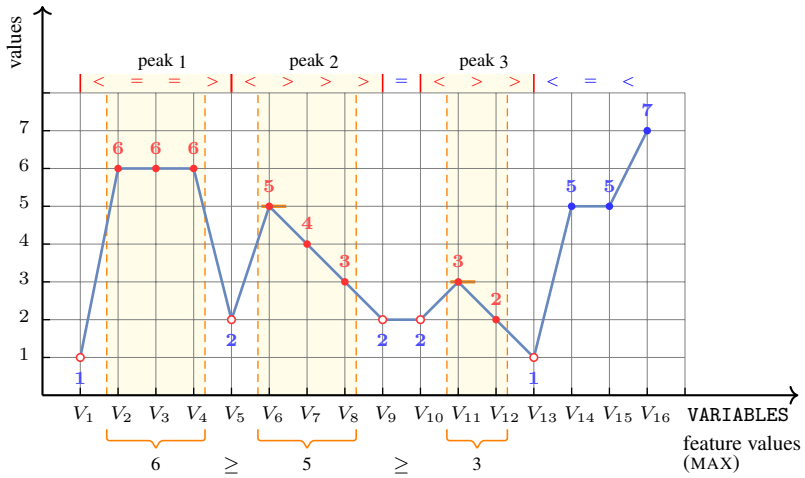


Figure 4.185: Illustrating the DECREASING_MAX_PEAK constraint of the **Example** slot

Automaton Figure 4.186 depicts the automaton associated with the constraint DECREASING_MAX_PEAK.

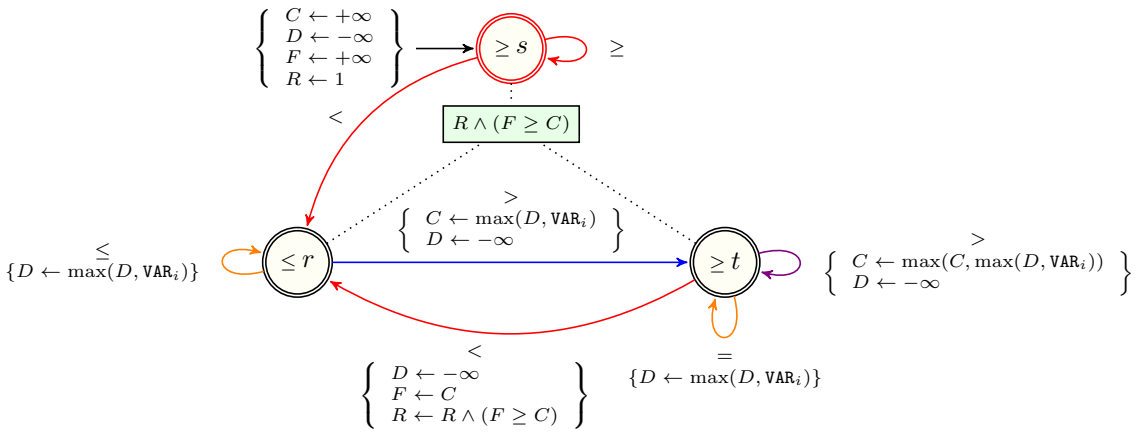


Figure 4.186: Automaton for the DECREASING_MAX_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 6,4,6,5,5,3,5,5,4,4,4,3,2,2,3,4\rangle)$

Figure 4.187 provides an example where the DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE $([6,4,6,5,5,3,5,5,4,4,4,3,2,2,3,4])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

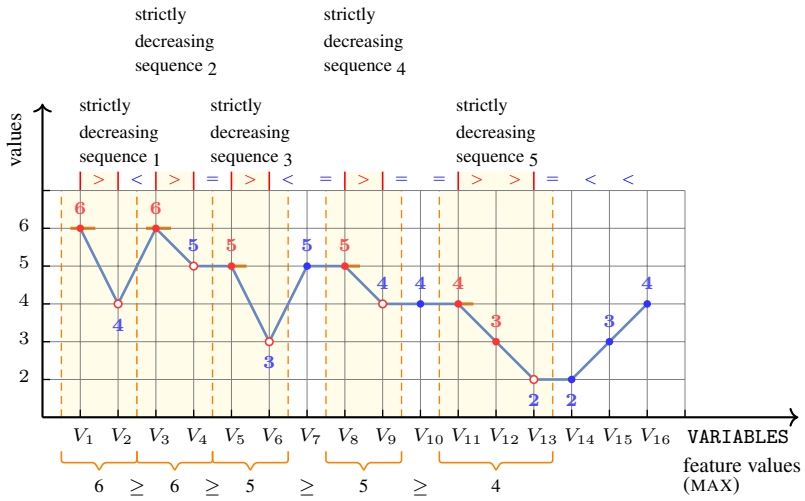


Figure 4.187: Illustrating the DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.188 depicts the automaton associated with the constraint DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE.

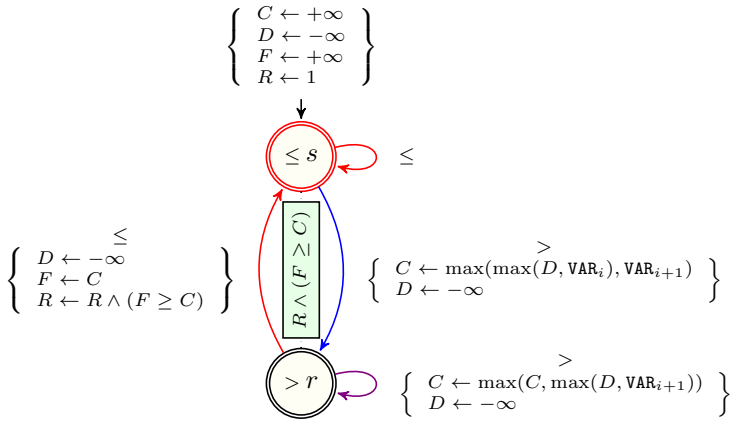


Figure 4.188: Automaton for the DECREASING_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 3,4,5,5,4,3,3,1,2,3,4,3,3,1,3,1\rangle)$

Figure 4.189 provides an example where the DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE $([3,4,5,5,4,3,3,1,2,3,4,3,3,1,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

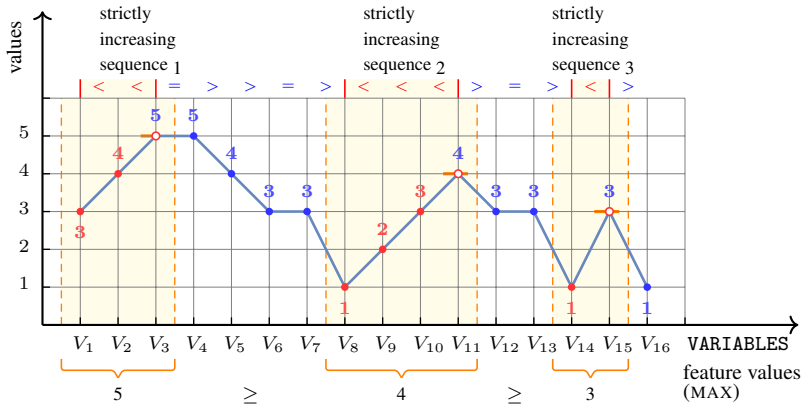


Figure 4.189: Illustrating the DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.190 depicts the automaton associated with the constraint DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE.

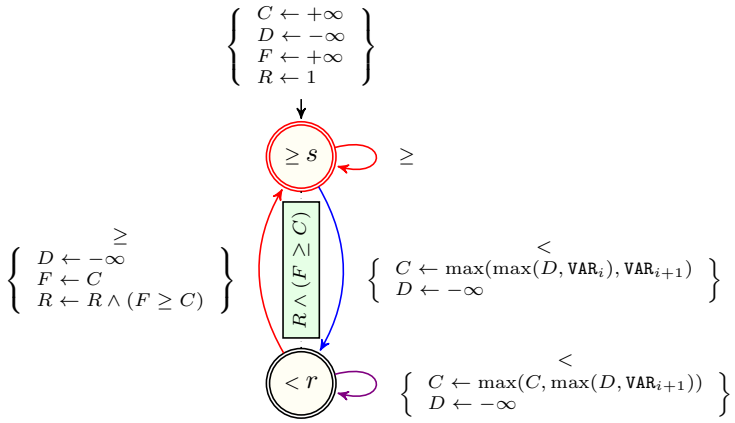


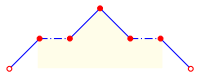
Figure 4.190: Automaton for the DECREASING_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MAX_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the SUMMIT pattern.

Constraint DECREASING_MAX_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2\rangle)$

Figure 4.191 provides an example where the DECREASING_MAX_SUMMIT $([1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

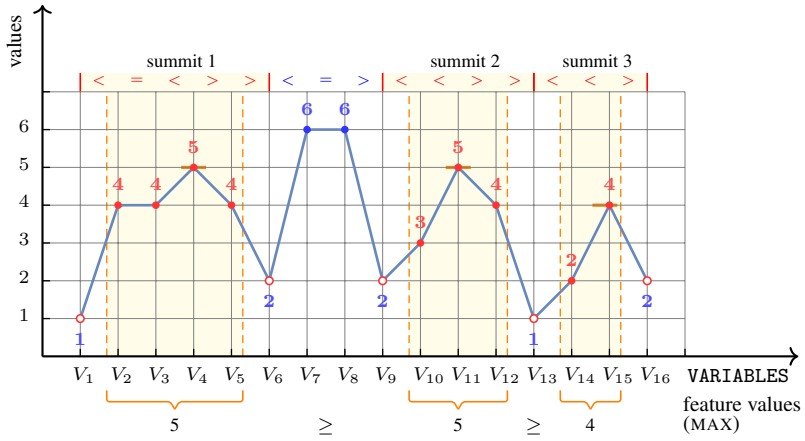


Figure 4.191: Illustrating the DECREASING_MAX_SUMMIT constraint of the **Example** slot

Automaton Figure 4.192 depicts the automaton associated with the constraint DECREASING_MAX_SUMMIT.

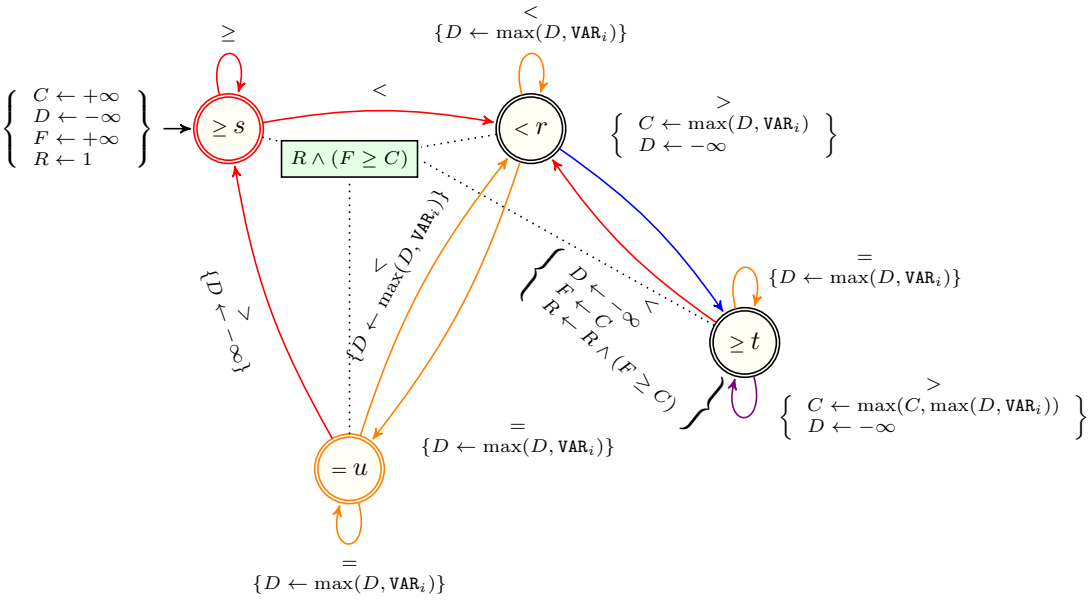
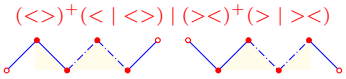


Figure 4.192: Automaton for the DECREASING_MAX_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_MAX_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint DECREASING_MAX_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 6, 5, 7, 4, 5, 1, 1, 6, 3, 6, 4, 3, 5, 2, 7, 7\rangle)$

Figure 4.193 provides an example where the DECREASING_MAX_ZIGZAG $([6, 5, 7, 4, 5, 1, 1, 6, 3, 6, 4, 3, 5, 2, 7, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

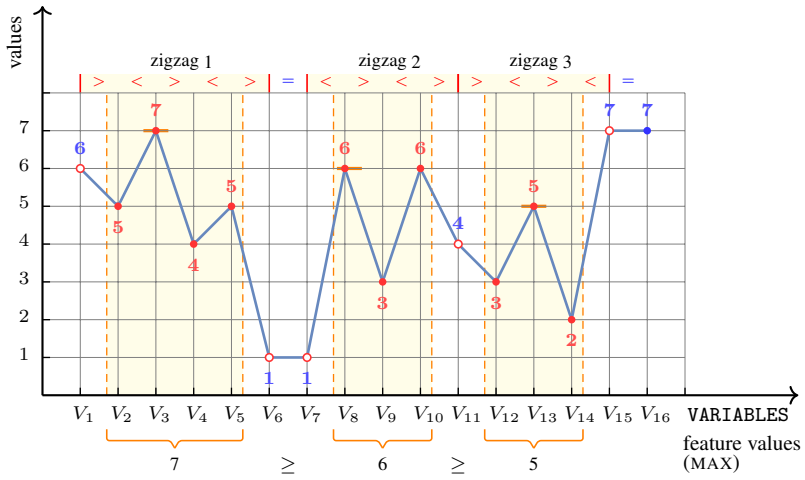


Figure 4.193: Illustrating the DECREASING_MAX_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.194 depicts the automaton associated with the constraint DECREASING_MAX_ZIGZAG.

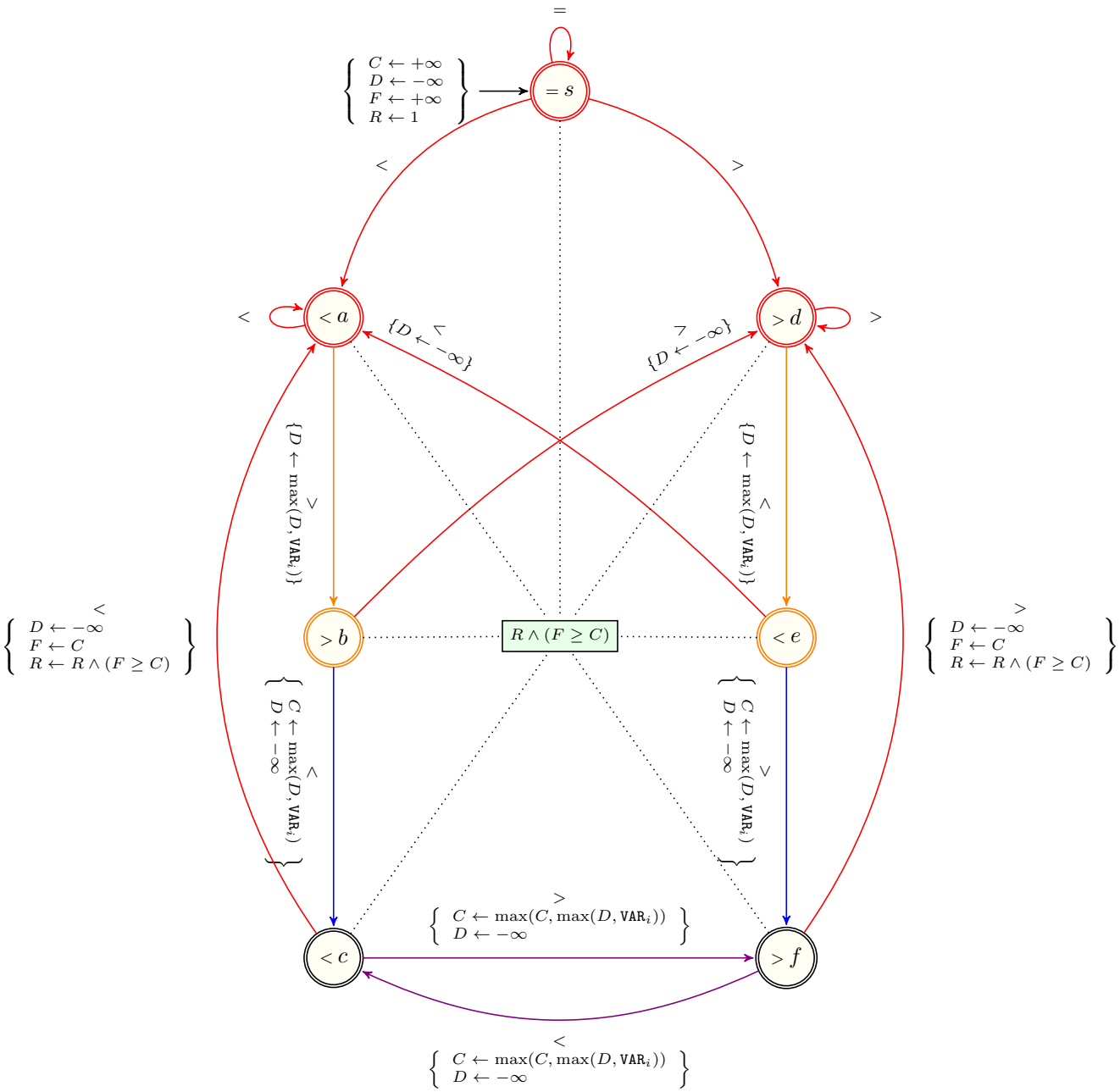
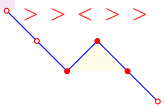


Figure 4.194: Automaton for the DECREASING_MAX_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 7,6,5,6,4,1,1,5,5,4,2,6,5,4,4,5\rangle)$

Figure 4.195 provides an example where the DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE $([7,6,5,6,4,1,1,5,5,4,2,6,5,4,4,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

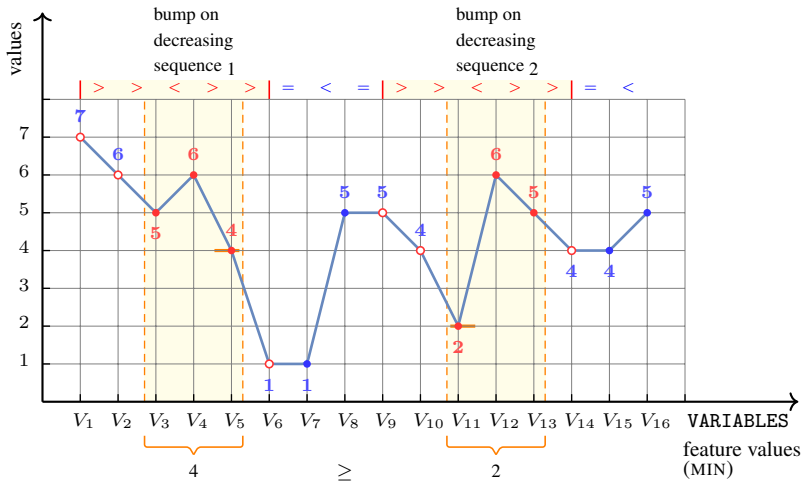


Figure 4.195: Illustrating the DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.196 depicts the automaton associated with the constraint DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE.

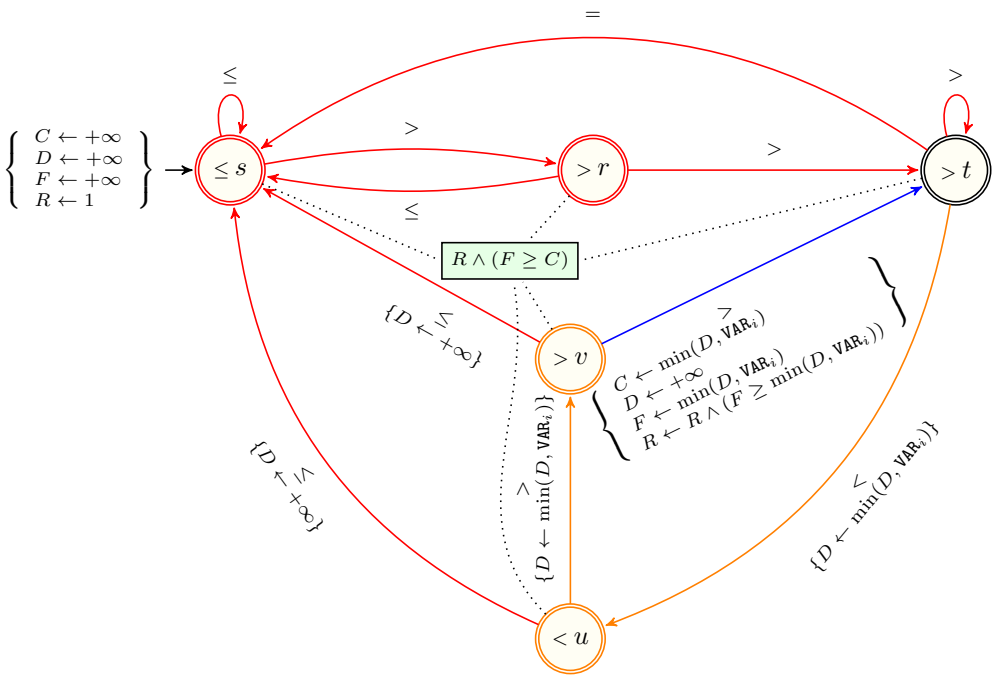


Figure 4.196: Automaton for the DECREASING_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint DECREASING_MIN_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,7,6,4,5,5,5,6,4,4,3,3,2,2,3,1\rangle)$

Figure 4.197 provides an example where the DECREASING_MIN_DECREASING $([4,7,6,4,5,5,5,6,4,4,3,3,2,2,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

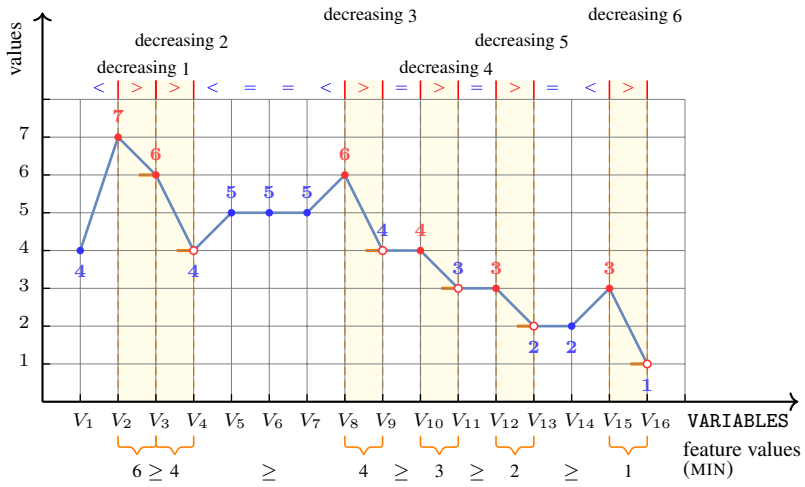


Figure 4.197: Illustrating the DECREASING_MIN_DECREASING constraint of the **Example** slot

Automaton

Figure 4.198 depicts the automaton associated with the constraint DECREASING_MIN_DECREASING.

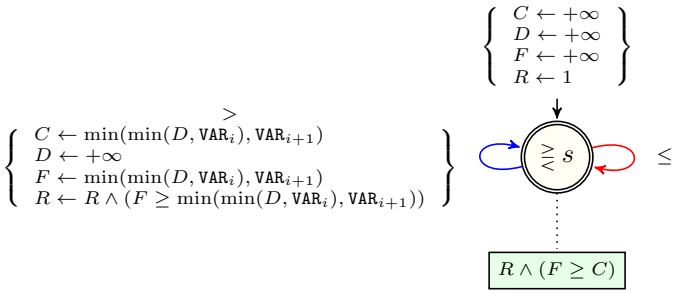
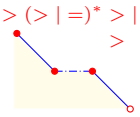


Figure 4.198: Automaton for the DECREASING_MIN_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 4, 7, 6, 4, 5, 5, 5, 6, 5, 5, 4, 4, 2, 2, 3, 1\rangle)$

Figure 4.199 provides an example where the DECREASING_MIN_DECREASING_SEQUENCE $([4, 7, 6, 4, 5, 5, 5, 6, 5, 5, 4, 4, 2, 2, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

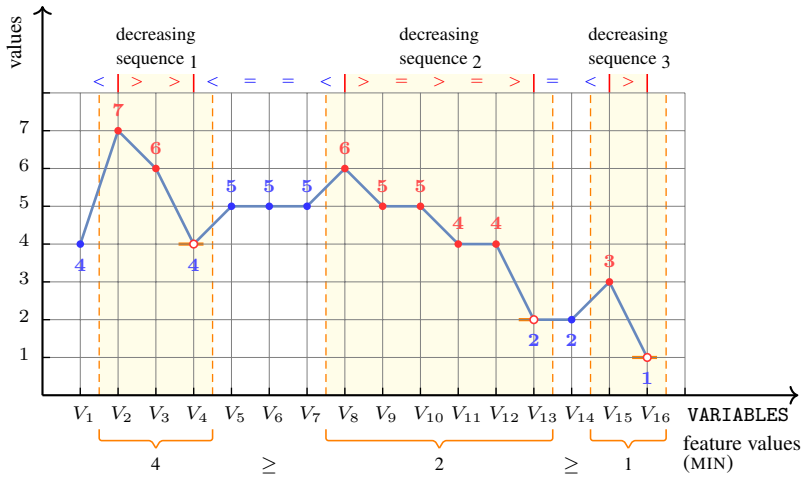


Figure 4.199: Illustrating the DECREASING_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.200 depicts the automaton associated with the constraint DECREASING_MIN_DECREASING_SEQUENCE.

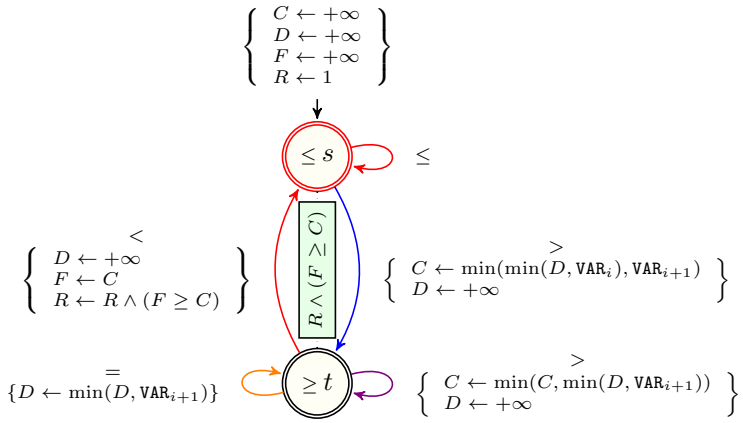
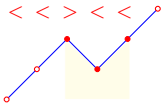


Figure 4.200: Automaton for the DECREASING_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 4,5,6,1,2,4,4,1,2,3,0,3,4,6,5,1\rangle)$

Figure 4.201 provides an example where the DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE $([4,5,6,1,2,4,4,1,2,3,0,3,4,6,5,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

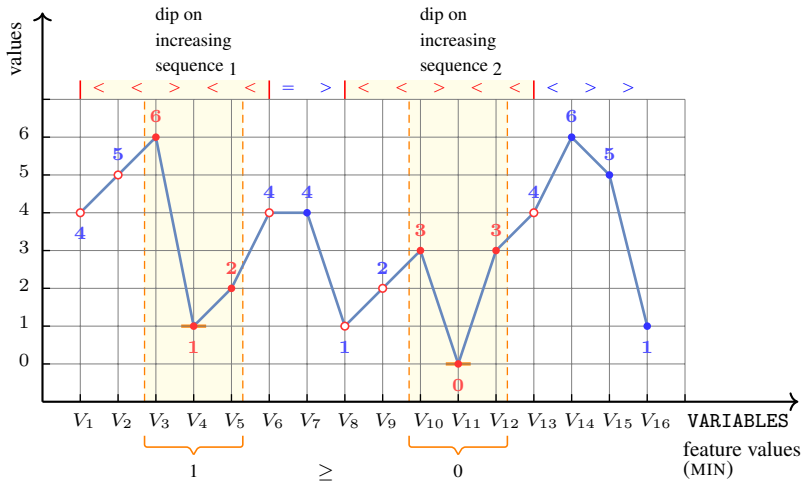


Figure 4.201: Illustrating the DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.202 depicts the automaton associated with the constraint DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE.

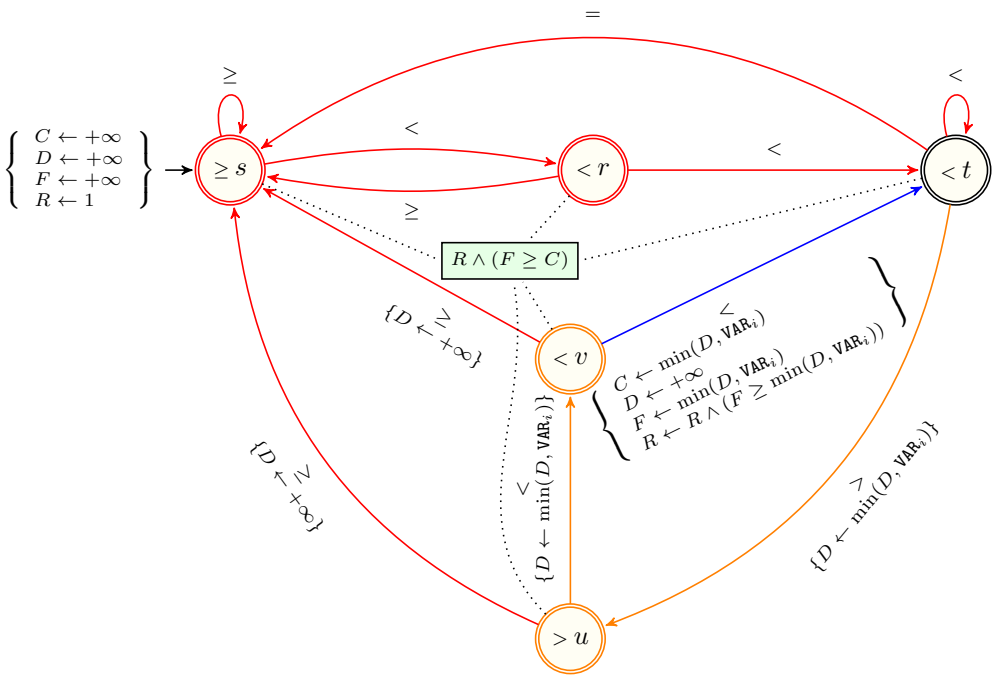
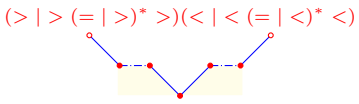


Figure 4.202: Automaton for the DECREASING_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint DECREASING_MIN_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5\rangle)$

Figure 4.203 provides an example where the DECREASING_MIN_GORGE $([6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

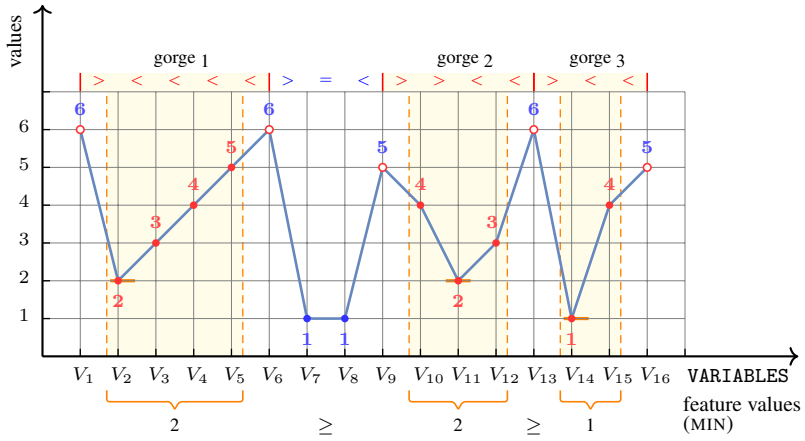


Figure 4.203: Illustrating the DECREASING_MIN_GORGE constraint of the **Example** slot

Automaton Figure 4.204 depicts the automaton associated with the constraint DECREASING_MIN_GORGE.

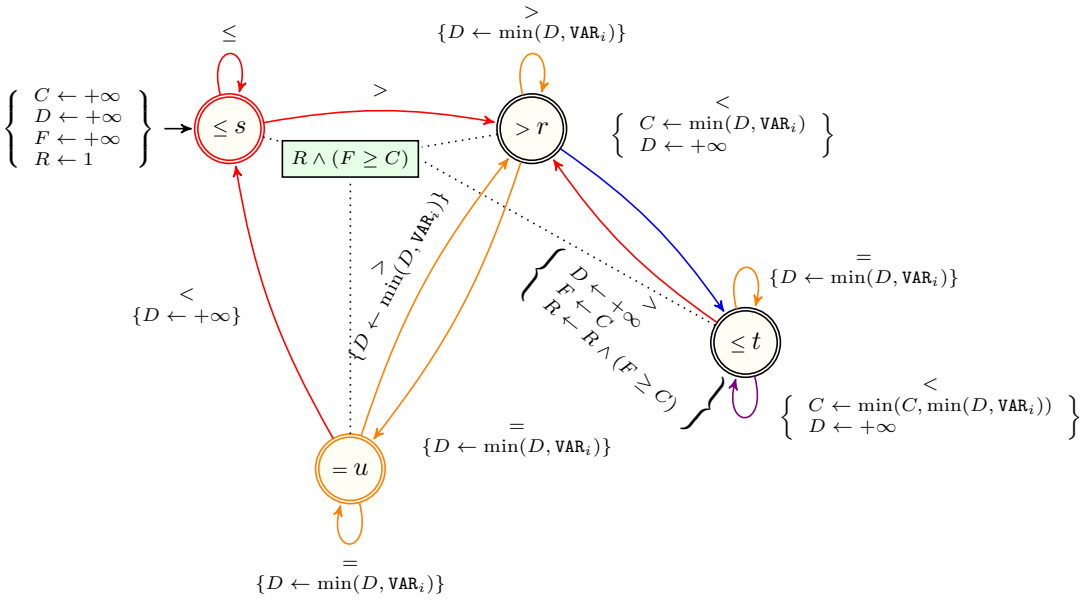


Figure 4.204: Automaton for the DECREASING_MIN_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint DECREASING_MIN_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,4,5,5,4,3,3,2,4,3,3,2,2,1,3,1\rangle)$

Figure 4.205 provides an example where the DECREASING_MIN_INCREASING $([4,4,5,5,4,3,3,2,4,3,3,2,2,1,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

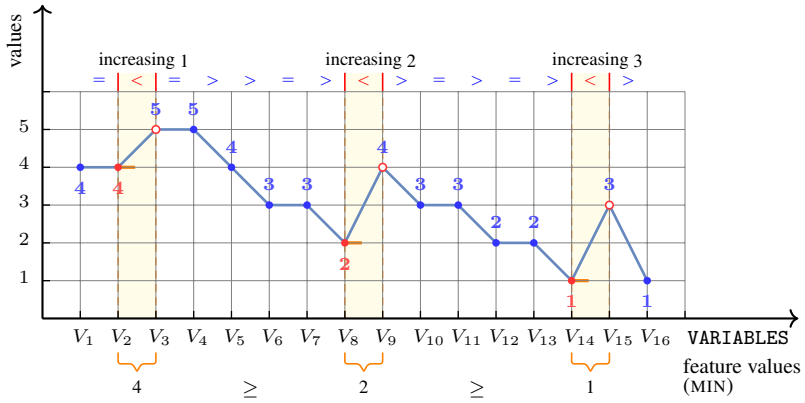


Figure 4.205: Illustrating the DECREASING_MIN_INCREASING constraint of the **Example** slot

Automaton

Figure 4.206 depicts the automaton associated with the constraint DECREASING_MIN_INCREASING.

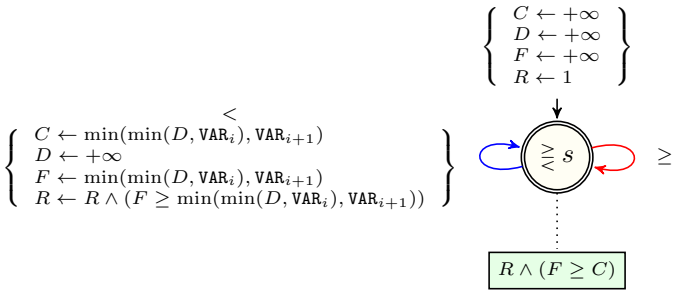
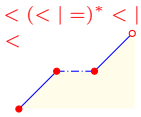


Figure 4.206: Automaton for the DECREASING_MIN_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1 \rangle)$

Figure 4.207 provides an example where the DECREASING_MIN_INCREASING_SEQUENCE $([3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

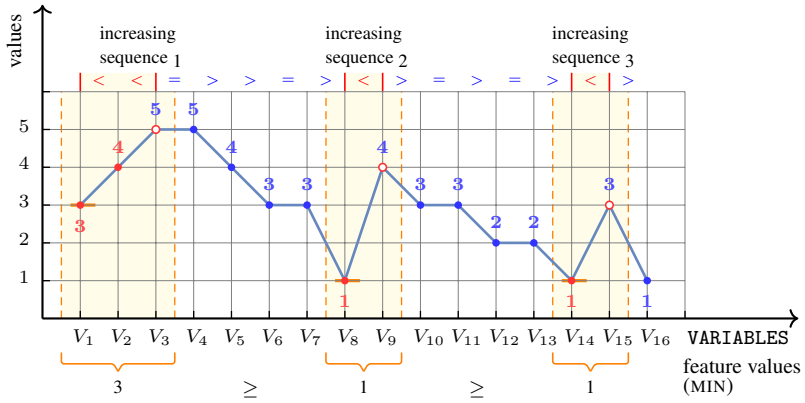


Figure 4.207: Illustrating the DECREASING_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.208 depicts the automaton associated with the constraint DECREASING_MIN_INCREASING_SEQUENCE.

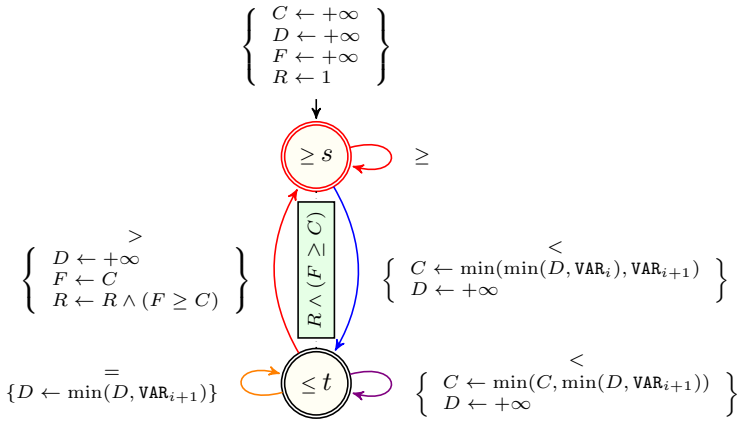
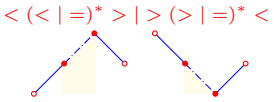


Figure 4.208: Automaton for the DECREASING_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint DECREASING_MIN_INFLEXION(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 3, 4, 3, 2, 2, 1, 1, 2, 2, 5, 5 \rangle)$

Figure 4.209 provides an example where the DECREASING_MIN_INFLEXION $([3, 4, 3, 2, 2, 1, 1, 2, 2, 5, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

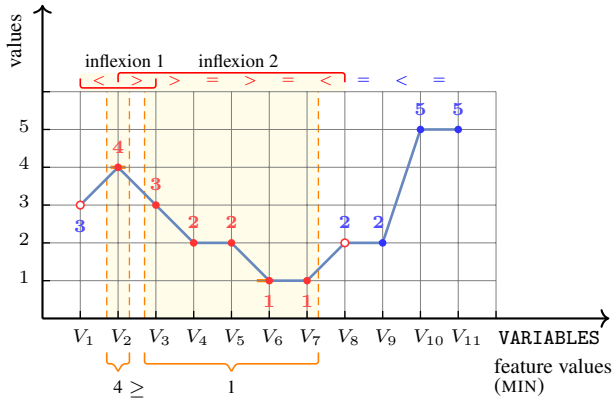


Figure 4.209: Illustrating the DECREASING_MIN_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.210 depicts the automaton associated with the constraint DECREASING_MIN_INFLEXION.

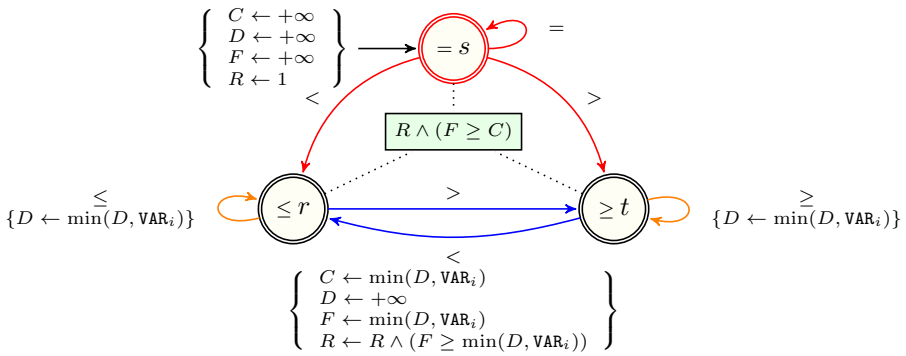


Figure 4.210: Automaton for the DECREASING_MIN_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 6,5,6,5,5,4,4,3,6,2,4,3,2,2,3,4\rangle)$

Figure 4.211 provides an example where the DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE $([6,5,6,5,5,4,4,3,6,2,4,3,2,2,3,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

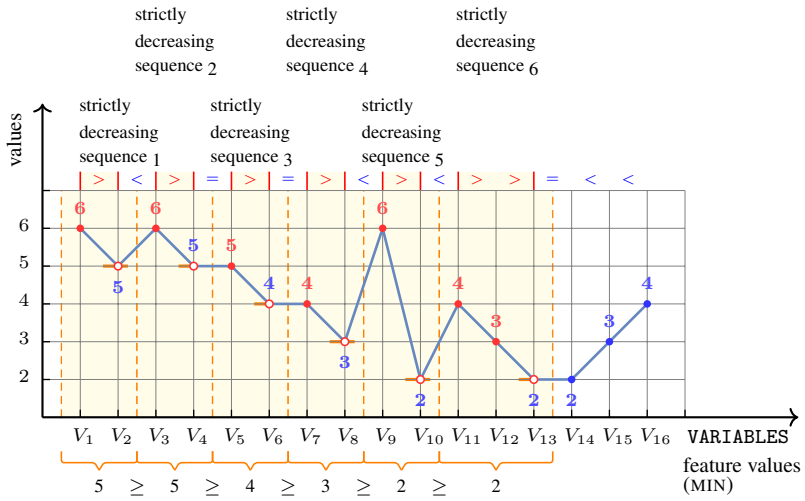


Figure 4.211: Illustrating the DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.212 depicts the automaton associated with the constraint DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE.

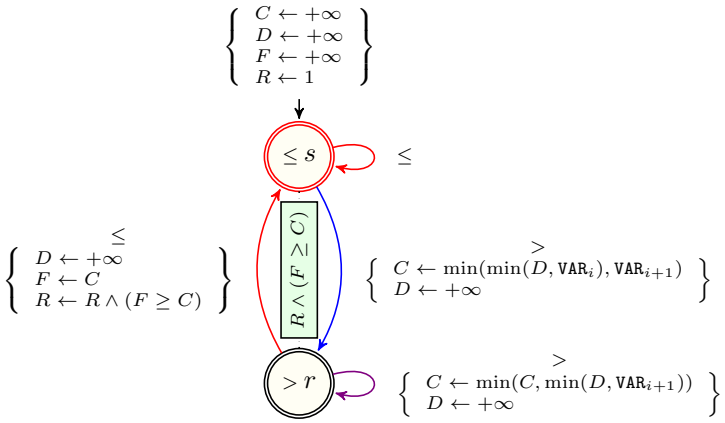


Figure 4.212: Automaton for the DECREASING_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 3,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1\rangle)$

Figure 4.213 provides an example where the DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE $([3,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

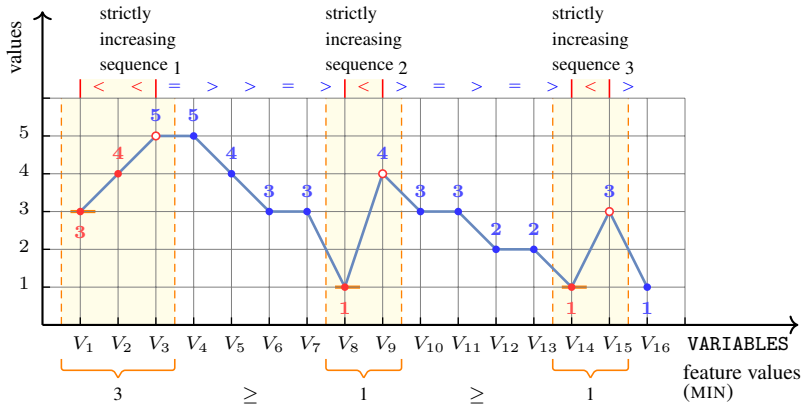


Figure 4.213: Illustrating the DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.214 depicts the automaton associated with the constraint DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE.

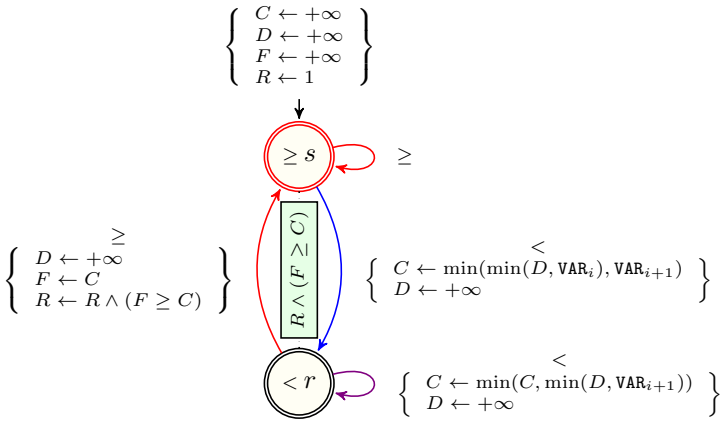
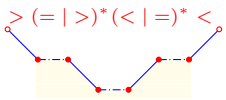


Figure 4.214: Automaton for the DECREASING_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint DECREASING_MIN_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> \ (= \mid >)^* \ (< \mid =)^* \ <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 1, 3, 3, 7, 6, 5, 4, 6, 6, 5, 4, 3, 6, 2, 2, 7 \rangle)$

Figure 4.215 provides an example where the DECREASING_MIN_VALLEY $([1, 3, 3, 7, 6, 5, 4, 6, 6, 5, 4, 3, 6, 2, 2, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

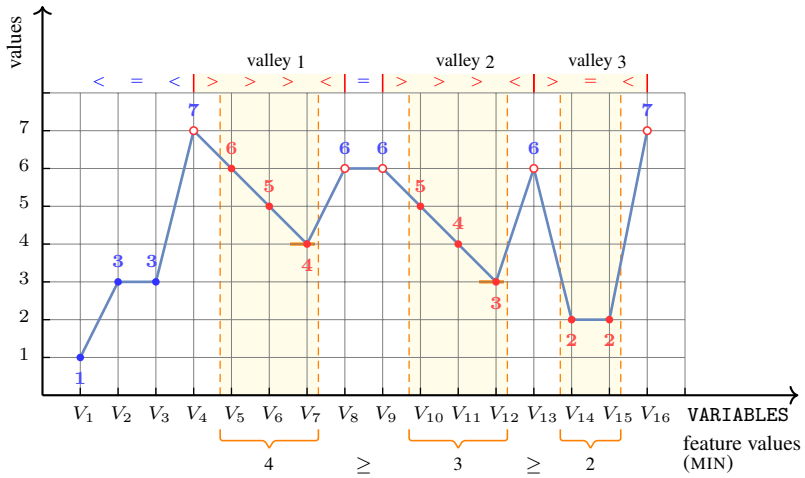


Figure 4.215: Illustrating the DECREASING_MIN_VALLEY constraint of the **Example** slot

Automaton Figure 4.216 depicts the automaton associated with the constraint DECREASING_MIN_VALLEY.

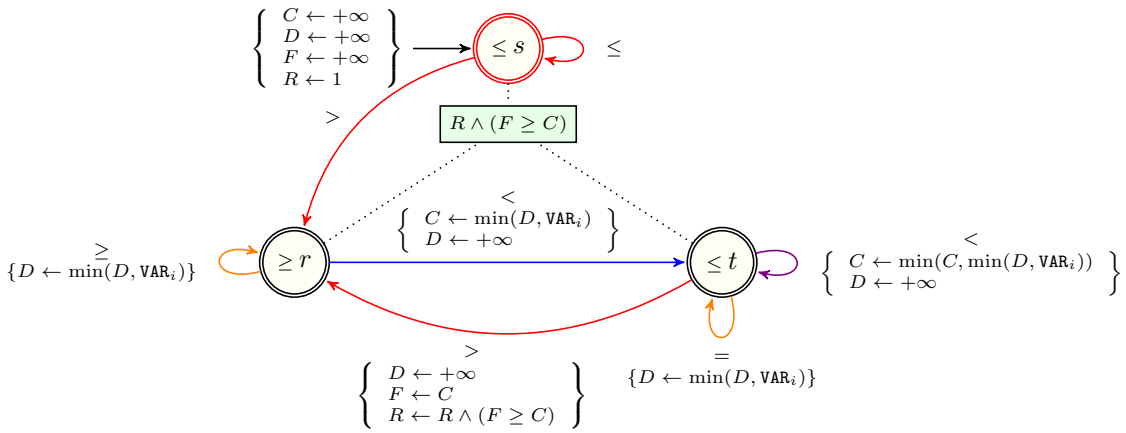
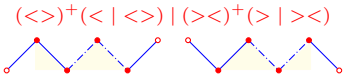


Figure 4.216: Automaton for the DECREASING_MIN_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

DECREASING_MIN_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint DECREASING_MIN_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 6,5,7,4,5,1,1,6,3,6,4,3,5,2,7,7\rangle)$

Figure 4.217 provides an example where the DECREASING_MIN_ZIGZAG $([6,5,7,4,5,1,1,6,3,6,4,3,5,2,7,7])$ constraint holds.

Typical
$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

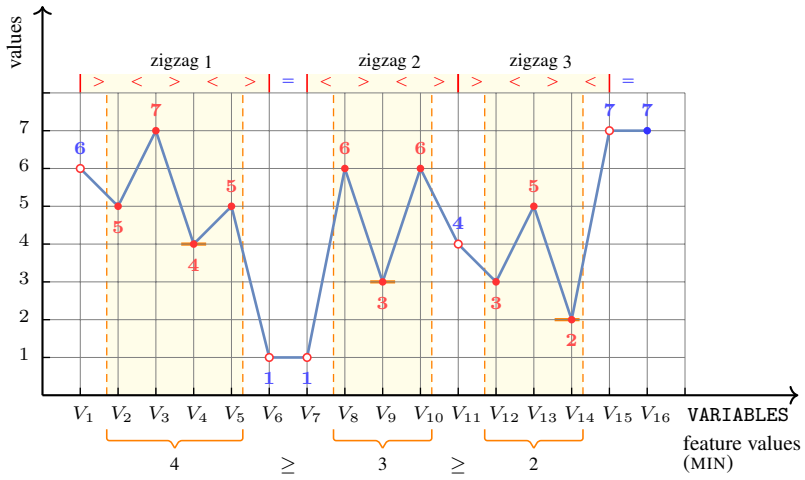


Figure 4.217: Illustrating the DECREASING_MIN_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.218 depicts the automaton associated with the constraint DECREASING_MIN_ZIGZAG.

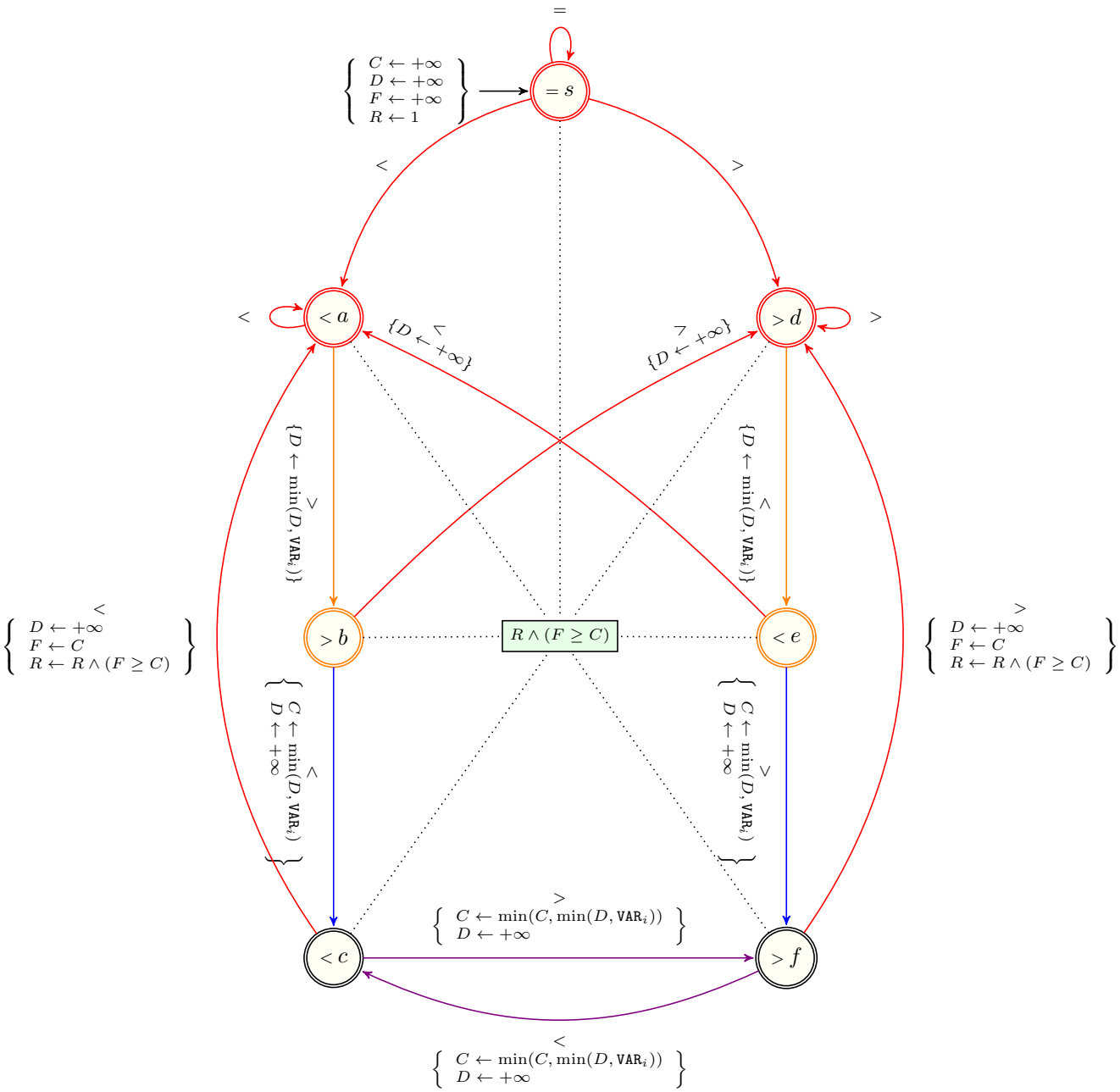


Figure 4.218: Automaton for the DECREASING_MIN_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

DECREASING_RANGE_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint DECREASING_RANGE_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 4,8,6,4,5,5,5,8,6,6,5,5,4,4,6,5\rangle)$

Figure 4.219 provides an example where the DECREASING_RANGE_DECREASING $([4,8,6,4,5,5,5,8,6,6,5,5,4,4,6,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

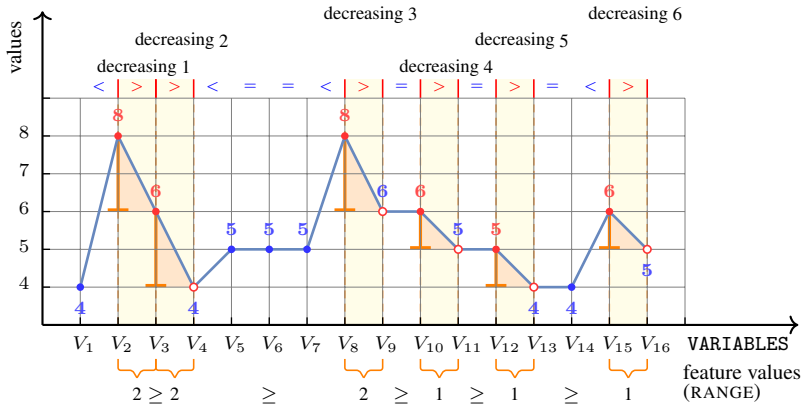


Figure 4.219: Illustrating the DECREASING_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figure 4.220 depicts the automaton associated with the constraint DECREASING_RANGE_DECREASING.

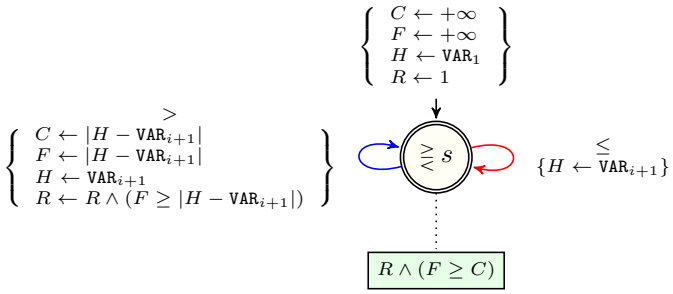
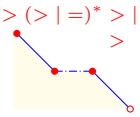


Figure 4.220: Automaton for the DECREASING_RANGE_DECREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

DECREASING_RANGE_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint DECREASING_RANGE_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 4, 7, 6, 1, 5, 5, 5, 6, 5, 5, 4, 4, 2, 2, 3, 1 \rangle)$

Figure 4.221 provides an example where the DECREASING_RANGE_DECREASING_SEQUENCE $([4, 7, 6, 1, 5, 5, 5, 6, 5, 5, 4, 4, 2, 2, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

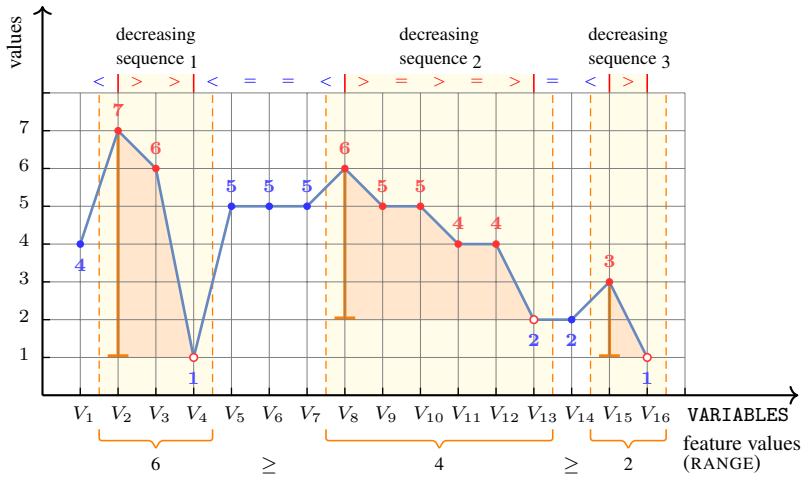


Figure 4.221: Illustrating the DECREASING_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.222 depicts the automaton associated with the constraint DECREASING_RANGE_DECREASING_SEQUENCE.

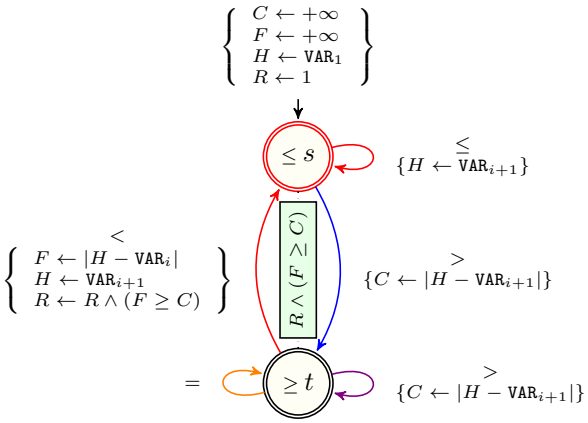


Figure 4.222: Automaton for the DECREASING_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_RANGE_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint DECREASING_RANGE_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 4,6,8,4,4,5,5,6,6,7,5,5,5,5,6,4\rangle)$

Figure 4.223 provides an example where the DECREASING_RANGE_INCREASING $([4,6,8,4,4,5,5,6,6,7,5,5,5,5,6,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

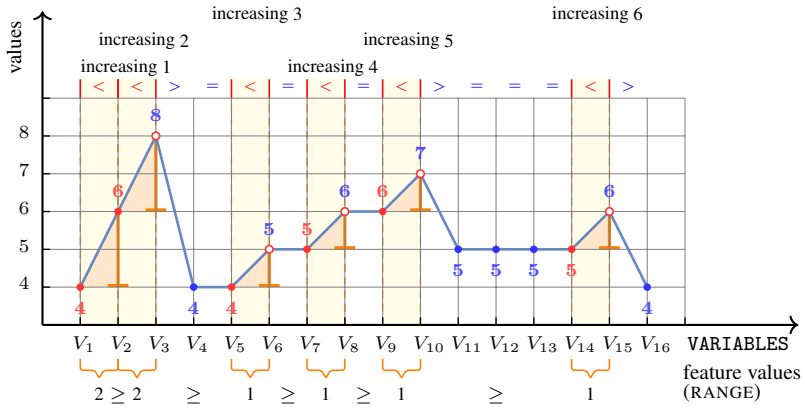


Figure 4.223: Illustrating the DECREASING_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figure 4.224 depicts the automaton associated with the constraint DECREASING_RANGE_INCREASING.

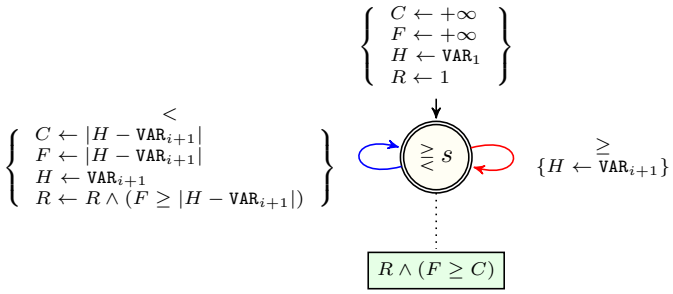
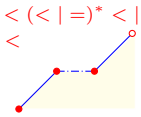


Figure 4.224: Automaton for the DECREASING_RANGE_INCREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

DECREASING_RANGE_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint DECREASING_RANGE_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 1, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1 \rangle)$

Figure 4.225 provides an example where the DECREASING_RANGE_INCREASING_SEQUENCE $([1, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

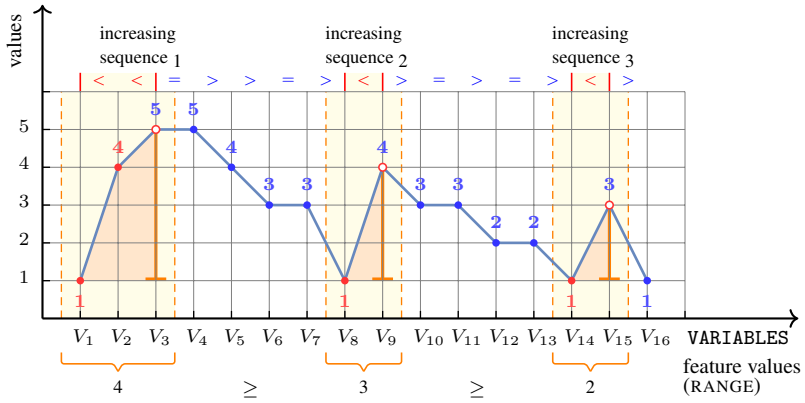


Figure 4.225: Illustrating the DECREASING_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.226 depicts the automaton associated with the constraint DECREASING_RANGE_INCREASING_SEQUENCE.

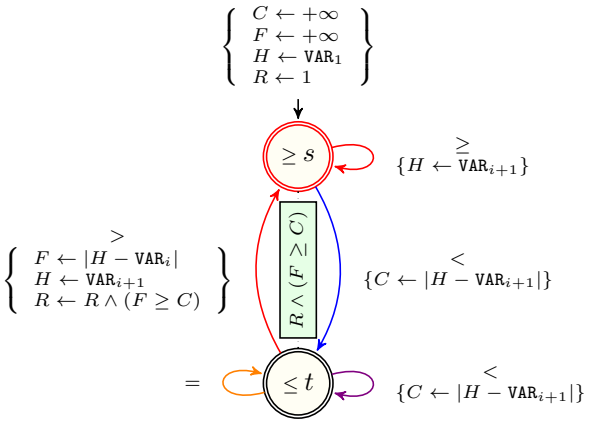


Figure 4.226: Automaton for the DECREASING_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 6,3,3,5,4,2,7,6,5,6,4,4,3,3,2,6\rangle)$

Figure 4.227 provides an example where the DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE $([6,3,3,5,4,2,7,6,5,6,4,4,3,3,2,6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

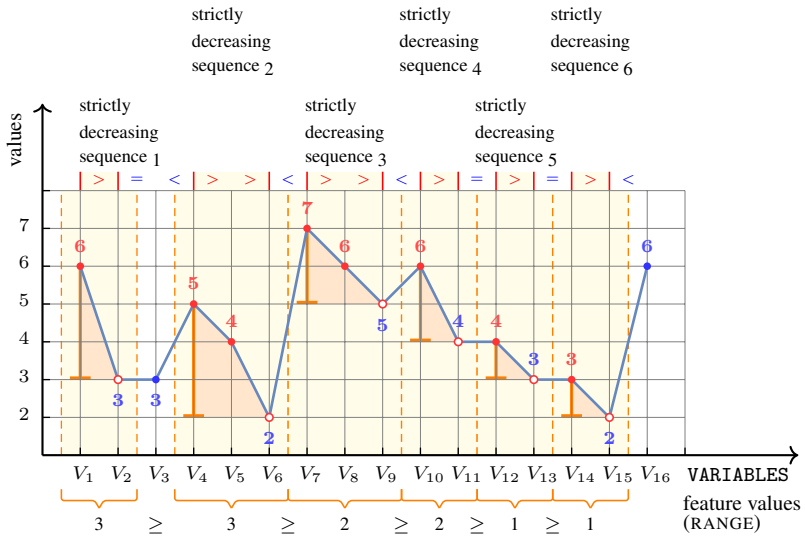


Figure 4.227: Illustrating the DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.228 depicts the automaton associated with the constraint DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE.

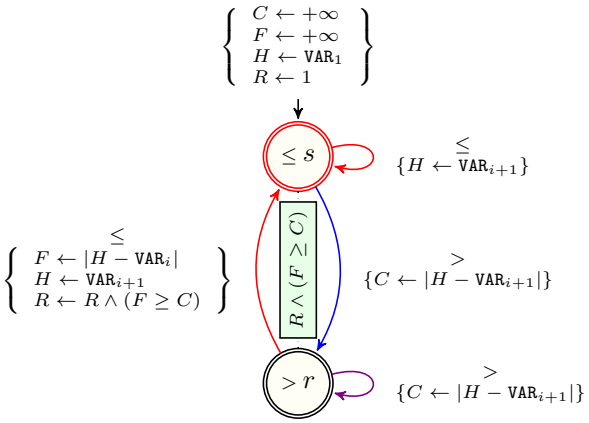


Figure 4.228: Automaton for the DECREASING_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 1, 4, 5, 5, 4, 1, 0, 2, 3, 3, 6, 2, 3, 4, 4, 6\rangle)$

Figure 4.229 provides an example where the DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE $([1, 4, 5, 5, 4, 1, 0, 2, 3, 3, 6, 2, 3, 4, 4, 6])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

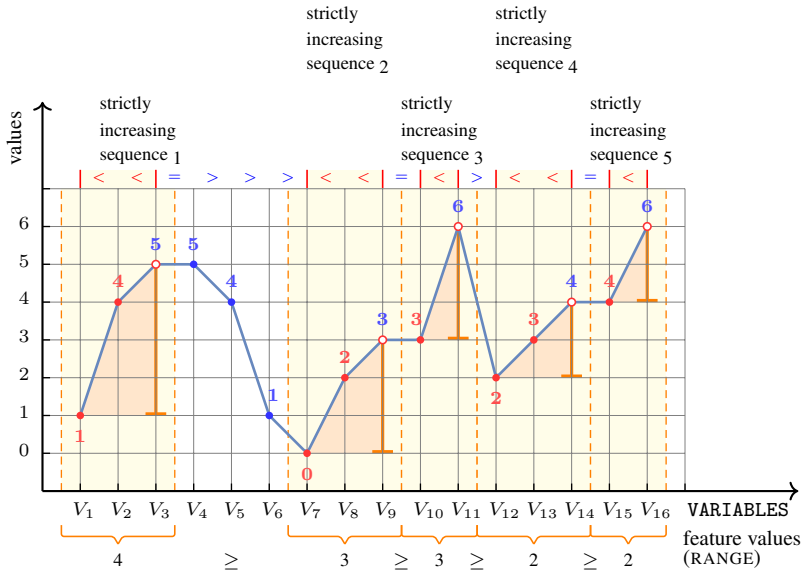


Figure 4.229: Illustrating the DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.230 depicts the automaton associated with the constraint DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE.

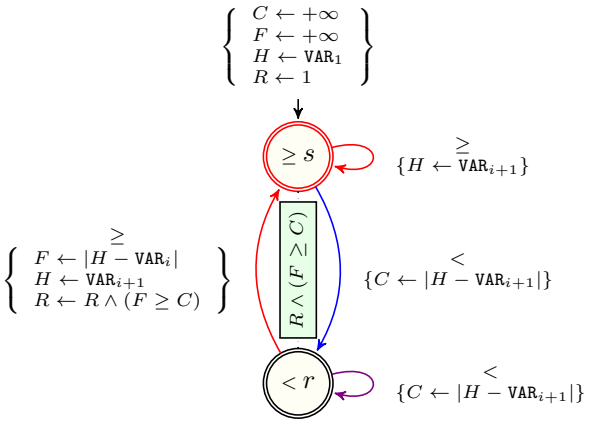
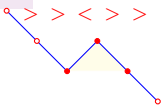


Figure 4.230: Automaton for the DECREASING_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+2$ to index j.

Example $(\langle 7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3\rangle)$

Figure 4.231 provides an example where the DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE $([7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

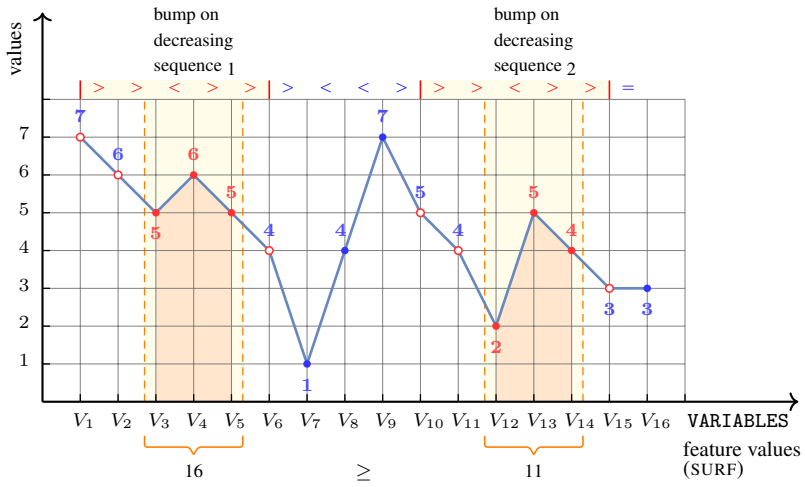


Figure 4.231: Illustrating the DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.232 depicts the automaton associated with the constraint DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE.

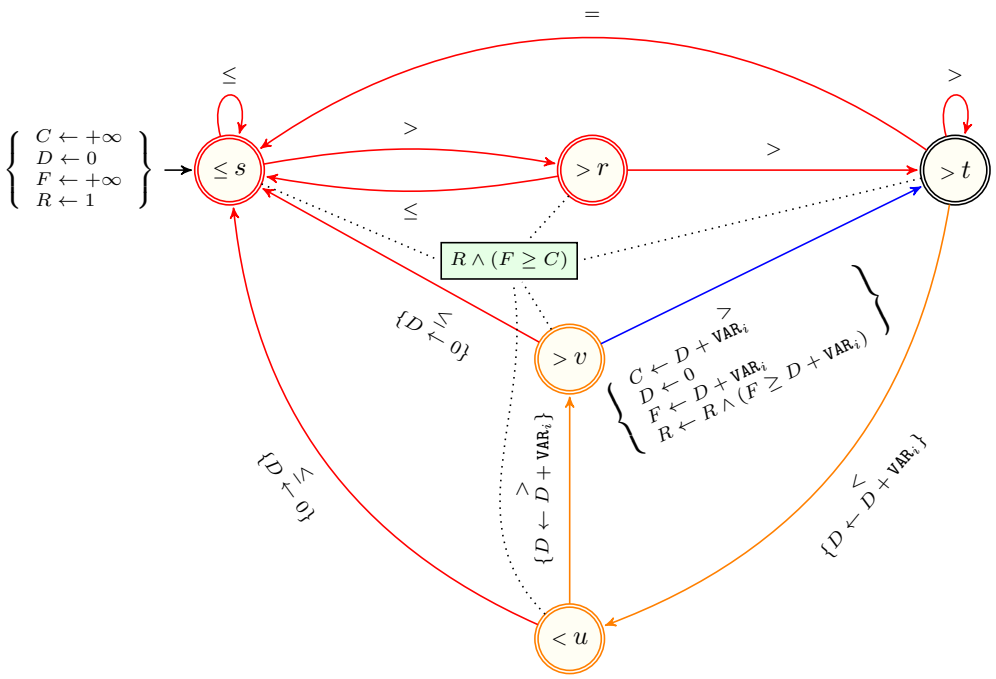


Figure 4.232: Automaton for the DECREASING_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint DECREASING_SURF_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '$>$'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 2,6,5,2,3,3,3,4,3,3,2,4,4,5,7,7\rangle)$

Figure 4.233 provides an example where the DECREASING_SURF_DECREASING $([2,6,5,2,3,3,3,4,3,3,2,4,4,5,7,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

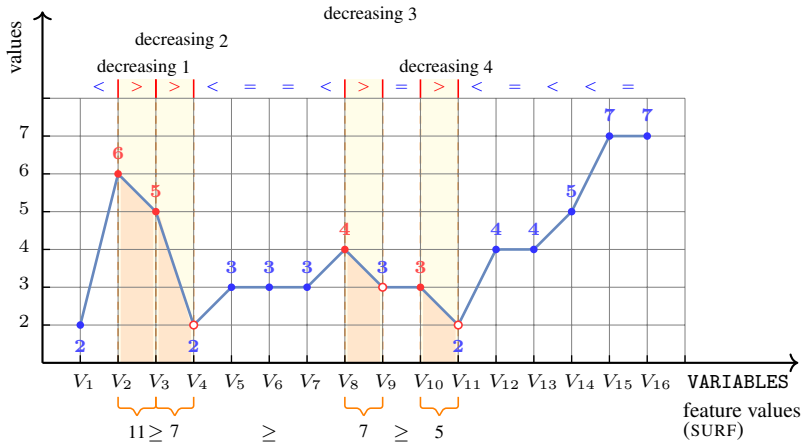


Figure 4.233: Illustrating the DECREASING_SURF_DECREASING constraint of the **Example** slot

Automaton

Figure 4.234 depicts the automaton associated with the constraint DECREASING_SURF_DECREASING.

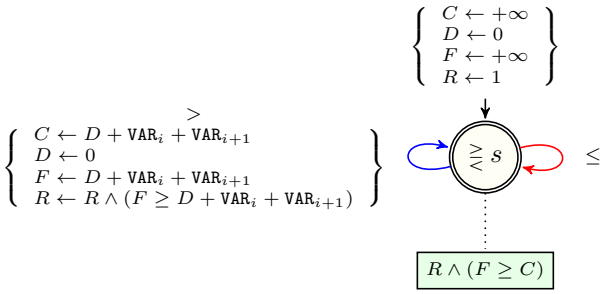
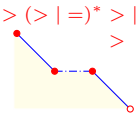


Figure 4.234: Automaton for the DECREASING_SURF_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 2, 6, 5, 2, 3, 3, 3, 4, 3, 3, 2, 4, 4, 5, 7, 5\rangle)$

Figure 4.235 provides an example where the DECREASING_SURF_DECREASING_SEQUENCE $([2, 6, 5, 2, 3, 3, 3, 4, 3, 3, 2, 4, 4, 5, 7, 5])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

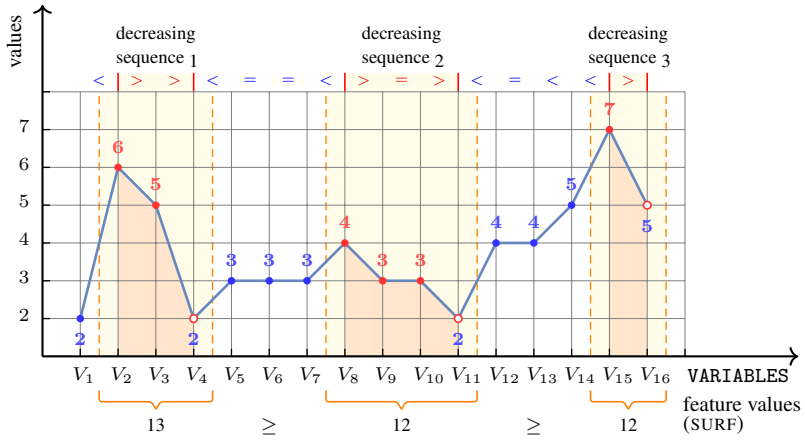


Figure 4.235: Illustrating the DECREASING_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.236 depicts the automaton associated with the constraint DECREASING_SURF_DECREASING_SEQUENCE.

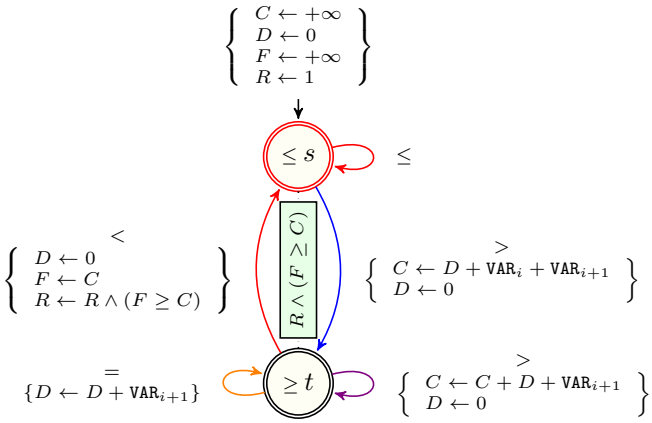
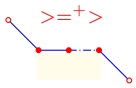


Figure 4.236: Automaton for the DECREASING_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the DECREASING_TERRACE pattern.

Constraint DECREASING_SURF_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,6,6,6,4,3,3,2,2,4,4,6,3,3,1,1\rangle)$

Figure 4.237 provides an example where the DECREASING_SURF_DECREASING_TERRACE $([7,6,6,6,4,3,3,2,2,4,4,6,3,3,1,1])$ constraint holds.

Typical |VARIABLES| > 3
range(VARIABLES.var) > 2

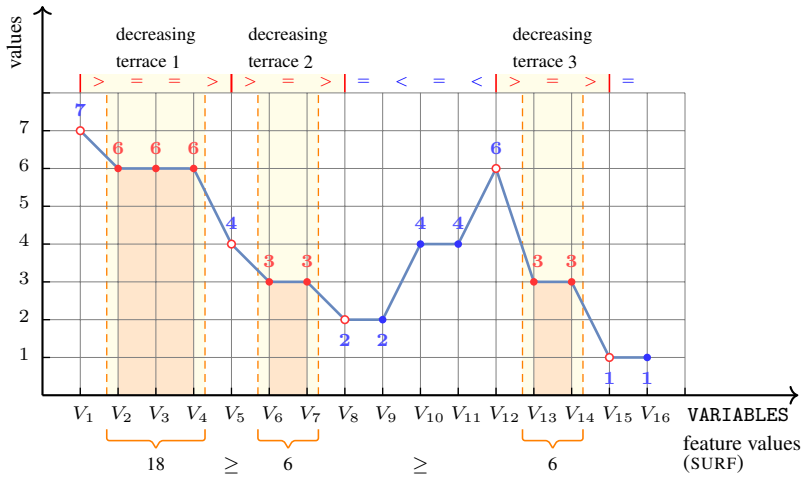


Figure 4.237: Illustrating the DECREASING_SURF_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.238 depicts the automaton associated with the constraint DECREASING_SURF_DECREASING_TERRACE.

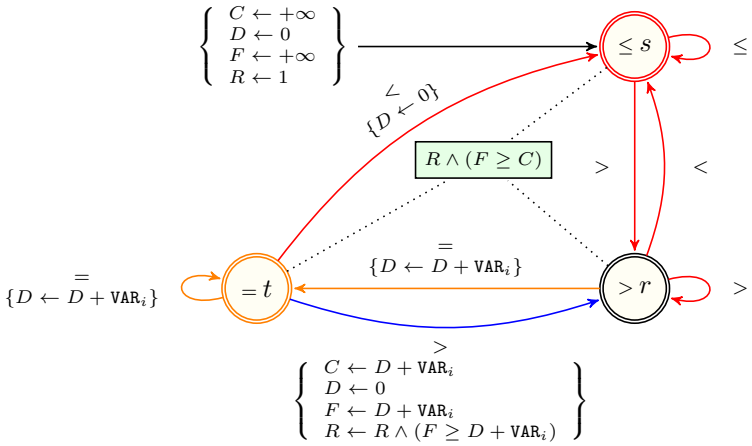
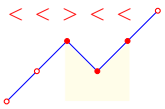


Figure 4.238: Automaton for the DECREASING_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+2$ to index j.

Example $(\langle 4,5,6,0,3,4,4,1,2,3,2,3,4,6,5,1\rangle)$

Figure 4.239 provides an example where the DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE $([4,5,6,0,3,4,4,1,2,3,2,3,4,6,5,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

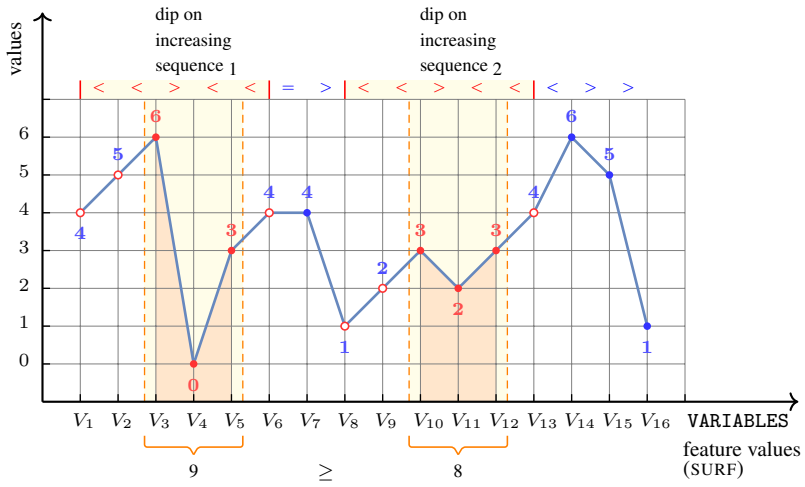


Figure 4.239: Illustrating the DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.240 depicts the automaton associated with the constraint DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE.

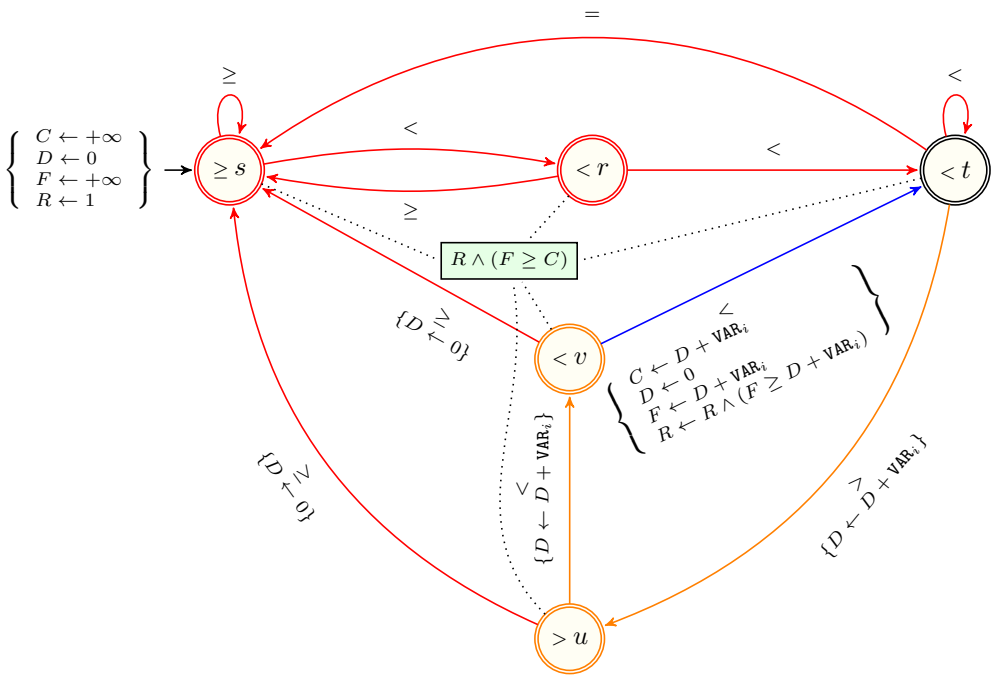
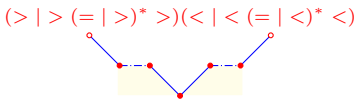


Figure 4.240: Automaton for the DECREASING_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint DECREASING_SURF_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5\rangle)$

Figure 4.241 provides an example where the DECREASING_SURF_GORGE $([6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

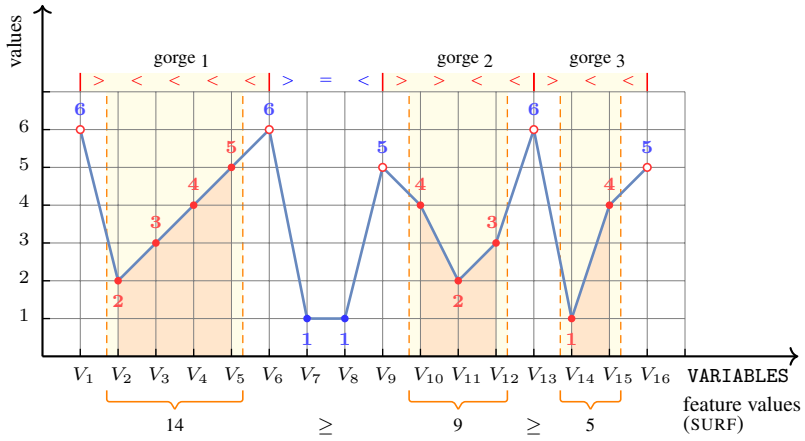


Figure 4.241: Illustrating the DECREASING_SURF_GORGE constraint of the **Example** slot

Automaton Figure 4.242 depicts the automaton associated with the constraint DECREASING_SURF_GORGE.

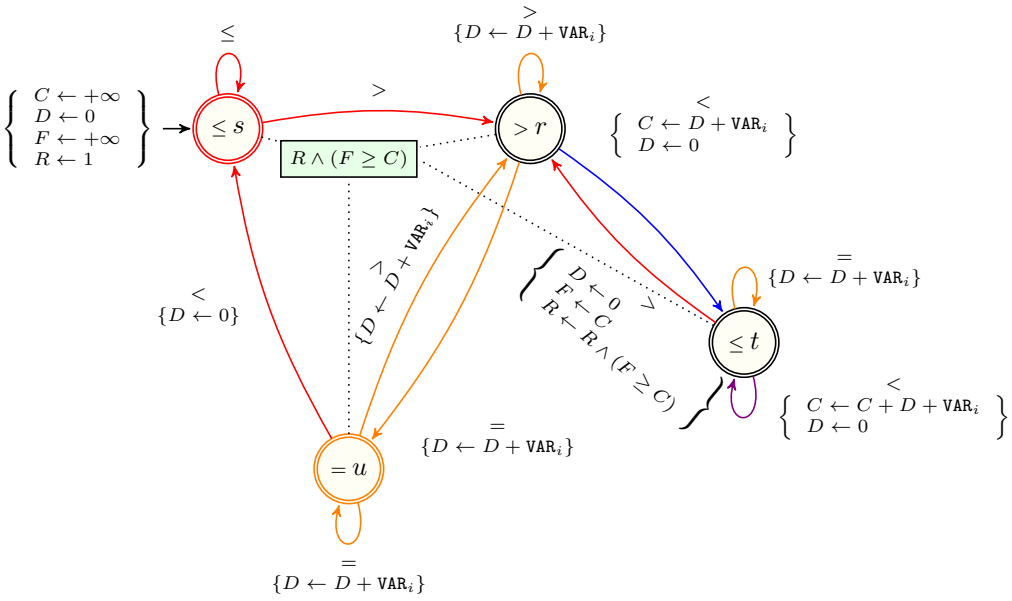


Figure 4.242: Automaton for the DECREASING_SURF_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint DECREASING_SURF_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 4,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1\rangle)$

Figure 4.243 provides an example where the DECREASING_SURF_INCREASING $([4,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

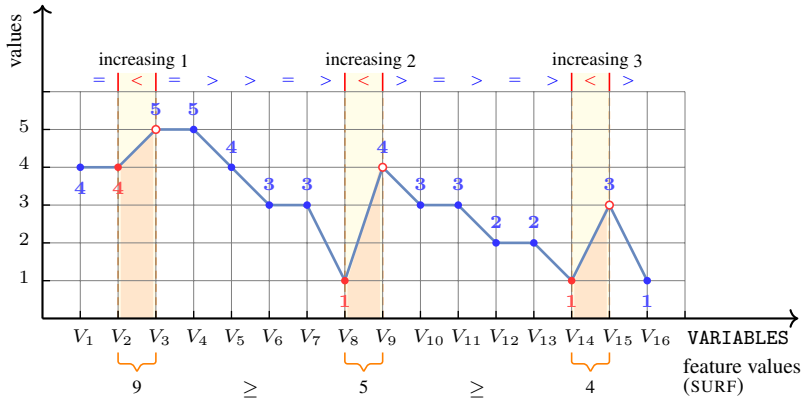


Figure 4.243: Illustrating the DECREASING_SURF_INCREASING constraint of the **Example** slot

Automaton

Figure 4.244 depicts the automaton associated with the constraint DECREASING_SURF_INCREASING.

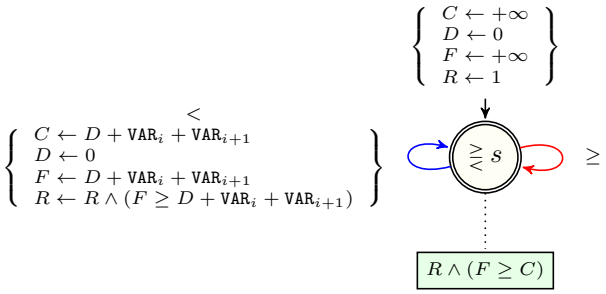
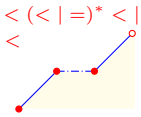


Figure 4.244: Automaton for the DECREASING_SURF_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1 \rangle)$

Figure 4.245 provides an example where the DECREASING_SURF_INCREASING_SEQUENCE $([3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

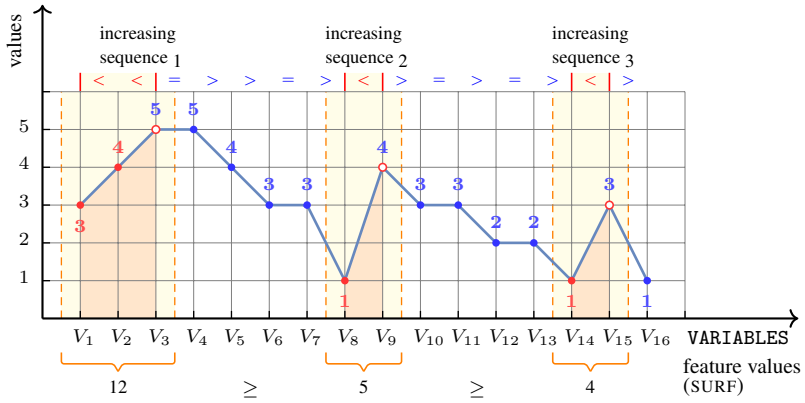


Figure 4.245: Illustrating the DECREASING_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.246 depicts the automaton associated with the constraint DECREASING_SURF_INCREASING_SEQUENCE.

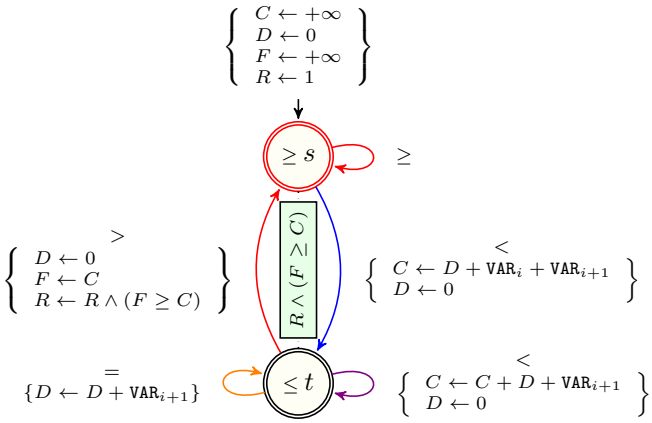
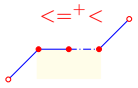


Figure 4.246: Automaton for the DECREASING_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint DECREASING_SURF_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,5,5,5,6,2,3,4,4,5,3,3,1,2,2,3\rangle)$

Figure 4.247 provides an example where the DECREASING_SURF_INCREASING_TERRACE $([2,5,5,5,6,2,3,4,4,5,3,3,1,2,2,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

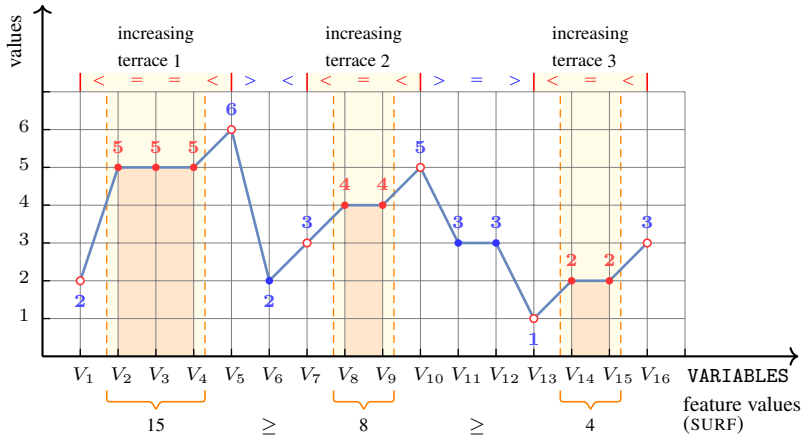


Figure 4.247: Illustrating the DECREASING_SURF_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.248 depicts the automaton associated with the constraint DECREASING_SURF_INCREASING_TERRACE.

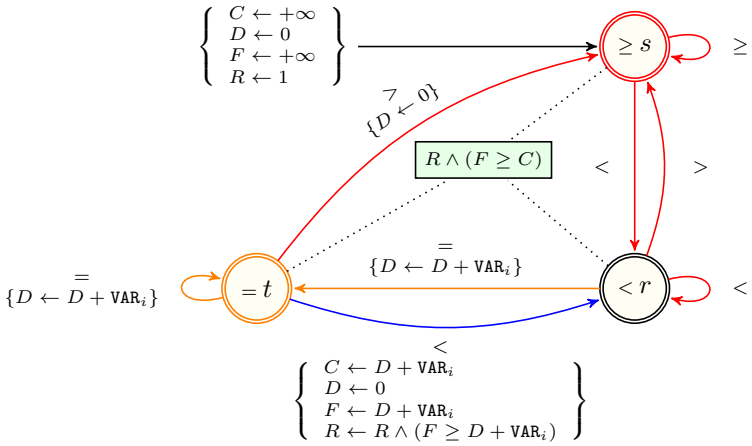
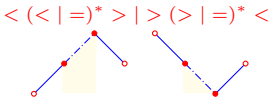


Figure 4.248: Automaton for the DECREASING_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

Constraint DECREASING_SURF_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 6,5,5,4,3,2,3,3,4,6,3,3,1,2,1,1\rangle)$

Figure 4.249 provides an example where the DECREASING_SURF_INFLEXION $([6,5,5,4,3,2,3,3,4,6,3,3,1,2,1,1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

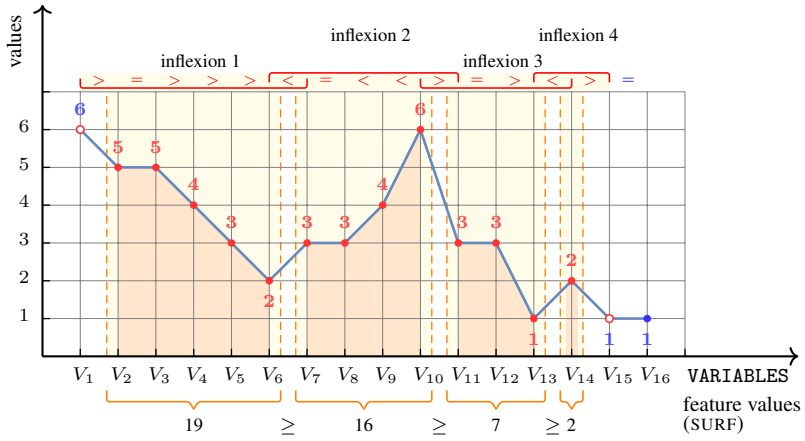


Figure 4.249: Illustrating the DECREASING_SURF_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.250 depicts the automaton associated with the constraint DECREASING_SURF_INFLEXION.

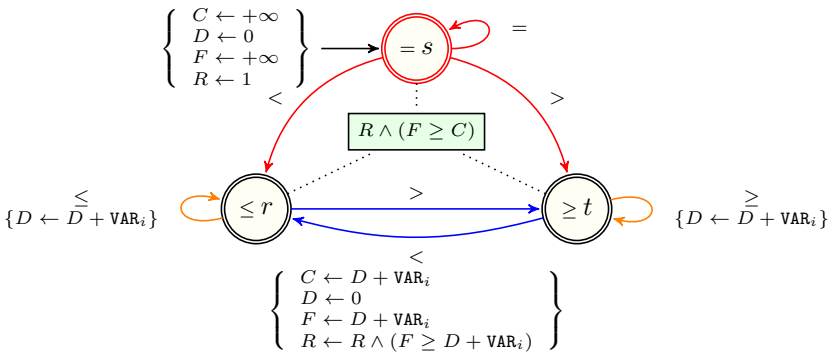
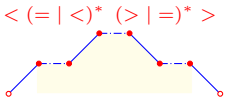


Figure 4.250: Automaton for the DECREASING_SURF_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

DECREASING_SURF_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint DECREASING_SURF_PEAK(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7 \rangle)$

Figure 4.251 provides an example where the DECREASING_SURF_PEAK $([1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

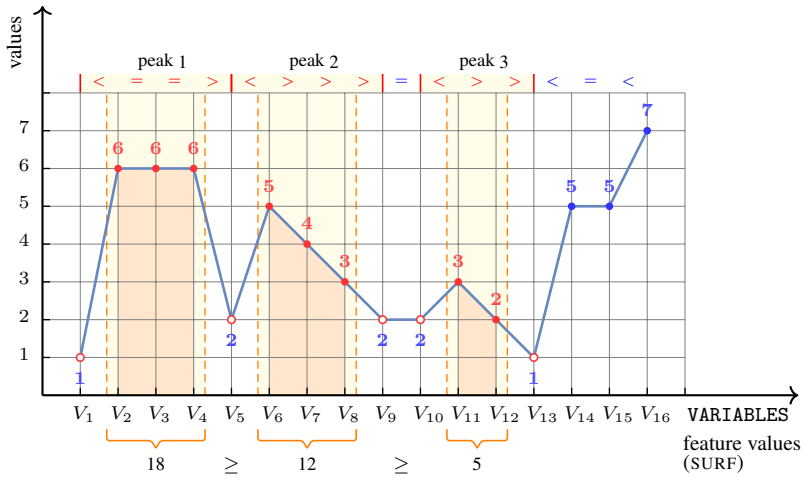


Figure 4.251: Illustrating the DECREASING_SURF_PEAK constraint of the **Example** slot

Automaton

Figure 4.252 depicts the automaton associated with the constraint DECREASING_SURF_PEAK.

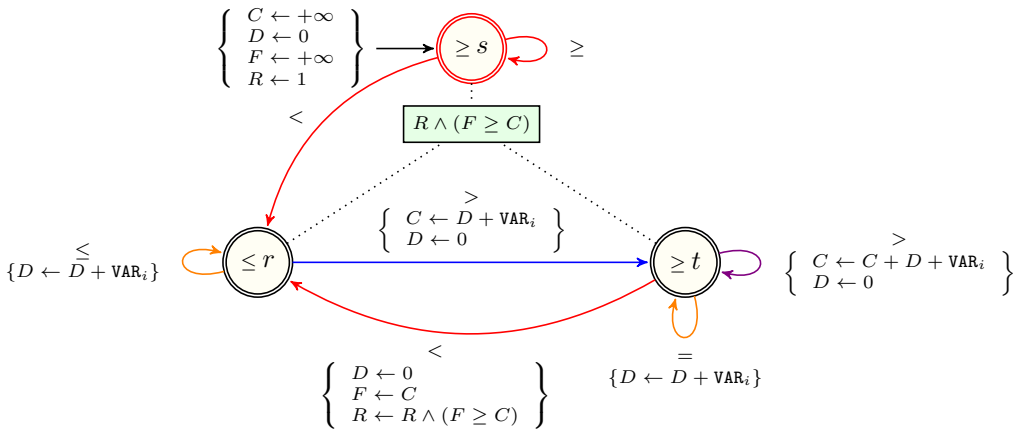


Figure 4.252: Automaton for the DECREASING_SURF_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint DECREASING_SURF_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3\rangle)$

Figure 4.253 provides an example where the DECREASING_SURF_PLAIN $([2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

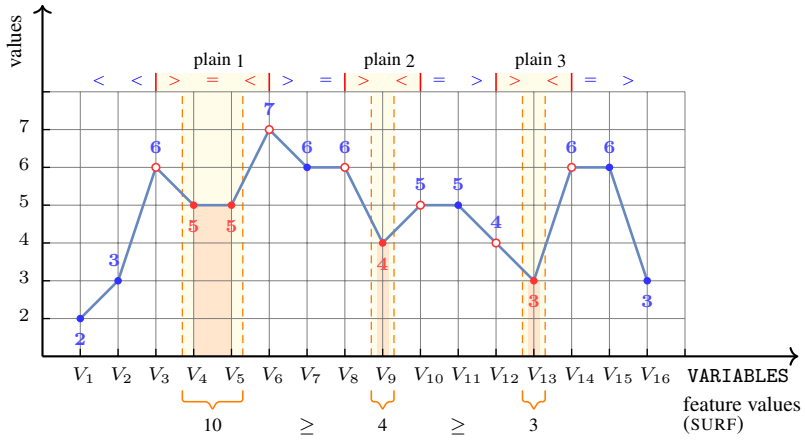


Figure 4.253: Illustrating the DECREASING_SURF_PLAIN constraint of the **Example** slot

Automaton

Figure 4.254 depicts the automaton associated with the constraint DECREASING_SURF_PLAIN.

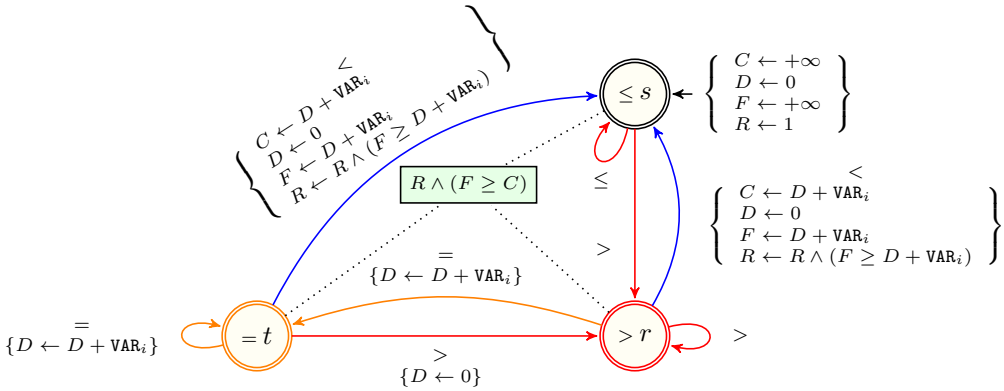


Figure 4.254: Automaton for the DECREASING_SURF_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint DECREASING_SURF_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 5,2,2,5,5,4,3,3,4,2,2,1,3,2,5,7\rangle)$

Figure 4.255 provides an example where the DECREASING_SURF_PLATEAU $([5,2,2,5,5,4,3,3,4,2,2,1,3,2,5,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

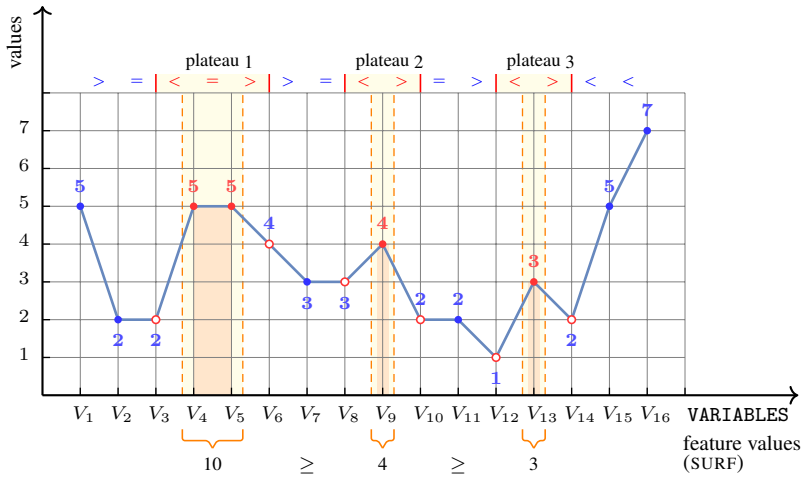


Figure 4.255: Illustrating the DECREASING_SURF_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.256 depicts the automaton associated with the constraint DECREASING_SURF_PLATEAU.

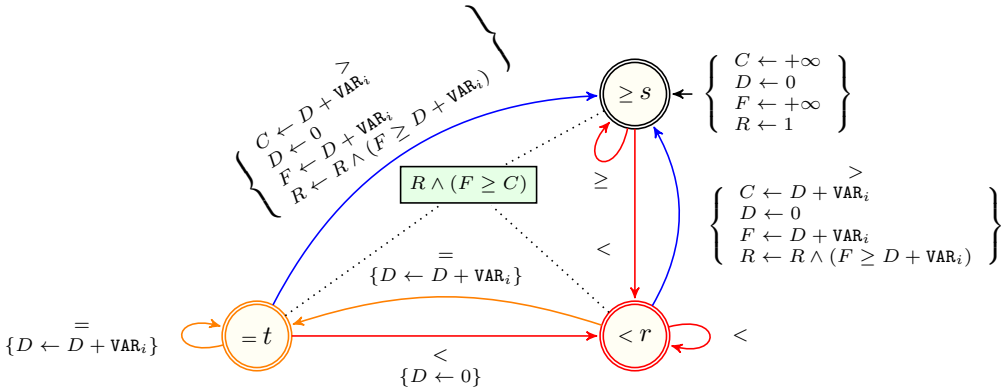
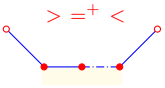


Figure 4.256: Automaton for the DECREASING_SURF_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint DECREASING_SURF_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 2,7,5,5,5,6,3,7,4,4,5,6,5,3,3,5\rangle)$

Figure 4.257 provides an example where the DECREASING_SURF_PROPER_PLAIN $([2,7,5,5,5,6,3,7,4,4,5,6,5,3,3,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

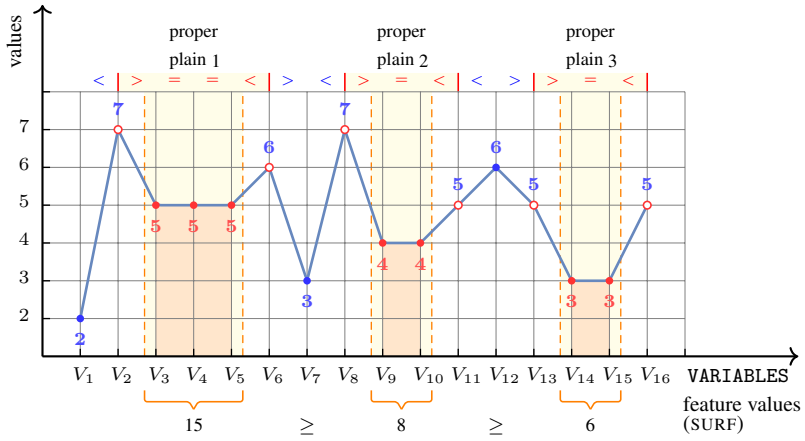


Figure 4.257: Illustrating the DECREASING_SURF_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.258 depicts the automaton associated with the constraint DECREASING_SURF_PROPER_PLAIN.

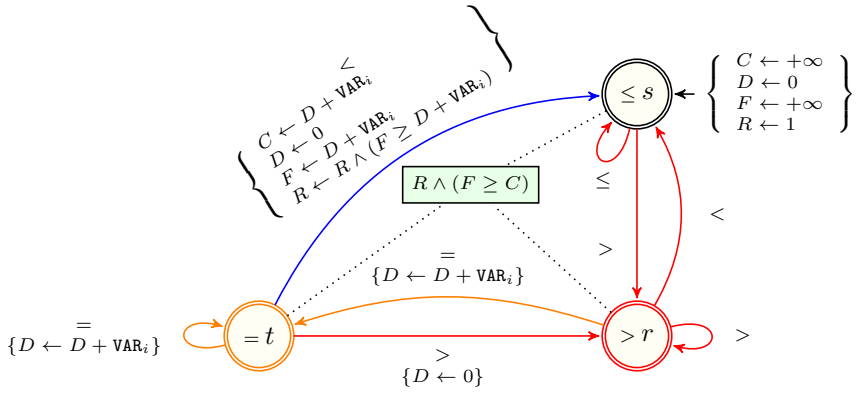
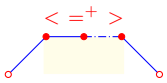


Figure 4.258: Automaton for the DECREASING_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLATEAU pattern.

Constraint DECREASING_SURF_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$< =^{+} >$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 3,5,5,5,3,2,3,4,4,1,5,2,3,3,1,7\rangle)$

Figure 4.259 provides an example where the DECREASING_SURF_PROPER_PLATEAU $([3,5,5,5,3,2,3,4,4,1,5,2,3,3,1,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

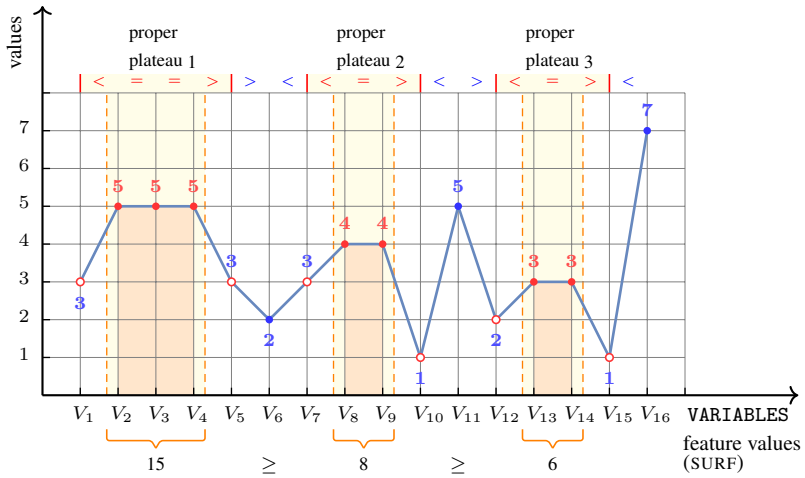


Figure 4.259: Illustrating the DECREASING_SURF_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.260 depicts the automaton associated with the constraint DECREASING_SURF_PROPER_PLATEAU.

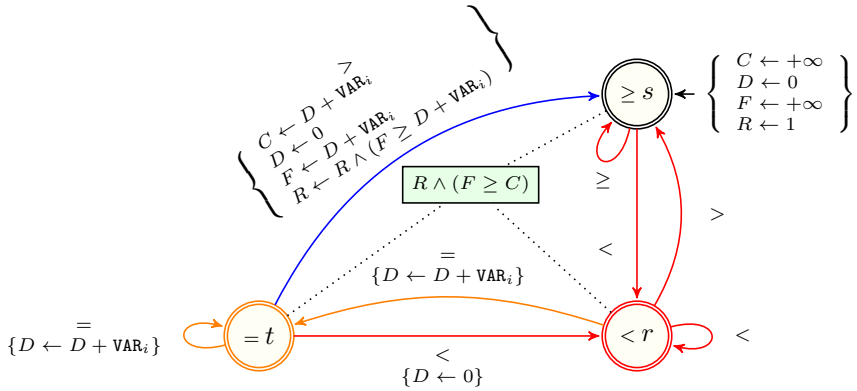


Figure 4.260: Automaton for the DECREASING_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint DECREASING_SURF_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 3,6,6,6,2,7,5,5,5,6,5,5,5,3,3,7\rangle)$

Figure 4.261 provides an example where the DECREASING_SURF_STEADY $([3,6,6,6,2,7,5,5,5,6,5,5,5,3,3,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

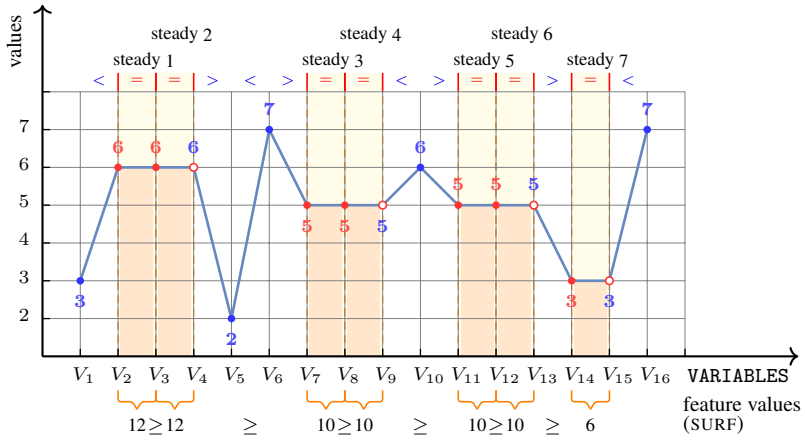


Figure 4.261: Illustrating the DECREASING_SURF_STEADY constraint of the **Example** slot

Automaton

Figure 4.262 depicts the automaton associated with the constraint DECREASING_SURF_STEADY.

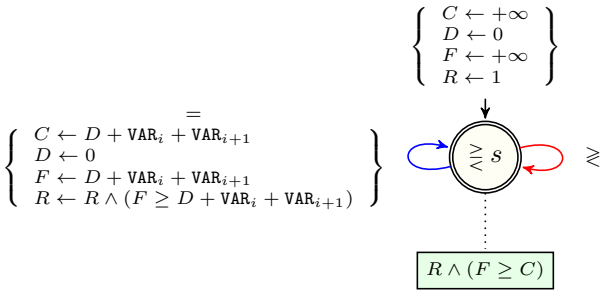
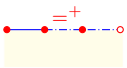


Figure 4.262: Automaton for the DECREASING_SURF_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_STEADY_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the STEADY_SEQUENCE pattern.

Constraint DECREASING_SURF_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 4,3,2,5,5,5,3,2,4,4,6,5,3,2,2,6\rangle)$

Figure 4.263 provides an example where the DECREASING_SURF_STEADY_SEQUENCE $([4,3,2,5,5,5,3,2,4,4,6,5,3,2,2,6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

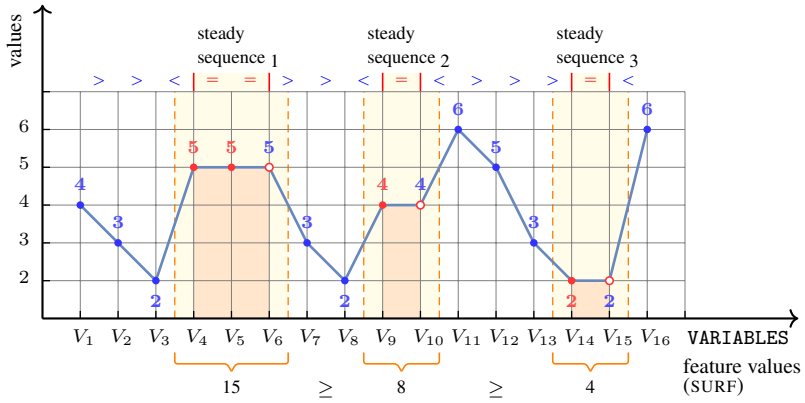


Figure 4.263: Illustrating the DECREASING_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.264 depicts the automaton associated with the constraint DECREASING_SURF_STEADY_SEQUENCE.

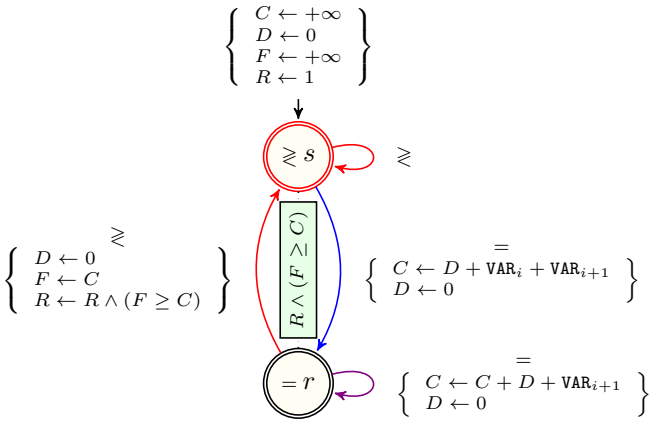


Figure 4.264: Automaton for the DECREASING_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 7,4,6,5,5,4,4,3,6,1,4,2,1,2,3,4\rangle)$

Figure 4.265 provides an example where the DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE $([7,4,6,5,5,4,4,3,6,1,4,2,1,2,3,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

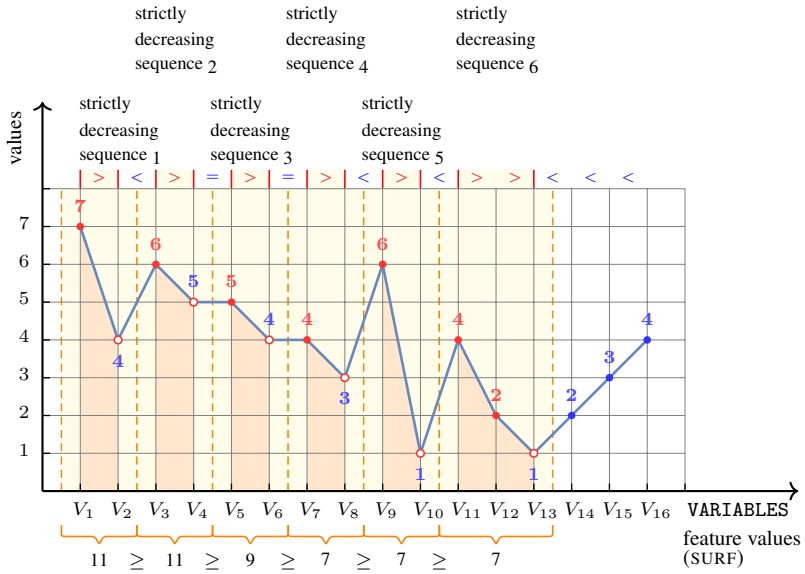


Figure 4.265: Illustrating the DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.266 depicts the automaton associated with the constraint DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE.

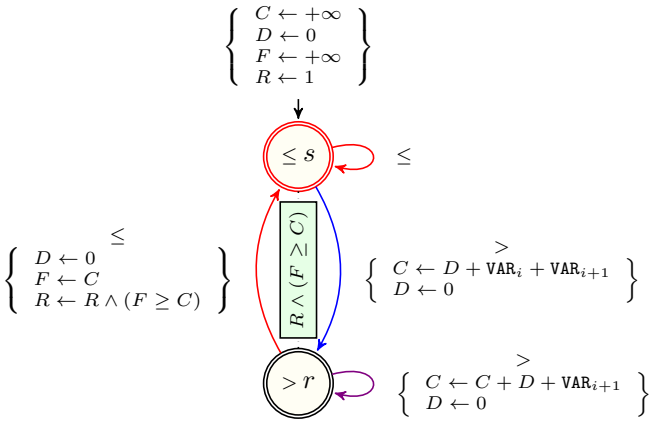


Figure 4.266: Automaton for the DECREASING_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 3,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1\rangle)$

Figure 4.267 provides an example where the DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE $([3,4,5,5,4,3,3,1,4,3,3,2,2,1,3,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

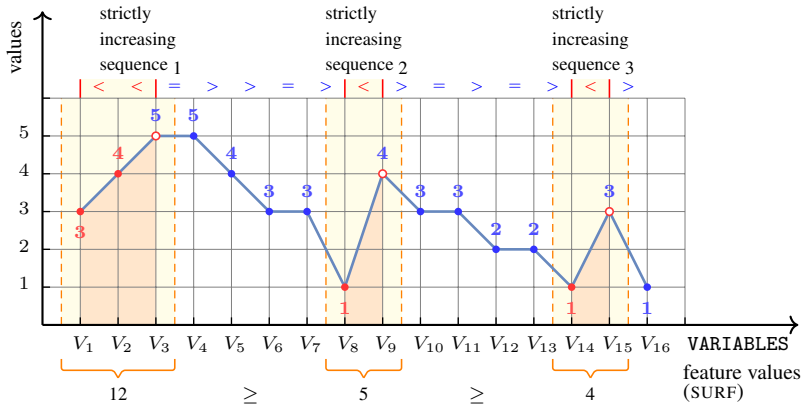


Figure 4.267: Illustrating the DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.268 depicts the automaton associated with the constraint DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE.

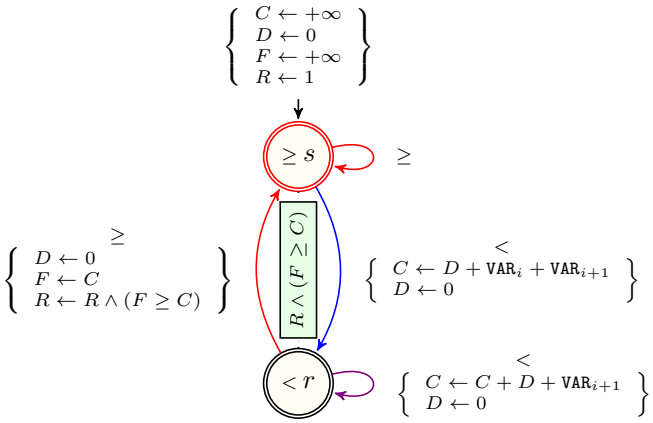
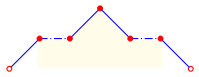


Figure 4.268: Automaton for the DECREASING_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint DECREASING_SURF_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2 \rangle)$

Figure 4.269 provides an example where the DECREASING_SURF_SUMMIT $([1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

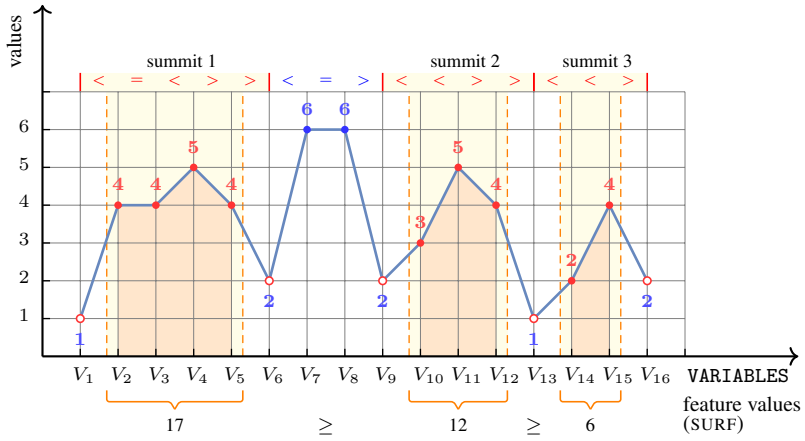


Figure 4.269: Illustrating the DECREASING_SURF_SUMMIT constraint of the **Example** slot

Automaton Figure 4.270 depicts the automaton associated with the constraint DECREASING_SURF_SUMMIT.

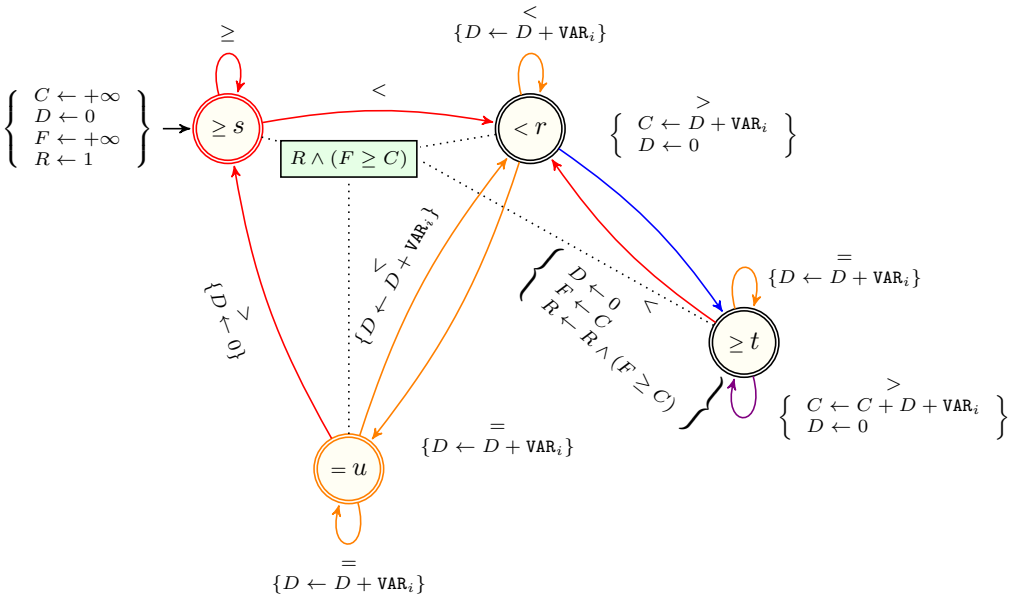
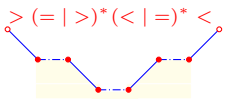


Figure 4.270: Automaton for the DECREASING_SURF_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_SURF_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint DECREASING_SURF_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1,3,3,7,6,5,4,6,6,5,4,3,6,2,2,7\rangle)$

Figure 4.271 provides an example where the DECREASING_SURF_VALLEY $([1,3,3,7,6,5,4,6,6,5,4,3,6,2,2,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

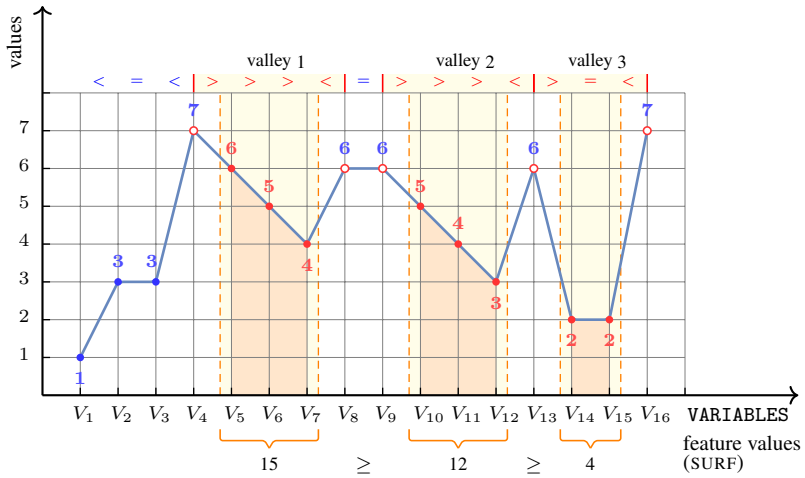


Figure 4.271: Illustrating the DECREASING_SURF_VALLEY constraint of the **Example** slot

Automaton Figure 4.272 depicts the automaton associated with the constraint DECREASING_SURF_VALLEY.

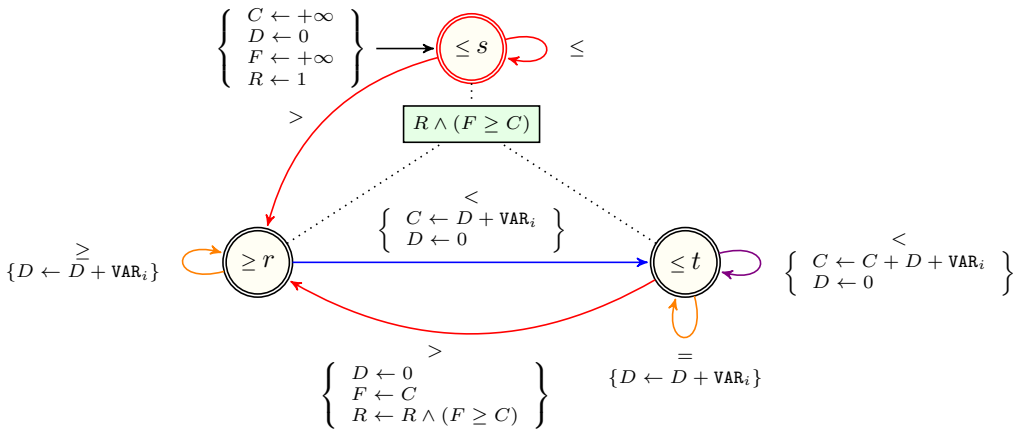
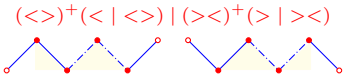


Figure 4.272: Automaton for the DECREASING_SURF_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

DECREASING_SURF_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint DECREASING_SURF_ZIGZAG(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^{+}(< \mid <>) \mid (><)^{+}(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 6,5,7,4,5,1,1,6,3,6,4,3,5,2,7,7\rangle)$

Figure 4.273 provides an example where the DECREASING_SURF_ZIGZAG $([6,5,7,4,5,1,1,6,3,6,4,3,5,2,7,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

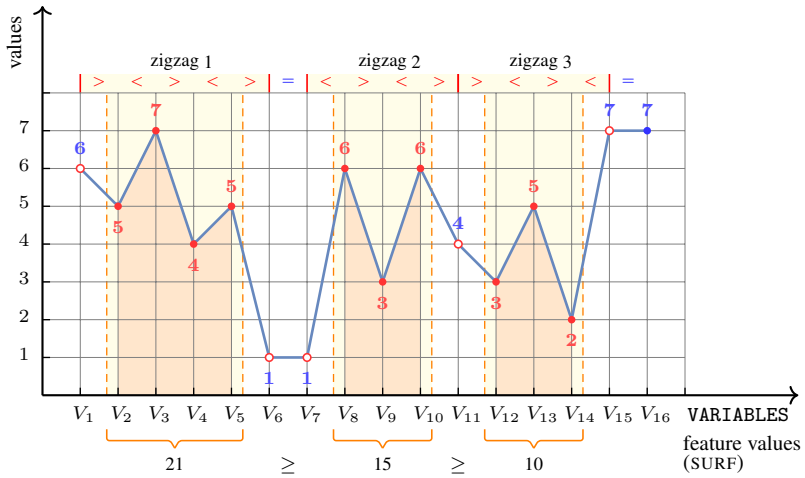


Figure 4.273: Illustrating the DECREASING_SURF_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.274 depicts the automaton associated with the constraint DECREASING_SURF_ZIGZAG.

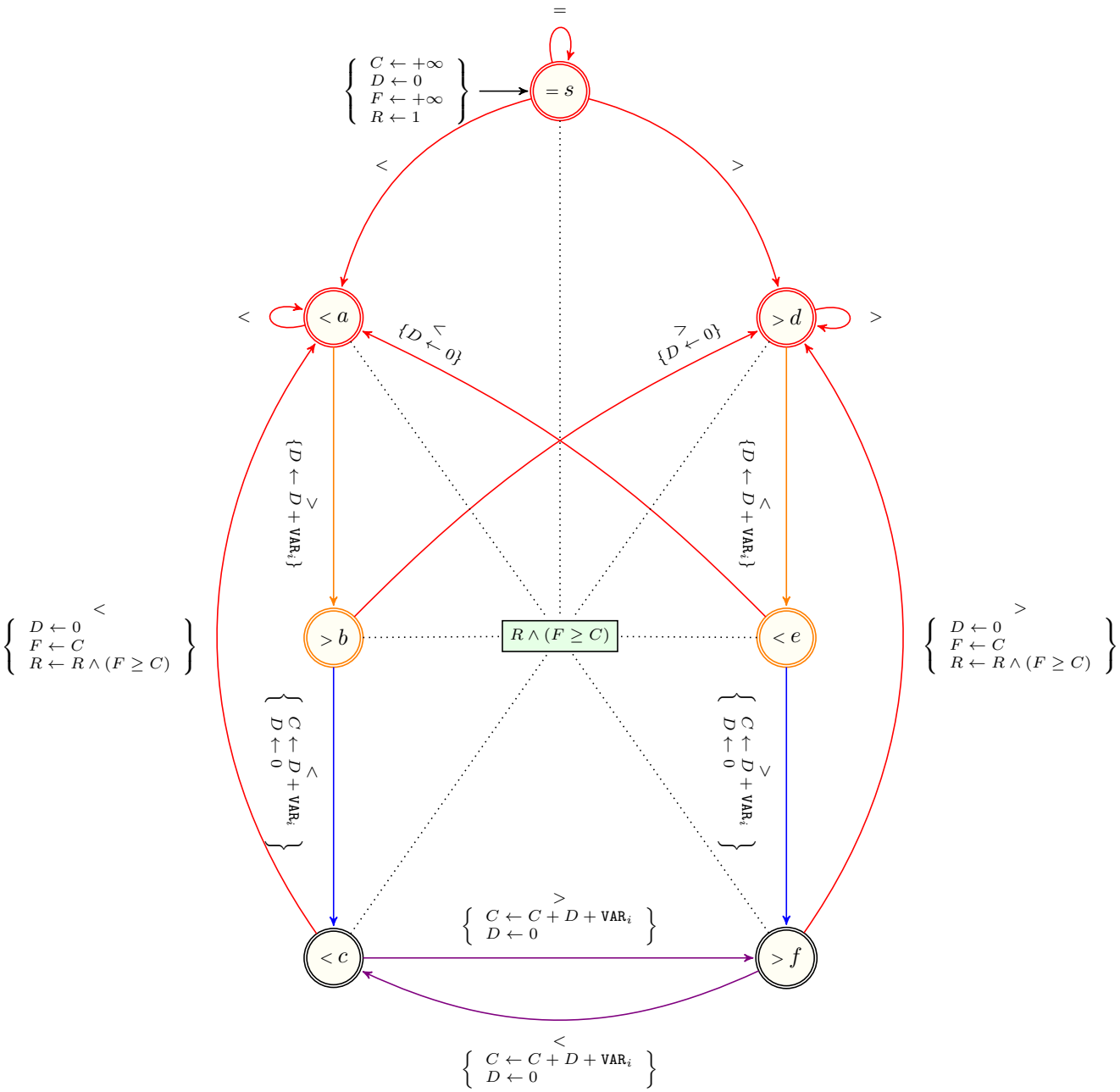
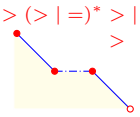


Figure 4.274: Automaton for the DECREASING_SURF_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

DECREASING_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint DECREASING_WIDTH_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 2, 6, 5, 2, 3, 3, 3, 4, 3, 2, 3, 4, 4, 5, 7, 5\rangle)$

Figure 4.275 provides an example where the DECREASING_WIDTH_DECREASING_SEQUENCE $([2, 6, 5, 2, 3, 3, 3, 4, 3, 2, 3, 4, 4, 5, 7, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

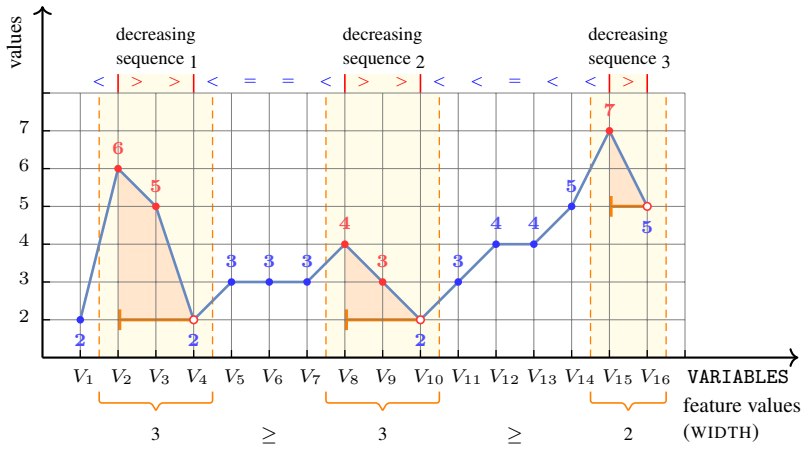


Figure 4.275: Illustrating the DECREASING_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.276 depicts the automaton associated with the constraint DECREASING_WIDTH_DECREASING_SEQUENCE.

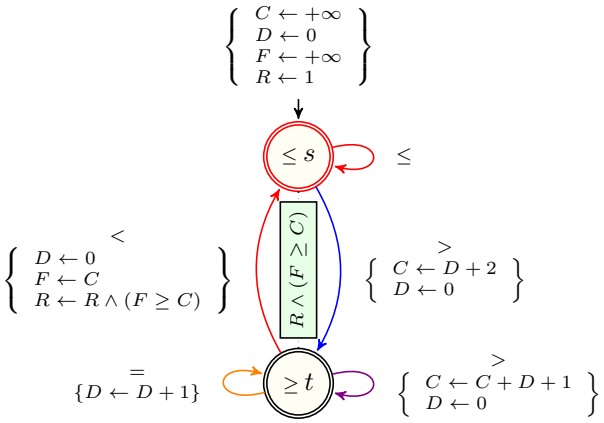
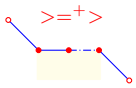


Figure 4.276: Automaton for the DECREASING_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the DECREASING_TERRACE pattern.

Constraint DECREASING_WIDTH_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 6, 6, 6, 4, 3, 3, 2, 2, 4, 4, 6, 3, 3, 1, 1\rangle)$

Figure 4.277 provides an example where the DECREASING_WIDTH_DECREASING_TERRACE $([7, 6, 6, 6, 4, 3, 3, 2, 2, 4, 4, 6, 3, 3, 1, 1])$ constraint holds.

Typical |VARIABLES| > 3
range(VARIABLES.var) > 2

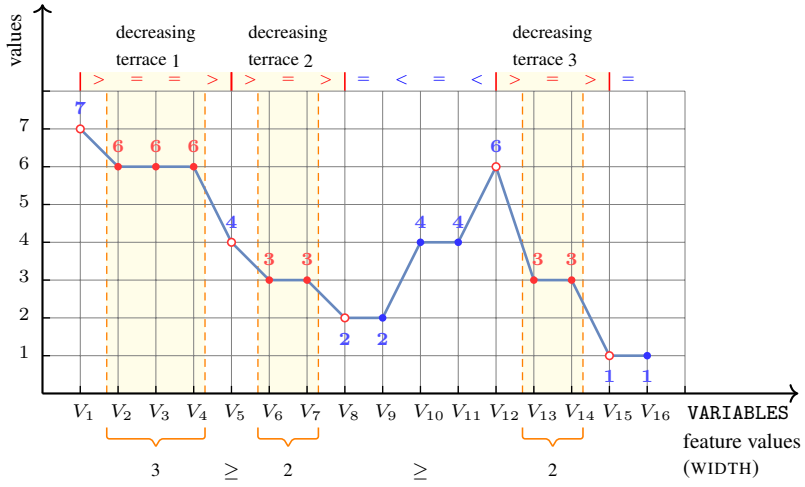


Figure 4.277: Illustrating the DECREASING_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.278 depicts the automaton associated with the constraint DECREASING_WIDTH_DECREASING_TERRACE.

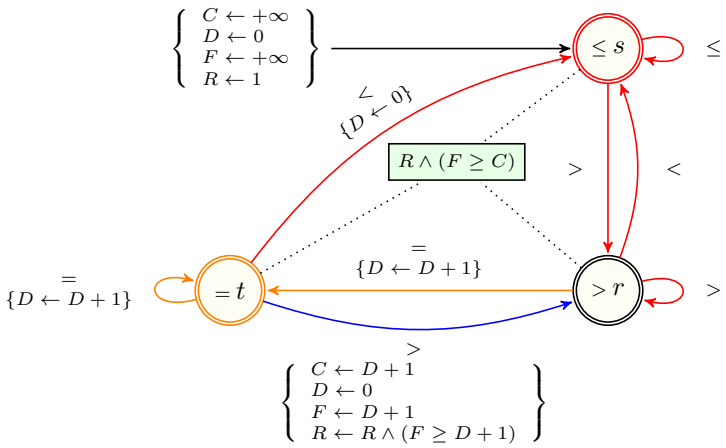
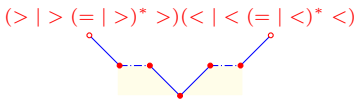


Figure 4.278: Automaton for the DECREASING_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint DECREASING_WIDTH_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5 \rangle)$

Figure 4.279 provides an example where the DECREASING_WIDTH_GORGE $([6, 2, 3, 4, 5, 6, 1, 1, 5, 4, 2, 3, 6, 1, 4, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

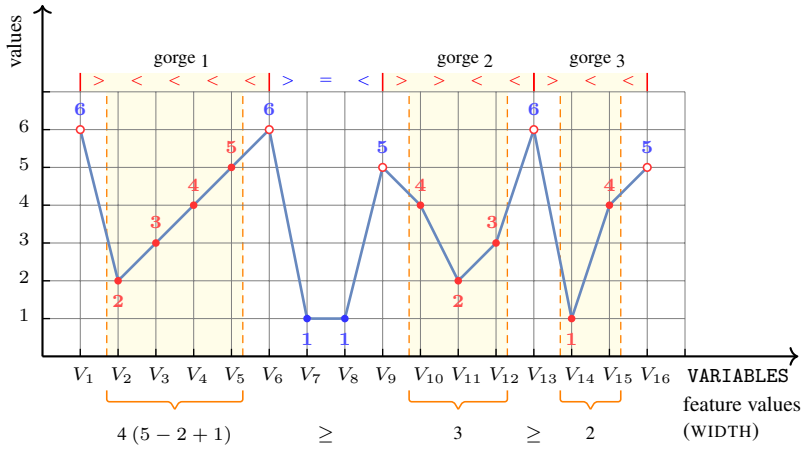


Figure 4.279: Illustrating the DECREASING_WIDTH_GORGE constraint of the **Example** slot

Automaton Figure 4.280 depicts the automaton associated with the constraint DECREASING_WIDTH_GORGE.

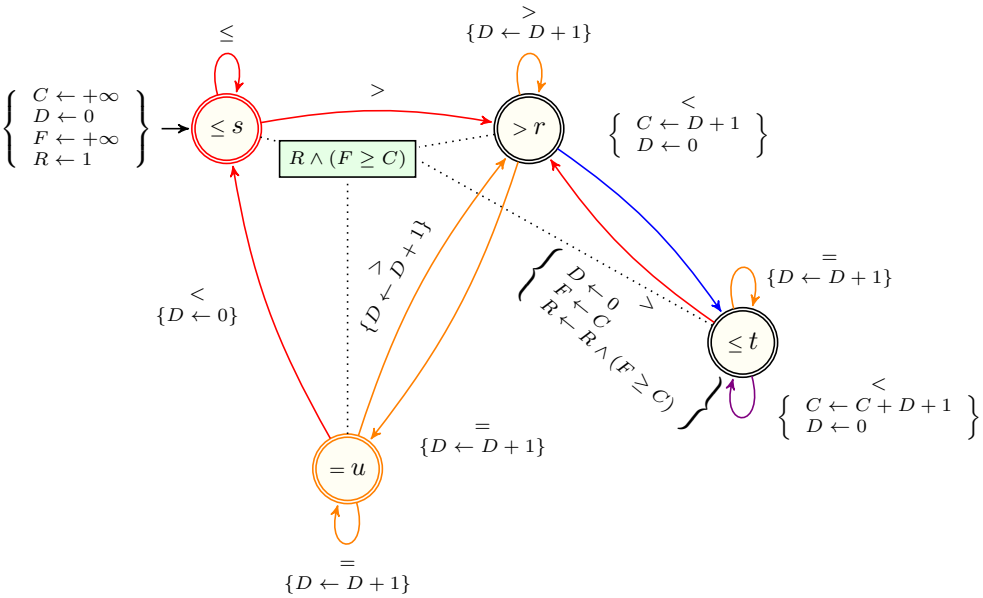
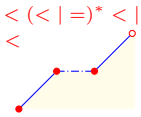


Figure 4.280: Automaton for the DECREASING_WIDTH_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint DECREASING_WIDTH_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1 \rangle)$

Figure 4.281 provides an example where the DECREASING_WIDTH_INCREASING_SEQUENCE $([3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

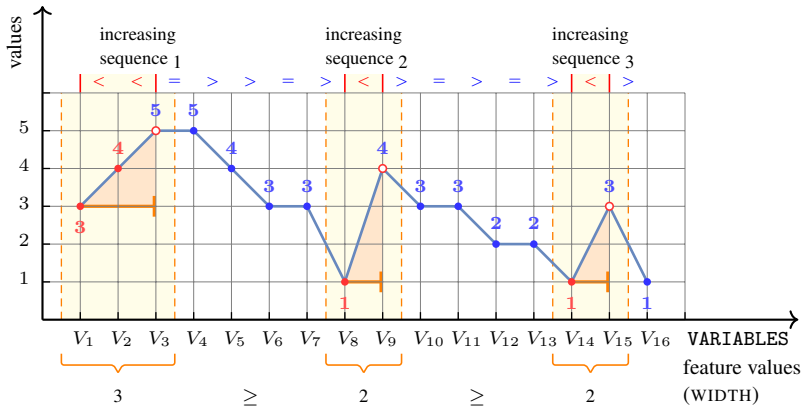


Figure 4.281: Illustrating the DECREASING_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.282 depicts the automaton associated with the constraint DECREASING_WIDTH_INCREASING_SEQUENCE.

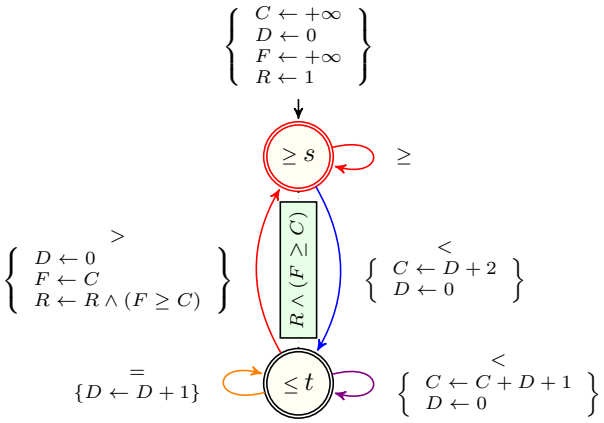
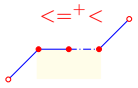


Figure 4.282: Automaton for the DECREASING_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INCREASING_TERRACE pattern.

Constraint DECREASING_WIDTH_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 2, 5, 5, 5, 6, 2, 3, 4, 4, 5, 3, 3, 1, 2, 2, 3\rangle)$

Figure 4.283 provides an example where the DECREASING_WIDTH_INCREASING_TERRACE $([2, 5, 5, 5, 6, 2, 3, 4, 4, 5, 3, 3, 1, 2, 2, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

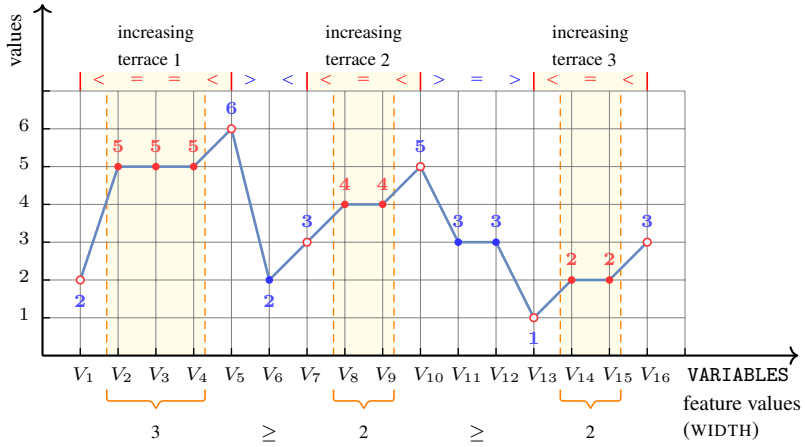


Figure 4.283: Illustrating the DECREASING_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.284 depicts the automaton associated with the constraint DECREASING_WIDTH_INCREASING_TERRACE.

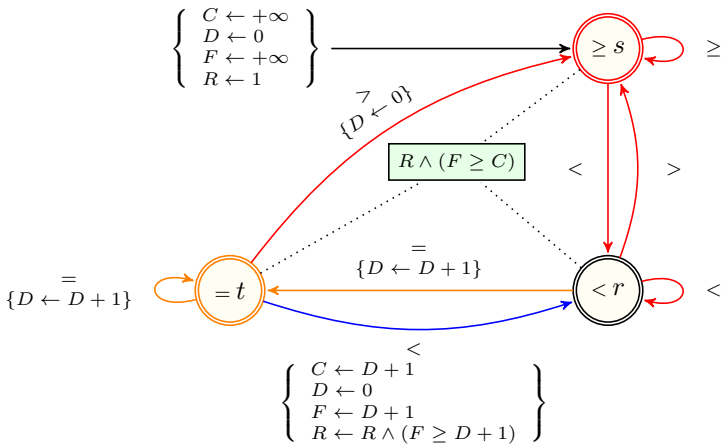
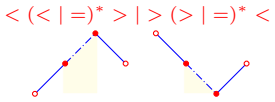


Figure 4.284: Automaton for the DECREASING_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

Constraint DECREASING_WIDTH_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 5, 5, 4, 3, 2, 3, 3, 4, 6, 3, 3, 1, 2, 1, 1 \rangle)$

Figure 4.285 provides an example where the DECREASING_WIDTH_INFLEXION $([6, 5, 5, 4, 3, 2, 3, 3, 4, 6, 3, 3, 1, 2, 1, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

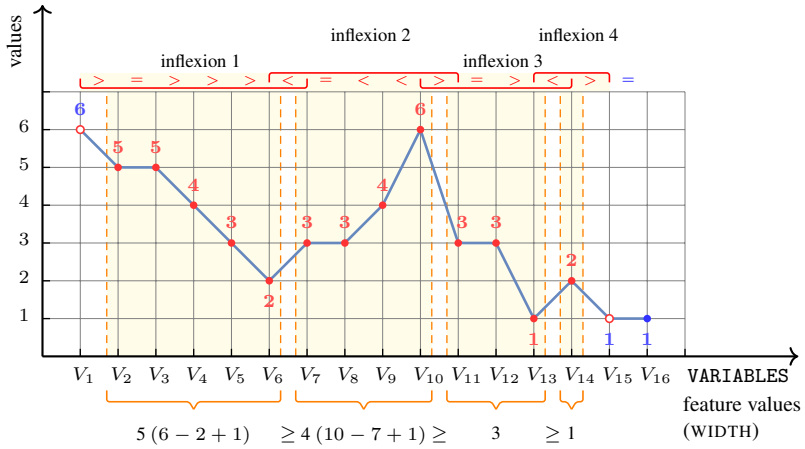


Figure 4.285: Illustrating the DECREASING_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.286 depicts the automaton associated with the constraint DECREASING_WIDTH_INFLEXION.

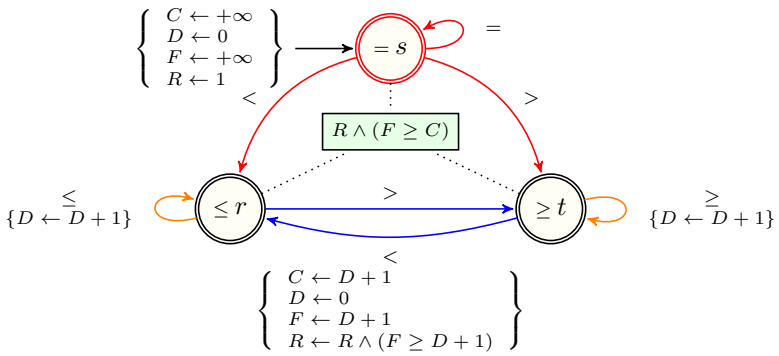
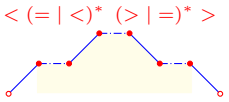


Figure 4.286: Automaton for the DECREASING_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

DECREASING_WIDTH_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint DECREASING_WIDTH_PEAK(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7 \rangle)$

Figure 4.287 provides an example where the DECREASING_WIDTH_PEAK $([1, 6, 6, 6, 2, 5, 4, 3, 2, 2, 3, 2, 1, 5, 5, 7])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

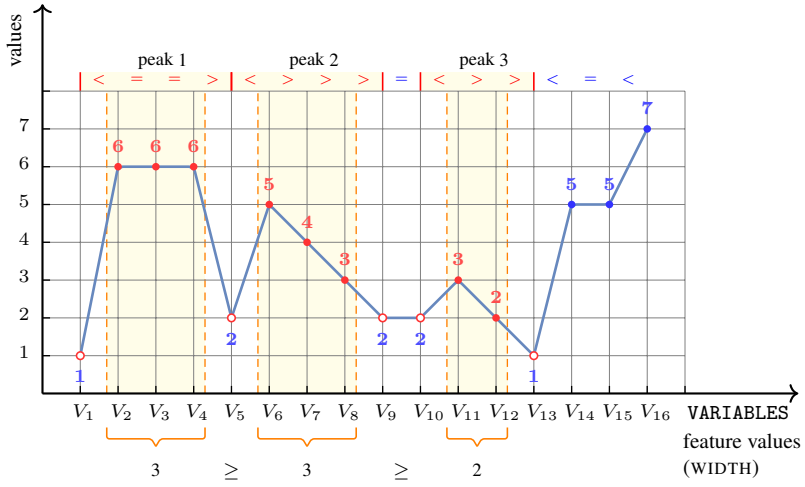


Figure 4.287: Illustrating the DECREASING_WIDTH_PEAK constraint of the **Example** slot

Automaton

Figure 4.288 depicts the automaton associated with the constraint DECREASING_WIDTH_PEAK.

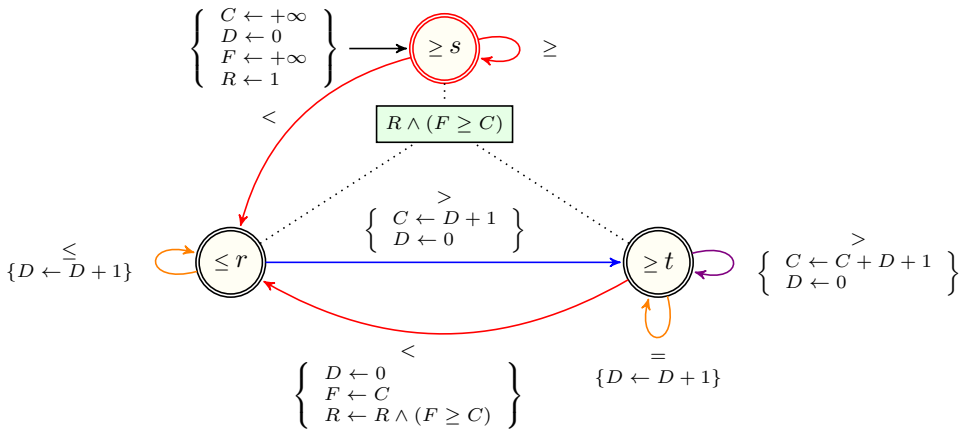


Figure 4.288: Automaton for the DECREASING_WIDTH_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint DECREASING_WIDTH_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j-i$.

Example $(\langle 2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3\rangle)$

Figure 4.289 provides an example where the DECREASING_WIDTH_PLAIN $([2,3,6,5,5,7,6,6,4,5,5,4,3,6,6,3])$ constraint holds.

Typical |VARIABLES| > 2
range(VARIABLES.var) > 1

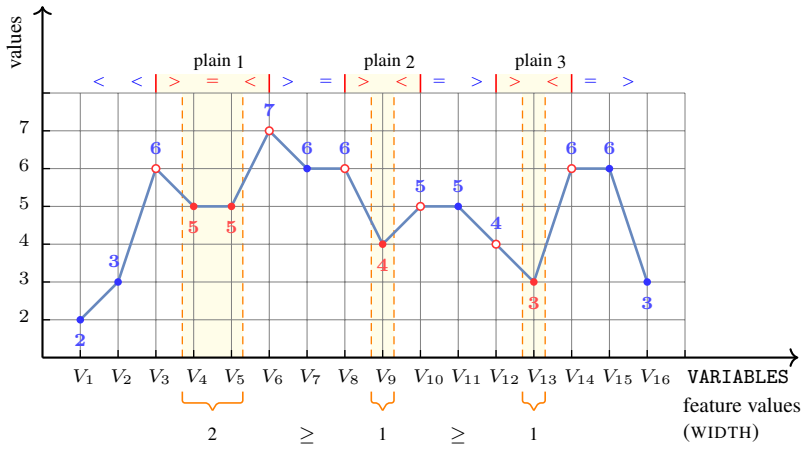


Figure 4.289: Illustrating the DECREASING_WIDTH_PLAIN constraint of the **Example** slot

Automaton Figure 4.290 depicts the automaton associated with the constraint DECREASING_WIDTH_PLAIN.

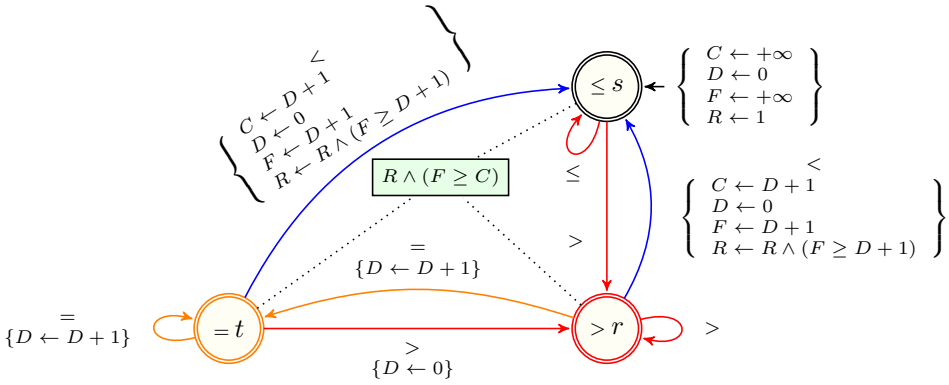


Figure 4.290: Automaton for the DECREASING_WIDTH_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint DECREASING_WIDTH_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 5, 2, 2, 5, 5, 4, 3, 3, 4, 2, 2, 1, 3, 2, 5, 7\rangle)$

Figure 4.291 provides an example where the DECREASING_WIDTH_PLATEAU $([5, 2, 2, 5, 5, 4, 3, 3, 4, 2, 2, 1, 3, 2, 5, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

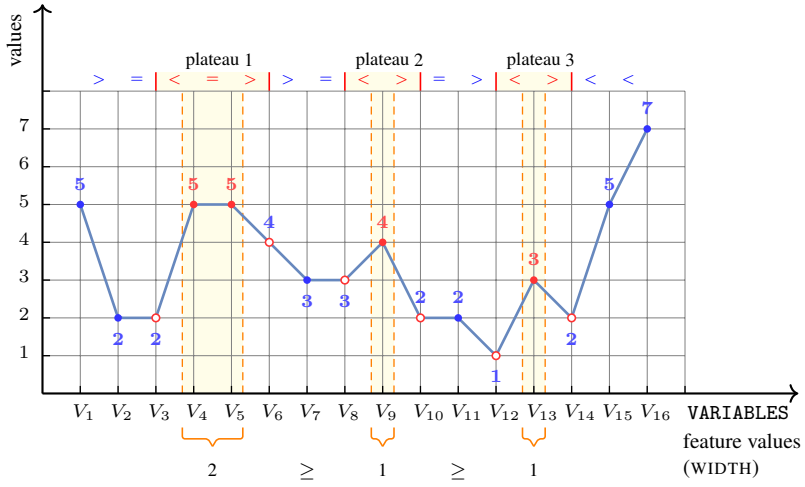


Figure 4.291: Illustrating the DECREASING_WIDTH_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.292 depicts the automaton associated with the constraint DECREASING_WIDTH_PLATEAU.

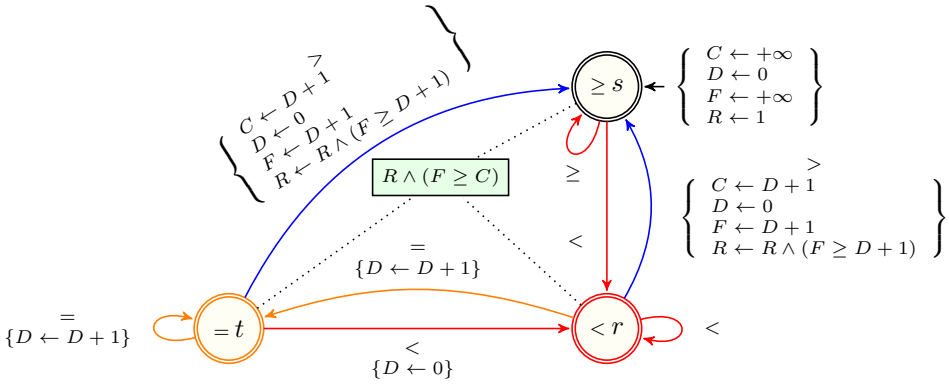
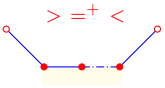


Figure 4.292: Automaton for the DECREASING_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint DECREASING_WIDTH_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 2, 7, 5, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 5\rangle)$

Figure 4.293 provides an example where the DECREASING_WIDTH_PROPER_PLAIN $([2, 7, 5, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

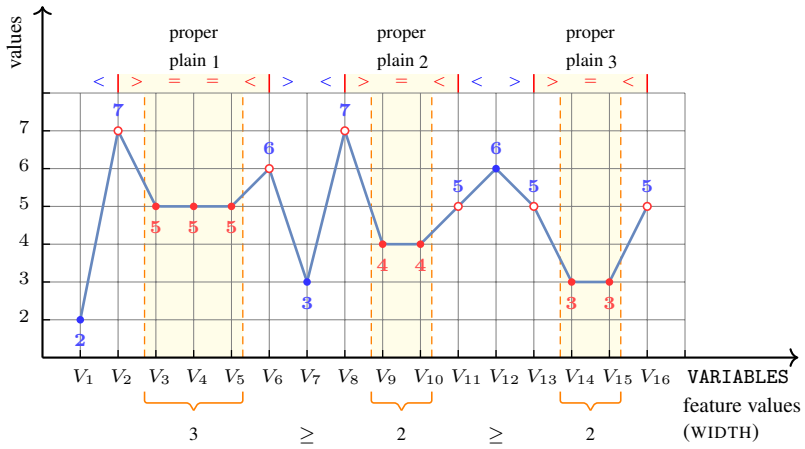


Figure 4.293: Illustrating the DECREASING_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.294 depicts the automaton associated with the constraint DECREASING_WIDTH_PROPER_PLAIN.

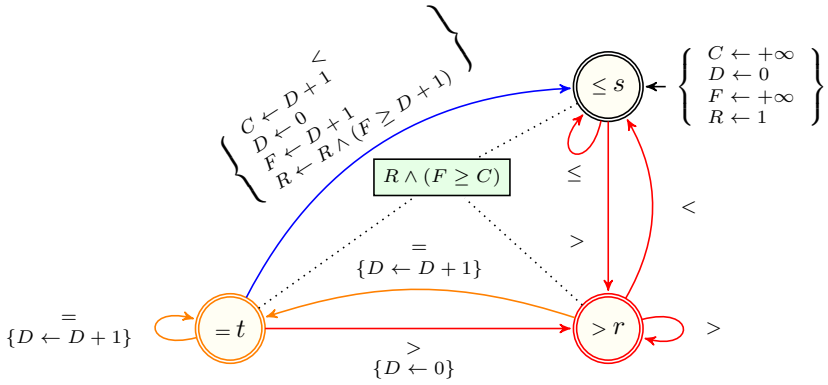
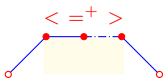


Figure 4.294: Automaton for the DECREASING_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLATEAU pattern.

Constraint DECREASING_WIDTH_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 3, 5, 5, 5, 3, 2, 3, 4, 4, 1, 5, 2, 3, 3, 1, 7\rangle)$

Figure 4.295 provides an example where the DECREASING_WIDTH_PROPER_PLATEAU $([3, 5, 5, 5, 3, 2, 3, 4, 4, 1, 5, 2, 3, 3, 1, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

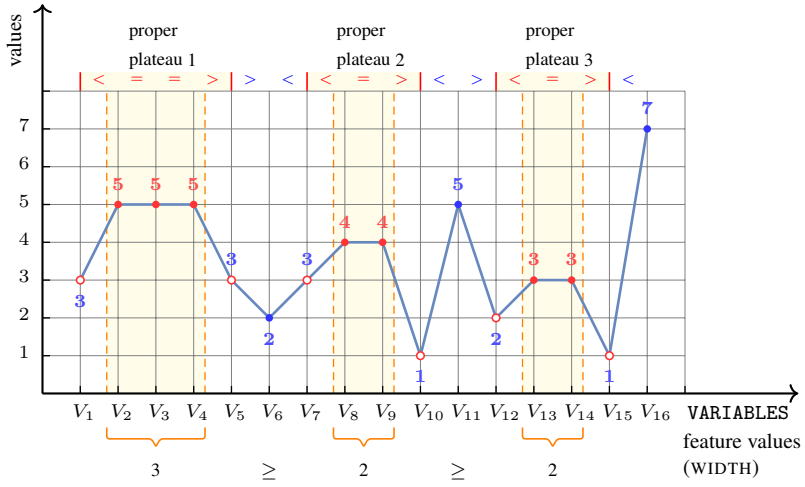


Figure 4.295: Illustrating the DECREASING_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.296 depicts the automaton associated with the constraint DECREASING_WIDTH_PROPER_PLATEAU.

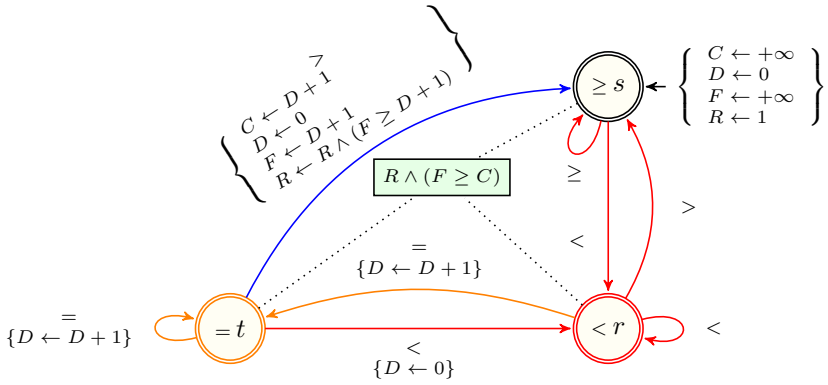
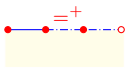


Figure 4.296: Automaton for the DECREASING_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint DECREASING_WIDTH_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 4, 3, 2, 5, 5, 5, 3, 2, 4, 4, 6, 5, 3, 2, 2, 6\rangle)$

Figure 4.297 provides an example where the DECREASING_WIDTH_STEADY_SEQUENCE $([4, 3, 2, 5, 5, 5, 3, 2, 4, 4, 6, 5, 3, 2, 2, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

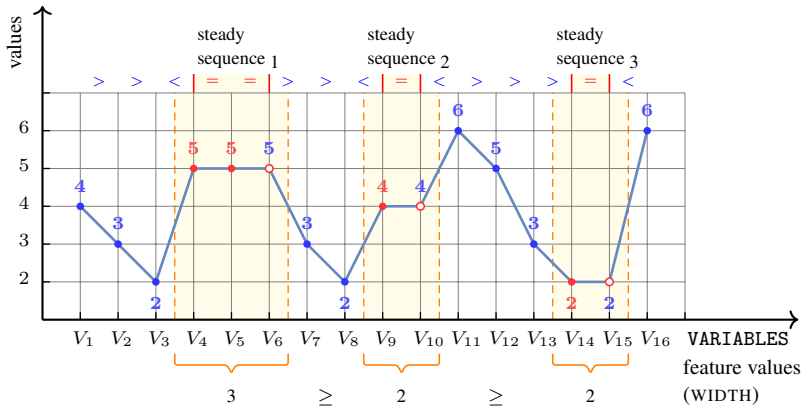


Figure 4.297: Illustrating the DECREASING_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.298 depicts the automaton associated with the constraint DECREASING_WIDTH_STEADY_SEQUENCE.

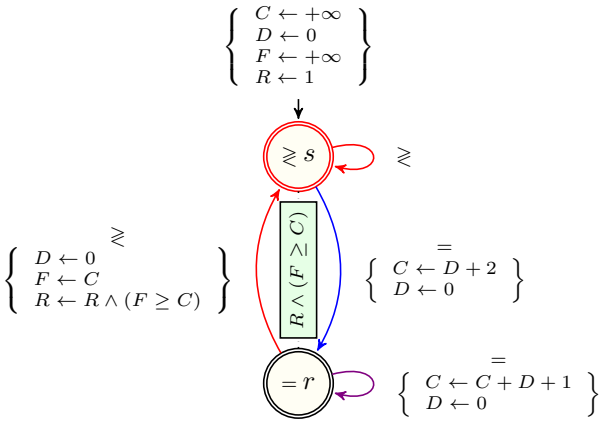


Figure 4.298: Automaton for the DECREASING_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^{+}$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 1, 1, 2, 2, 3, 2, 1, 5, 6, 6, 5, 4, 4, 6, 6, 1\rangle)$

Figure 4.299 provides an example where the DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE $([1, 1, 2, 2, 3, 2, 1, 5, 6, 6, 5, 4, 4, 6, 6, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

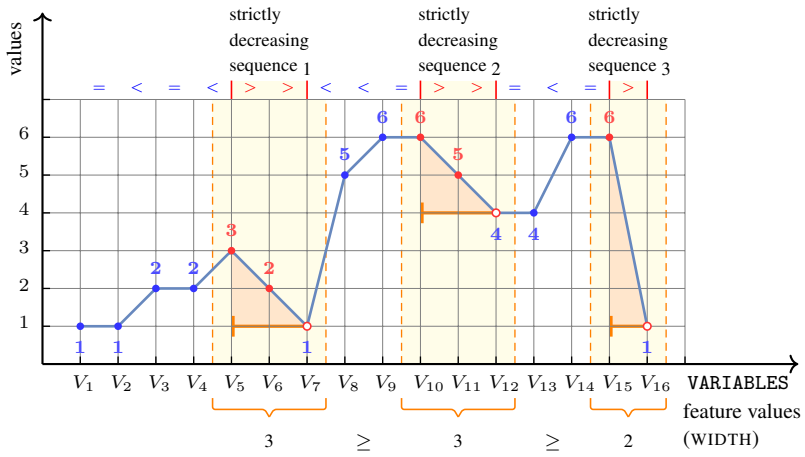


Figure 4.299: Illustrating the DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.300 depicts the automaton associated with the constraint DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE.

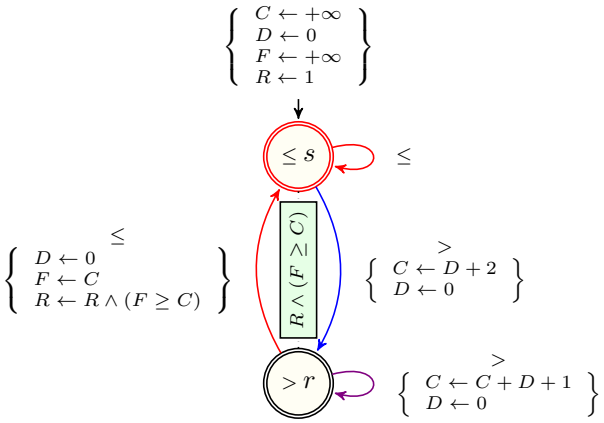


Figure 4.300: Automaton for the DECREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1 \rangle)$

Figure 4.301 provides an example where the DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE $([3, 4, 5, 5, 4, 3, 3, 1, 4, 3, 3, 2, 2, 1, 3, 1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

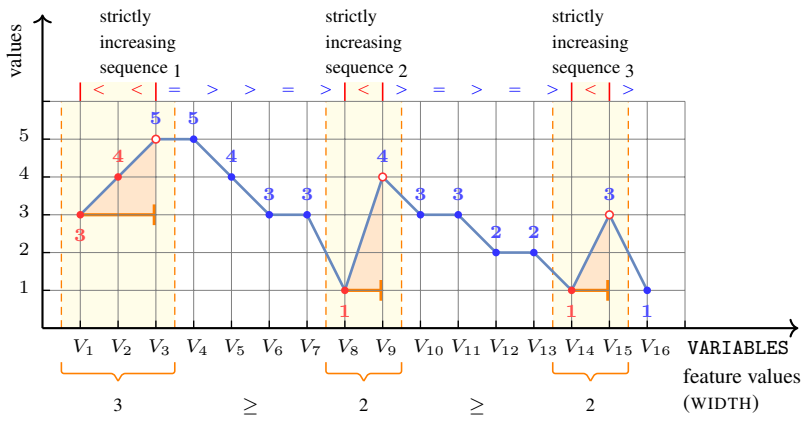


Figure 4.301: Illustrating the DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.302 depicts the automaton associated with the constraint DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE.

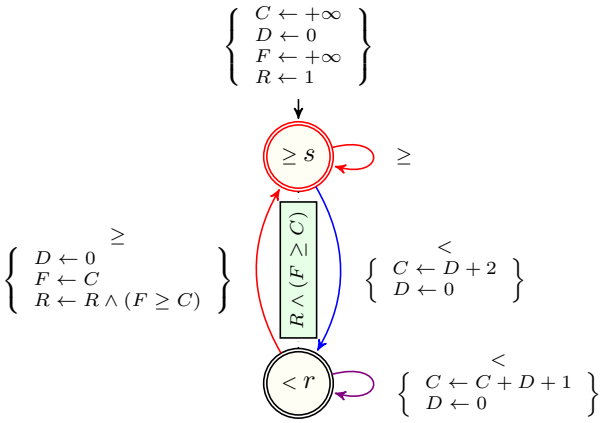


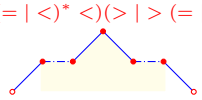
Figure 4.302: Automaton for the DECREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the SUMMIT pattern.

Constraint DECREASING_WIDTH_SUMMIT(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2 \rangle)$

Figure 4.303 provides an example where the DECREASING_WIDTH_SUMMIT $([1, 4, 4, 5, 4, 2, 6, 6, 2, 3, 5, 4, 1, 2, 4, 2])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

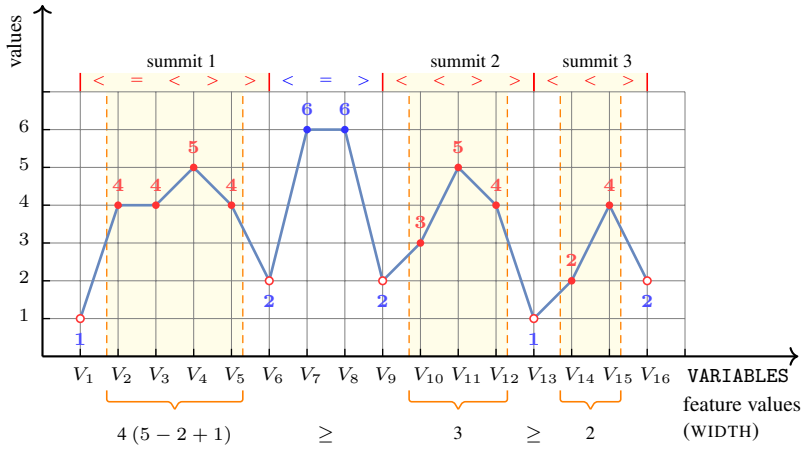


Figure 4.303: Illustrating the DECREASING_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figure 4.304 depicts the automaton associated with the constraint DECREASING_WIDTH_SUMMIT.

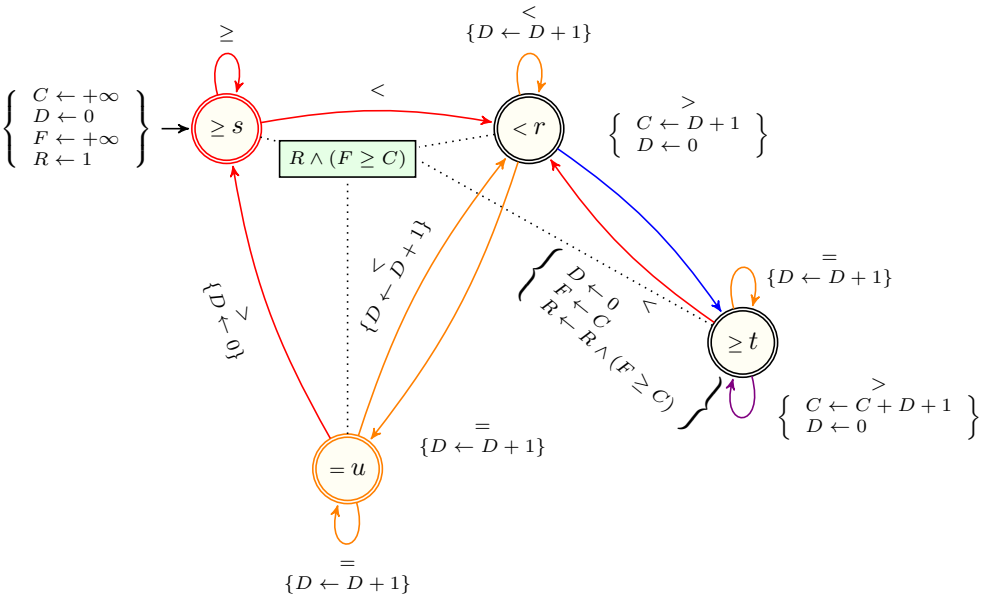
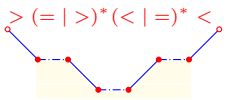


Figure 4.304: Automaton for the DECREASING_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

DECREASING_WIDTH_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint DECREASING_WIDTH_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> \ (= \mid >)^* (< \mid =)^* \ <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 3, 3, 7, 6, 5, 4, 6, 6, 5, 4, 3, 6, 2, 2, 7 \rangle)$

Figure 4.305 provides an example where the DECREASING_WIDTH_VALLEY $([1, 3, 3, 7, 6, 5, 4, 6, 6, 5, 4, 3, 6, 2, 2, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

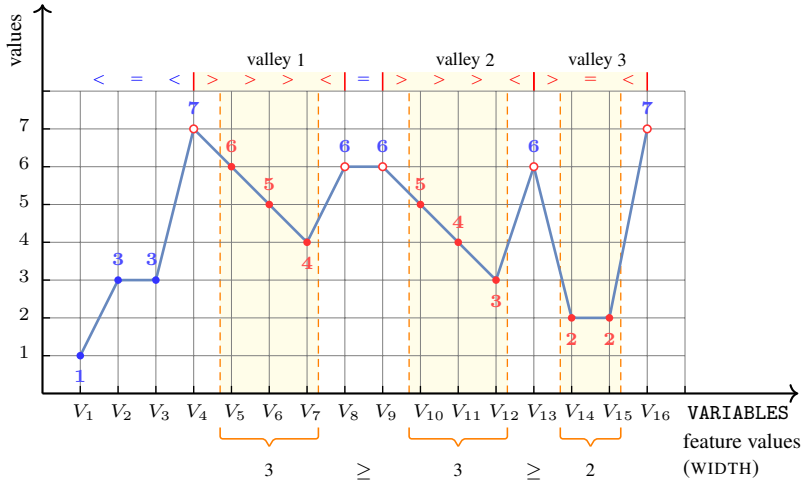


Figure 4.305: Illustrating the DECREASING_WIDTH_VALLEY constraint of the **Example** slot

Automaton

Figure 4.306 depicts the automaton associated with the constraint DECREASING_WIDTH_VALLEY.

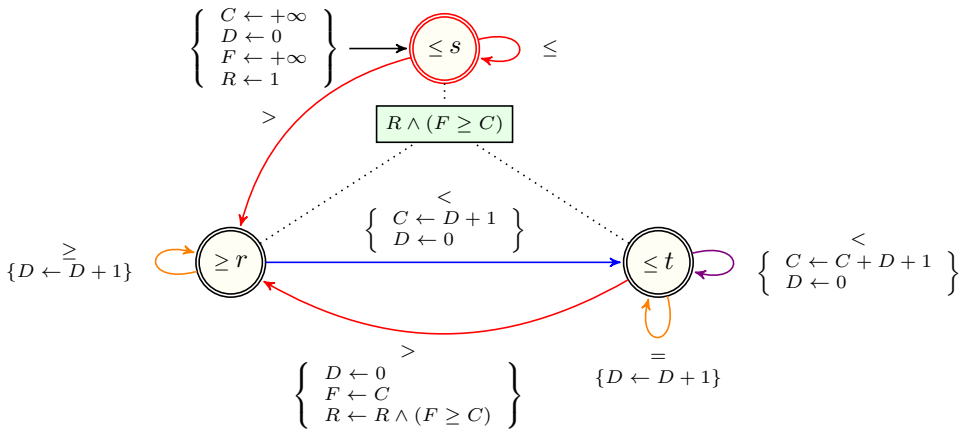
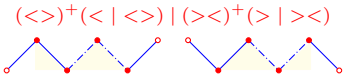


Figure 4.306: Automaton for the DECREASING_WIDTH_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

DECREASING_WIDTH_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint DECREASING_WIDTH_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are decreasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^{+}(< \mid <>) \mid (><)^{+}(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 5, 7, 4, 5, 1, 1, 6, 3, 6, 4, 3, 5, 2, 7, 7 \rangle)$

Figure 4.307 provides an example where the DECREASING_WIDTH_ZIGZAG $([6, 5, 7, 4, 5, 1, 1, 6, 3, 6, 4, 3, 5, 2, 7, 7])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

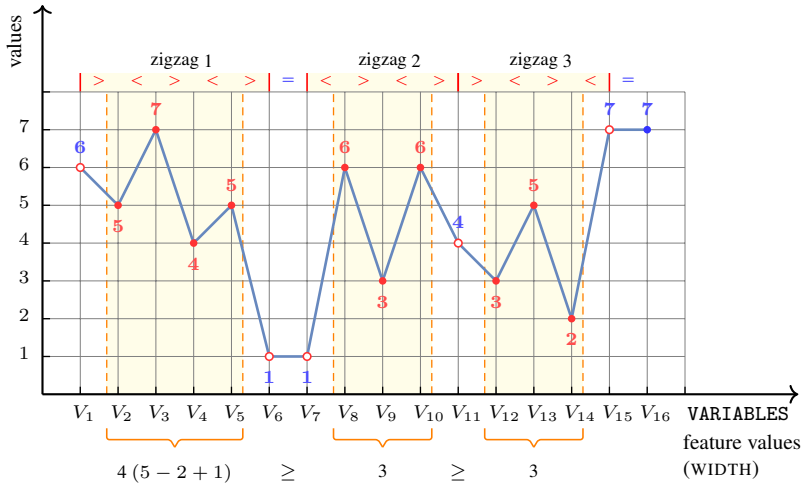


Figure 4.307: Illustrating the DECREASING_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.308 depicts the automaton associated with the constraint DECREASING_WIDTH_ZIGZAG.

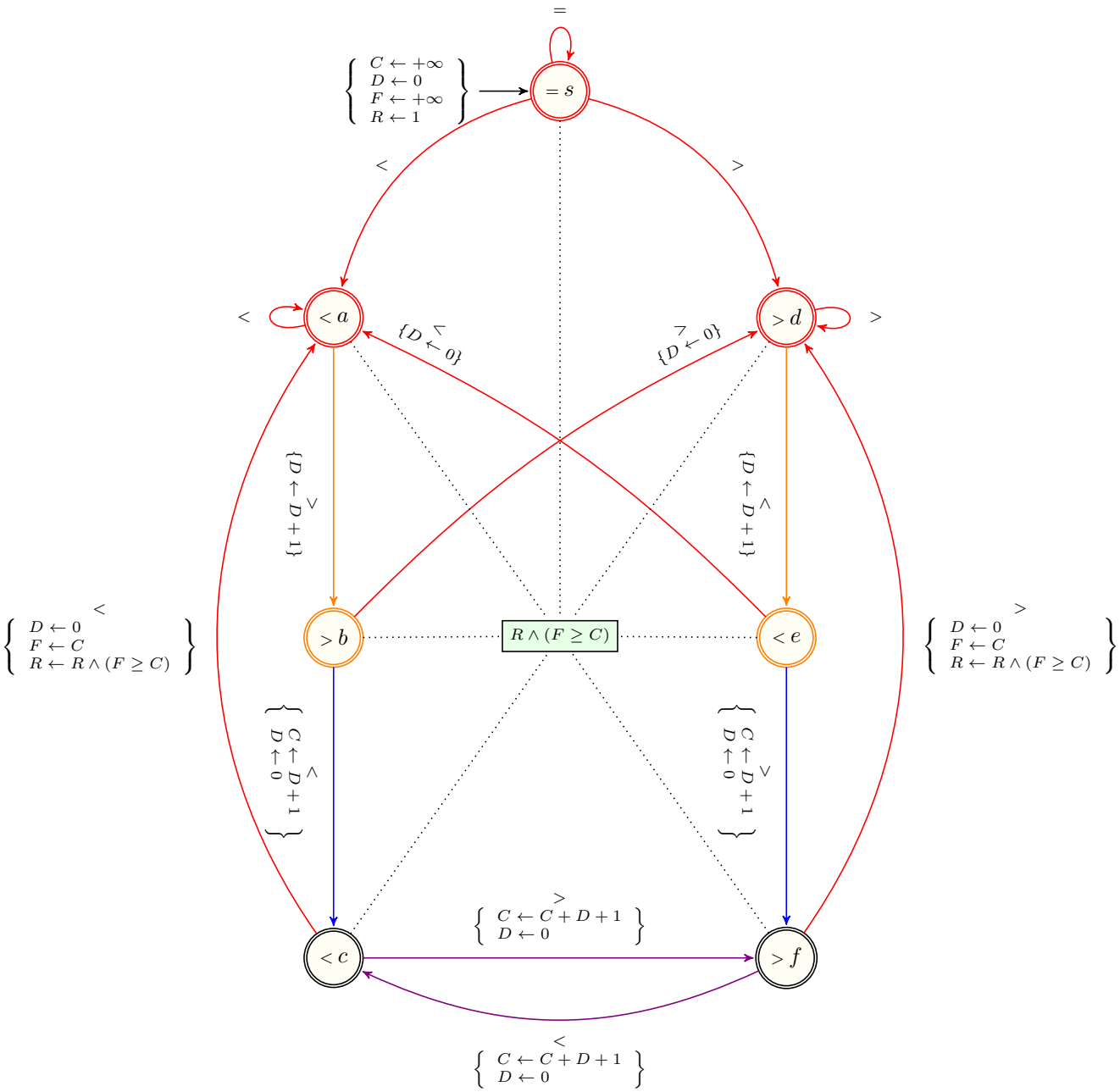
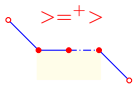


Figure 4.308: Automaton for the DECREASING_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

FEATURE PATTERN

HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint HEIGHT_DECREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv + 1 ∨ DEFAULT > maxv − 1

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_TERRACE.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.309 provides an example where the HEIGHT_DECREASING_TERRACE $([6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3],[0,0,0,0,0,0,0,2,0,0,0,0,0,4,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

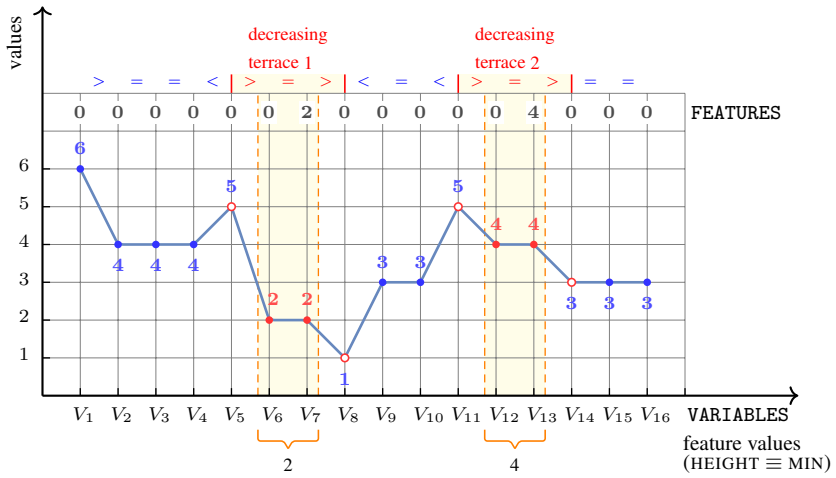


Figure 4.309: Illustrating the HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

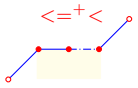
Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint HEIGHT_INCREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv + 1 ∨ DEFAULT > maxv − 1

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_TERRACE is identified then FEATURES[i] is the default value DEFAULT; otherwise FEATURES[i] gives the feature value of the corresponding occurrence of INCREASING_TERRACE.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.310 provides an example where the HEIGHT_INCREASING_TERRACE $([1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4],[0,0,0,0,0,0,5,0,0,0,0,0,0,3,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

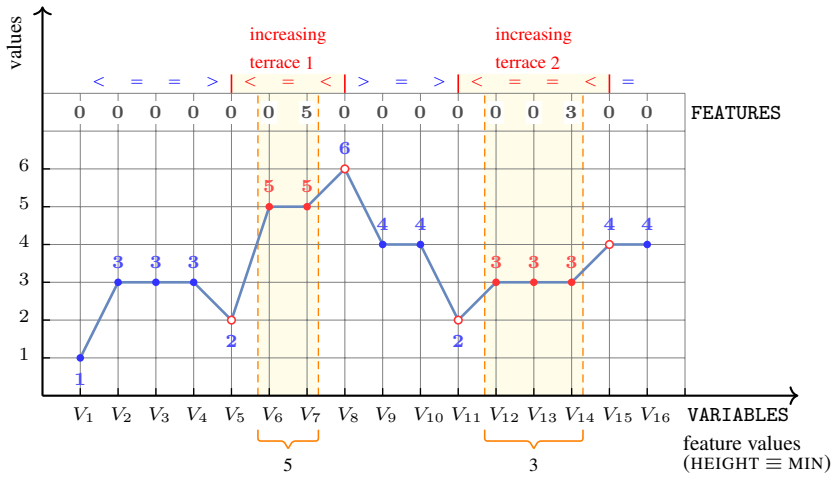


Figure 4.310: Illustrating the HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Arg. properties

Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint HEIGHT_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLAIN.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.311 provides an example where the HEIGHT_PLAIN $([2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3],[0,0,0,5,0,0,0,4,0,0,0,0,3,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

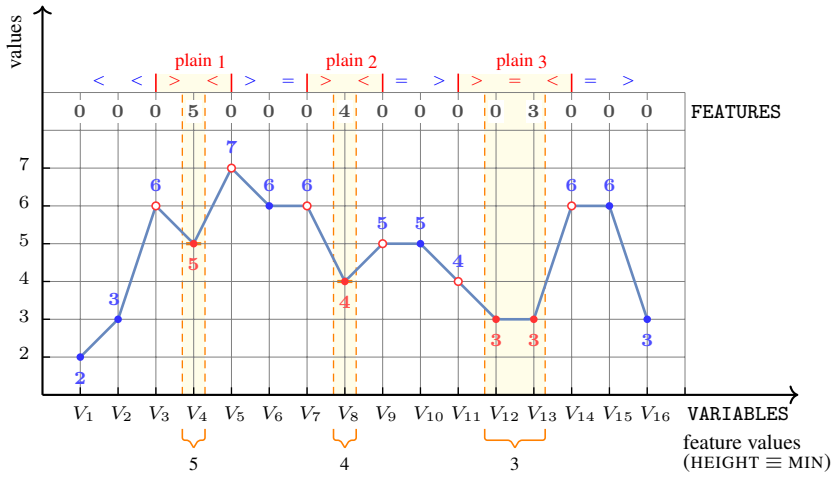


Figure 4.311: Illustrating the HEIGHT_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint HEIGHT_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLATEAU.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.312 provides an example where the HEIGHT_PLATEAU $([7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5],[0,0,0,3,0,0,0,4,0,0,0,0,5,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

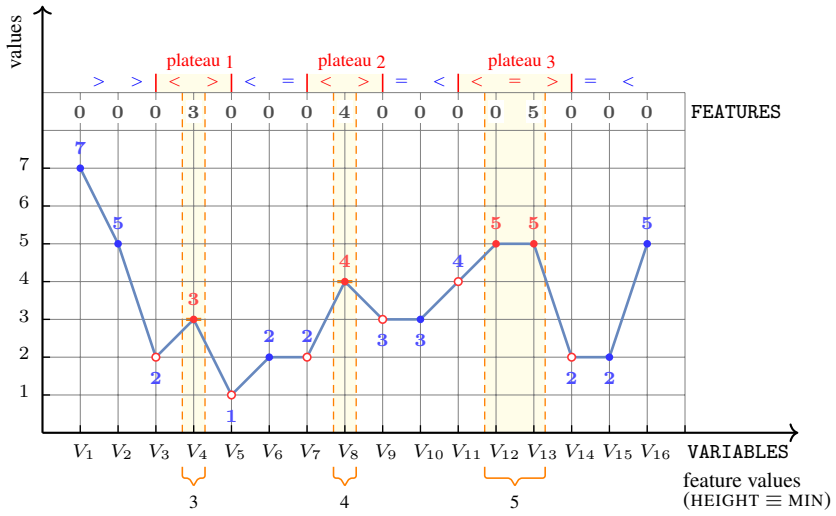


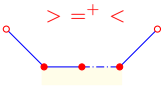
Figure 4.312: Illustrating the HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint HEIGHT_PROPER_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLAIN.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.313 provides an example where the HEIGHT_PROPER_PLAIN $([2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5],[0,0,0,5,0,0,0,0,4,0,0,0,0,0,3,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

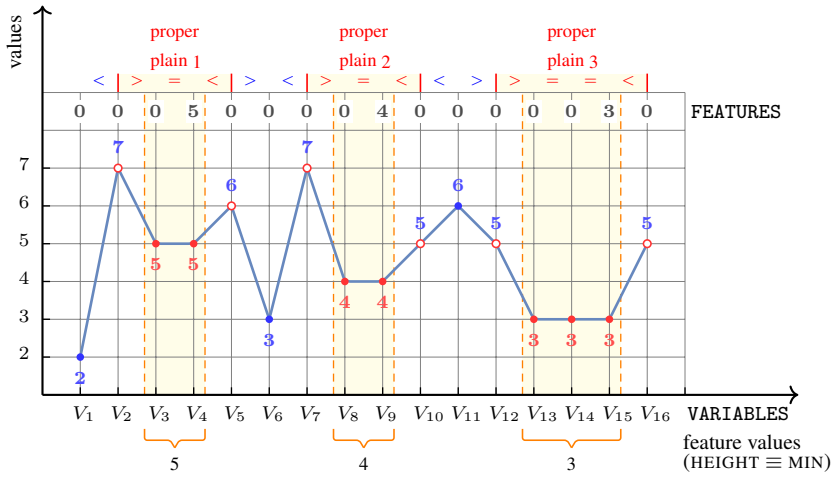


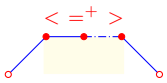
Figure 4.313: Illustrating the HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint HEIGHT_PROPER_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLATEAU.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.314 provides an example where the HEIGHT_PROPER_PLATEAU $([7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3],[0,0,0,3,0,0,0,0,4,0,0,0,0,0,5,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

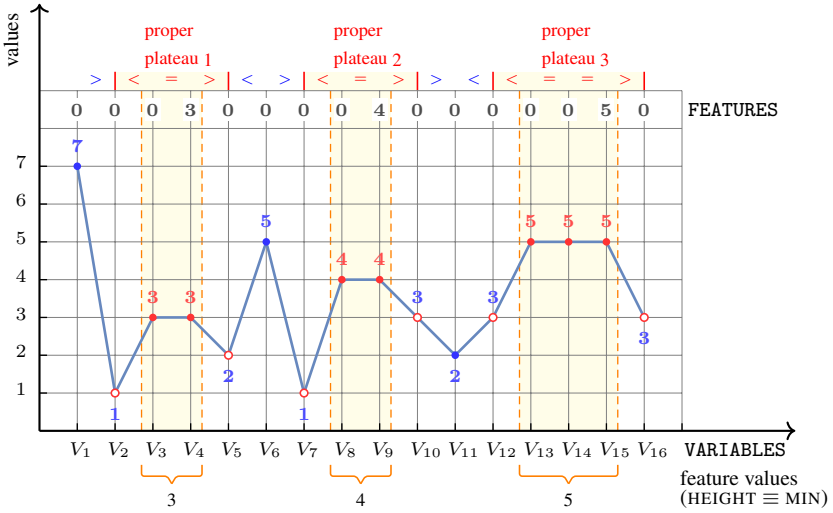


Figure 4.314: Illustrating the HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON 

Origin Based on the STEADY pattern.

Constraint HEIGHT_STEADY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.315 provides an example where the HEIGHT_STEADY $([1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6], [1, 0, 0, 3, 0, 5, 5, 0, 0, 5, 5, 0, 0, 0, 6, 0], 0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

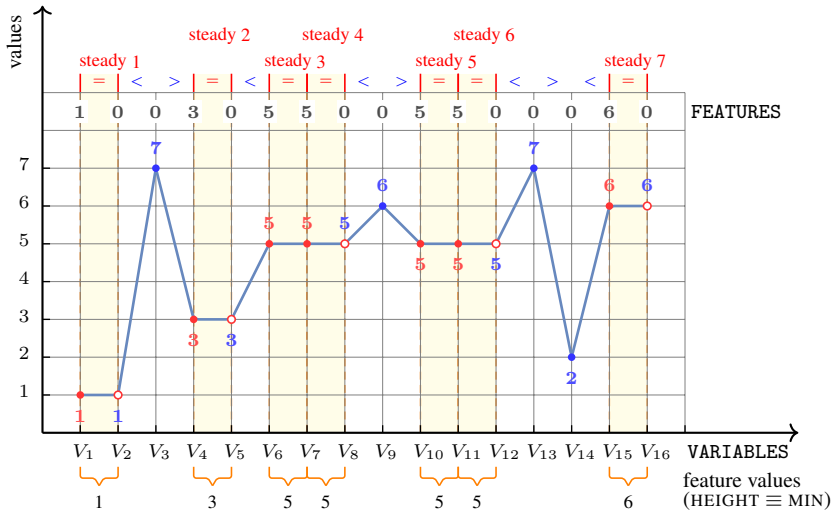


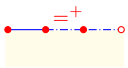
Figure 4.315: Illustrating the HEIGHT_STEADY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint HEIGHT_STEADY_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY_SEQUENCE.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example Figure 4.316 provides an example where the HEIGHT_STEADY_SEQUENCE $([3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1],[0,1,0,0,5,0,0,0,2,0,4,0,0,0,1,0],0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

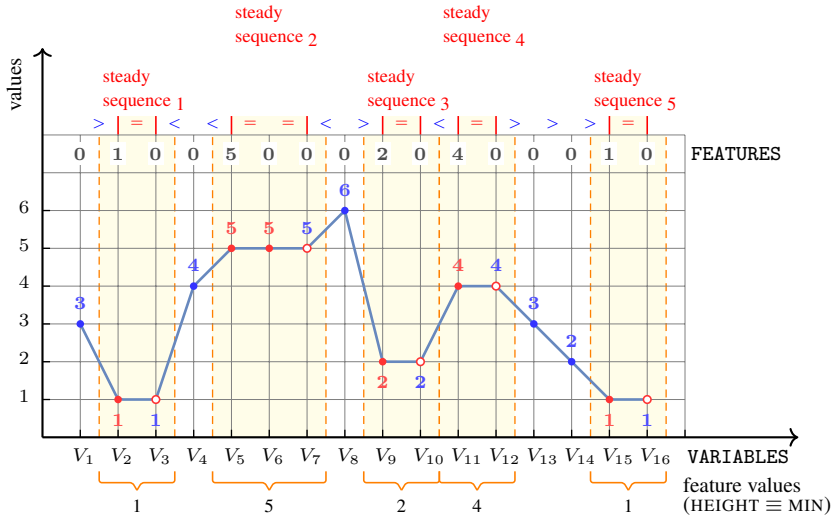


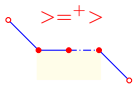
Figure 4.316: Illustrating the HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the DECREASING_TERRACE pattern.

Constraint INCREASING_HEIGHT_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 5,2,2,1,5,4,3,3,2,4,4,6,5,5,5,4\rangle)$

Figure 4.317 provides an example where the INCREASING_HEIGHT_DECREASING_TERRACE $([5,2,2,1,5,4,3,3,2,4,4,6,5,5,5,4])$ constraint holds.

Typical |VARIABLES| > 3
range(VARIABLES.var) > 2

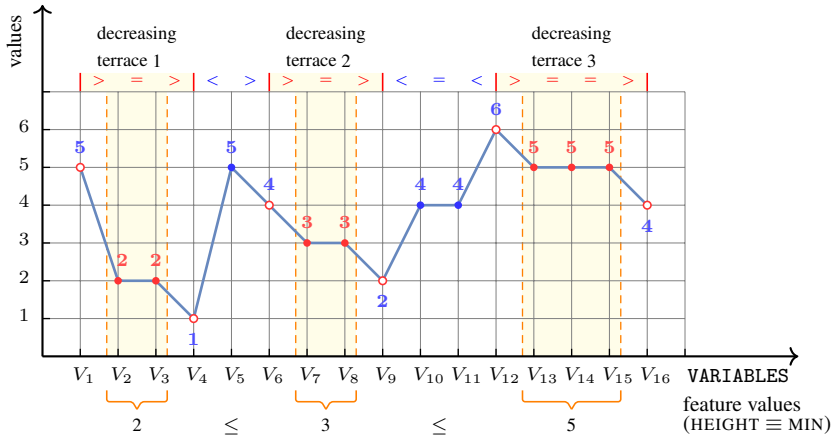


Figure 4.317: Illustrating the INCREASING_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.318 depicts the automaton associated with the constraint INCREASING_HEIGHT_DECREASING_TERRACE.

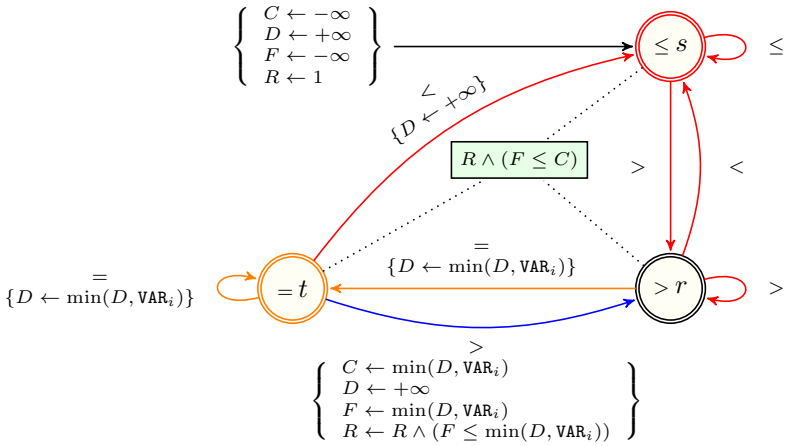
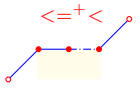


Figure 4.318: Automaton for the INCREASING_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INCREASING_TERRACE pattern.

Constraint INCREASING_HEIGHT_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 1,2,2,4,5,5,6,6,4,4,2,5,5,5,7,7\rangle)$

Figure 4.319 provides an example where the INCREASING_HEIGHT_INCREASING_TERRACE $([1,2,2,4,5,5,6,6,4,4,2,5,5,5,7,7])$ constraint holds.

Typical |VARIABLES| > 3
range(VARIABLES.var) > 2

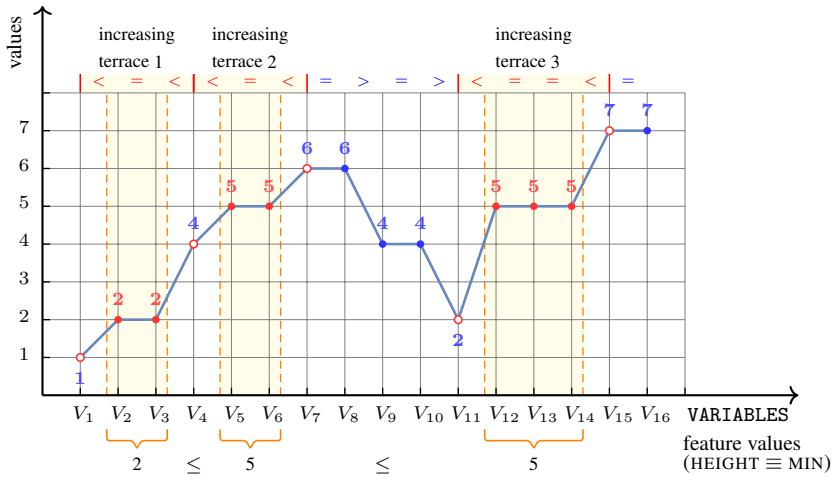


Figure 4.319: Illustrating the INCREASING_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.320 depicts the automaton associated with the constraint INCREASING_HEIGHT_INCREASING_TERRACE.

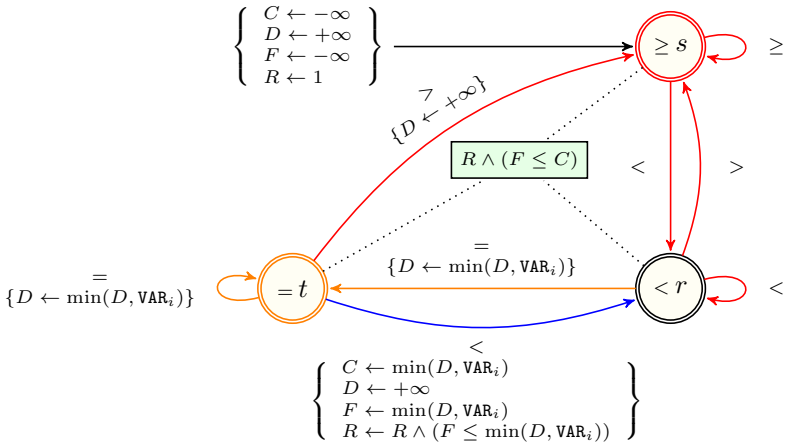


Figure 4.320: Automaton for the INCREASING_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint INCREASING_HEIGHT_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2\rangle)$

Figure 4.321 provides an example where the INCREASING_HEIGHT_PLAIN $([3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

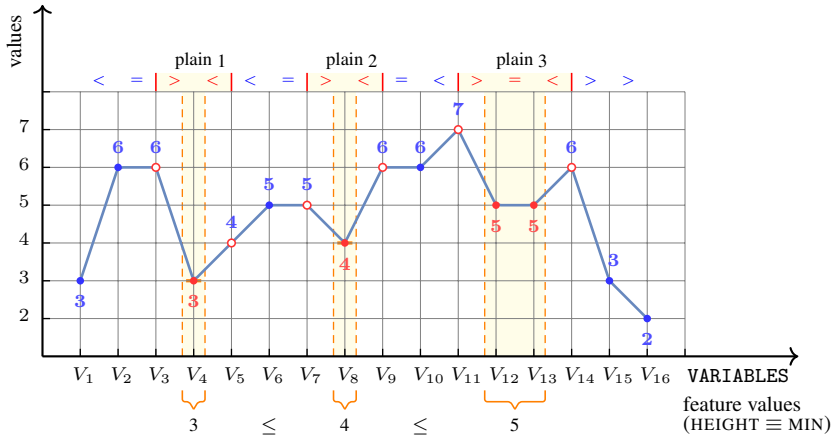


Figure 4.321: Illustrating the INCREASING_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figure 4.322 depicts the automaton associated with the constraint INCREASING_HEIGHT_PLAIN.

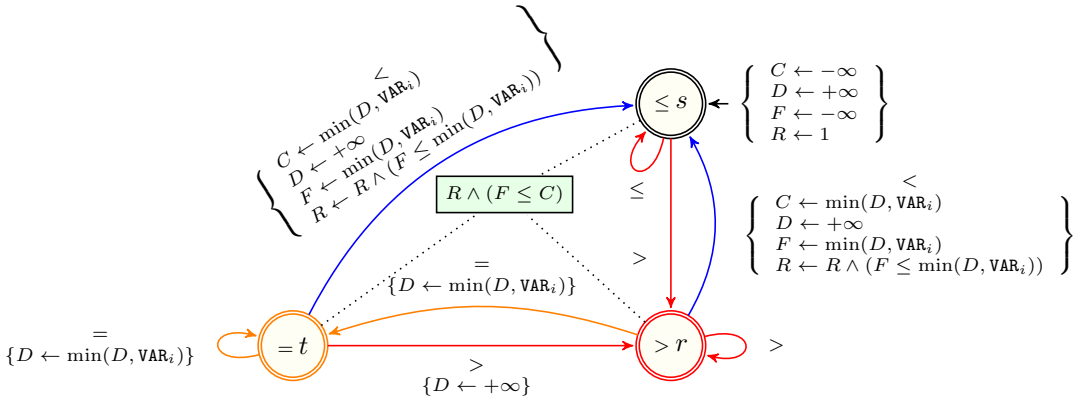


Figure 4.322: Automaton for the INCREASING_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint INCREASING_HEIGHT_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle)$

Figure 4.323 provides an example where the INCREASING_HEIGHT_PLATEAU $([7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

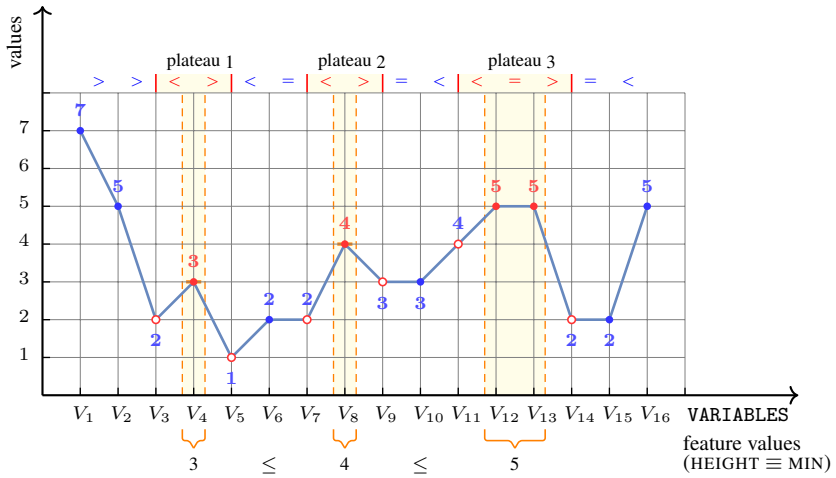


Figure 4.323: Illustrating the INCREASING_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figure 4.324 depicts the automaton associated with the constraint INCREASING_HEIGHT_PLATEAU.

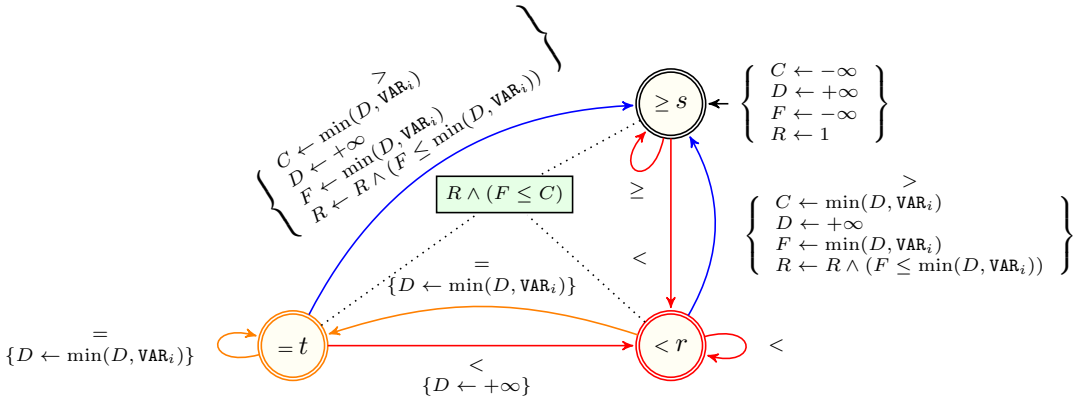
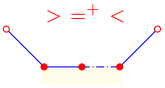


Figure 4.324: Automaton for the INCREASING_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_PROPER_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLAIN pattern.

Constraint INCREASING_HEIGHT_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2\rangle)$

Figure 4.325 provides an example where the INCREASING_HEIGHT_PROPER_PLAIN $([5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

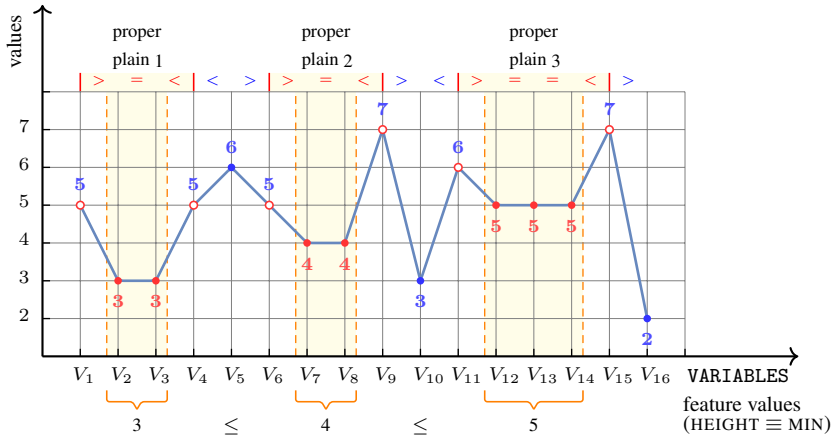


Figure 4.325: Illustrating the INCREASING_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.326 depicts the automaton associated with the constraint INCREASING_HEIGHT_PROPER_PLAIN.

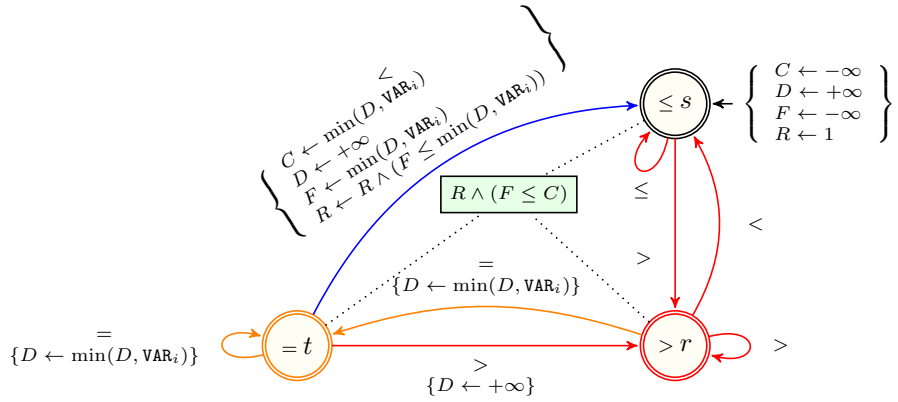
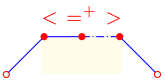


Figure 4.326: Automaton for the INCREASING_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLATEAU pattern.

Constraint INCREASING_HEIGHT_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3\rangle)$

Figure 4.327 provides an example where the INCREASING_HEIGHT_PROPER_PLATEAU $([7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

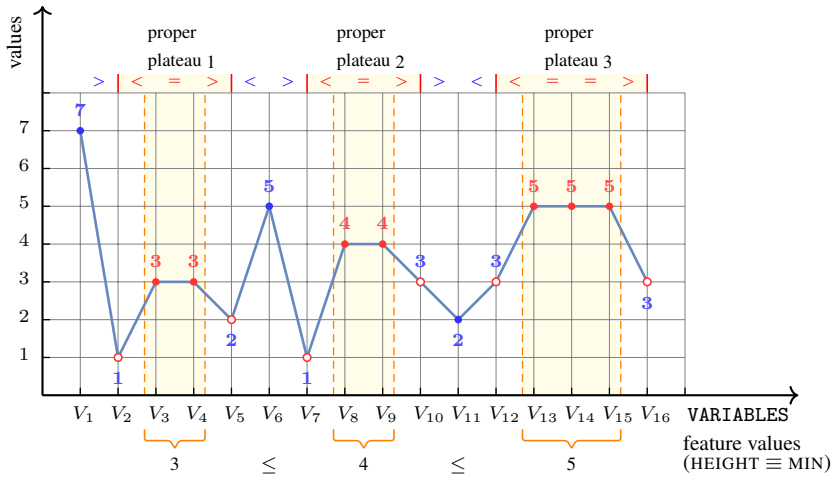


Figure 4.327: Illustrating the INCREASING_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton Figure 4.328 depicts the automaton associated with the constraint INCREASING_HEIGHT_PROPER_PLATEAU.

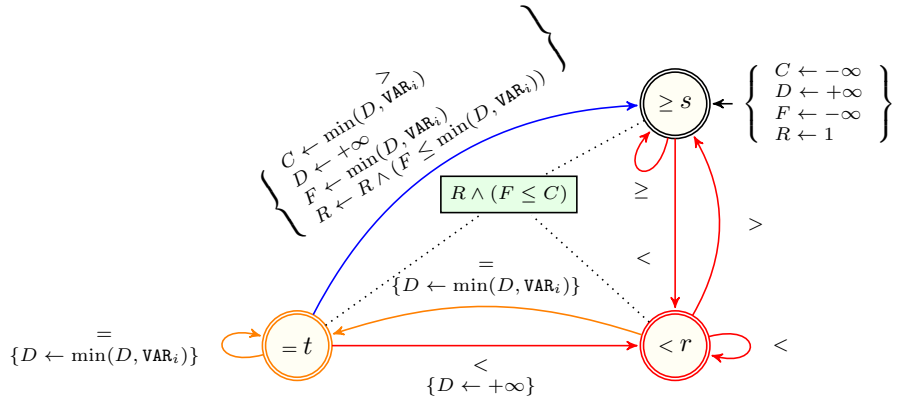


Figure 4.328: Automaton for the INCREASING_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_STEADY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STEADY pattern.

Constraint INCREASING_HEIGHT_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6\rangle)$

Figure 4.329 provides an example where the INCREASING_HEIGHT_STEADY $([1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

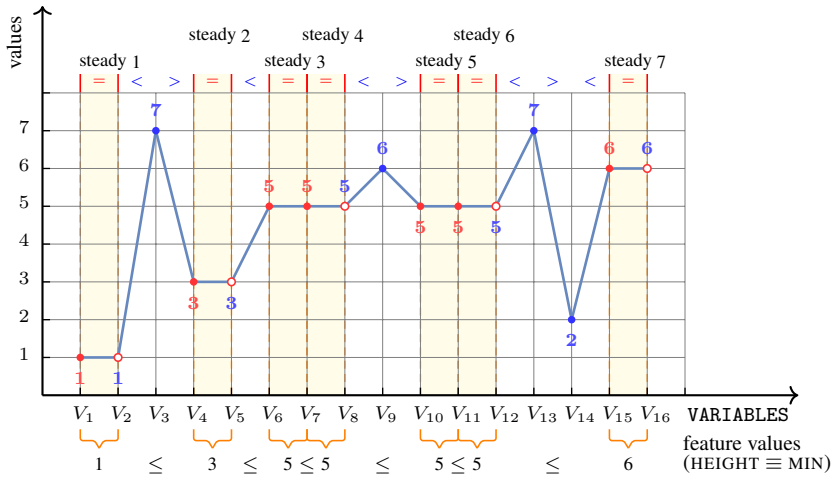


Figure 4.329: Illustrating the INCREASING_HEIGHT_STEADY constraint of the **Example** slot

Automaton Figure 4.330 depicts the automaton associated with the constraint INCREASING_HEIGHT_STEADY.

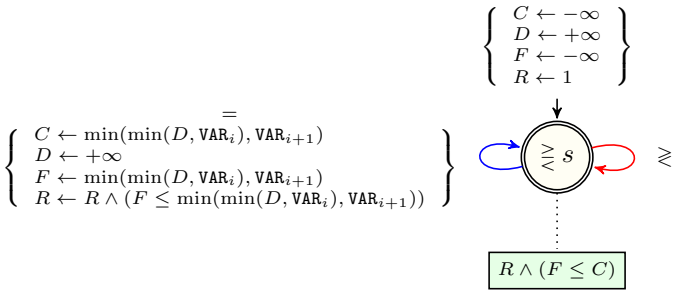
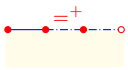


Figure 4.330: Automaton for the INCREASING_HEIGHT_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

INCREASING_HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint INCREASING_HEIGHT_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(\langle 6,2,2,3,5,6,4,4,2,3,5,5,5,2,3,4\rangle)$

Figure 4.331 provides an example where the INCREASING_HEIGHT_STEADY_SEQUENCE $([6,2,2,3,5,6,4,4,2,3,5,5,5,2,3,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

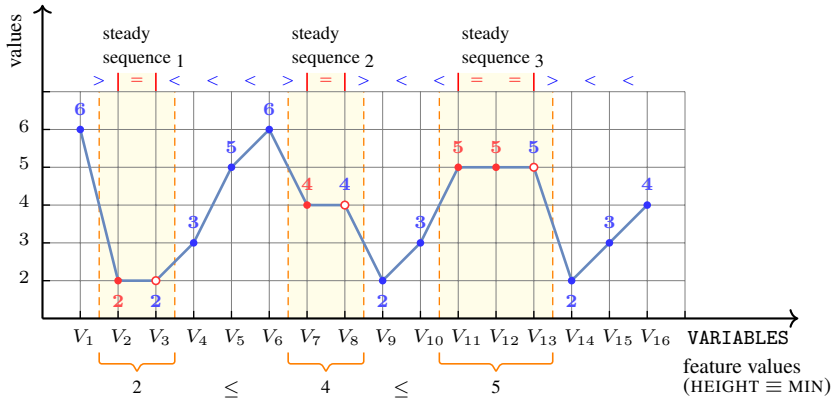


Figure 4.331: Illustrating the INCREASING_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.332 depicts the automaton associated with the constraint INCREASING_HEIGHT_STEADY_SEQUENCE.

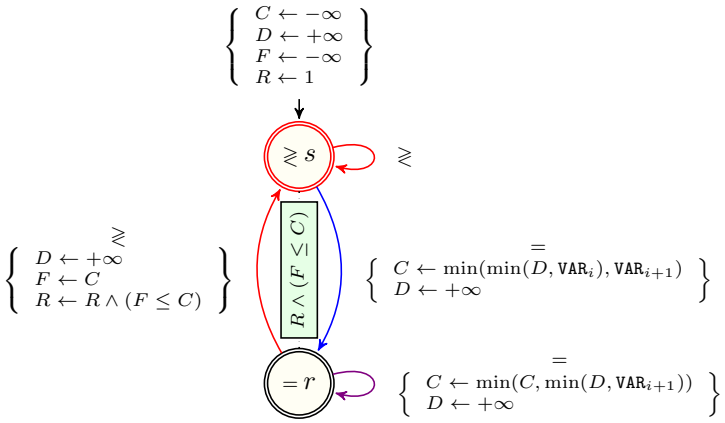
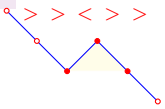


Figure 4.332: Automaton for the INCREASING_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(\langle 7, 5, 4, 2, 5, 4, 3, 3, 5, 7, 6, 5, 6, 5, 4, 1\rangle)$

Figure 4.333 provides an example where the INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE $([7, 5, 4, 2, 5, 4, 3, 3, 5, 7, 6, 5, 6, 5, 4, 1])$ constraint holds.

Typical |VARIABLES| > 5
range(VARIABLES.var) > 2

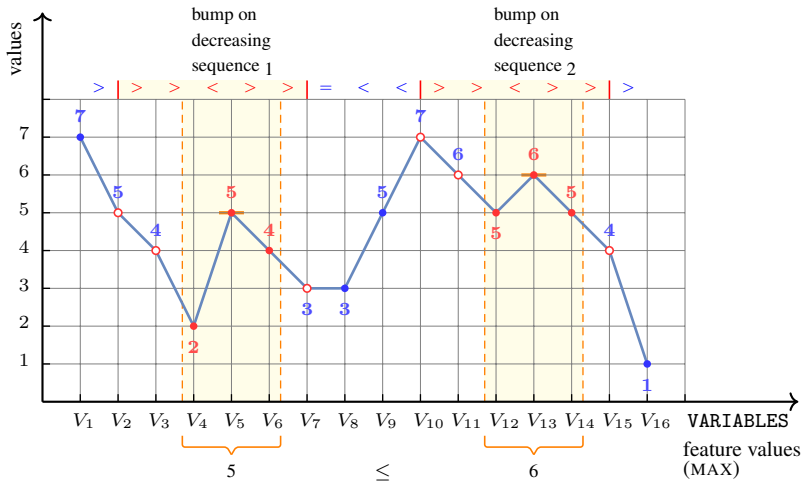


Figure 4.333: Illustrating the INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.334 depicts the automaton associated with the constraint INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE.

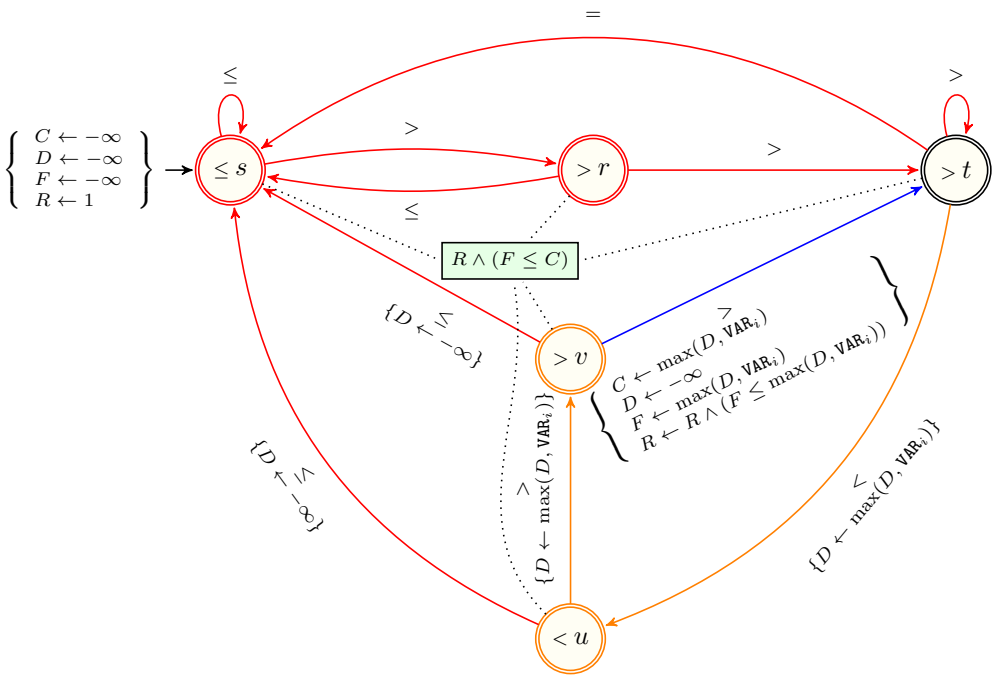


Figure 4.334: Automaton for the INCREASING_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint INCREASING_MAX_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 2,1,1,1,3,2,2,3,1,1,5,3,4,5,6,5\rangle)$

Figure 4.335 provides an example where the INCREASING_MAX_DECREASING $([2,1,1,1,3,2,2,3,1,1,5,3,4,5,6,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

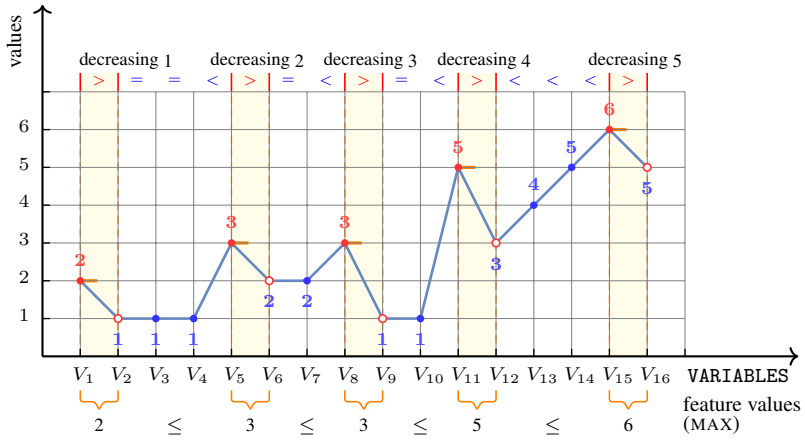


Figure 4.335: Illustrating the INCREASING_MAX_DECREASING constraint of the **Example** slot

Automaton

Figure 4.336 depicts the automaton associated with the constraint INCREASING_MAX_DECREASING.

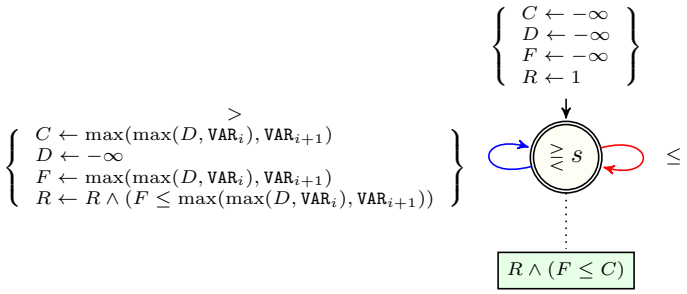
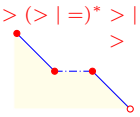


Figure 4.336: Automaton for the INCREASING_MAX_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 4, 3, 2, 2, 3, 4, 4, 6, 5, 5, 3, 4, 4, 6, 4, 6 \rangle)$

Figure 4.337 provides an example where the INCREASING_MAX_DECREASING_SEQUENCE $([4, 3, 2, 2, 3, 4, 4, 6, 5, 5, 3, 4, 4, 6, 4, 6])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

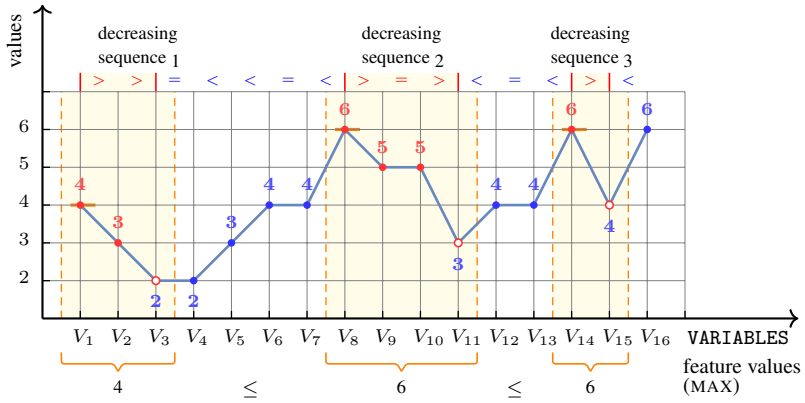


Figure 4.337: Illustrating the INCREASING_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.338 depicts the automaton associated with the constraint INCREASING_MAX_DECREASING_SEQUENCE.

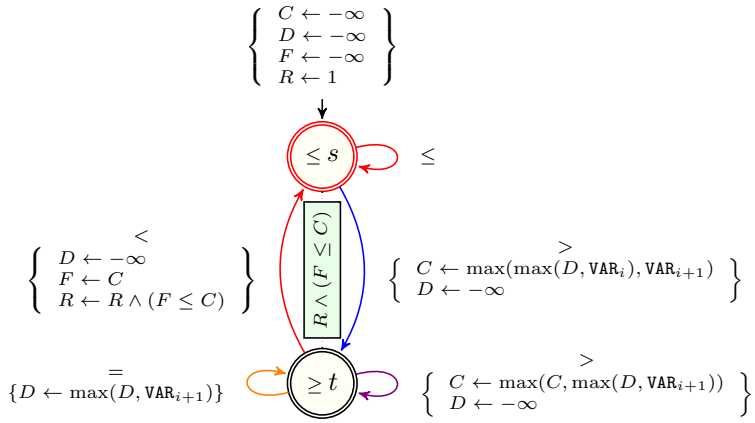
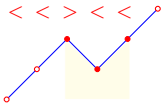


Figure 4.338: Automaton for the INCREASING_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(\langle 1, 2, 3, 2, 3, 4, 6, 5, 1, 4, 5, 6, 0, 2, 4, 4\rangle)$

Figure 4.339 provides an example where the INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE $([1, 2, 3, 2, 3, 4, 6, 5, 1, 4, 5, 6, 0, 2, 4, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

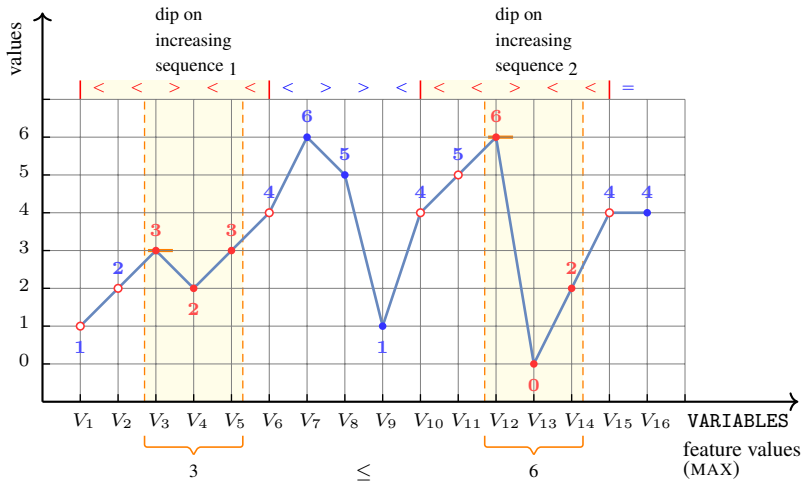


Figure 4.339: Illustrating the INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.340 depicts the automaton associated with the constraint INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE.

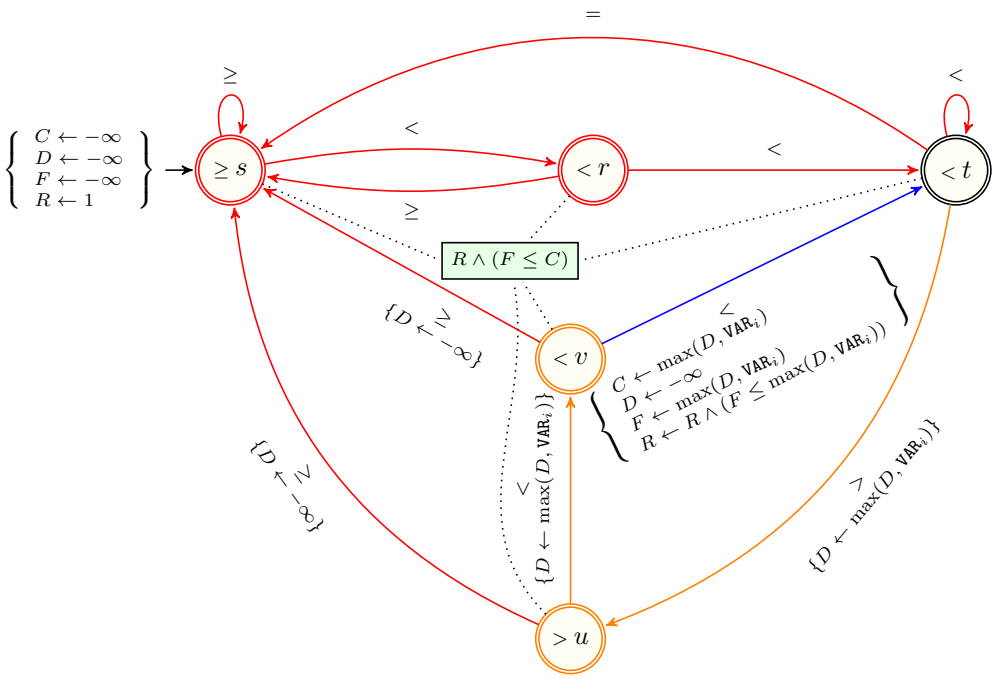


Figure 4.340: Automaton for the INCREASING_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_INCREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING pattern.

Constraint INCREASING_MAX_INCREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 2,1,2,0,2,3,1,1,3,4,2,1,1,0,6,7\rangle)$

Figure 4.341 provides an example where the INCREASING_MAX_INCREASING $([2,1,2,0,2,3,1,1,3,4,2,1,1,0,6,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

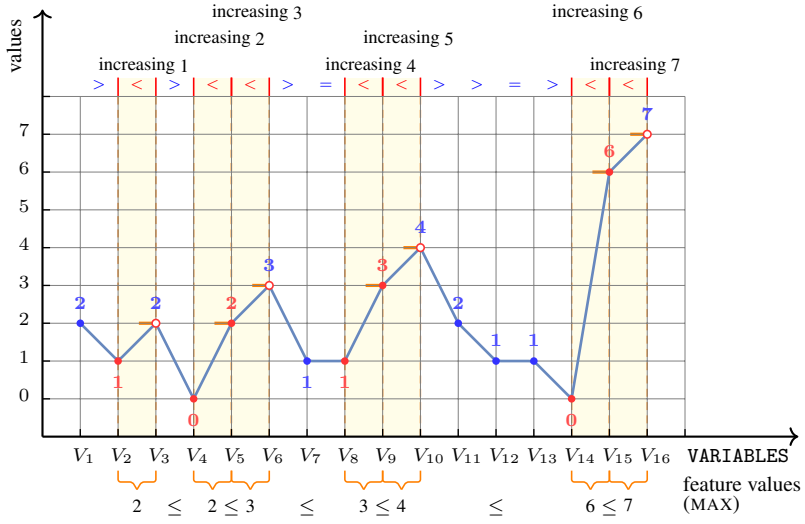


Figure 4.341: Illustrating the INCREASING_MAX_INCREASING constraint of the **Example** slot

Automaton

Figure 4.342 depicts the automaton associated with the constraint INCREASING_MAX_INCREASING.

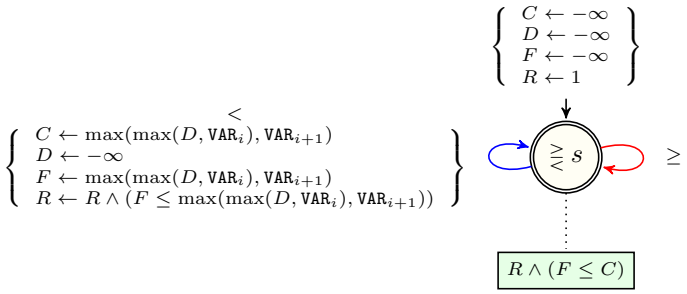
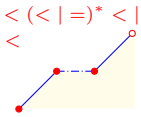


Figure 4.342: Automaton for the INCREASING_MAX_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(\langle 4, 1, 4, 5, 5, 4, 4, 2, 3, 3, 4, 5, 6, 6, 5, 6 \rangle)$

Figure 4.343 provides an example where the INCREASING_MAX_INCREASING_SEQUENCE $([4, 1, 4, 5, 5, 4, 4, 2, 3, 3, 4, 5, 6, 6, 5, 6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

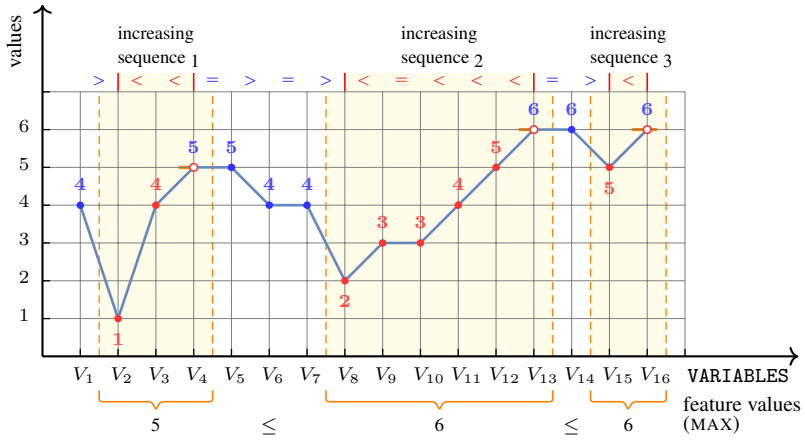


Figure 4.343: Illustrating the INCREASING_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.344 depicts the automaton associated with the constraint INCREASING_MAX_INCREASING_SEQUENCE.

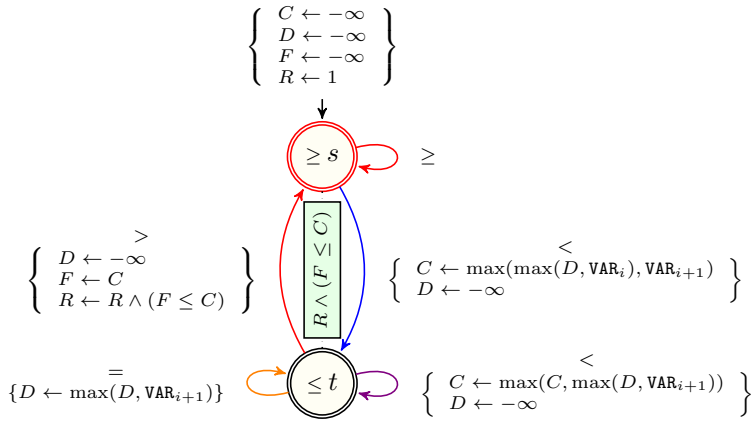
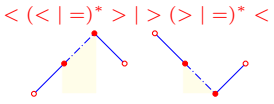


Figure 4.344: Automaton for the INCREASING_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INFLEXION pattern.

Constraint INCREASING_MAX_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 5, 5, 2, 2, 1, 1, 2, 2, 3, 4, 3\rangle)$

Figure 4.345 provides an example where the INCREASING_MAX_INFLEXION $([5, 5, 2, 2, 1, 1, 2, 2, 3, 4, 3])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

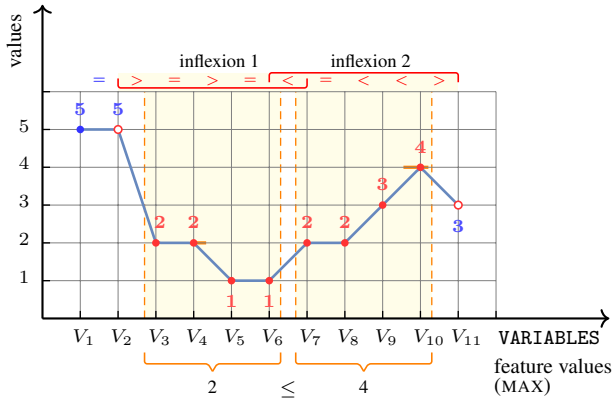


Figure 4.345: Illustrating the INCREASING_MAX_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.346 depicts the automaton associated with the constraint INCREASING_MAX_INFLEXION.

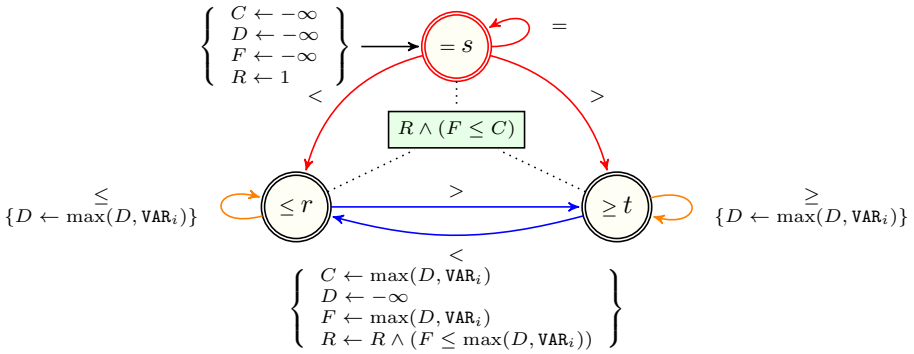
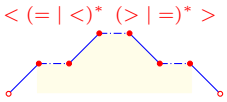


Figure 4.346: Automaton for the INCREASING_MAX_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

INCREASING_MAX_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint INCREASING_MAX_PEAK(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1\rangle)$

Figure 4.347 provides an example where the INCREASING_MAX_PEAK $([7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

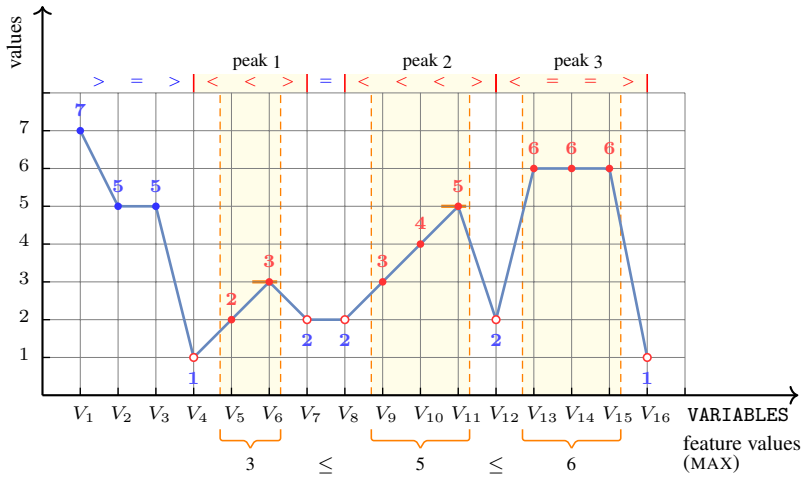


Figure 4.347: Illustrating the INCREASING_MAX_PEAK constraint of the **Example** slot

Automaton Figure 4.348 depicts the automaton associated with the constraint INCREASING_MAX_PEAK.

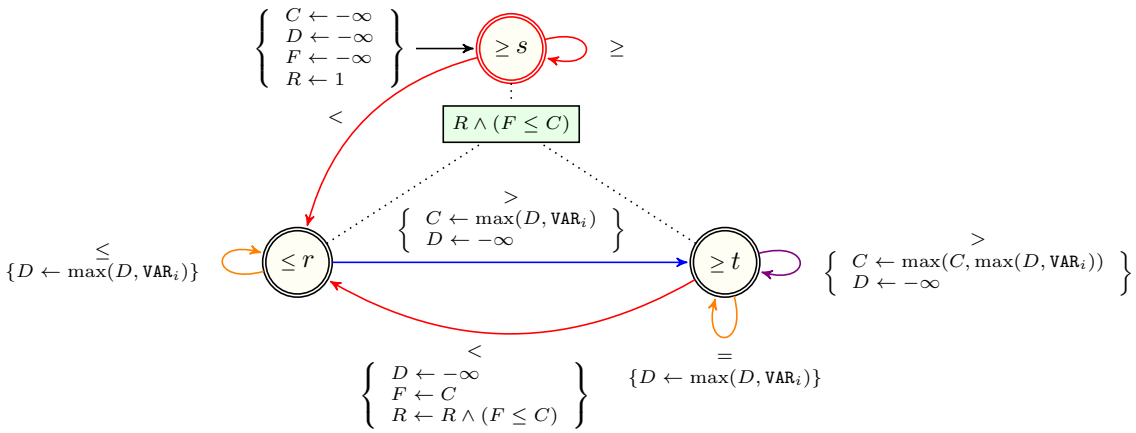


Figure 4.348: Automaton for the INCREASING_MAX_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 2,3,2,2,3,4,5,5,6,5,4,3,2,4,7,4\rangle)$

Figure 4.349 provides an example where the INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE $([2,3,2,2,3,4,5,5,6,5,4,3,2,4,7,4])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

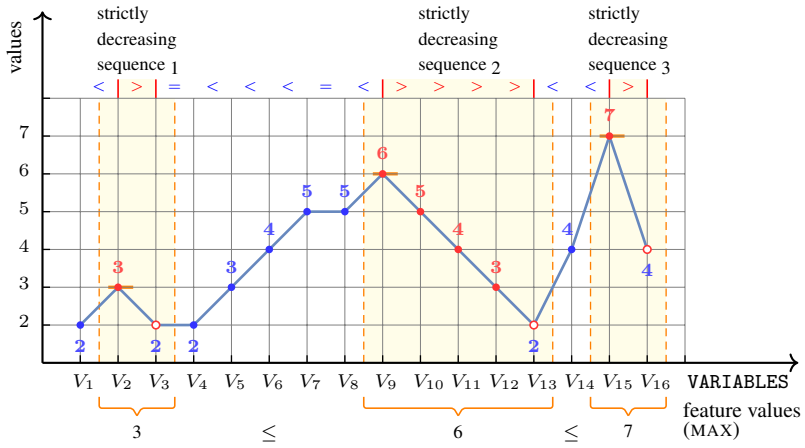


Figure 4.349: Illustrating the INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.350 depicts the automaton associated with the constraint INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE.

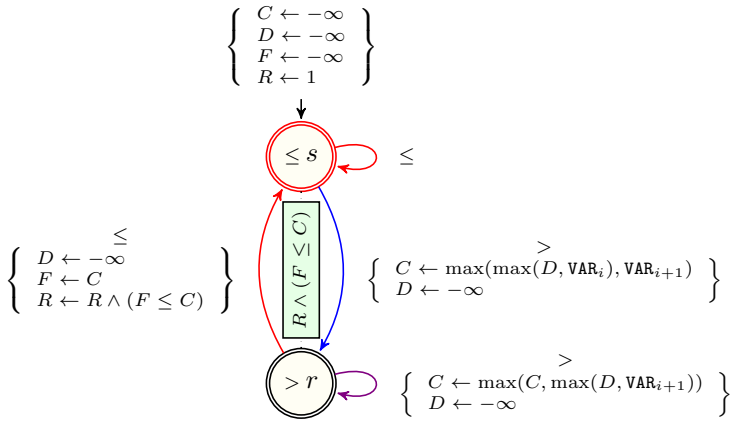


Figure 4.350: Automaton for the INCREASING_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example $(\langle 4,1,4,5,5,4,4,2,2,3,4,5,6,6,5,6\rangle)$

Figure 4.351 provides an example where the INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE $([4,1,4,5,5,4,4,2,2,3,4,5,6,6,5,6])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

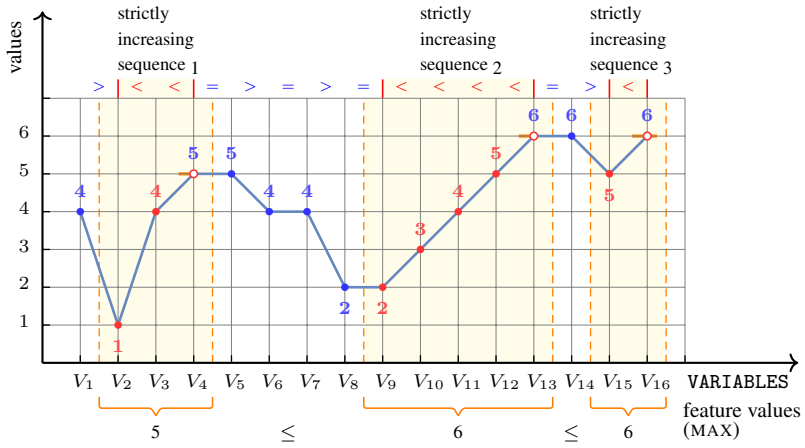


Figure 4.351: Illustrating the INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.352 depicts the automaton associated with the constraint INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE.

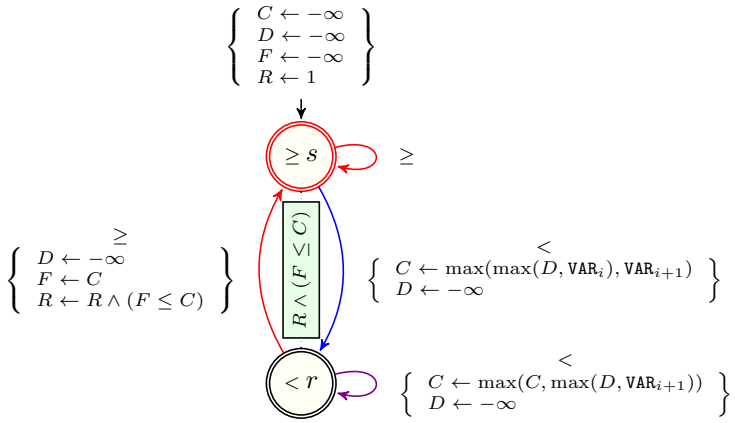
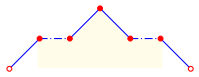


Figure 4.352: Automaton for the INCREASING_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MAX_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint INCREASING_MAX_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2 \rangle)$

Figure 4.353 provides an example where the INCREASING_MAX_SUMMIT $([1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

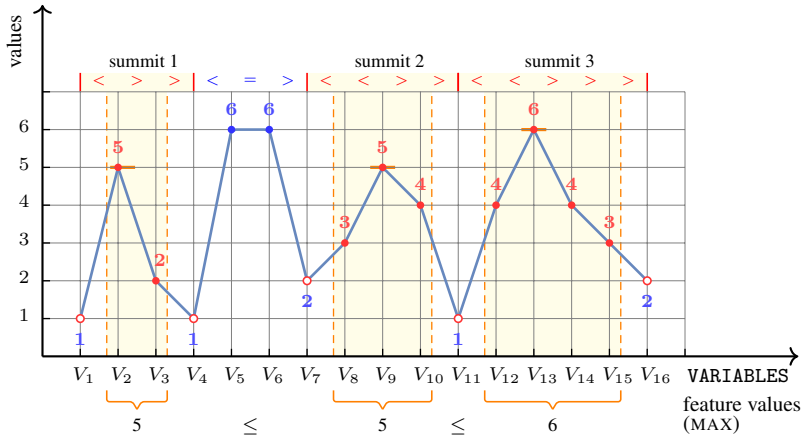


Figure 4.353: Illustrating the INCREASING_MAX_SUMMIT constraint of the **Example** slot

Automaton Figure 4.354 depicts the automaton associated with the constraint INCREASING_MAX_SUMMIT.

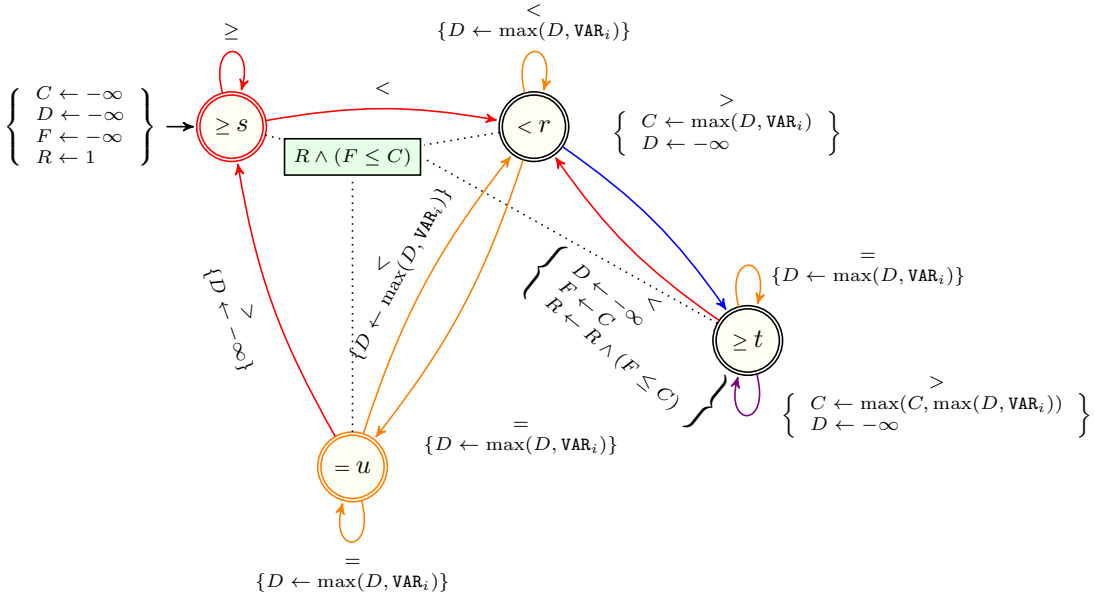
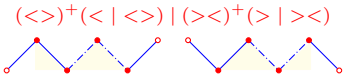


Figure 4.354: Automaton for the INCREASING_MAX_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_MAX_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint INCREASING_MAX_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the maxima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(\langle 7, 7, 2, 5, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6\rangle)$

Figure 4.355 provides an example where the INCREASING_MAX_ZIGZAG $([7, 7, 2, 5, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

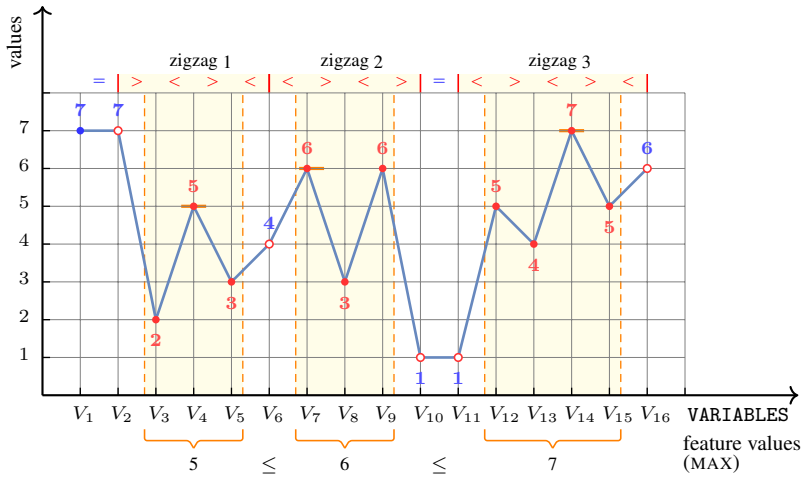


Figure 4.355: Illustrating the INCREASING_MAX_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.356 depicts the automaton associated with the constraint INCREASING_MAX_ZIGZAG.

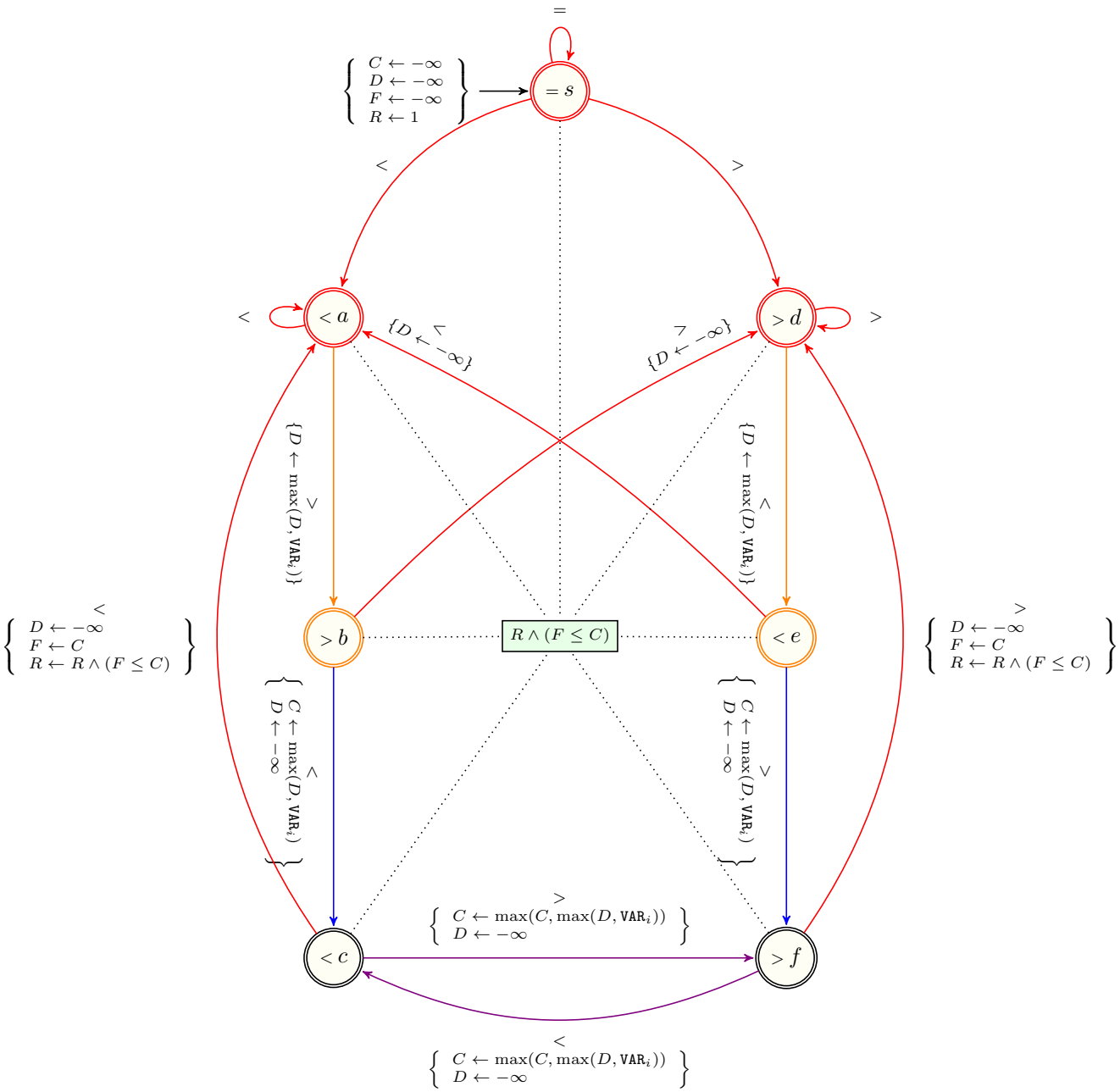
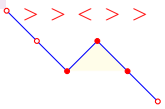


Figure 4.356: Automaton for the INCREASING_MAX_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 5,5,4,2,6,5,4,4,5,7,6,5,6,4,1,1\rangle)$

Figure 4.357 provides an example where the INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE $([5,5,4,2,6,5,4,4,5,7,6,5,6,4,1,1])$ constraint holds.

Typical |VARIABLES| > 5
range(VARIABLES.var) > 2

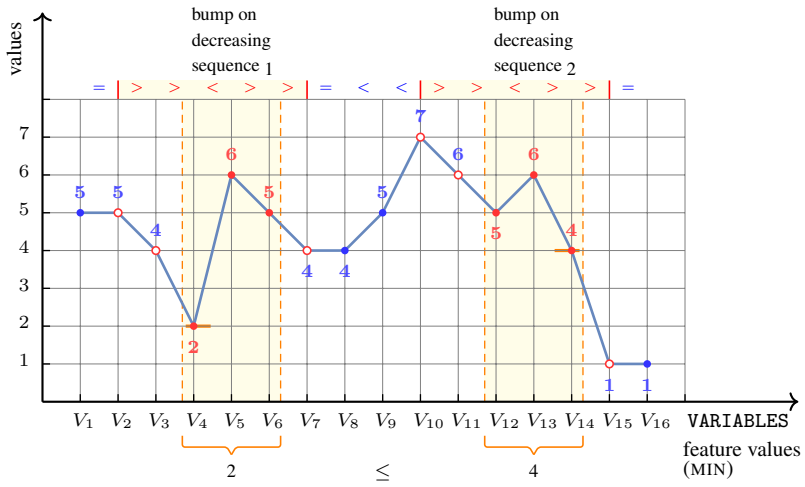


Figure 4.357: Illustrating the INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.358 depicts the automaton associated with the constraint INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE.

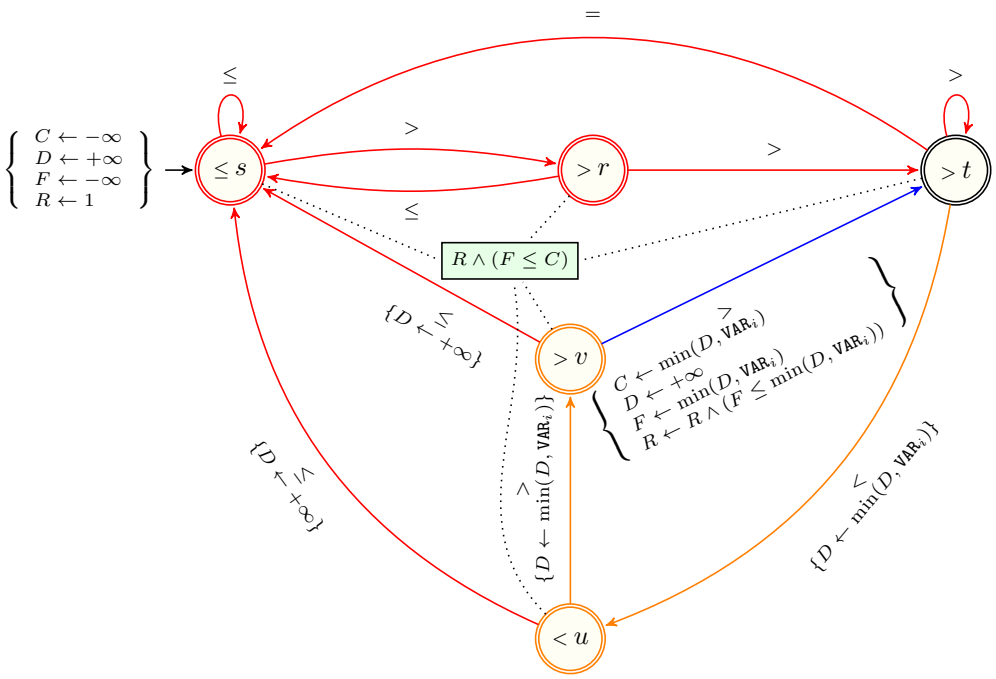


Figure 4.358: Automaton for the INCREASING_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint INCREASING_MIN_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 3,6,6,1,1,3,3,4,5,6,1,5,3,3,6,5\rangle)$

Figure 4.359 provides an example where the INCREASING_MIN_DECREASING $([3,6,6,1,1,3,3,4,5,6,1,5,3,3,6,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

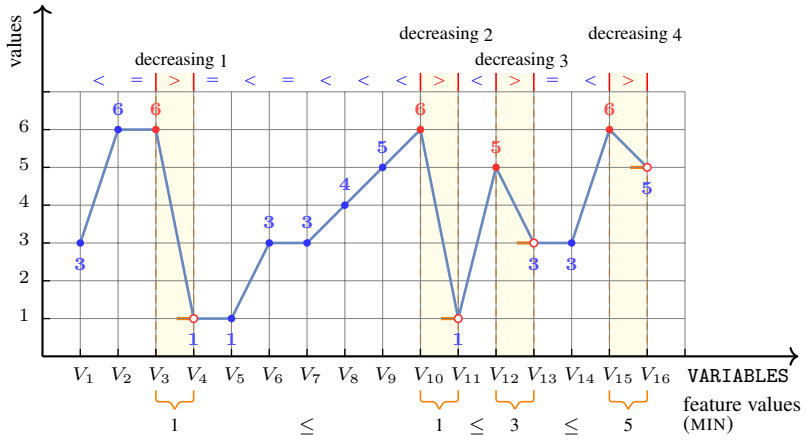


Figure 4.359: Illustrating the INCREASING_MIN_DECREASING constraint of the **Example** slot

Automaton

Figure 4.360 depicts the automaton associated with the constraint INCREASING_MIN_DECREASING.

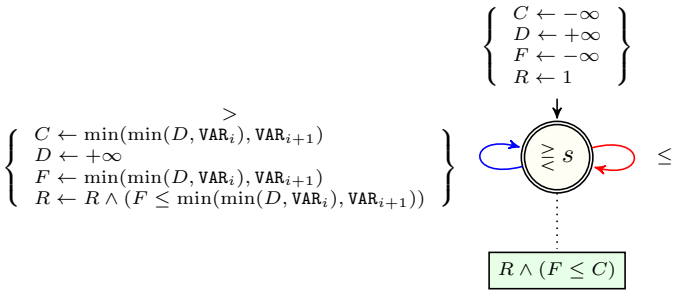
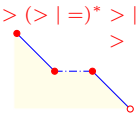


Figure 4.360: Automaton for the INCREASING_MIN_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 4, 3, 2, 2, 3, 4, 4, 6, 3, 4, 4, 5, 5, 6, 4, 6\rangle)$

Figure 4.361 provides an example where the INCREASING_MIN_DECREASING_SEQUENCE $([4, 3, 2, 2, 3, 4, 4, 6, 3, 4, 4, 5, 5, 6, 4, 6])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

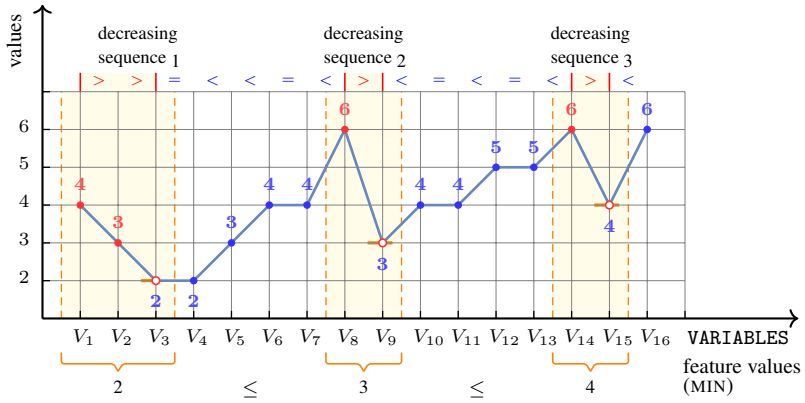


Figure 4.361: Illustrating the INCREASING_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.362 depicts the automaton associated with the constraint INCREASING_MIN_DECREASING_SEQUENCE.

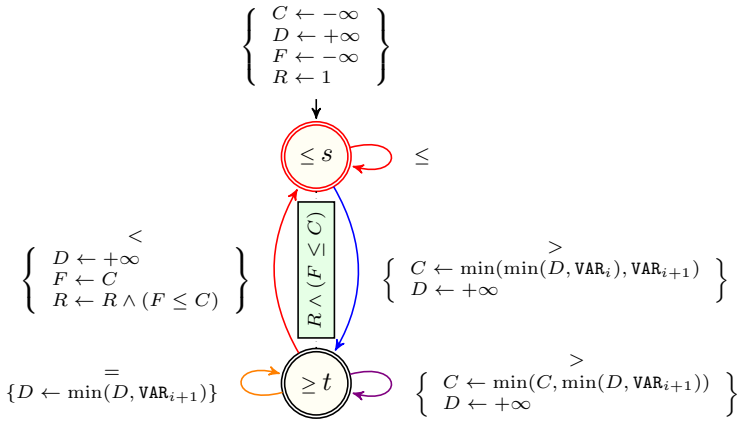
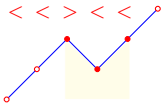


Figure 4.362: Automaton for the INCREASING_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example $(\langle 1,2,3,0,3,4,6,5,1,4,5,6,1,2,4,4\rangle)$

Figure 4.363 provides an example where the INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE $([1,2,3,0,3,4,6,5,1,4,5,6,1,2,4,4])$ constraint holds.

Typical |VARIABLES| > 5
range(VARIABLES.var) > 2

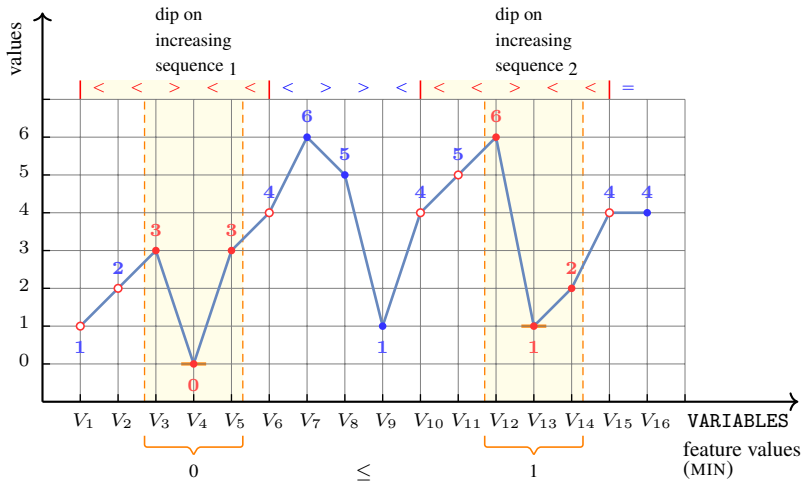


Figure 4.363: Illustrating the INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.364 depicts the automaton associated with the constraint INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE.

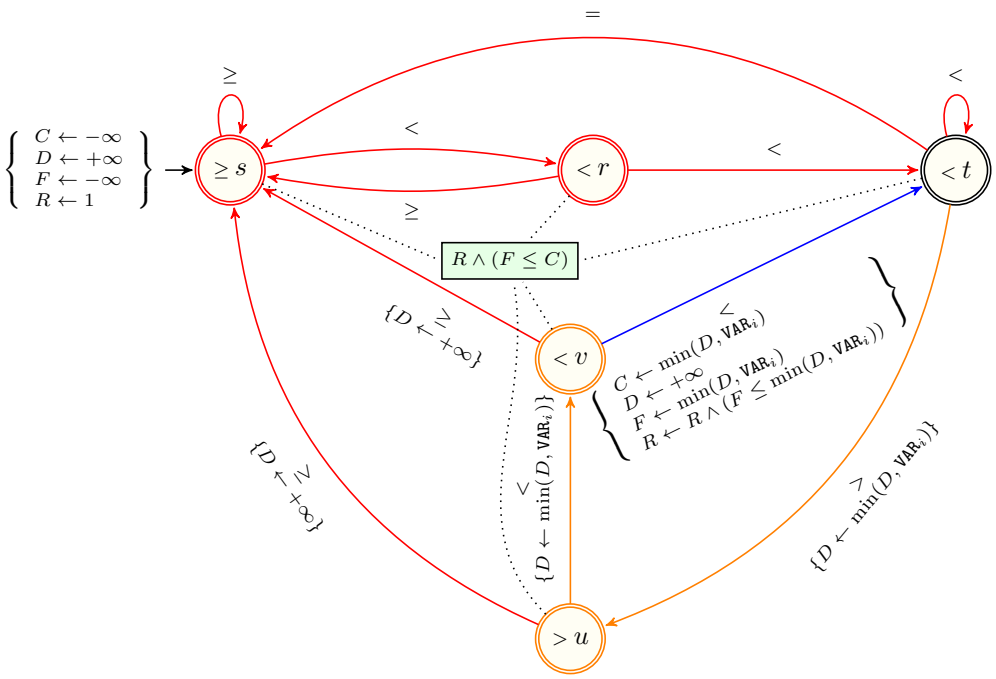
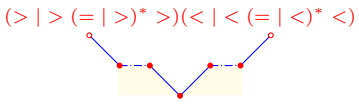


Figure 4.364: Automaton for the INCREASING_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint INCREASING_MIN_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5 \rangle)$

Figure 4.365 provides an example where the INCREASING_MIN_GORGE $([6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

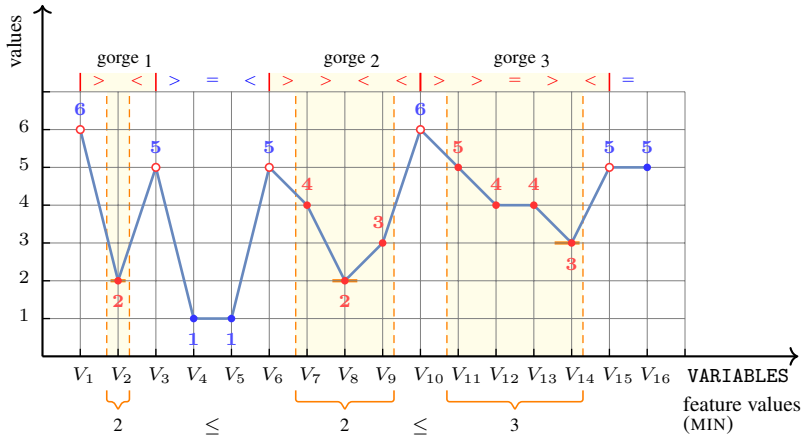


Figure 4.365: Illustrating the INCREASING_MIN_GORGE constraint of the **Example** slot

Automaton Figure 4.366 depicts the automaton associated with the constraint INCREASING_MIN_GORGE.

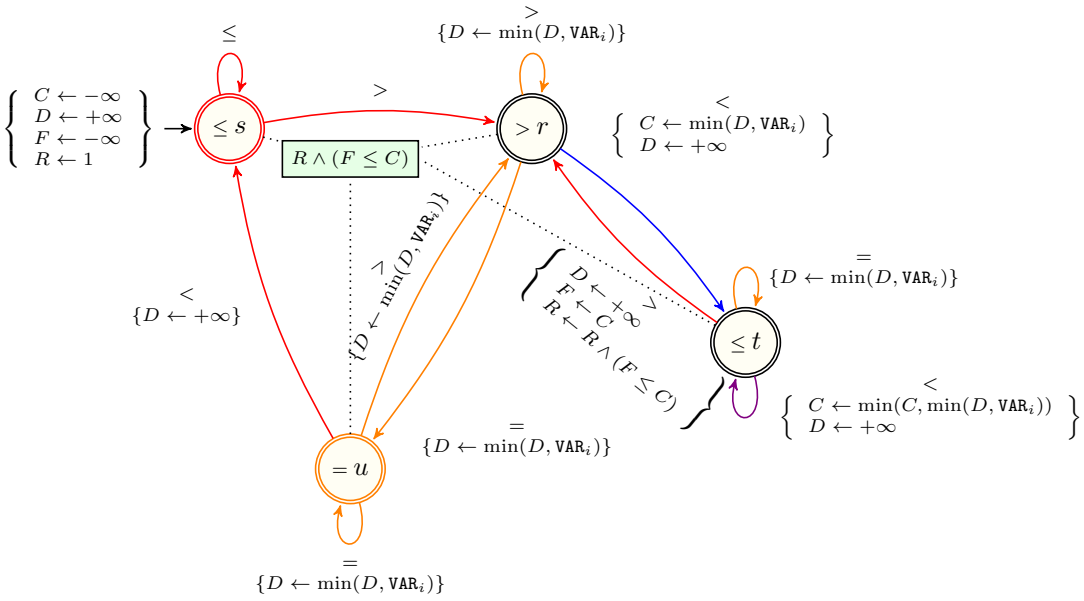


Figure 4.366: Automaton for the INCREASING_MIN_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint INCREASING_MIN_INCREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,1,2,4,3,3,3,2,3,3,4,4,6,6,5,7\rangle)$

Figure 4.367 provides an example where the INCREASING_MIN_INCREASING $([4,1,2,4,3,3,3,2,3,3,4,4,6,6,5,7])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

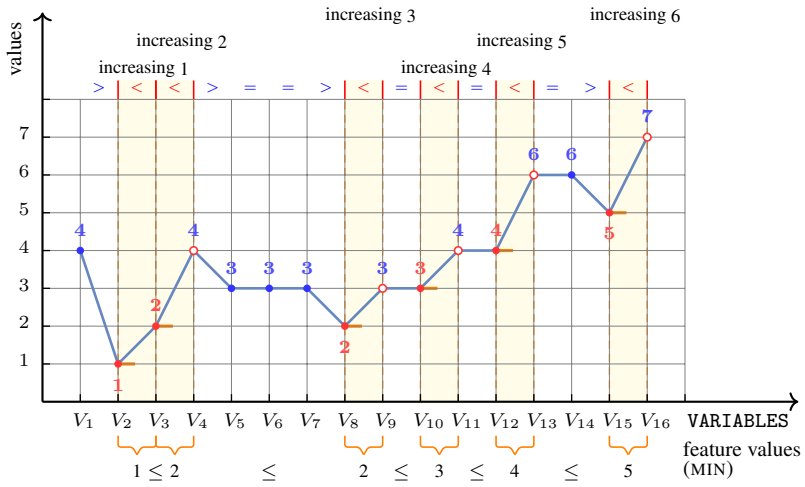


Figure 4.367: Illustrating the INCREASING_MIN_INCREASING constraint of the **Example** slot

Automaton

Figure 4.368 depicts the automaton associated with the constraint INCREASING_MIN_INCREASING.

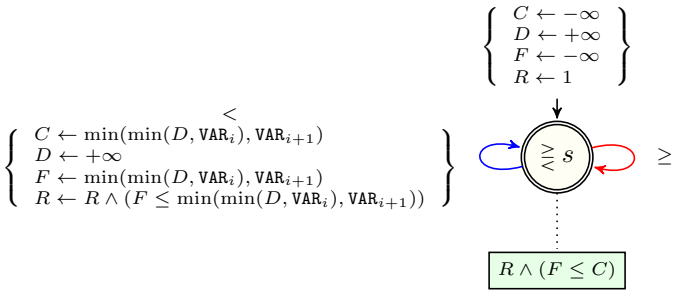
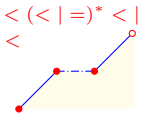


Figure 4.368: Automaton for the INCREASING_MIN_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(\langle 4, 1, 2, 4, 3, 3, 3, 2, 3, 3, 4, 4, 6, 6, 5, 7 \rangle)$

Figure 4.369 provides an example where the INCREASING_MIN_INCREASING_SEQUENCE $([4, 1, 2, 4, 3, 3, 3, 2, 3, 3, 4, 4, 6, 6, 5, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

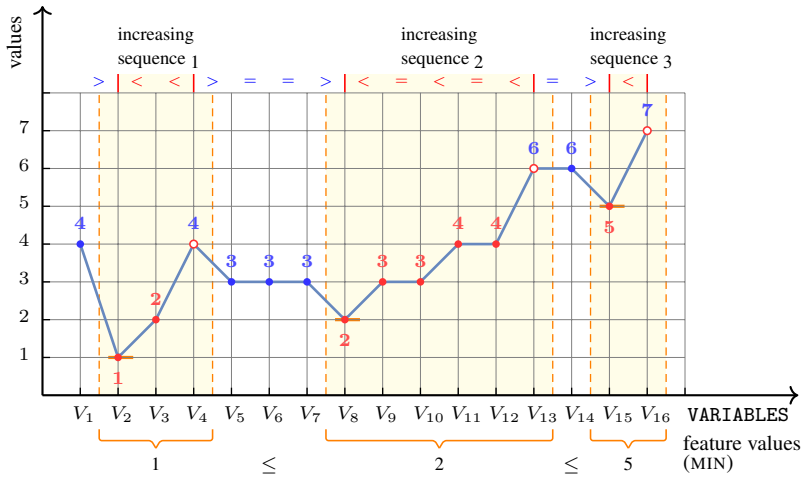


Figure 4.369: Illustrating the INCREASING_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.370 depicts the automaton associated with the constraint INCREASING_MIN_INCREASING_SEQUENCE.

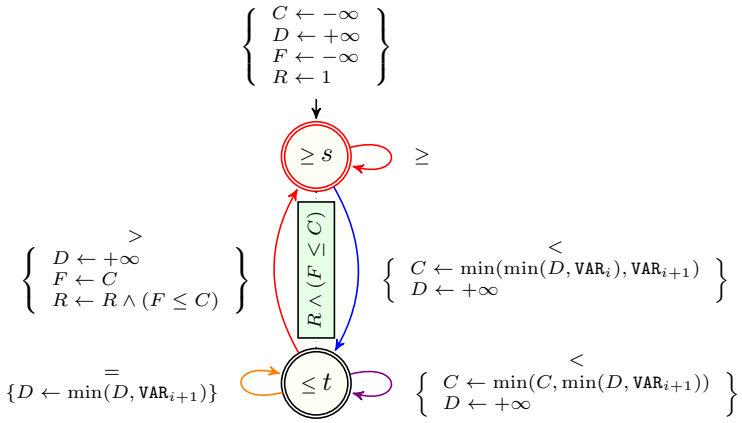
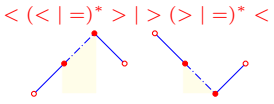


Figure 4.370: Automaton for the INCREASING_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint INCREASING_MIN_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 1$ to index j.

Example $(\langle 1, 1, 2, 4, 4, 5, 3, 3, 3, 4, 5, 6 \rangle)$

Figure 4.371 provides an example where the INCREASING_MIN_INFLEXION $([1, 1, 2, 4, 4, 5, 3, 3, 3, 4, 5, 6])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

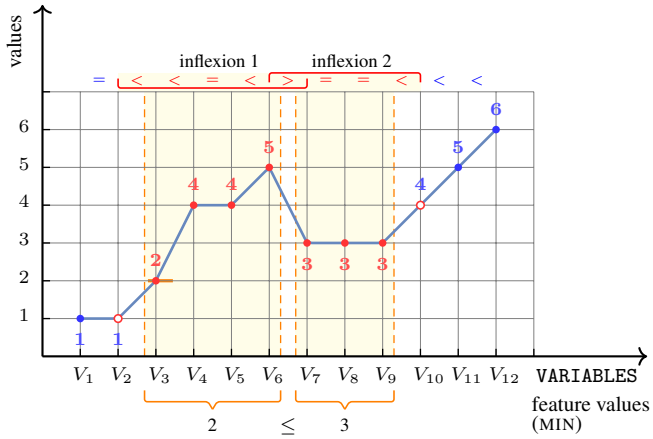


Figure 4.371: Illustrating the INCREASING_MIN_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.372 depicts the automaton associated with the constraint INCREASING_MIN_INFLEXION.

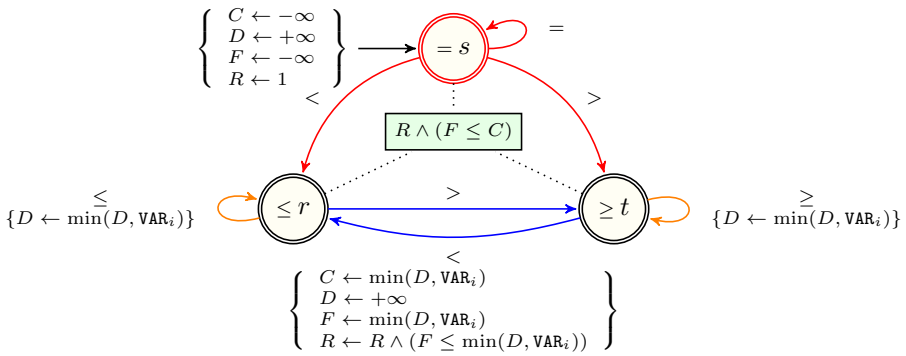


Figure 4.372: Automaton for the INCREASING_MIN_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 1,3,2,2,4,4,5,5,6,3,5,5,4,6,7,4\rangle)$

Figure 4.373 provides an example where the INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE $([1,3,2,2,4,4,5,5,6,3,5,5,4,6,7,4])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

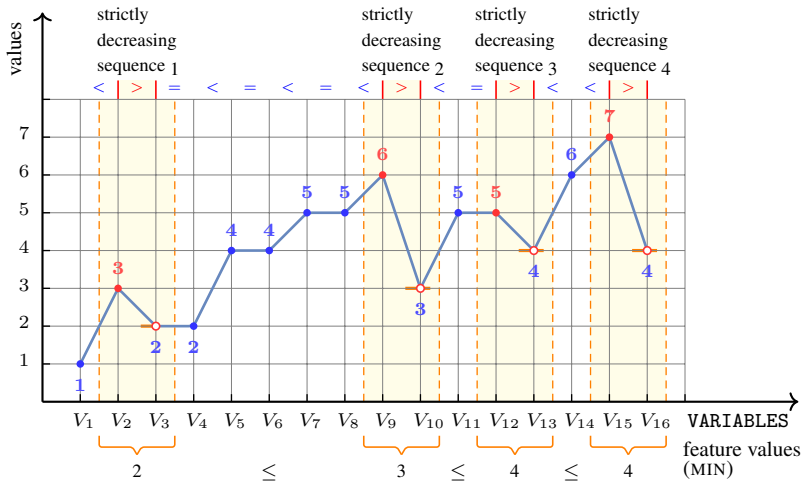


Figure 4.373: Illustrating the INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.374 depicts the automaton associated with the constraint INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE.

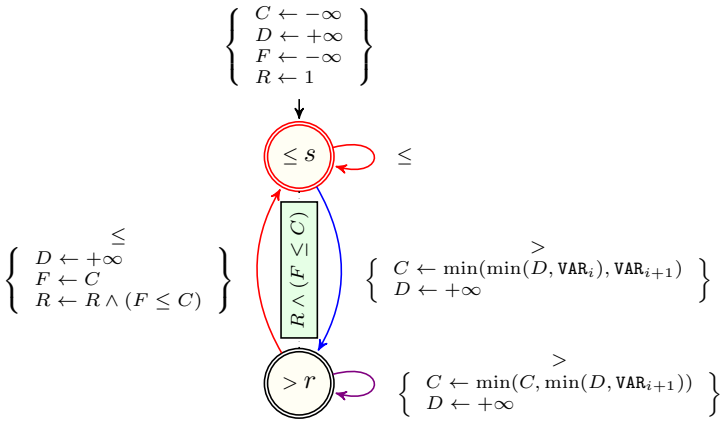


Figure 4.374: Automaton for the INCREASING_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(\langle 4,1,2,4,3,3,3,2,3,3,4,5,6,6,5,7\rangle)$

Figure 4.375 provides an example where the INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE $([4,1,2,4,3,3,3,2,3,3,4,5,6,6,5,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

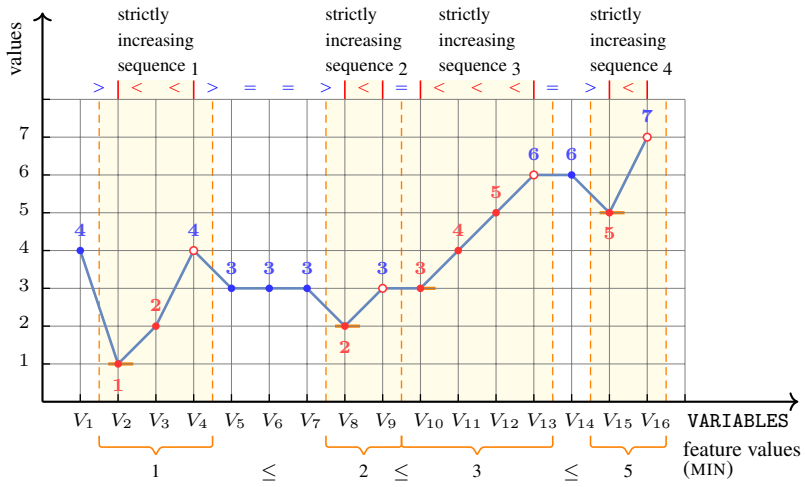


Figure 4.375: Illustrating the INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.376 depicts the automaton associated with the constraint INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE.

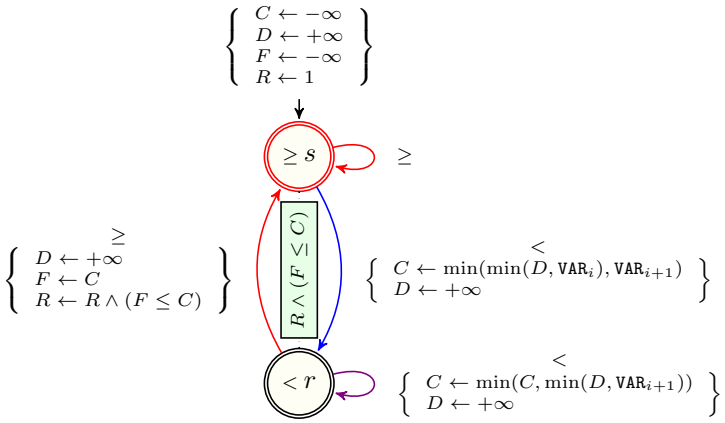
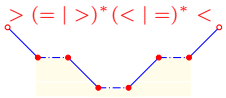


Figure 4.376: Automaton for the INCREASING_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint INCREASING_MIN_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1 \rangle)$

Figure 4.377 provides an example where the INCREASING_MIN_VALLEY $([7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

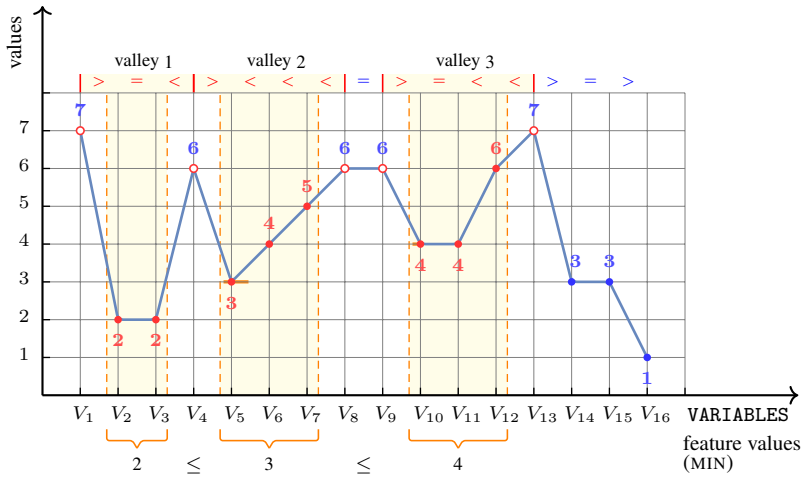


Figure 4.377: Illustrating the INCREASING_MIN_VALLEY constraint of the **Example** slot

Automaton Figure 4.378 depicts the automaton associated with the constraint INCREASING_MIN_VALLEY.

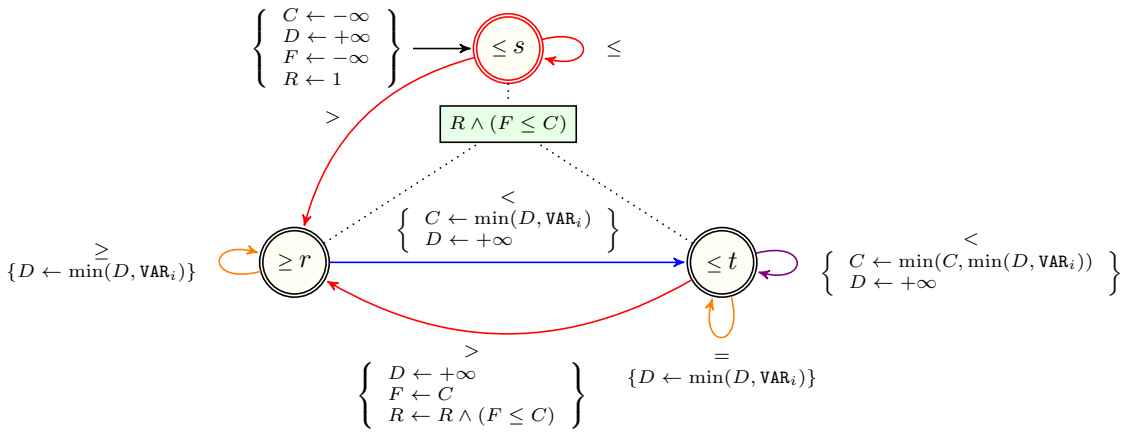
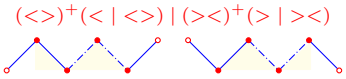


Figure 4.378: Automaton for the INCREASING_MIN_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

INCREASING_MIN_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint INCREASING_MIN_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the minima of the values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(\langle 7,7,2,5,3,4,6,3,6,1,1,5,4,7,5,6\rangle)$

Figure 4.379 provides an example where the INCREASING_MIN_ZIGZAG $([7,7,2,5,3,4,6,3,6,1,1,5,4,7,5,6])$ constraint holds.

Typical
$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

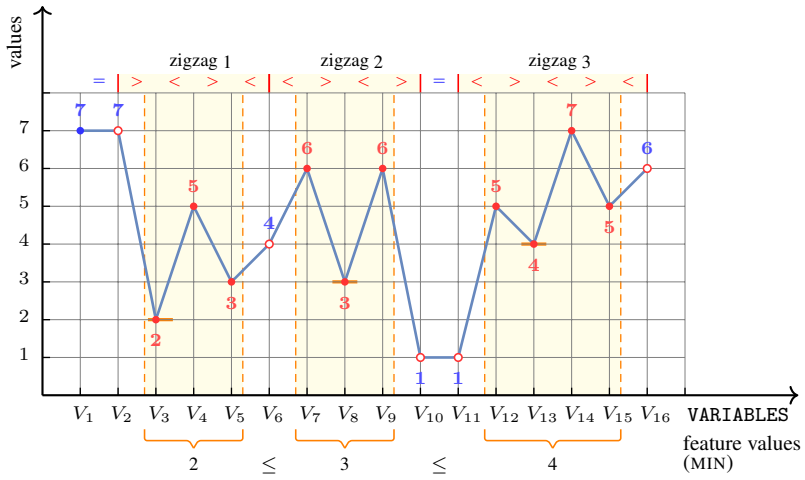


Figure 4.379: Illustrating the INCREASING_MIN_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.380 depicts the automaton associated with the constraint INCREASING_MIN_ZIGZAG.

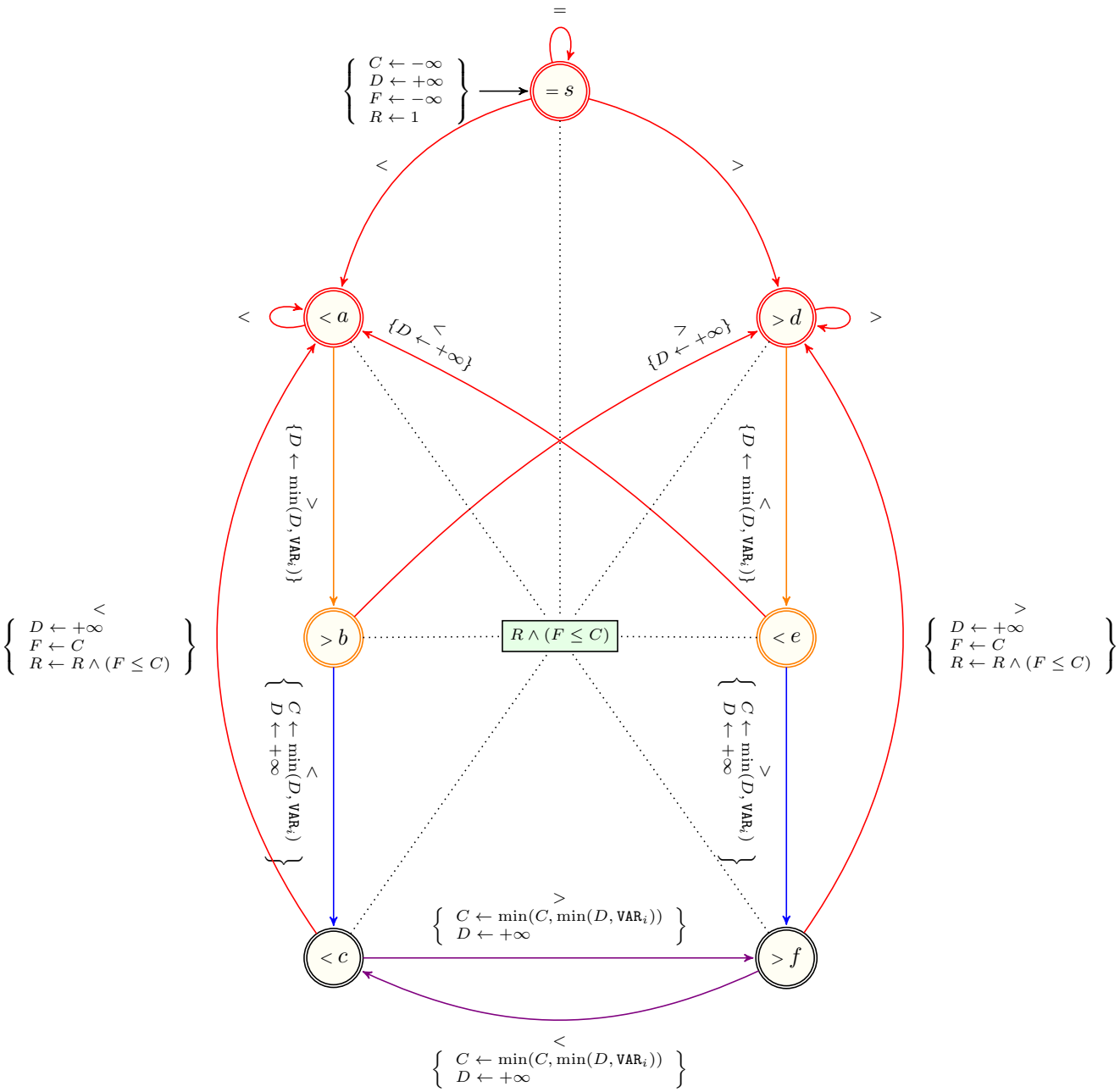


Figure 4.380: Automaton for the INCREASING_MIN_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

INCREASING_RANGE_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint INCREASING_RANGE_DECREASING(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 3, 6, 6, 5, 4, 4, 6, 4, 2, 2, 0, 5, 2, 6, 3, 3\rangle)$

Figure 4.381 provides an example where the INCREASING_RANGE_DECREASING $([3, 6, 6, 5, 4, 4, 6, 4, 2, 2, 0, 5, 2, 6, 3, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

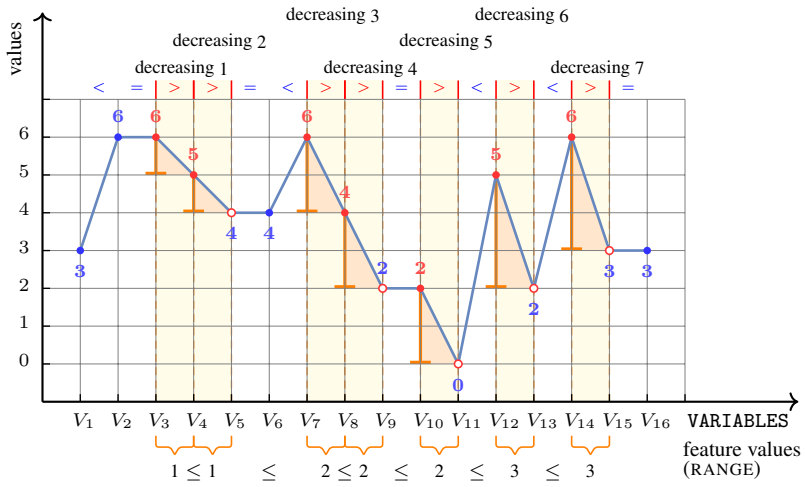


Figure 4.381: Illustrating the INCREASING_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figure 4.382 depicts the automaton associated with the constraint INCREASING_RANGE_DECREASING.

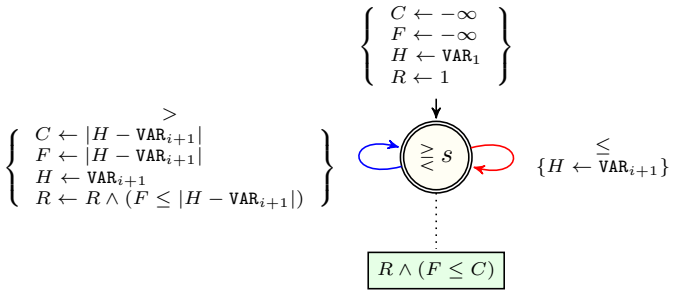
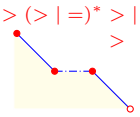


Figure 4.382: Automaton for the INCREASING_RANGE_DECREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

INCREASING_RANGE_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INCREASING_RANGE_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 4, 3, 2, 2, 0, 0, 3, 4, 4, 6, 2, 4, 4, 6, 5, 1\rangle)$

Figure 4.383 provides an example where the INCREASING_RANGE_DECREASING_SEQUENCE $([4, 3, 2, 2, 0, 0, 3, 4, 4, 6, 2, 4, 4, 6, 5, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

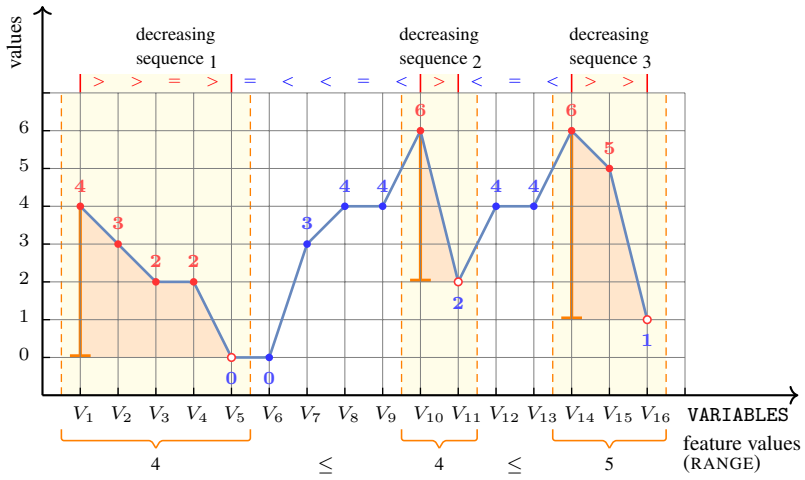


Figure 4.383: Illustrating the INCREASING_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.384 depicts the automaton associated with the constraint INCREASING_RANGE_DECREASING_SEQUENCE.

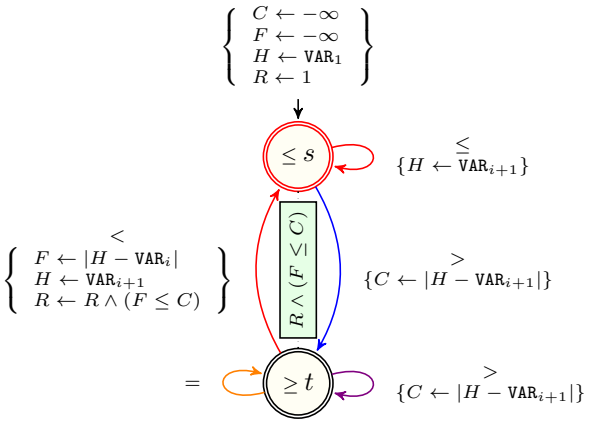


Figure 4.384: Automaton for the INCREASING_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_RANGE_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint INCREASING_RANGE_INCREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 4,5,6,6,3,3,4,6,2,2,1,4,7,0,5,5\rangle)$

Figure 4.385 provides an example where the INCREASING_RANGE_INCREASING $([4,5,6,6,3,3,4,6,2,2,1,4,7,0,5,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

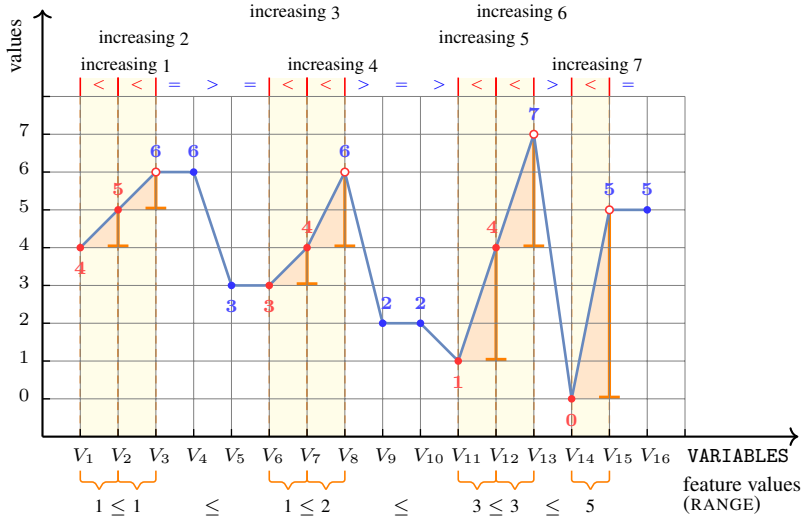


Figure 4.385: Illustrating the INCREASING_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figure 4.386 depicts the automaton associated with the constraint INCREASING_RANGE_INCREASING.

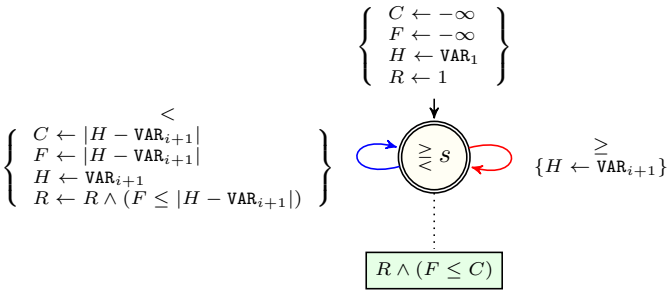
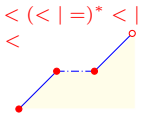


Figure 4.386: Automaton for the INCREASING_RANGE_INCREASING constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

INCREASING_RANGE_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INCREASING_RANGE_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 4, 1, 2, 4, 3, 3, 3, 2, 3, 3, 4, 4, 6, 6, 3, 7\rangle)$

Figure 4.387 provides an example where the INCREASING_RANGE_INCREASING_SEQUENCE $([4, 1, 2, 4, 3, 3, 3, 2, 3, 3, 4, 4, 6, 6, 3, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

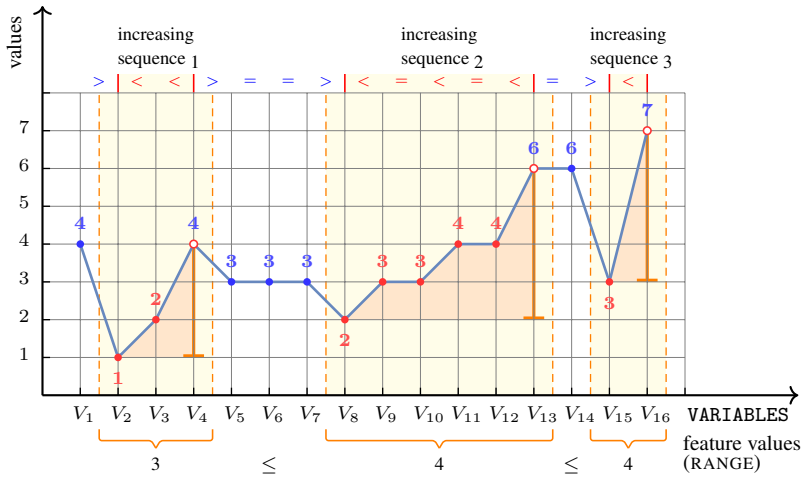


Figure 4.387: Illustrating the INCREASING_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.388 depicts the automaton associated with the constraint INCREASING_RANGE_INCREASING_SEQUENCE.

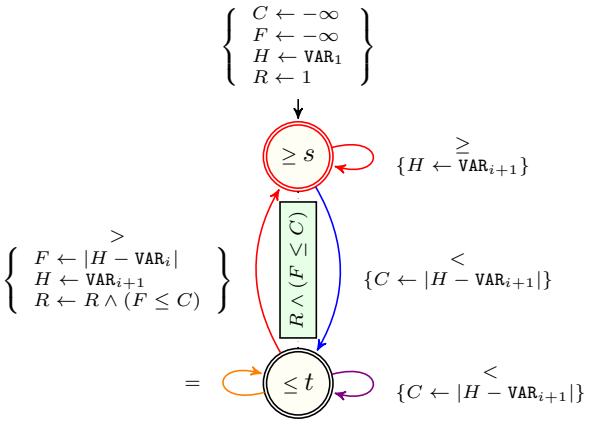


Figure 4.388: Automaton for the INCREASING_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(\langle 1,3,2,2,5,6,5,3,3,0,1,3,5,1,1,2\rangle)$

Figure 4.389 provides an example where the INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE $([1,3,2,2,5,6,5,3,3,0,1,3,5,1,1,2])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

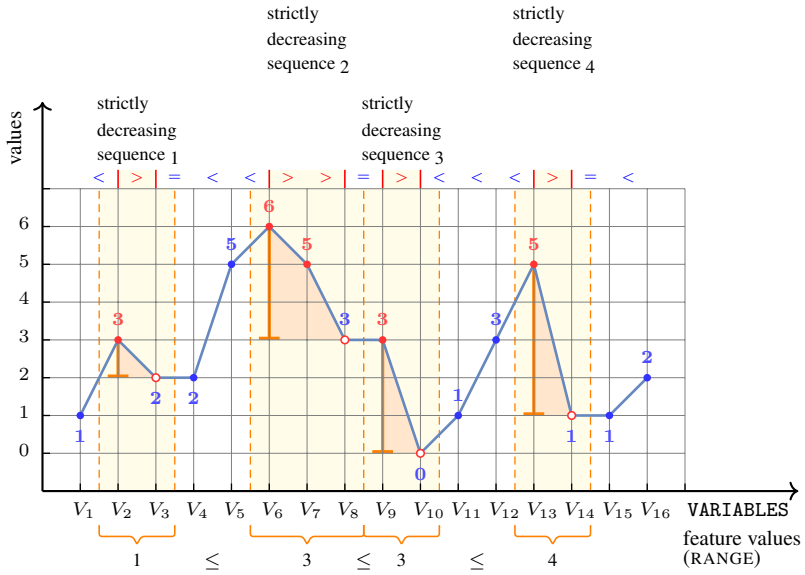


Figure 4.389: Illustrating the INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.390 depicts the automaton associated with the constraint INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE.

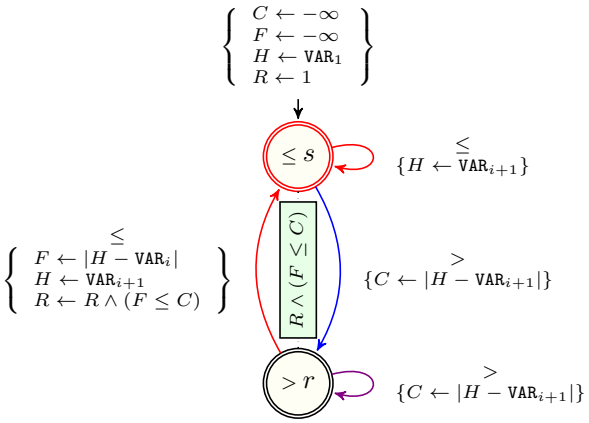


Figure 4.390: Automaton for the INCREASING_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(\langle 4, 1, 2, 2, 3, 0, 1, 1, 4, 1, 1, 2, 3, 4, 4, 8\rangle)$

Figure 4.391 provides an example where the INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE $([4, 1, 2, 2, 3, 0, 1, 1, 4, 1, 1, 2, 3, 4, 4, 8])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

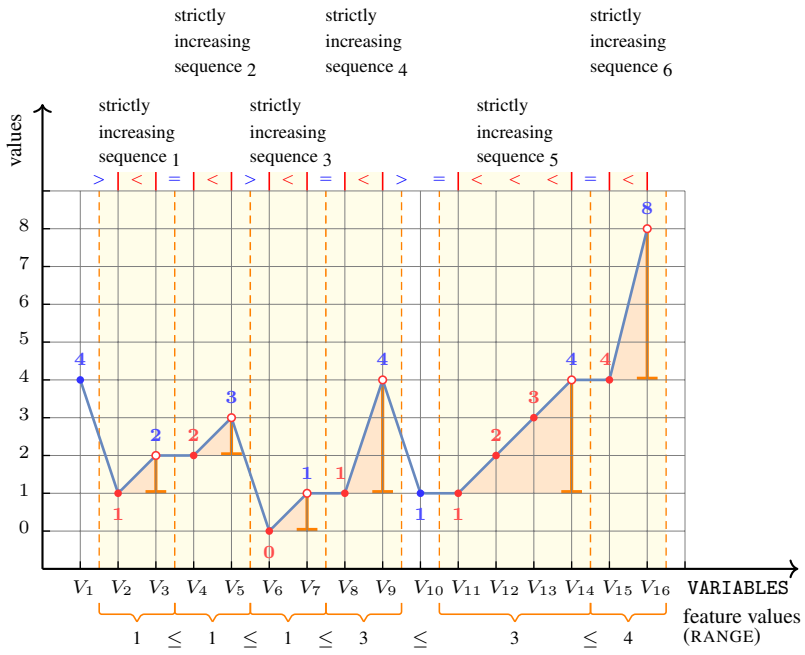


Figure 4.391: Illustrating the INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.392 depicts the automaton associated with the constraint INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE.

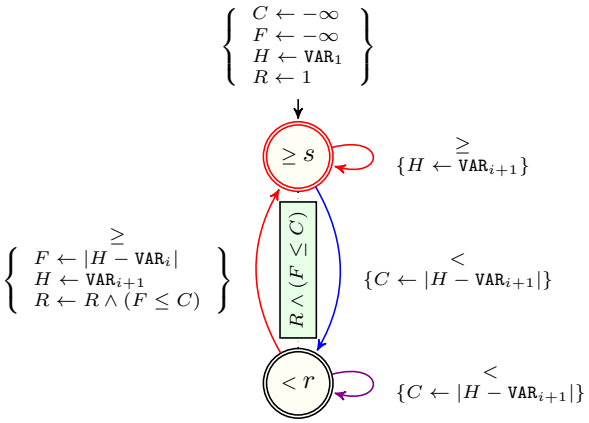
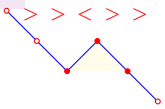


Figure 4.392: Automaton for the INCREASING_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.49 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+2$ to index j.

Example $(\langle 7,6,4,2,5,4,3,3,5,7,6,5,6,5,4,1\rangle)$

Figure 4.393 provides an example where the INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE $([7,6,4,2,5,4,3,3,5,7,6,5,6,5,4,1])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

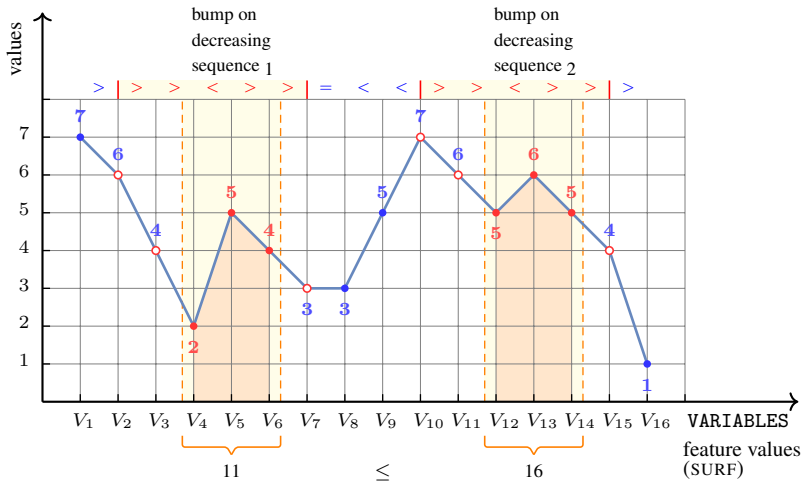


Figure 4.393: Illustrating the INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.394 depicts the automaton associated with the constraint INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE.

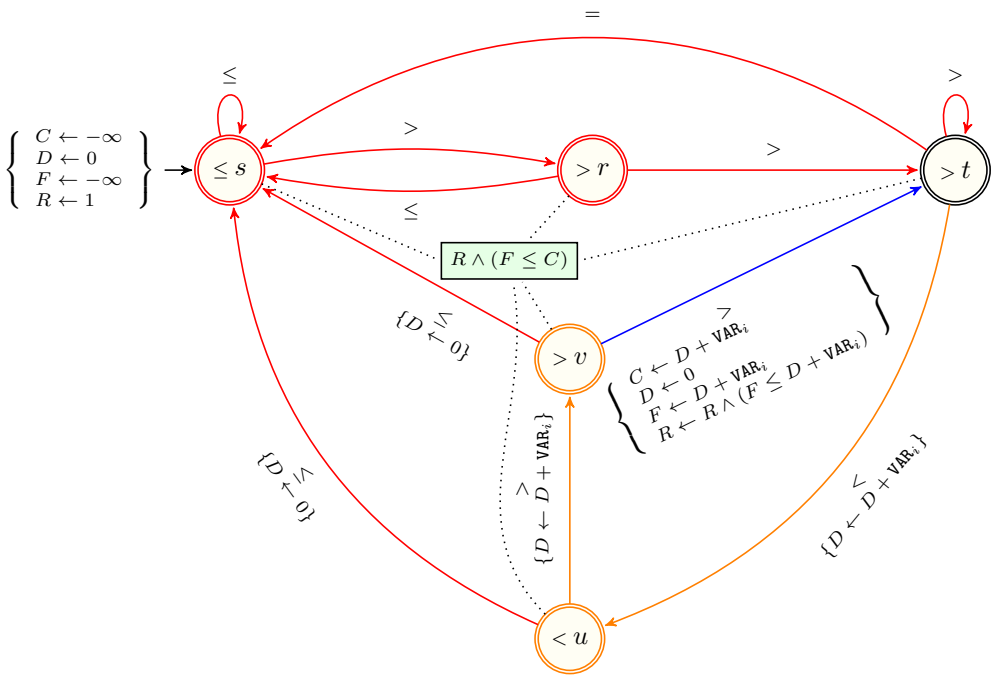


Figure 4.394: Automaton for the INCREASING_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_DECREASING

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING pattern.

Constraint INCREASING_SURF_DECREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 3,3,2,2,3,4,4,6,3,4,4,5,5,6,4,6\rangle)$

Figure 4.395 provides an example where the INCREASING_SURF_DECREASING $([3,3,2,2,3,4,4,6,3,4,4,5,5,6,4,6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

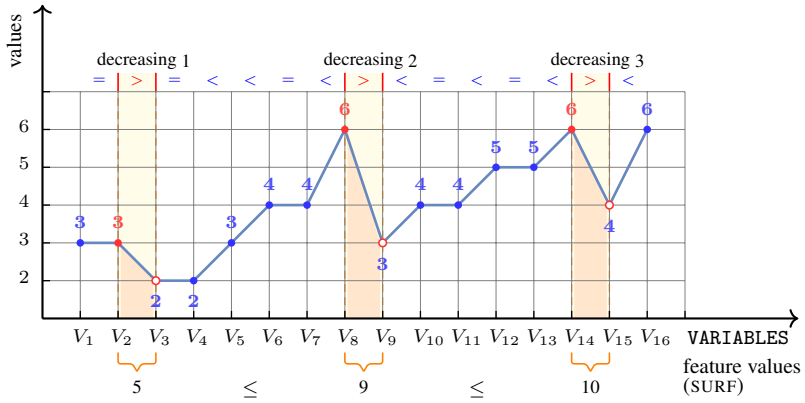


Figure 4.395: Illustrating the INCREASING_SURF_DECREASING constraint of the **Example** slot

Automaton

Figure 4.396 depicts the automaton associated with the constraint INCREASING_SURF_DECREASING.

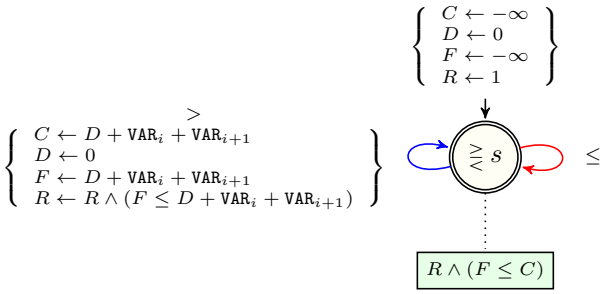
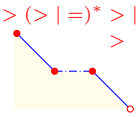


Figure 4.396: Automaton for the INCREASING_SURF_DECREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 4, 3, 2, 2, 3, 4, 4, 6, 3, 4, 4, 5, 5, 6, 4, 6\rangle)$

Figure 4.397 provides an example where the INCREASING_SURF_DECREASING_SEQUENCE $([4, 3, 2, 2, 3, 4, 4, 6, 3, 4, 4, 5, 5, 6, 4, 6])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

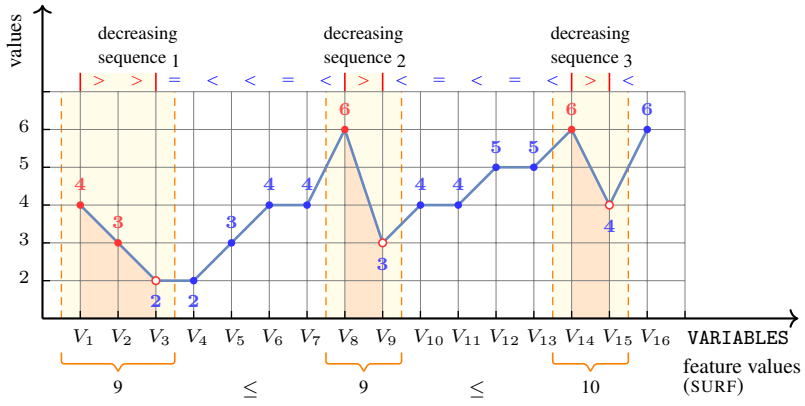


Figure 4.397: Illustrating the INCREASING_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.398 depicts the automaton associated with the constraint INCREASING_SURF_DECREASING_SEQUENCE.

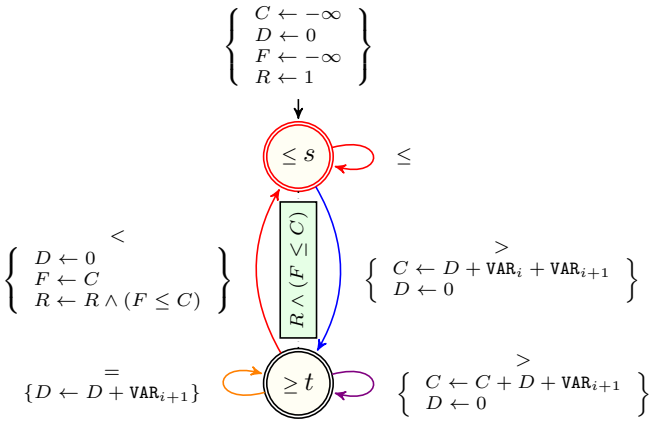
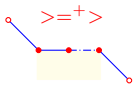


Figure 4.398: Automaton for the INCREASING_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the DECREASING_TERRACE pattern.

Constraint INCREASING_SURF_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 5,2,2,1,5,4,3,3,2,4,4,6,5,5,5,4\rangle)$

Figure 4.399 provides an example where the INCREASING_SURF_DECREASING_TERRACE $([5,2,2,1,5,4,3,3,2,4,4,6,5,5,5,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

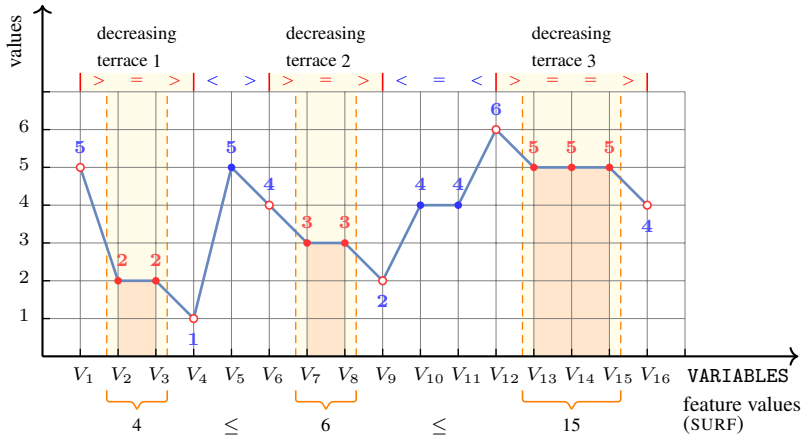


Figure 4.399: Illustrating the INCREASING_SURF_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.400 depicts the automaton associated with the constraint INCREASING_SURF_DECREASING_TERRACE.

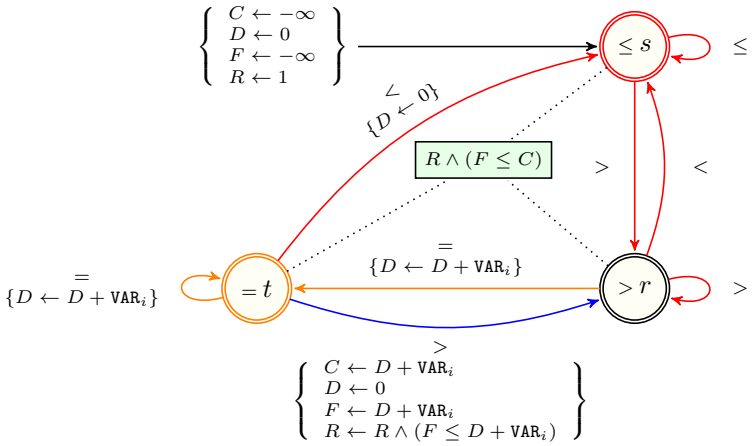
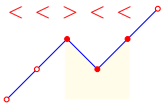


Figure 4.400: Automaton for the INCREASING_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+2$ to index j.

Example $(\langle 1,2,3,0,3,4,6,5,1,4,5,6,1,2,4,4\rangle)$

Figure 4.401 provides an example where the INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE $([1,2,3,0,3,4,6,5,1,4,5,6,1,2,4,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

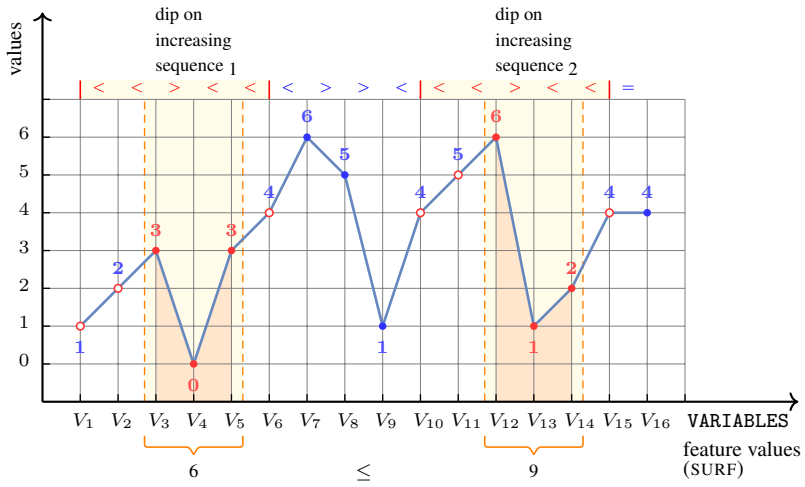


Figure 4.401: Illustrating the INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.402 depicts the automaton associated with the constraint INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE.

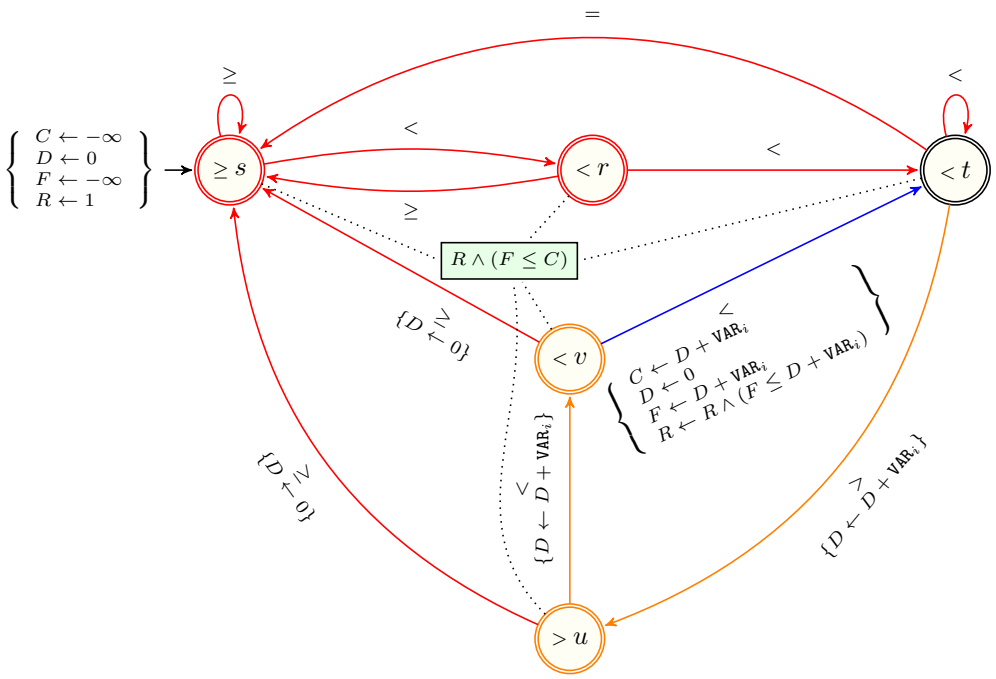
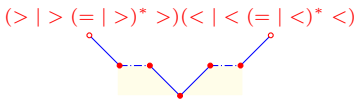


Figure 4.402: Automaton for the INCREASING_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint INCREASING_SURF_GORGE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5 \rangle)$

Figure 4.403 provides an example where the INCREASING_SURF_GORGE $([6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

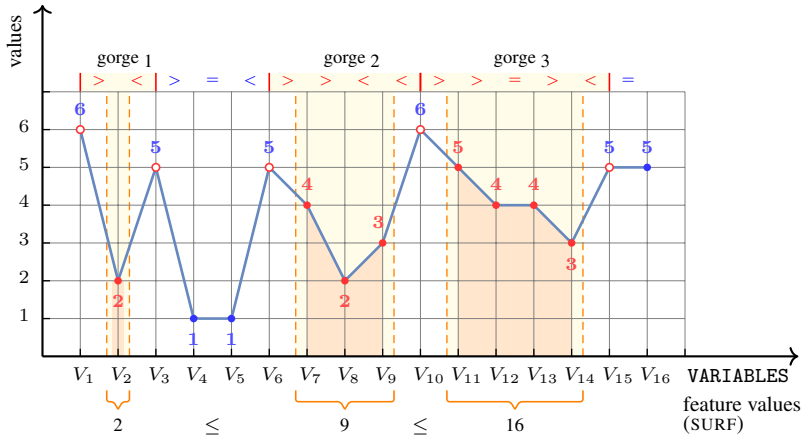


Figure 4.403: Illustrating the INCREASING_SURF_GORGE constraint of the **Example** slot

Automaton Figure 4.404 depicts the automaton associated with the constraint INCREASING_SURF_GORGE.

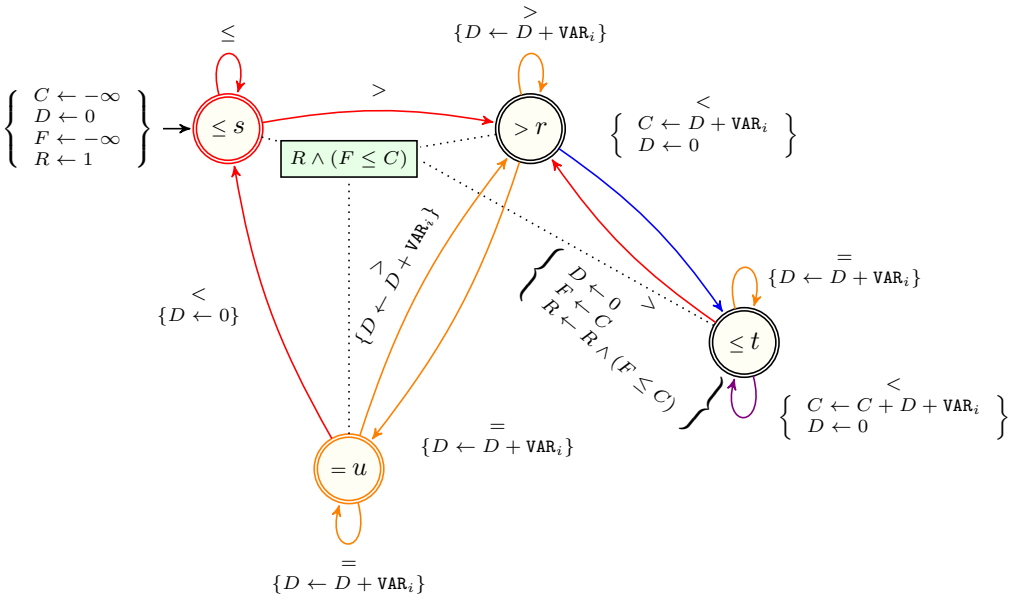


Figure 4.404: Automaton for the INCREASING_SURF_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint INCREASING_SURF_INCREASING(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 6,1,2,3,5,5,5,4,5,6,4,3,3,2,1,1\rangle)$

Figure 4.405 provides an example where the INCREASING_SURF_INCREASING $([6,1,2,3,5,5,5,4,5,6,4,3,3,2,1,1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

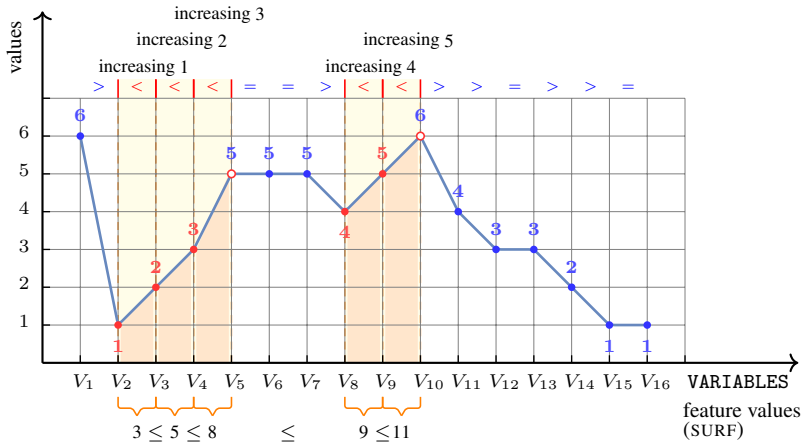


Figure 4.405: Illustrating the INCREASING_SURF_INCREASING constraint of the **Example** slot

Automaton

Figure 4.406 depicts the automaton associated with the constraint INCREASING_SURF_INCREASING.

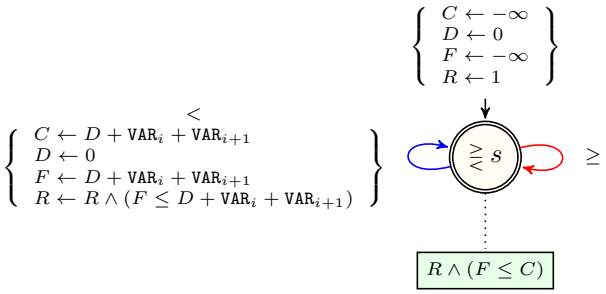
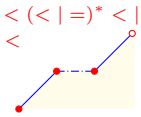


Figure 4.406: Automaton for the INCREASING_SURF_INCREASING constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 3, 3, 4, 5, 5 \rangle)$

Figure 4.407 provides an example where the INCREASING_SURF_INCREASING_SEQUENCE $([6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 3, 3, 4, 5, 5])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

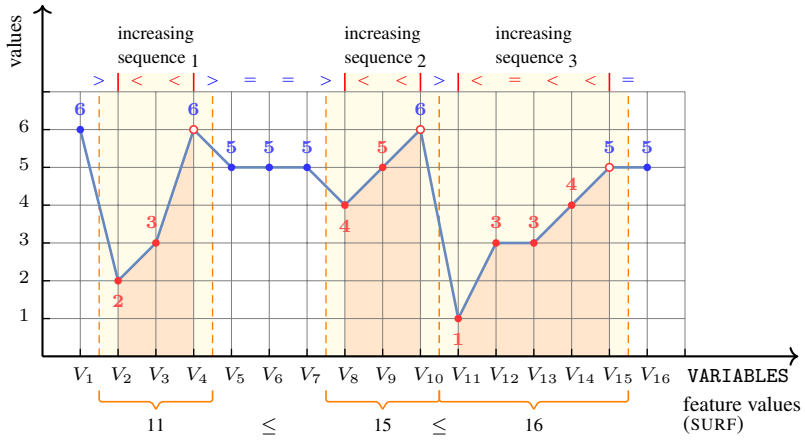


Figure 4.407: Illustrating the INCREASING_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.408 depicts the automaton associated with the constraint INCREASING_SURF_INCREASING_SEQUENCE.

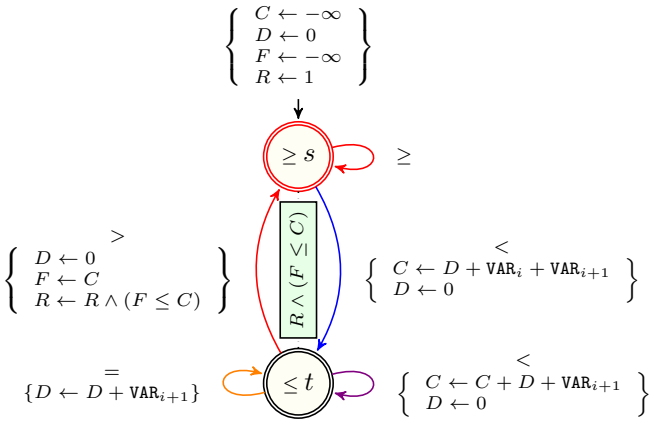
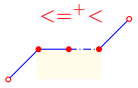


Figure 4.408: Automaton for the INCREASING_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint INCREASING_SURF_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1,2,2,4,5,5,6,6,4,4,2,5,5,5,7,7\rangle)$

Figure 4.409 provides an example where the INCREASING_SURF_INCREASING_TERRACE $([1,2,2,4,5,5,6,6,4,4,2,5,5,5,7,7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

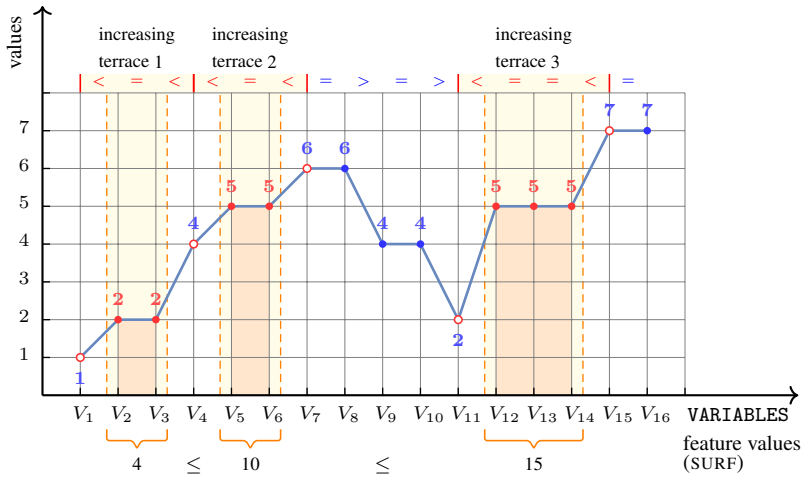


Figure 4.409: Illustrating the INCREASING_SURF_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.410 depicts the automaton associated with the constraint INCREASING_SURF_INCREASING_TERRACE.

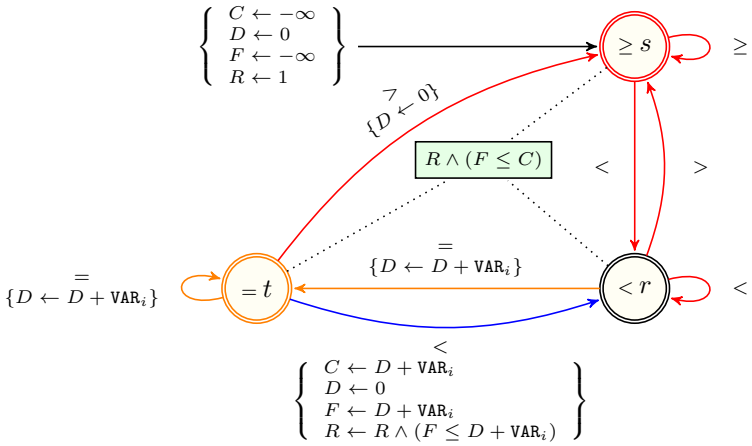
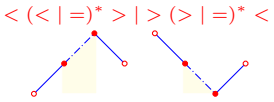


Figure 4.410: Automaton for the INCREASING_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INFLEXION pattern.

Constraint INCREASING_SURF_INFLEXION(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1,1,2,3,3,2,2,2,2,2,1,6,6,5,5,5\rangle)$

Figure 4.411 provides an example where the INCREASING_SURF_INFLEXION $([1,1,2,3,3,2,2,2,2,2,1,6,6,5,5,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

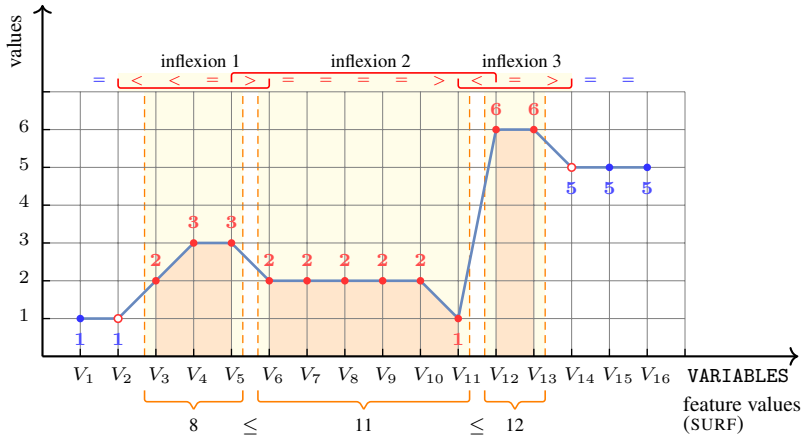


Figure 4.411: Illustrating the INCREASING_SURF_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.412 depicts the automaton associated with the constraint INCREASING_SURF_INFLEXION.

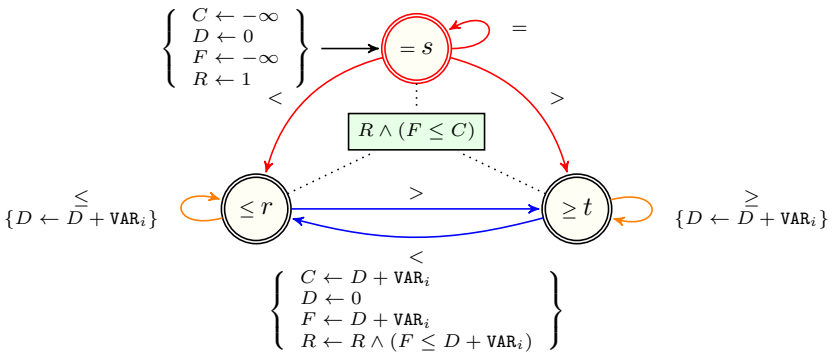
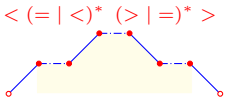


Figure 4.412: Automaton for the INCREASING_SURF_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

INCREASING_SURF_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint INCREASING_SURF_PEAK(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1\rangle)$

Figure 4.413 provides an example where the INCREASING_SURF_PEAK $([7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

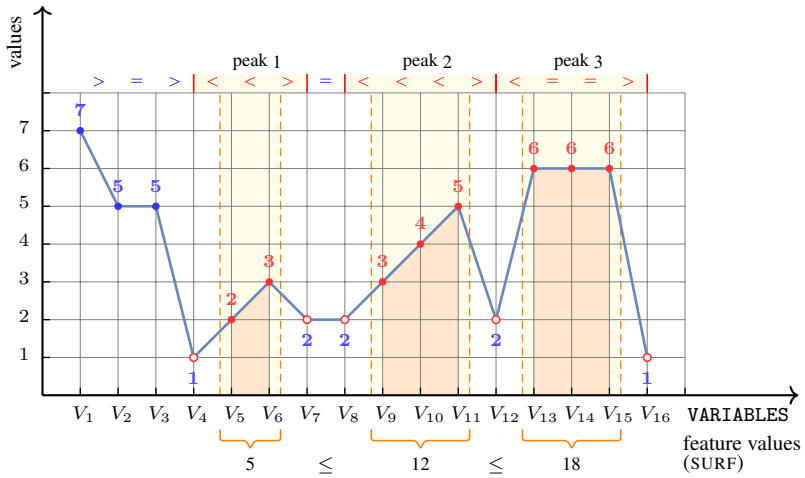


Figure 4.413: Illustrating the INCREASING_SURF_PEAK constraint of the **Example** slot

Automaton Figure 4.414 depicts the automaton associated with the constraint INCREASING_SURF_PEAK.

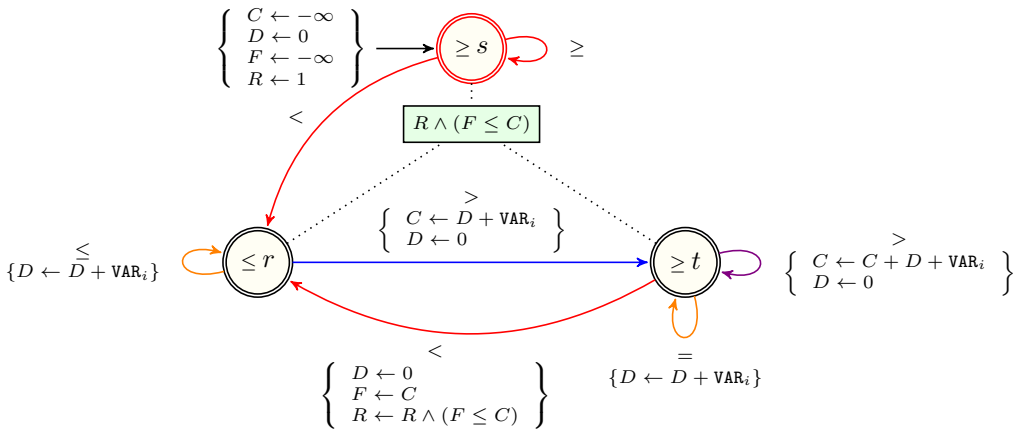


Figure 4.414: Automaton for the INCREASING_SURF_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint INCREASING_SURF_PLAIN(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2\rangle)$

Figure 4.415 provides an example where the INCREASING_SURF_PLAIN $([3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

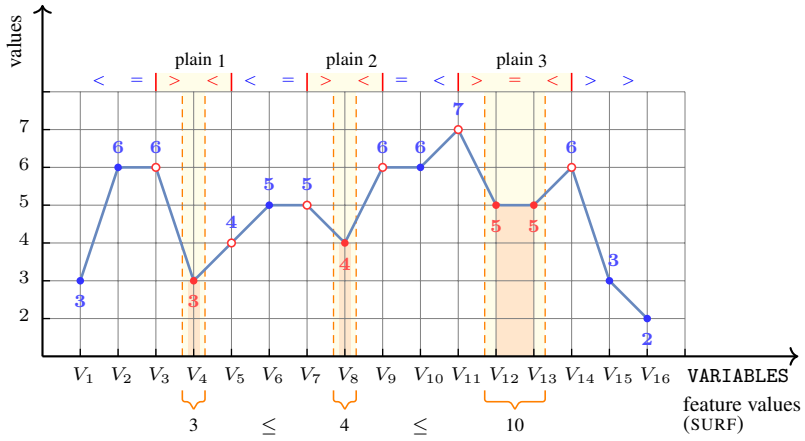


Figure 4.415: Illustrating the INCREASING_SURF_PLAIN constraint of the **Example** slot

Automaton

Figure 4.416 depicts the automaton associated with the constraint INCREASING_SURF_PLAIN.

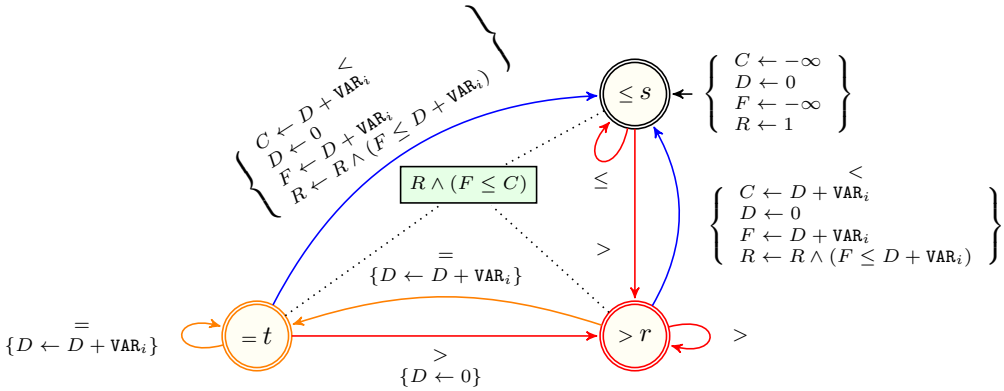


Figure 4.416: Automaton for the INCREASING_SURF_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLATEAU pattern.

Constraint INCREASING_SURF_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle)$

Figure 4.417 provides an example where the INCREASING_SURF_PLATEAU $([7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

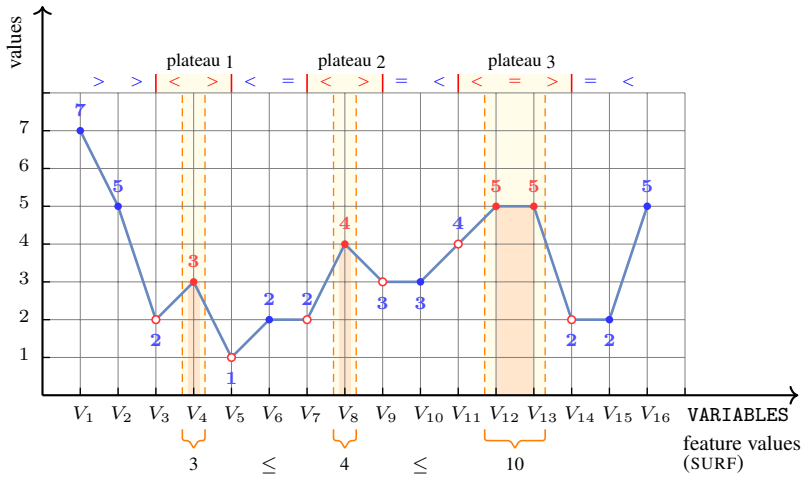


Figure 4.417: Illustrating the INCREASING_SURF_PLATEAU constraint of the **Example** slot

Automaton Figure 4.418 depicts the automaton associated with the constraint INCREASING_SURF_PLATEAU.

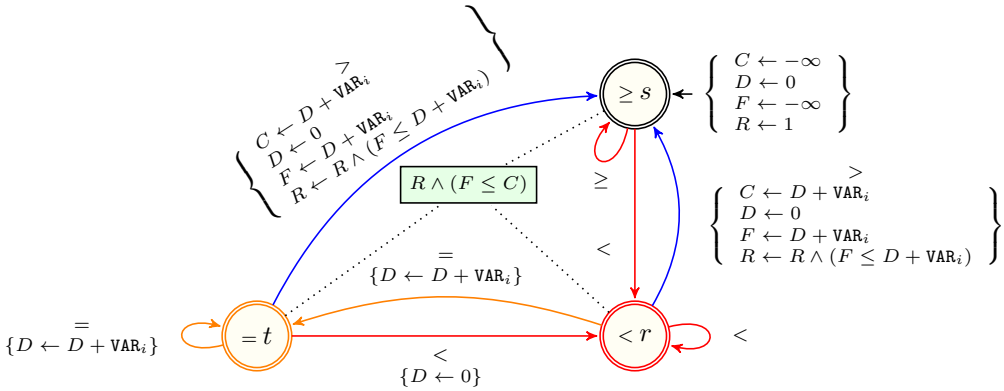
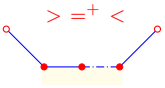


Figure 4.418: Automaton for the INCREASING_SURF_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint INCREASING_SURF_PROPER_PLAIN(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2\rangle)$

Figure 4.419 provides an example where the INCREASING_SURF_PROPER_PLAIN $([5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

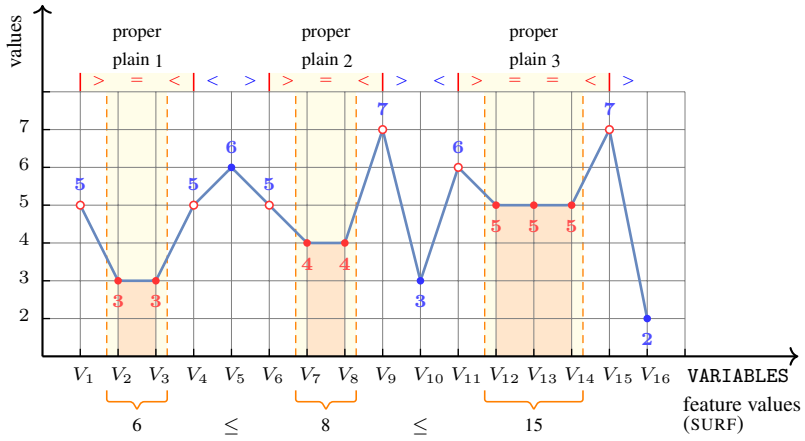


Figure 4.419: Illustrating the INCREASING_SURF_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.420 depicts the automaton associated with the constraint INCREASING_SURF_PROPER_PLAIN.

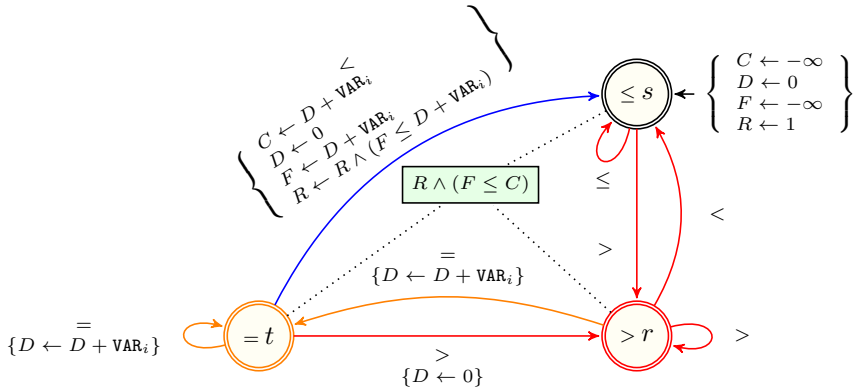
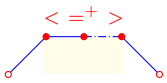


Figure 4.420: Automaton for the INCREASING_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint INCREASING_SURF_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3\rangle)$

Figure 4.421 provides an example where the INCREASING_SURF_PROPER_PLATEAU $([7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

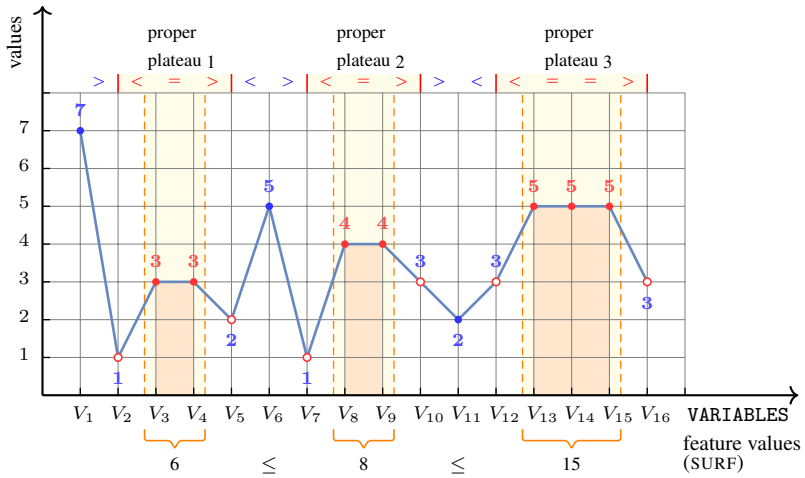


Figure 4.421: Illustrating the INCREASING_SURF_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.422 depicts the automaton associated with the constraint INCREASING_SURF_PROPER_PLATEAU.

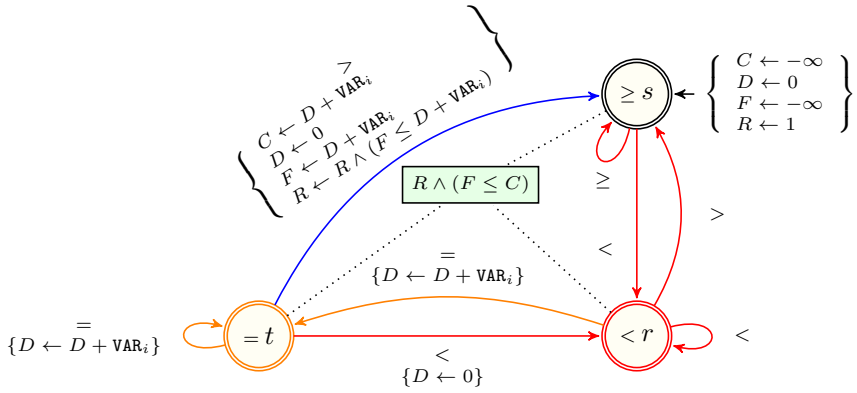


Figure 4.422: Automaton for the INCREASING_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_STEADY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STEADY pattern.

Constraint INCREASING_SURF_STEADY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6\rangle)$

Figure 4.423 provides an example where the INCREASING_SURF_STEADY $([1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

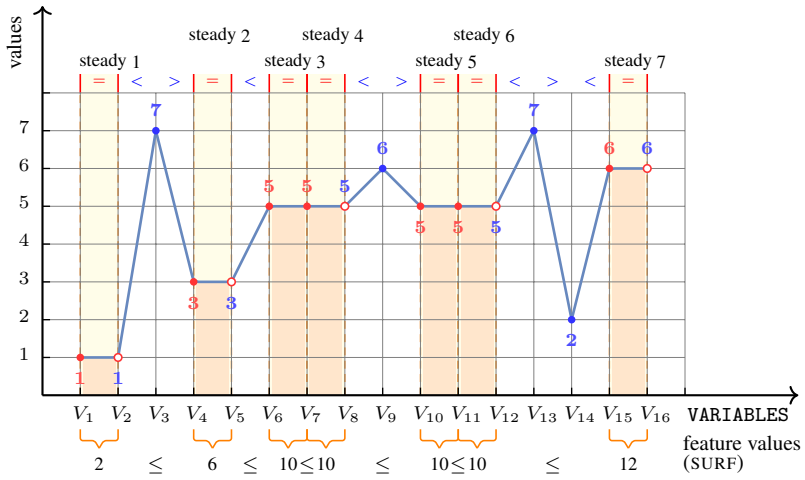


Figure 4.423: Illustrating the INCREASING_SURF_STEADY constraint of the **Example** slot

Automaton

Figure 4.424 depicts the automaton associated with the constraint INCREASING_SURF_STEADY.

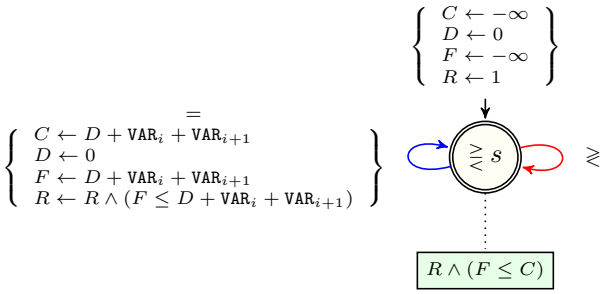
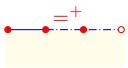


Figure 4.424: Automaton for the INCREASING_SURF_STEADY constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint INCREASING_SURF_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 6,2,2,3,5,6,4,4,2,3,5,5,5,2,3,4\rangle)$

Figure 4.425 provides an example where the INCREASING_SURF_STEADY_SEQUENCE $([6,2,2,3,5,6,4,4,2,3,5,5,5,2,3,4])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$

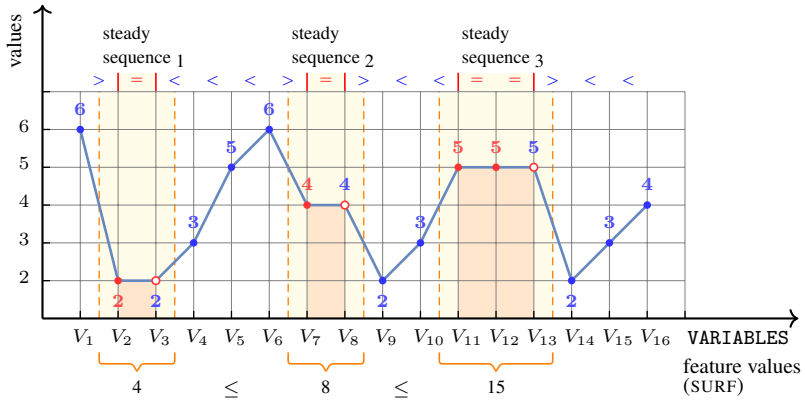


Figure 4.425: Illustrating the INCREASING_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.426 depicts the automaton associated with the constraint INCREASING_SURF_STEADY_SEQUENCE.

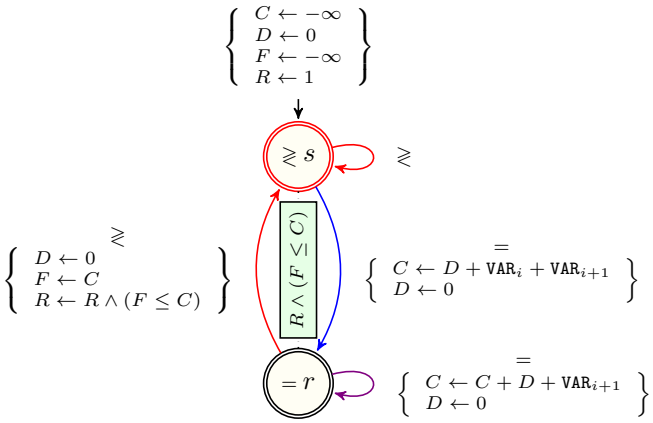


Figure 4.426: Automaton for the INCREASING_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(\langle 1,4,3,2,7,3,3,7,3,3,3,4,4,5,7,5\rangle)$

Figure 4.427 provides an example where the INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE $([1,4,3,2,7,3,3,7,3,3,3,4,4,5,7,5])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

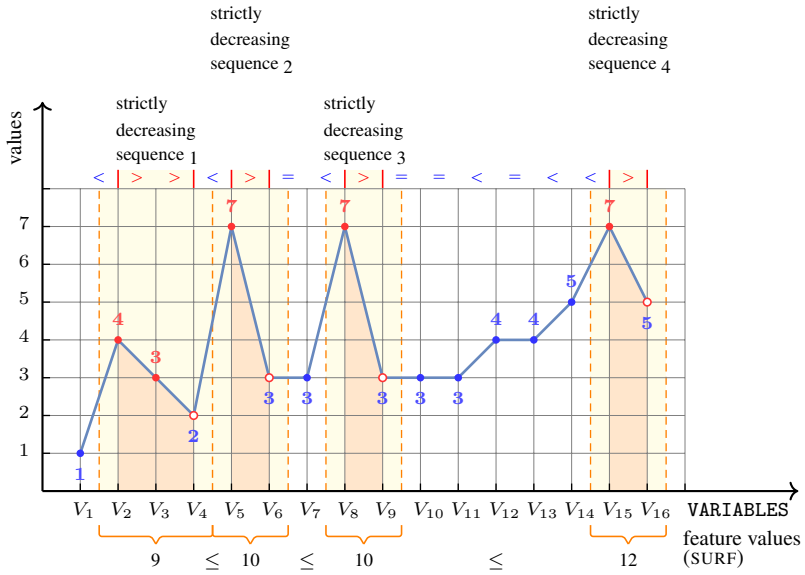


Figure 4.427: Illustrating the INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.428 depicts the automaton associated with the constraint INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE.

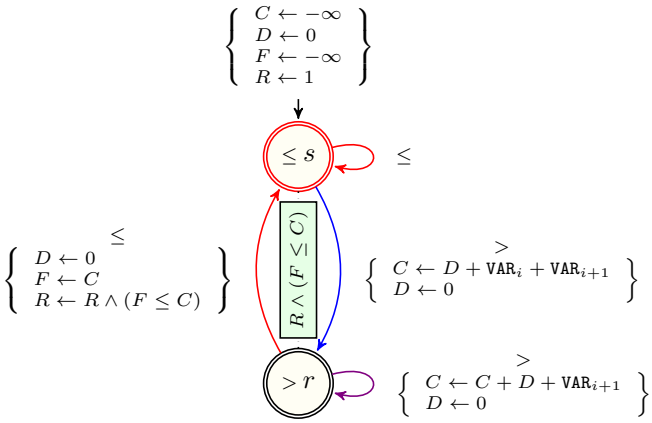


Figure 4.428: Automaton for the INCREASING_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(\langle 6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 2, 3, 4, 5, 5\rangle)$

Figure 4.429 provides an example where the INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE $([6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 2, 3, 4, 5, 5])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

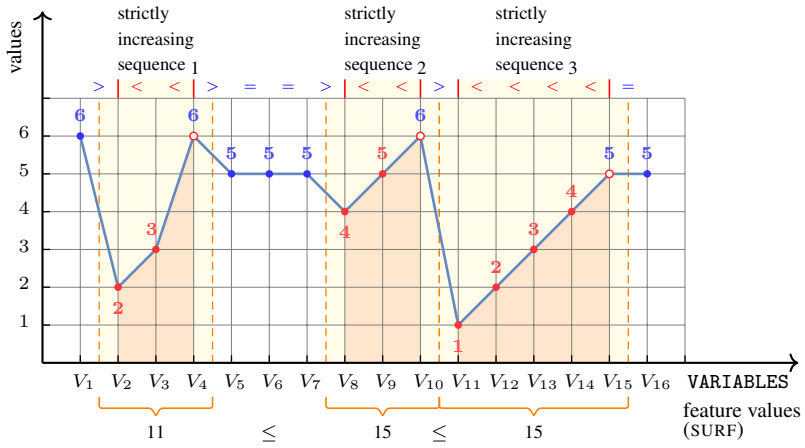


Figure 4.429: Illustrating the INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.430 depicts the automaton associated with the constraint INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE.

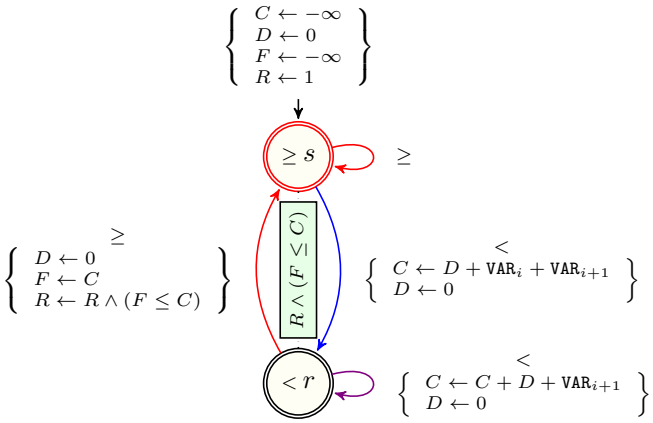
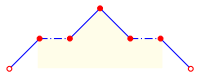


Figure 4.430: Automaton for the INCREASING_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint INCREASING_SURF_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2 \rangle)$

Figure 4.431 provides an example where the INCREASING_SURF_SUMMIT $([1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

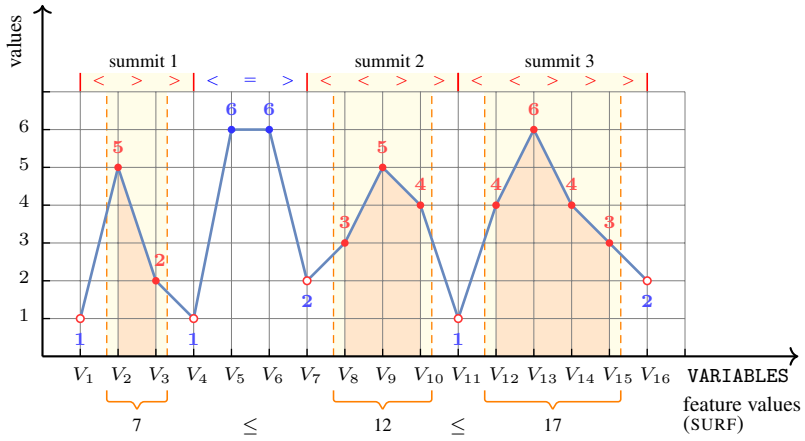


Figure 4.431: Illustrating the INCREASING_SURF_SUMMIT constraint of the **Example** slot

Automaton Figure 4.432 depicts the automaton associated with the constraint INCREASING_SURF_SUMMIT.

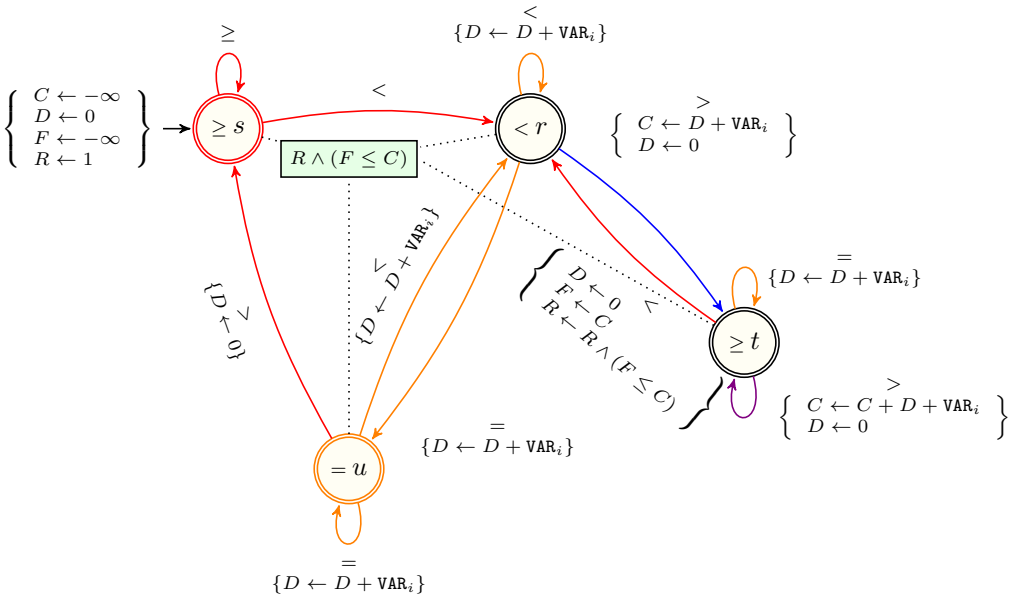
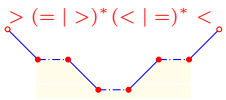


Figure 4.432: Automaton for the INCREASING_SURF_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_SURF_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint INCREASING_SURF_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1 \rangle)$

Figure 4.433 provides an example where the INCREASING_SURF_VALLEY $([7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

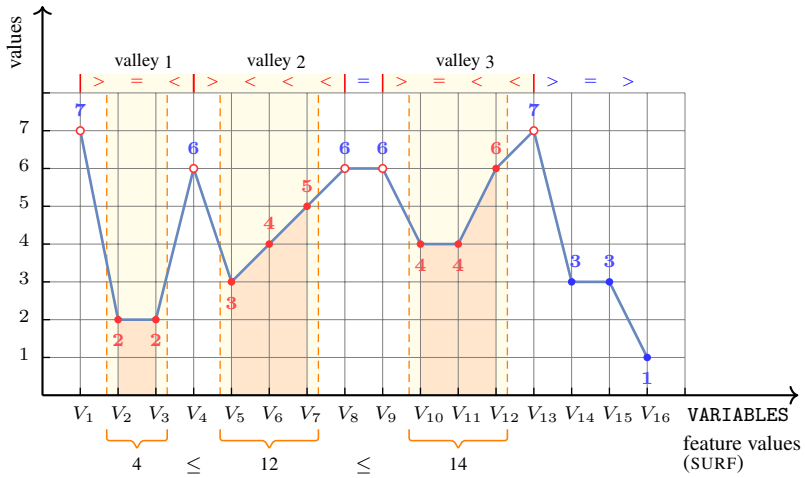


Figure 4.433: Illustrating the INCREASING_SURF_VALLEY constraint of the **Example** slot

Automaton Figure 4.434 depicts the automaton associated with the constraint INCREASING_SURF_VALLEY.

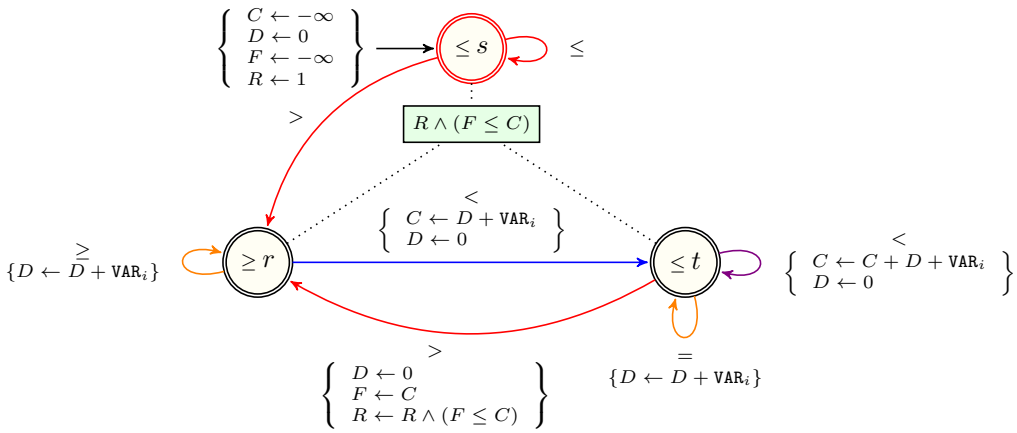
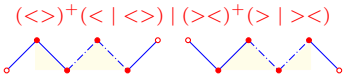


Figure 4.434: Automaton for the INCREASING_SURF_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

INCREASING_SURF_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint INCREASING_SURF_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the surface of each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^{+}(< \mid <>) \mid (><)^{+}(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(\langle 7, 7, 2, 5, 3, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6 \rangle)$

Figure 4.435 provides an example where the INCREASING_SURF_ZIGZAG $([7, 7, 2, 5, 3, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

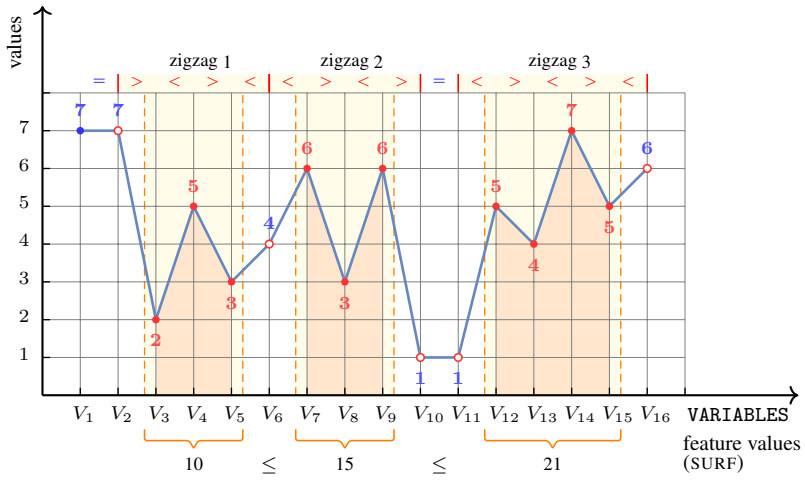


Figure 4.435: Illustrating the INCREASING_SURF_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.436 depicts the automaton associated with the constraint INCREASING_SURF_ZIGZAG.

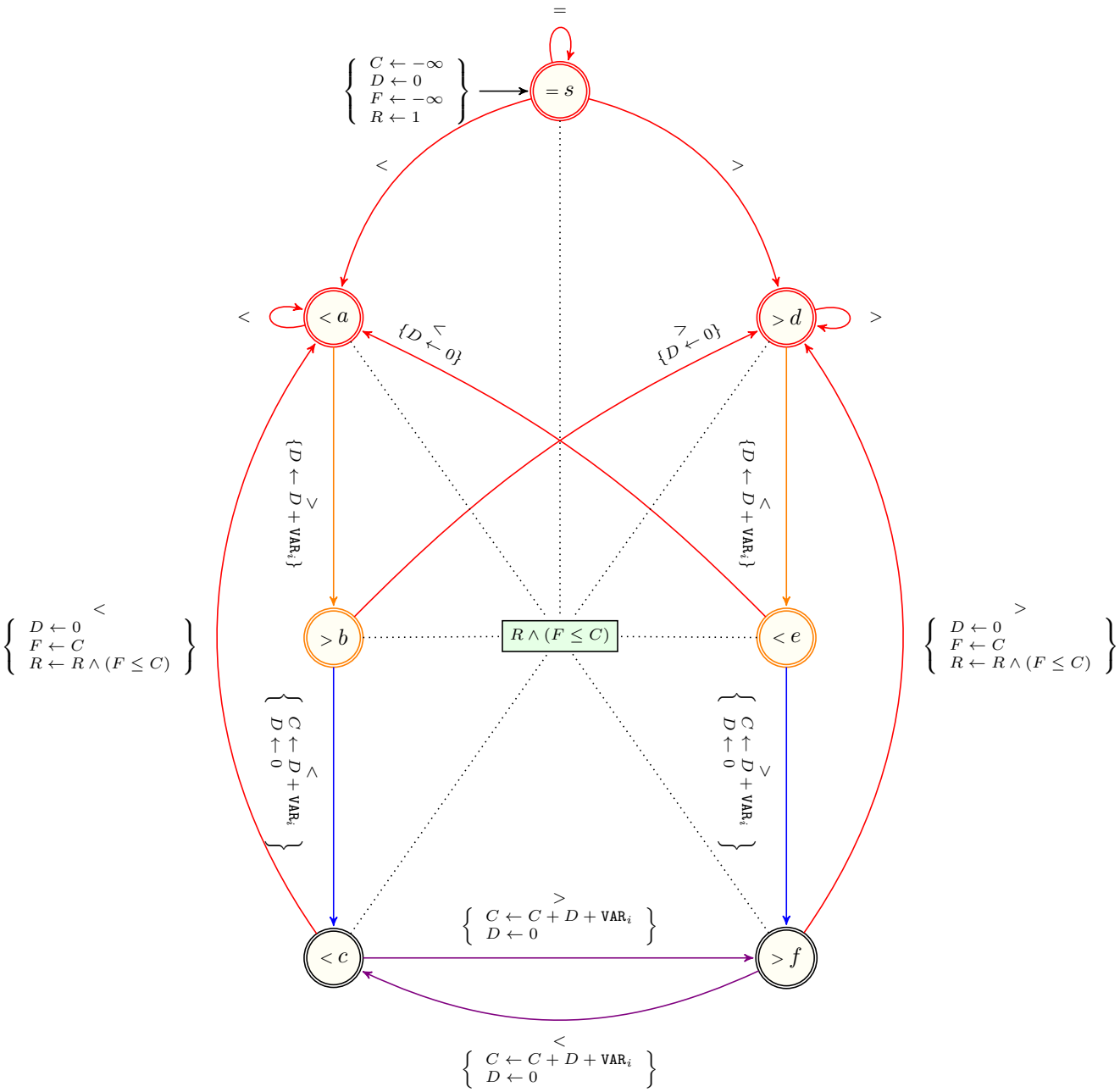
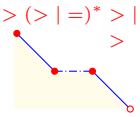


Figure 4.436: Automaton for the INCREASING_SURF_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

CONDITION FEATURE PATTERN

INCREASING_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INCREASING_WIDTH_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 1, 6, 6, 4, 4, 5, 6, 6, 5, 2, 3, 3, 2, 2, 1, 1\rangle)$

Figure 4.437 provides an example where the INCREASING_WIDTH_DECREASING_SEQUENCE $([1, 6, 6, 4, 4, 5, 6, 6, 5, 2, 3, 3, 2, 2, 1, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

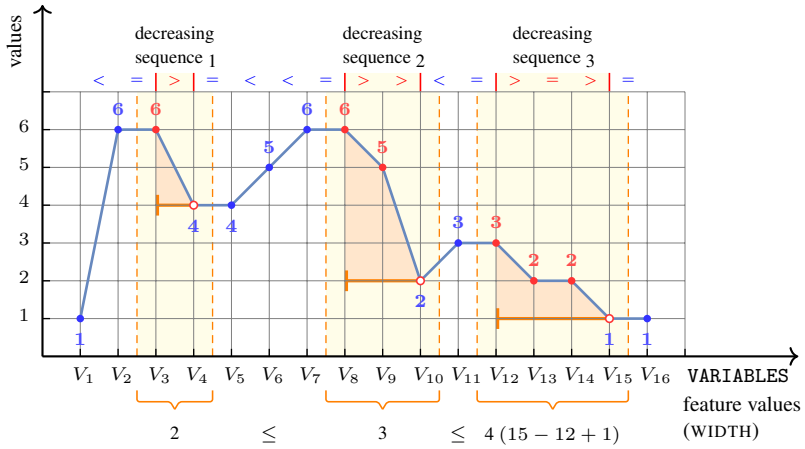


Figure 4.437: Illustrating the INCREASING_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.438 depicts the automaton associated with the constraint INCREASING_WIDTH_DECREASING_SEQUENCE.

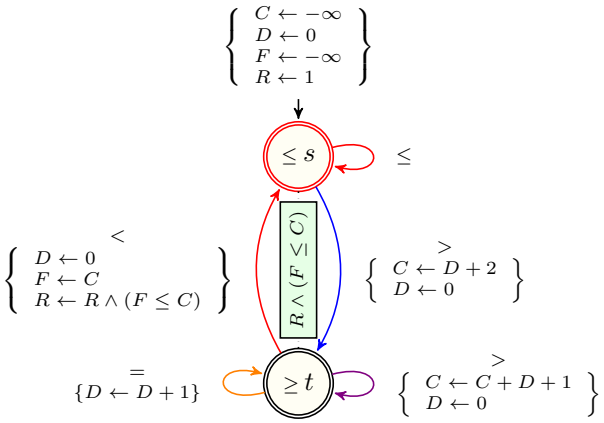
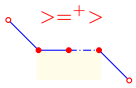


Figure 4.438: Automaton for the INCREASING_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the DECREASING_TERRACE pattern.

Constraint INCREASING_WIDTH_DECREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 5, 2, 2, 1, 5, 4, 3, 3, 2, 4, 4, 6, 5, 5, 5, 4\rangle)$

Figure 4.439 provides an example where the INCREASING_WIDTH_DECREASING_TERRACE $([5, 2, 2, 1, 5, 4, 3, 3, 2, 4, 4, 6, 5, 5, 5, 4])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

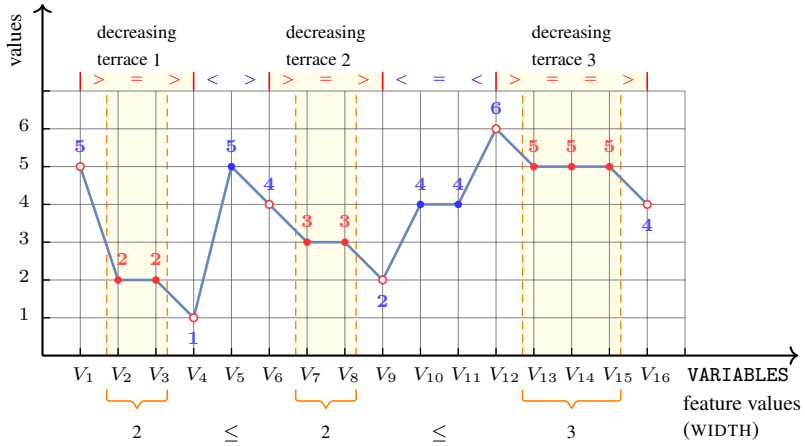


Figure 4.439: Illustrating the INCREASING_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.440 depicts the automaton associated with the constraint INCREASING_WIDTH_DECREASING_TERRACE.

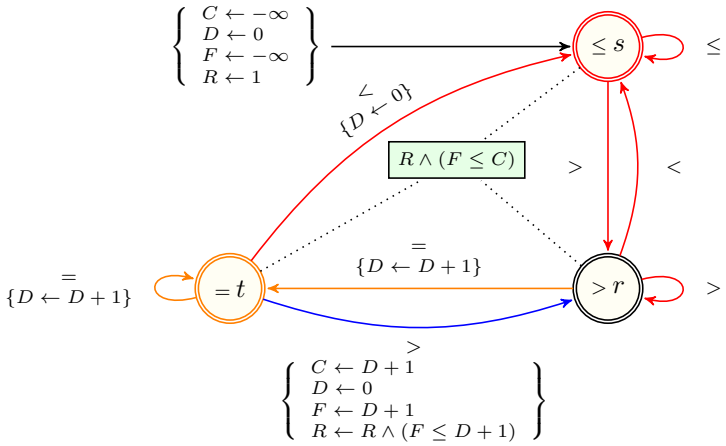
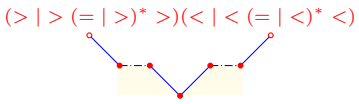


Figure 4.440: Automaton for the INCREASING_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the DECREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint INCREASING_WIDTH_GORGE(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5 \rangle)$

Figure 4.441 provides an example where the INCREASING_WIDTH_GORGE $([6, 2, 5, 1, 1, 5, 4, 2, 3, 6, 5, 4, 4, 3, 5, 5])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

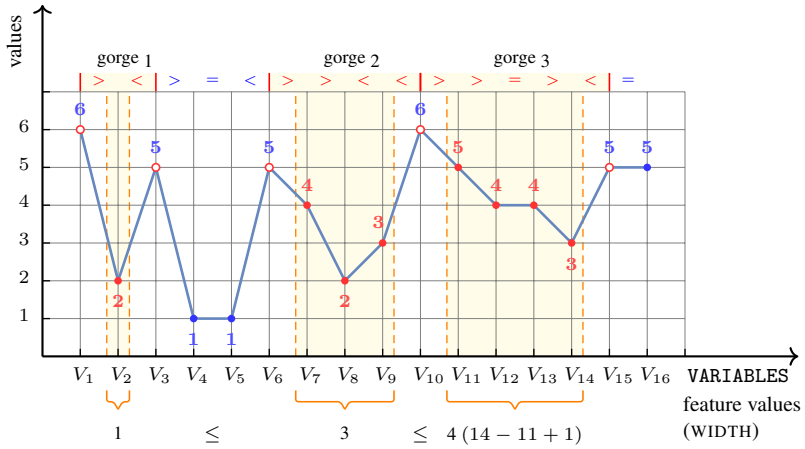


Figure 4.441: Illustrating the INCREASING_WIDTH_GORGE constraint of the **Example** slot

Automaton Figure 4.442 depicts the automaton associated with the constraint INCREASING_WIDTH_GORGE.

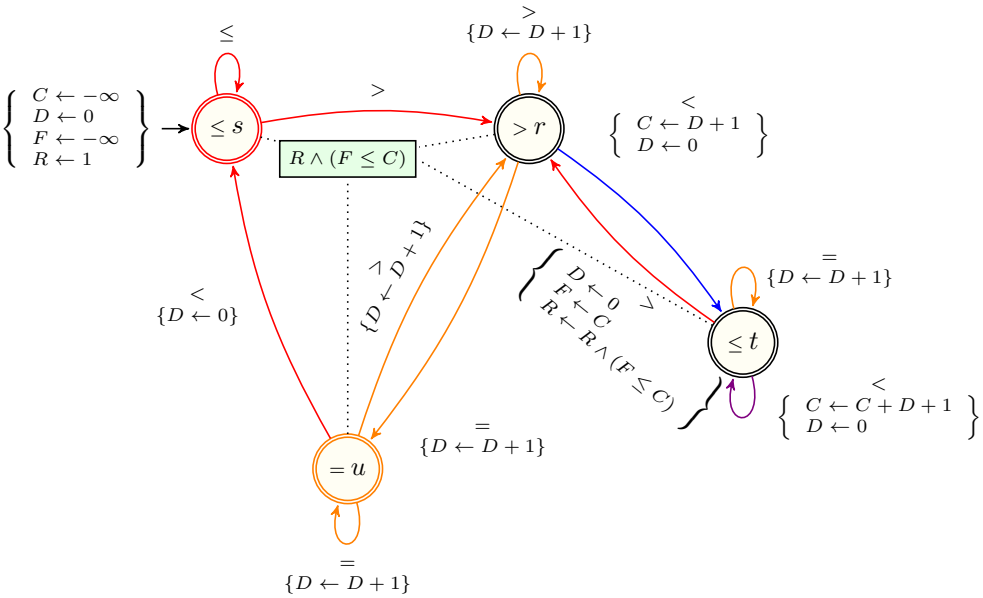
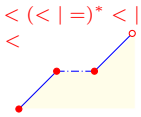


Figure 4.442: Automaton for the INCREASING_WIDTH_GORGE constraint obtained by applying decoration Table 3.38 to the seed transducer of the GORGE pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INCREASING_WIDTH_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 3, 3, 4, 5, 5 \rangle)$

Figure 4.443 provides an example where the INCREASING_WIDTH_INCREASING_SEQUENCE constraint holds. $([6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 3, 3, 4, 5, 5])$

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

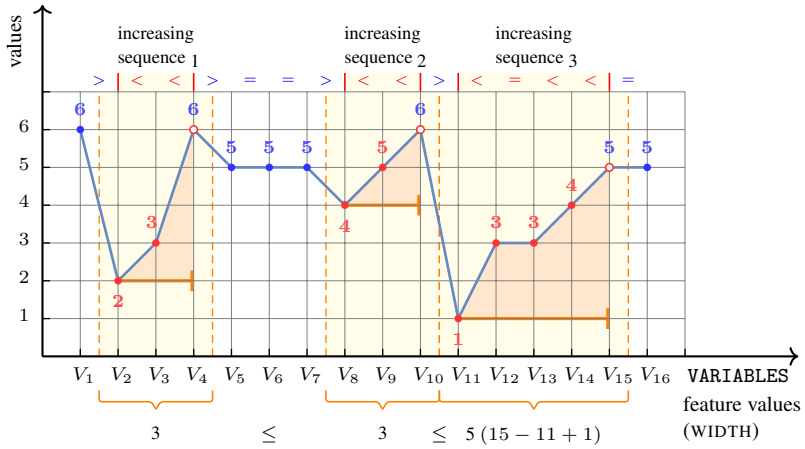


Figure 4.443: Illustrating the INCREASING_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.444 depicts the automaton associated with the constraint INCREASING_WIDTH_INCREASING_SEQUENCE.

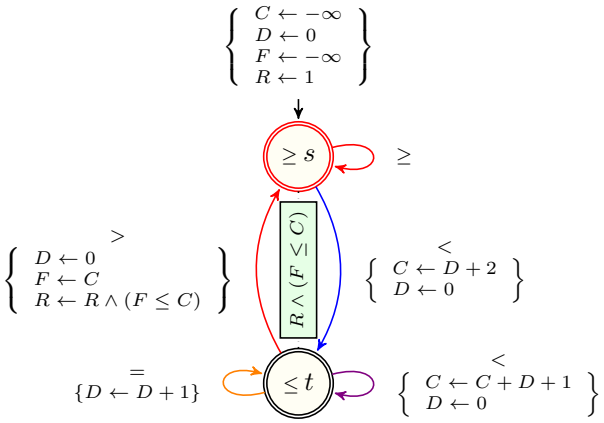
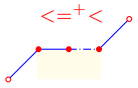


Figure 4.444: Automaton for the INCREASING_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INCREASING_TERRACE pattern.

Constraint INCREASING_WIDTH_INCREASING_TERRACE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 2, 2, 4, 5, 5, 6, 6, 4, 4, 2, 5, 5, 5, 7, 7\rangle)$

Figure 4.445 provides an example where the INCREASING_WIDTH_INCREASING_TERRACE $([1, 2, 2, 4, 5, 5, 6, 6, 4, 4, 2, 5, 5, 5, 7, 7])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 2$

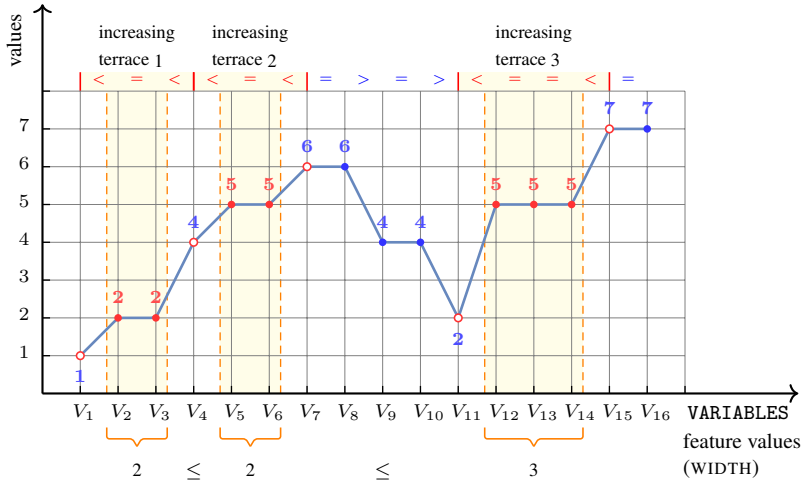


Figure 4.445: Illustrating the INCREASING_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figure 4.446 depicts the automaton associated with the constraint INCREASING_WIDTH_INCREASING_TERRACE.

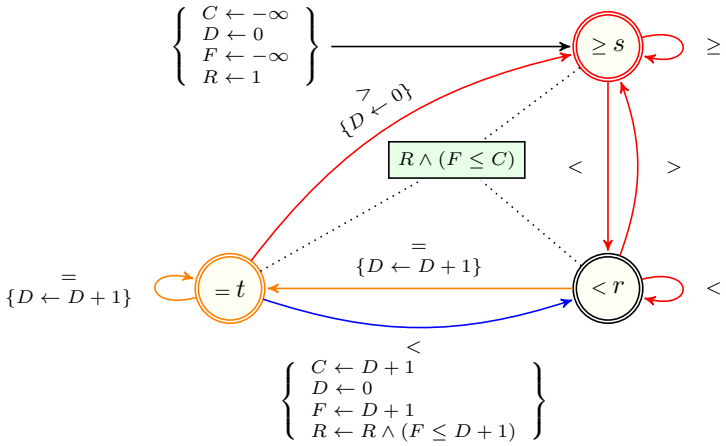
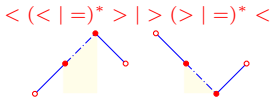


Figure 4.446: Automaton for the INCREASING_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.38 to the seed transducer of the INCREASING_TERRACE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint INCREASING_WIDTH_INFLEXION(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 1, 2, 1, 3, 3, 6, 4, 3, 3, 2, 3, 4, 5, 5, 6 \rangle)$

Figure 4.447 provides an example where the INCREASING_WIDTH_INFLEXION $([1, 1, 2, 1, 3, 3, 6, 4, 3, 3, 2, 3, 4, 5, 5, 6])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

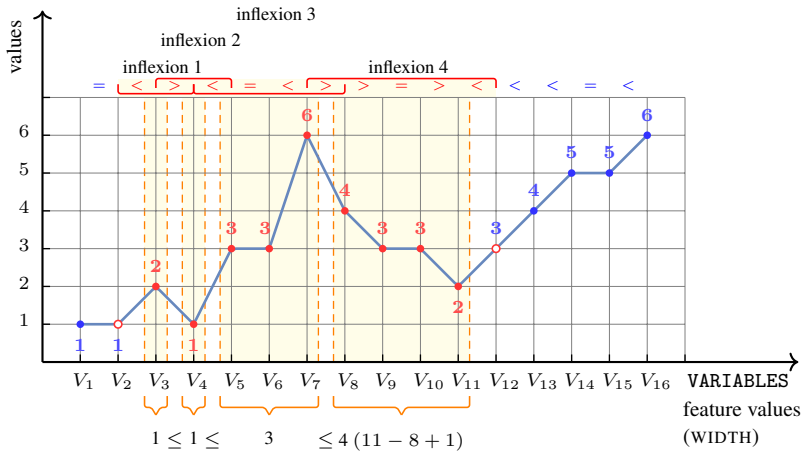


Figure 4.447: Illustrating the INCREASING_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figure 4.448 depicts the automaton associated with the constraint INCREASING_WIDTH_INFLEXION.

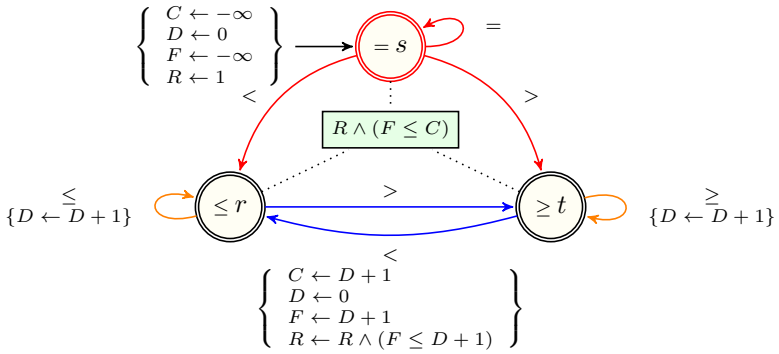
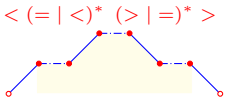


Figure 4.448: Automaton for the INCREASING_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.38 to the seed transducer of the INFLEXION pattern (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

CONDITION FEATURE PATTERN

INCREASING_WIDTH_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint INCREASING_WIDTH_PEAK(VARIABLES)

Argument

VARIABLES	:	collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1\rangle)$

Figure 4.449 provides an example where the INCREASING_WIDTH_PEAK $([7, 5, 5, 1, 2, 3, 2, 2, 3, 4, 5, 2, 6, 6, 6, 1])$ constraint holds.

Typical

$|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

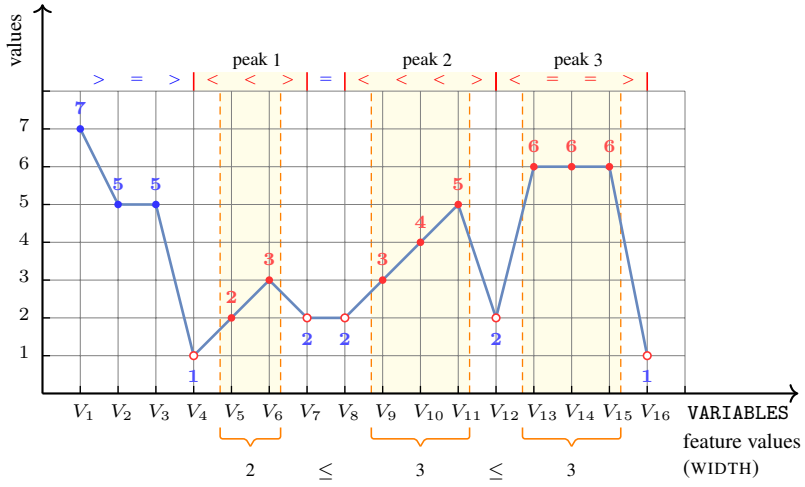


Figure 4.449: Illustrating the INCREASING_WIDTH_PEAK constraint of the **Example** slot

Automaton

Figure 4.450 depicts the automaton associated with the constraint INCREASING_WIDTH_PEAK.

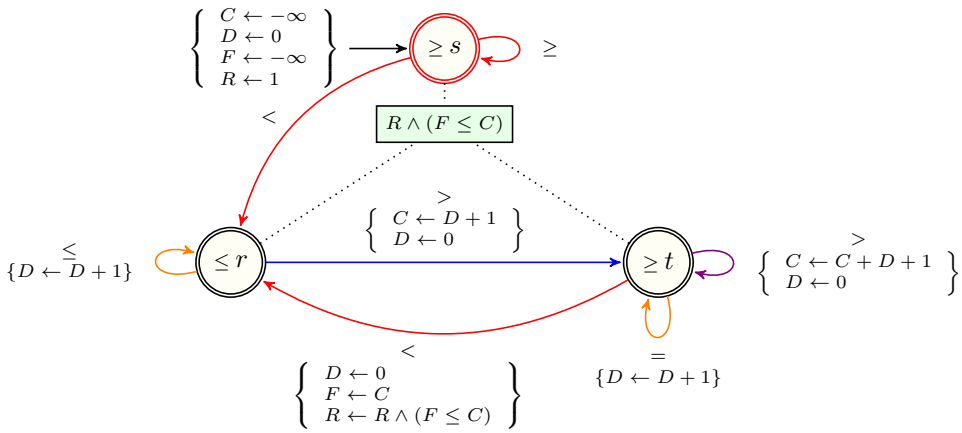


Figure 4.450: Automaton for the INCREASING_WIDTH_PEAK constraint obtained by applying decoration Table 3.38 to the seed transducer of the PEAK pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint INCREASING_WIDTH_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j-i$.

Example $(\langle 3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2\rangle)$

Figure 4.451 provides an example where the INCREASING_WIDTH_PLAIN $([3,6,6,3,4,5,5,4,6,6,7,5,5,6,3,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

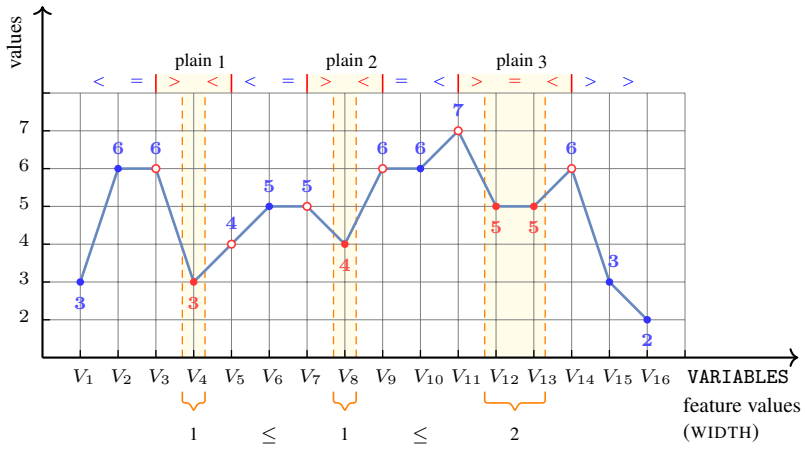


Figure 4.451: Illustrating the INCREASING_WIDTH_PLAIN constraint of the **Example** slot

Automaton Figure 4.452 depicts the automaton associated with the constraint INCREASING_WIDTH_PLAIN.

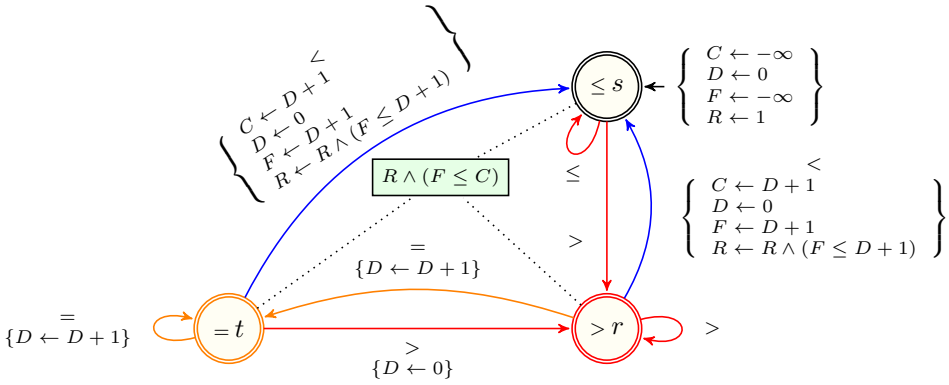


Figure 4.452: Automaton for the INCREASING_WIDTH_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint INCREASING_WIDTH_PLATEAU(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5\rangle)$

Figure 4.453 provides an example where the INCREASING_WIDTH_PLATEAU $([7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

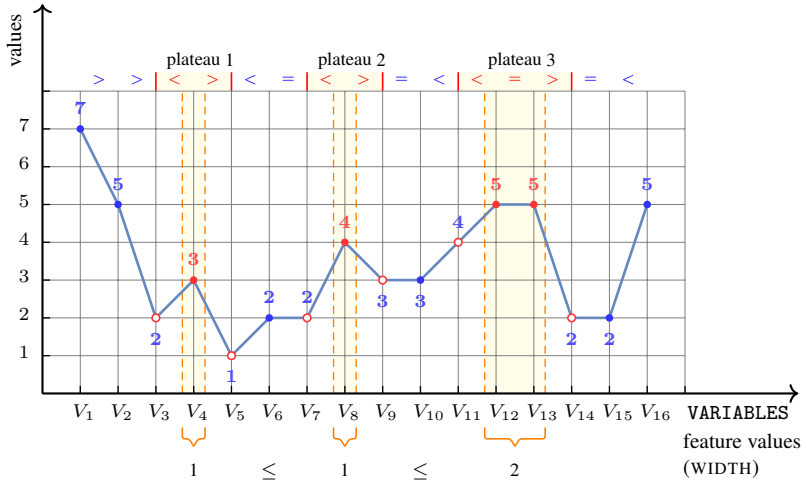


Figure 4.453: Illustrating the INCREASING_WIDTH_PLATEAU constraint of the **Example** slot

Automaton Figure 4.454 depicts the automaton associated with the constraint INCREASING_WIDTH_PLATEAU.

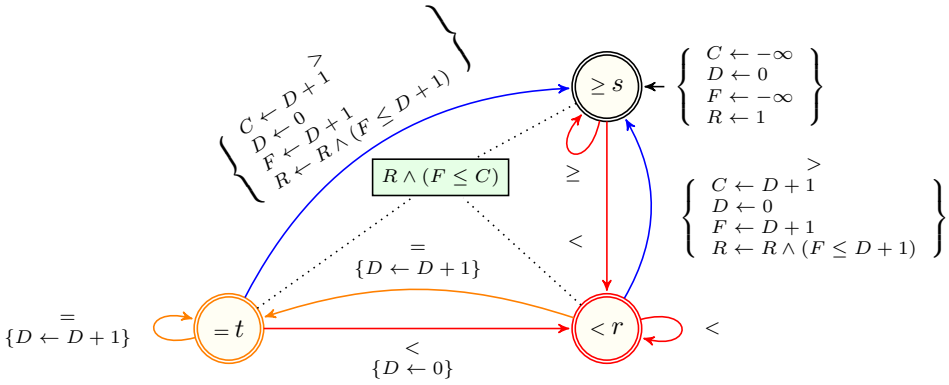
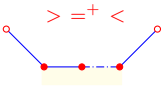


Figure 4.454: Automaton for the INCREASING_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint INCREASING_WIDTH_PROPER_PLAIN(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2\rangle)$

Figure 4.455 provides an example where the INCREASING_WIDTH_PROPER_PLAIN $([5,3,3,5,6,5,4,4,7,3,6,5,5,5,7,2])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

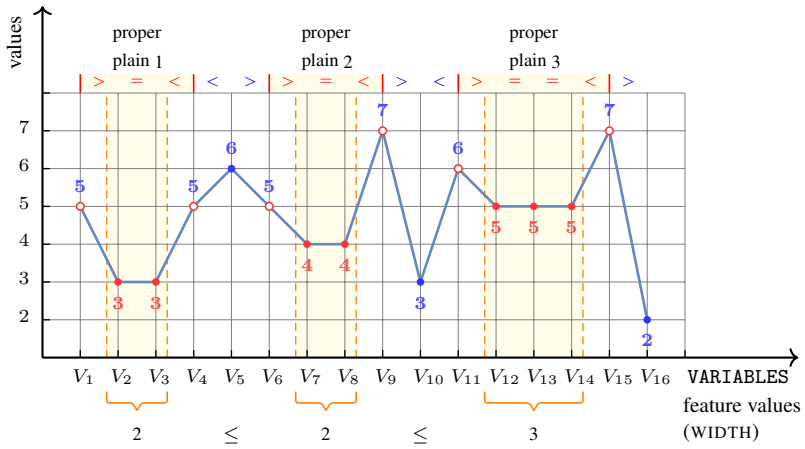


Figure 4.455: Illustrating the INCREASING_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figure 4.456 depicts the automaton associated with the constraint INCREASING_WIDTH_PROPER_PLAIN.

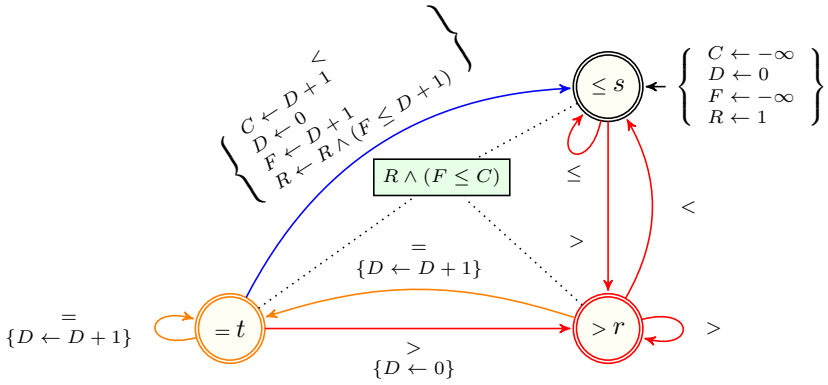
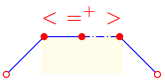


Figure 4.456: Automaton for the INCREASING_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLAIN pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLATEAU pattern.

Constraint INCREASING_WIDTH_PROPER_PLATEAU(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3\rangle)$

Figure 4.457 provides an example where the INCREASING_WIDTH_PROPER_PLATEAU $([7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical $|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

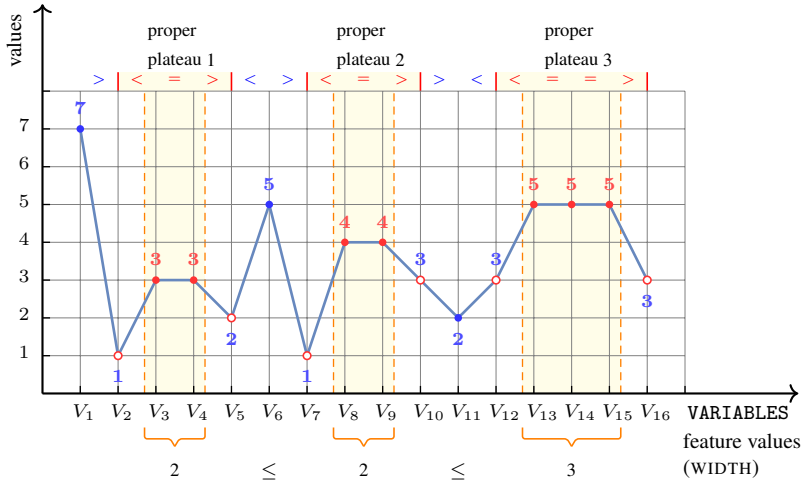


Figure 4.457: Illustrating the INCREASING_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figure 4.458 depicts the automaton associated with the constraint INCREASING_WIDTH_PROPER_PLATEAU.

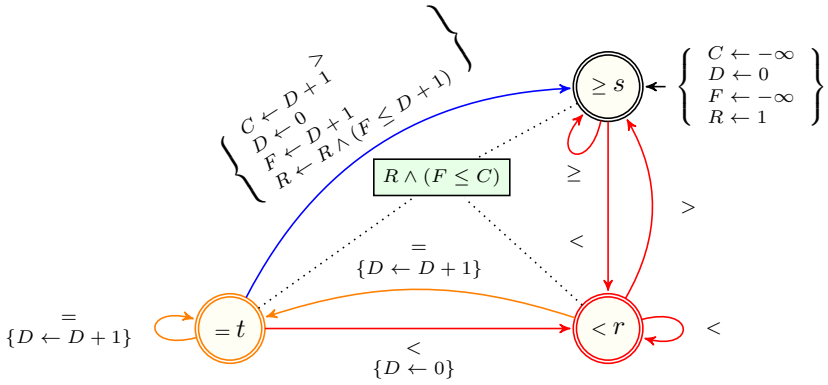
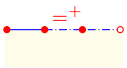


Figure 4.458: Automaton for the INCREASING_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.38 to the seed transducer of the PROPER_PLATEAU pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the STEADY_SEQUENCE pattern.

Constraint INCREASING_WIDTH_STEADY_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 6, 2, 2, 3, 5, 6, 4, 4, 2, 3, 5, 5, 5, 2, 3, 4\rangle)$

Figure 4.459 provides an example where the INCREASING_WIDTH_STEADY_SEQUENCE $([6, 2, 2, 3, 5, 6, 4, 4, 2, 3, 5, 5, 5, 2, 3, 4])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

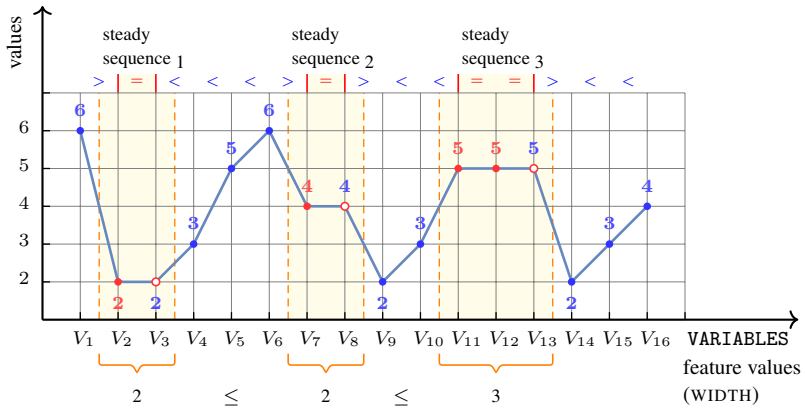


Figure 4.459: Illustrating the INCREASING_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.460 depicts the automaton associated with the constraint INCREASING_WIDTH_STEADY_SEQUENCE.

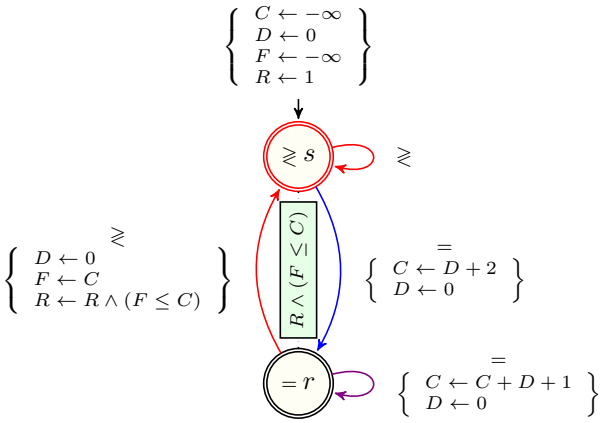


Figure 4.460: Automaton for the INCREASING_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STEADY_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^{+}$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 5, 7, 4, 4, 2, 3, 4, 3, 2, 2, 2, 4, 5, 6, 3, 2 \rangle)$

Figure 4.461 provides an example where the INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE $([5, 7, 4, 4, 2, 3, 4, 3, 2, 2, 2, 4, 5, 6, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

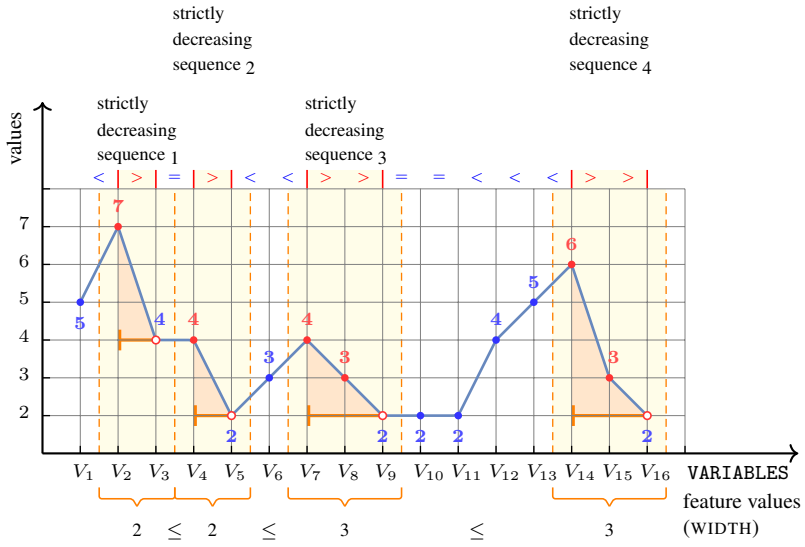


Figure 4.461: Illustrating the INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.462 depicts the automaton associated with the constraint INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE.

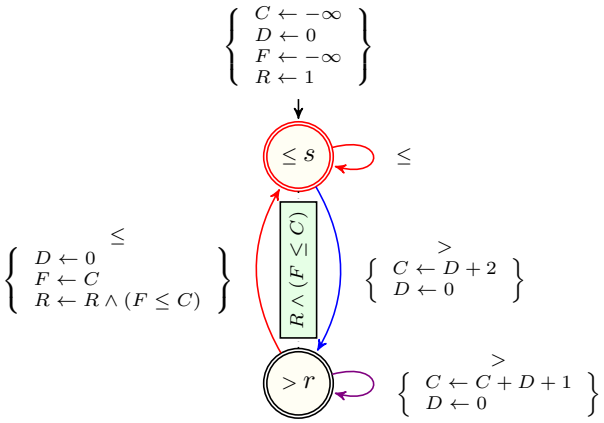


Figure 4.462: Automaton for the INCREASING_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(\langle 6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 2, 3, 4, 5, 5 \rangle)$

Figure 4.463 provides an example where the INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE $([6, 2, 3, 6, 5, 5, 5, 4, 5, 6, 1, 2, 3, 4, 5, 5])$ constraint holds.

Typical $|\text{VARIABLES}| > 1$
$\text{range}(\text{VARIABLES.var}) > 1$

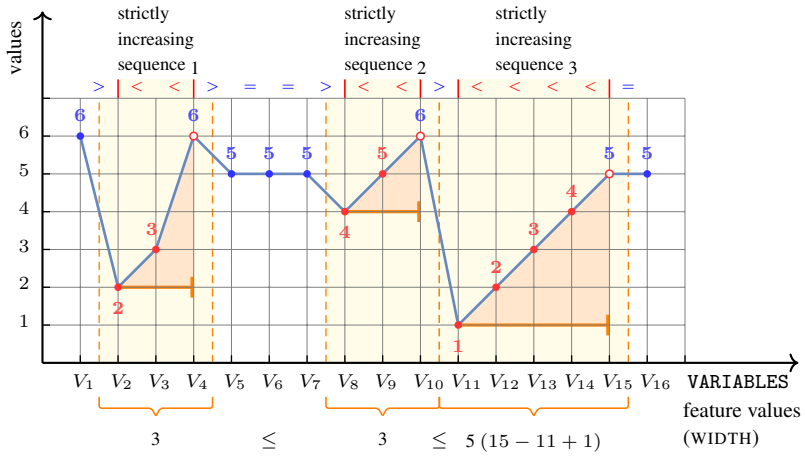


Figure 4.463: Illustrating the INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figure 4.464 depicts the automaton associated with the constraint INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE.

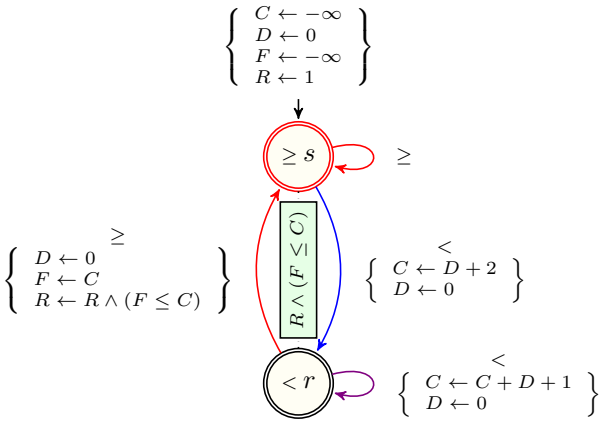
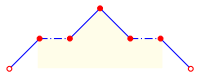


Figure 4.464: Automaton for the INCREASING_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.38 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint INCREASING_WIDTH_SUMMIT(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2 \rangle)$

Figure 4.465 provides an example where the INCREASING_WIDTH_SUMMIT $([1, 5, 2, 1, 6, 6, 2, 3, 5, 4, 1, 4, 6, 4, 3, 2])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

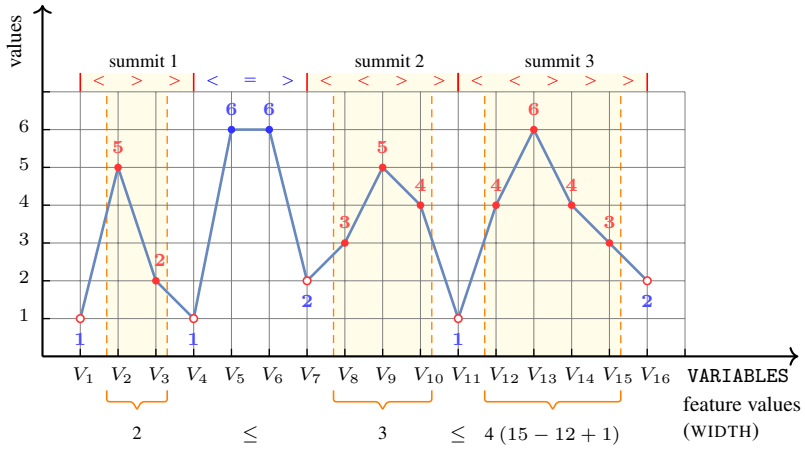


Figure 4.465: Illustrating the INCREASING_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figure 4.466 depicts the automaton associated with the constraint INCREASING_WIDTH_SUMMIT.

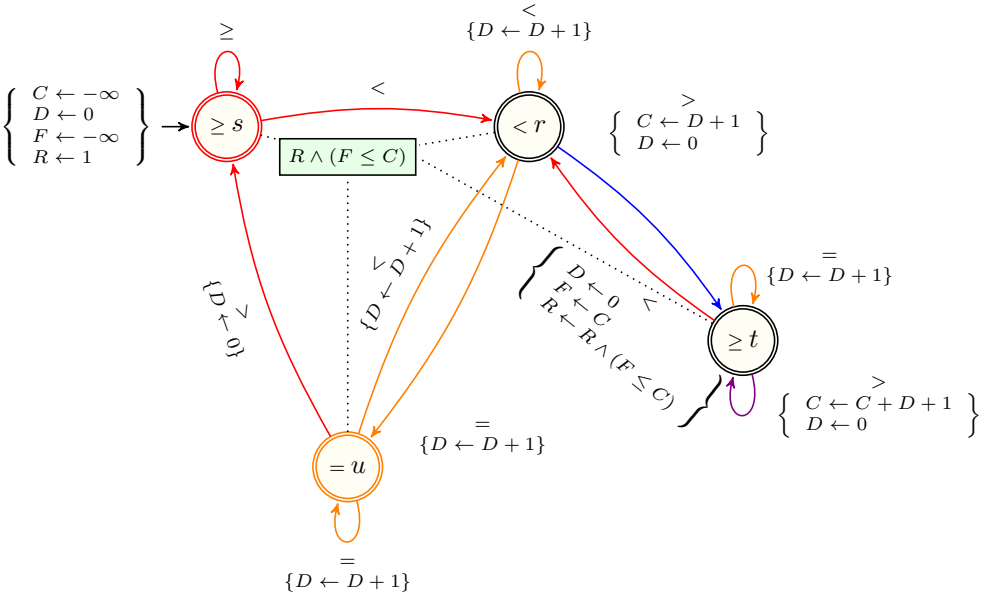
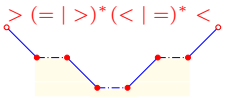


Figure 4.466: Automaton for the INCREASING_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.38 to the seed transducer of the SUMMIT pattern (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

CONDITION FEATURE PATTERN

INCREASING_WIDTH_VALLEY



Origin Based on the VALLEY pattern.

Constraint INCREASING_WIDTH_VALLEY(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1 \rangle)$

Figure 4.467 provides an example where the INCREASING_WIDTH_VALLEY $([7, 2, 2, 6, 3, 4, 5, 6, 6, 4, 4, 6, 7, 3, 3, 1])$ constraint holds.

Typical $|\text{VARIABLES}| > 2$
$\text{range}(\text{VARIABLES.var}) > 1$

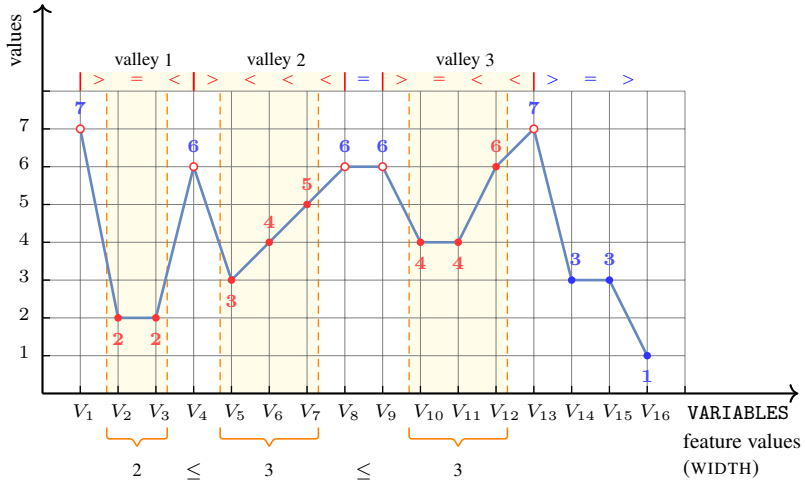


Figure 4.467: Illustrating the INCREASING_WIDTH_VALLEY constraint of the **Example** slot

Automaton

Figure 4.468 depicts the automaton associated with the constraint INCREASING_WIDTH_VALLEY.

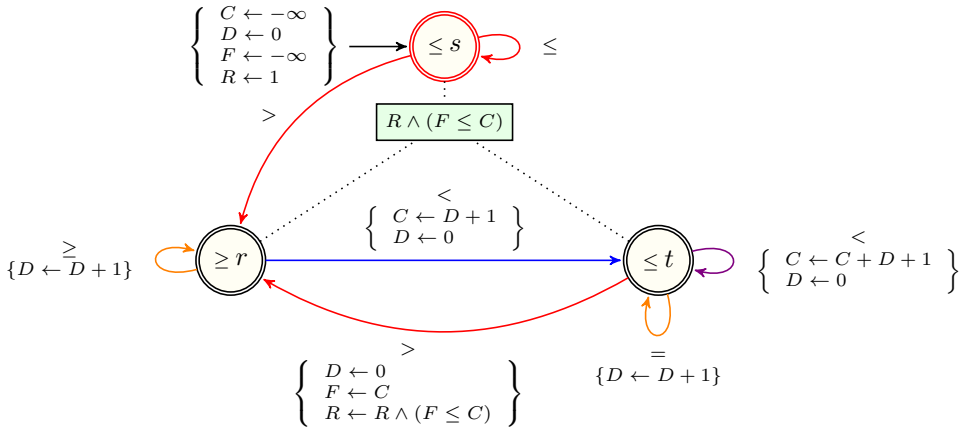
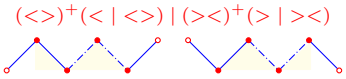


Figure 4.468: Automaton for the INCREASING_WIDTH_VALLEY constraint obtained by applying decoration Table 3.38 to the seed transducer of the VALLEY pattern

CONDITION FEATURE PATTERN

INCREASING_WIDTH_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint INCREASING_WIDTH_ZIGZAG(VARIABLES)

Argument

VARIABLES	:	collection	(var−dvar)

Restriction required(VARIABLES, var)

Purpose Succeeds if the values denoting the width of each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection are increasing.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^{+}(< \mid <>) \mid (><)^{+}(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(\langle 7, 7, 2, 5, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6 \rangle)$

Figure 4.469 provides an example where the INCREASING_WIDTH_ZIGZAG $([7, 7, 2, 5, 3, 4, 6, 3, 6, 1, 1, 5, 4, 7, 5, 6])$ constraint holds.

Typical

$|\text{VARIABLES}| > 3$
$\text{range}(\text{VARIABLES.var}) > 1$

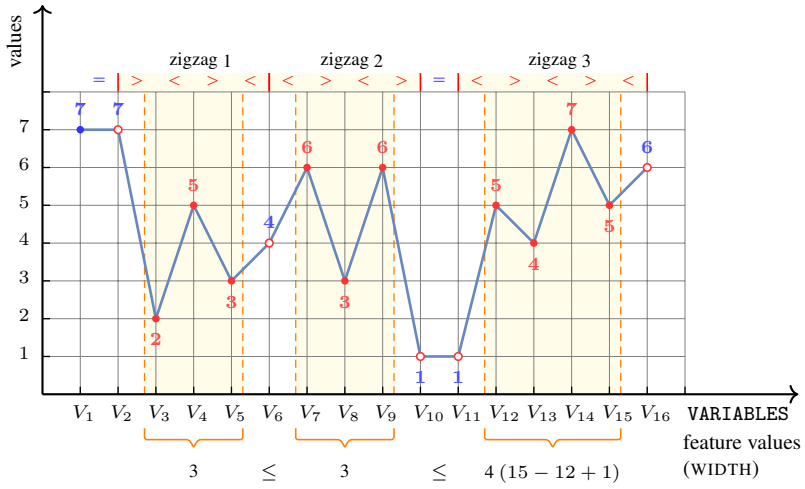


Figure 4.469: Illustrating the INCREASING_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figure 4.470 depicts the automaton associated with the constraint INCREASING_WIDTH_ZIGZAG.

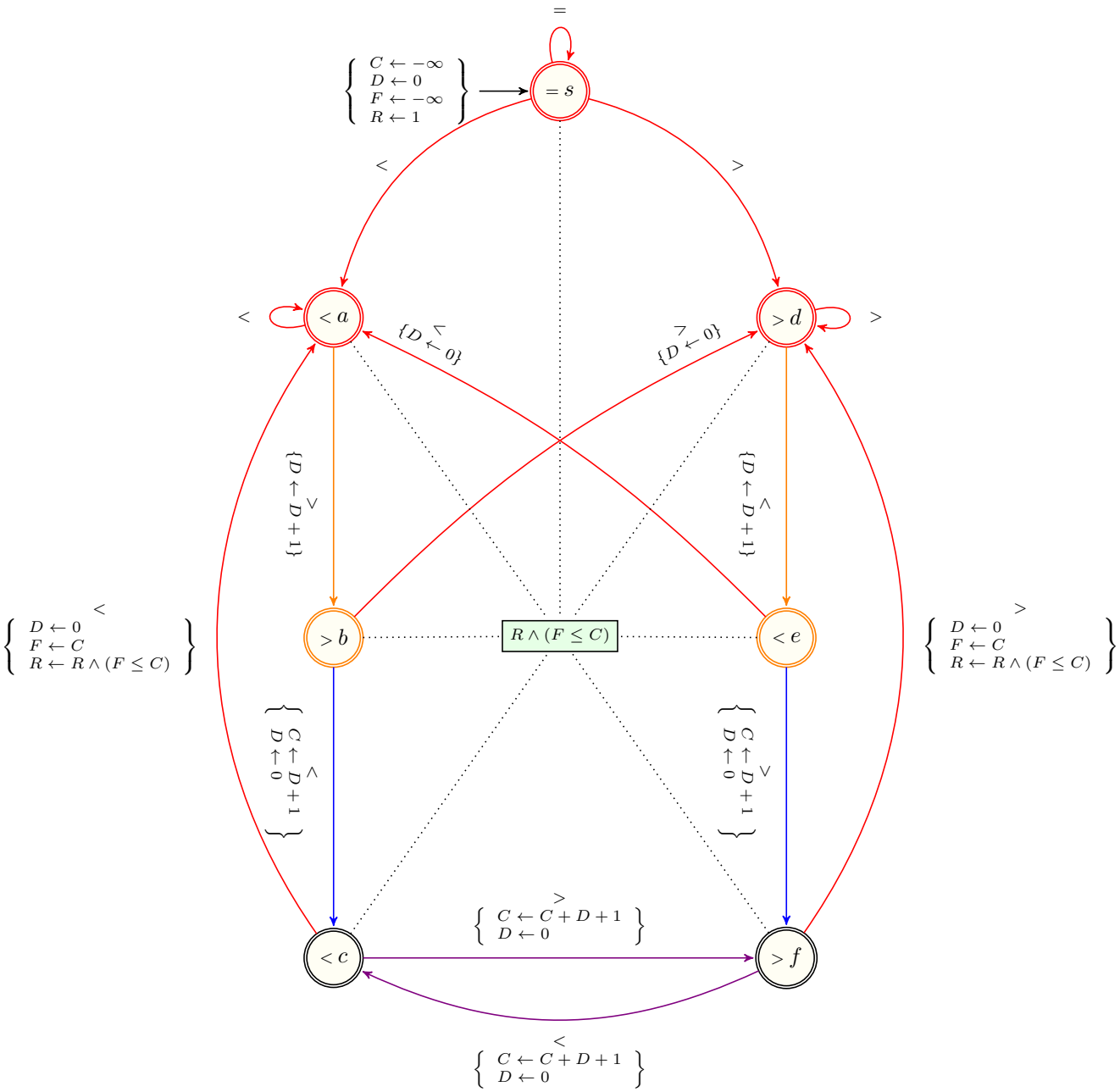
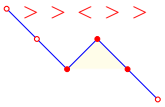


Figure 4.470: Automaton for the INCREASING_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.38 to the seed transducer of the ZIGZAG pattern; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register F is reset to C, and the register R is updated wrt C and F

FEATURE PATTERN

INDEX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint INDEX_BUMP_ON_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of BUMP_ON_DECREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of BUMP_ON_DECREASING_SEQUENCE.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.471 provides an example where the INDEX_BUMP_ON_DECREASING_SEQUENCE ([7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

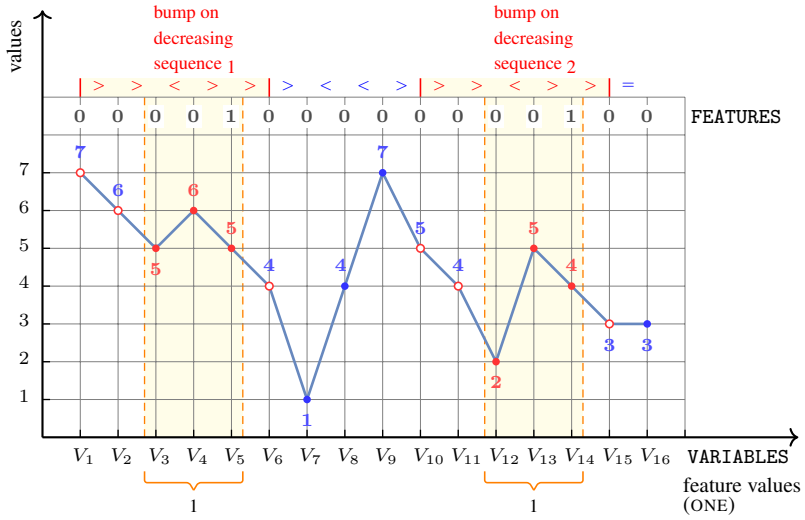


Figure 4.471: Illustrating the INDEX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint INDEX_DECREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.472 provides an example where the INDEX_DECREASING $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

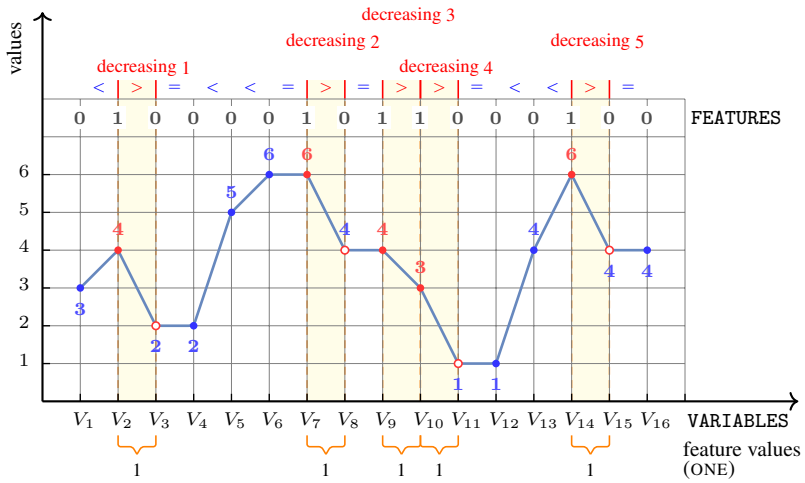
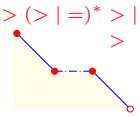


Figure 4.472: Illustrating the INDEX_DECREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint INDEX_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_SEQUENCE.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.473 provides an example where the INDEX_DECREASING_SEQUENCE $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

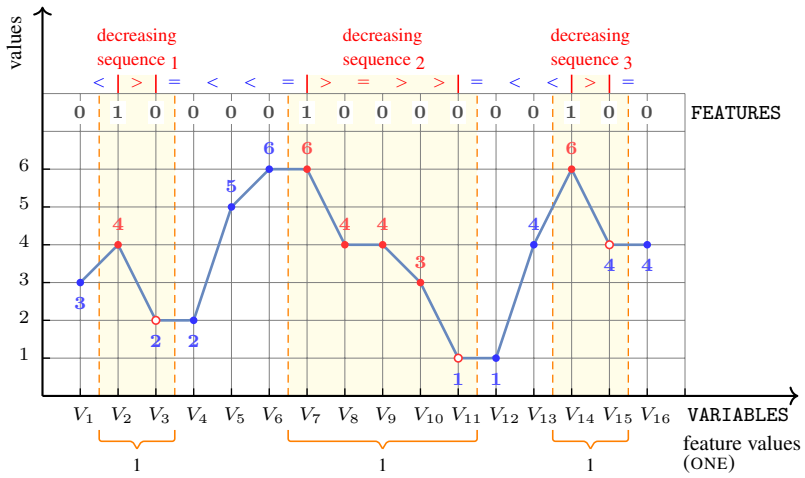


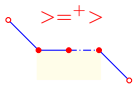
Figure 4.473: Illustrating the INDEX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint INDEX_DECREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_TERRACE.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.474 provides an example where the INDEX_DECREASING_TERRACE $([6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3],[0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

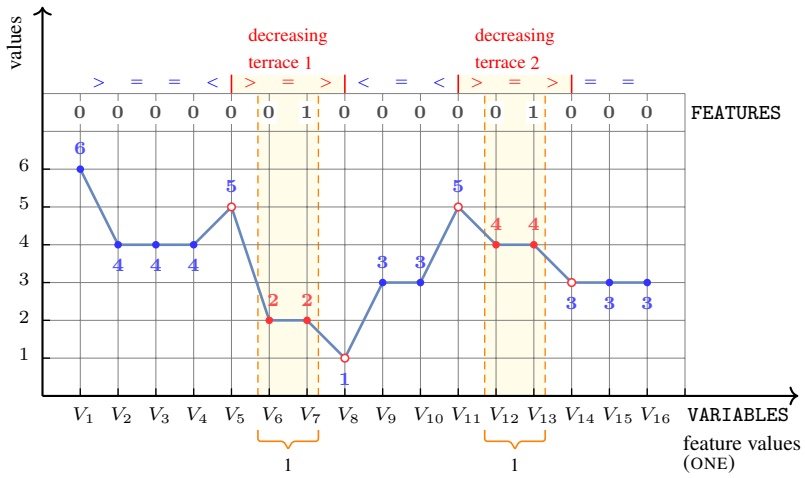
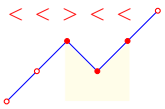


Figure 4.474: Illustrating the INDEX_DECREASING_TERRACE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint INDEX_DIP_ON_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DIP_ON_INCREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DIP_ON_INCREASING_SEQUENCE.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.475 provides an example where the INDEX_DIP_ON_INCREASING_SEQUENCE $([1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4],[0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

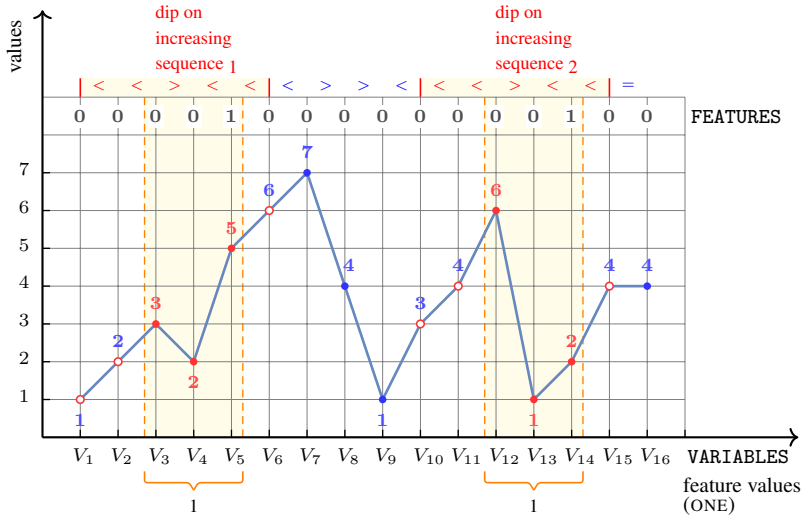


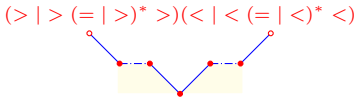
Figure 4.475: Illustrating the INDEX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint INDEX_GORGE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of GORGE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of GORGE.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.476 provides an example where the INDEX_GORGE $([1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7],[0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

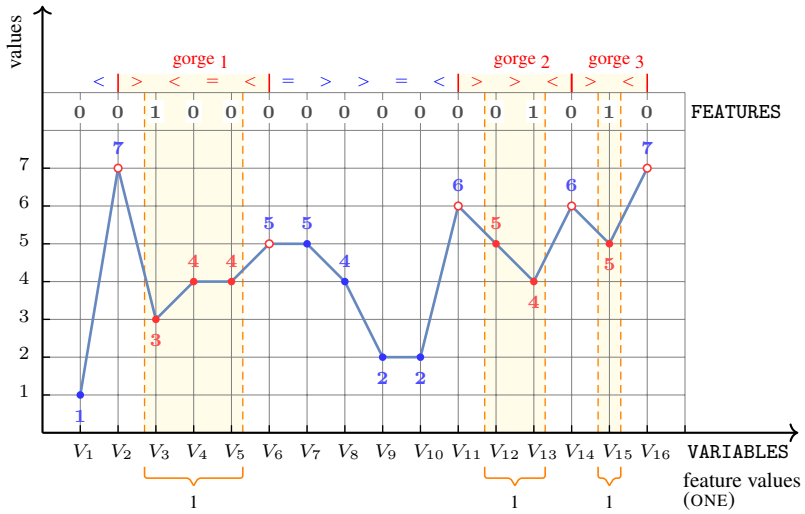


Figure 4.476: Illustrating the INDEX_GORGE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint INDEX_INCREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.477 provides an example where the INDEX_INCREASING $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

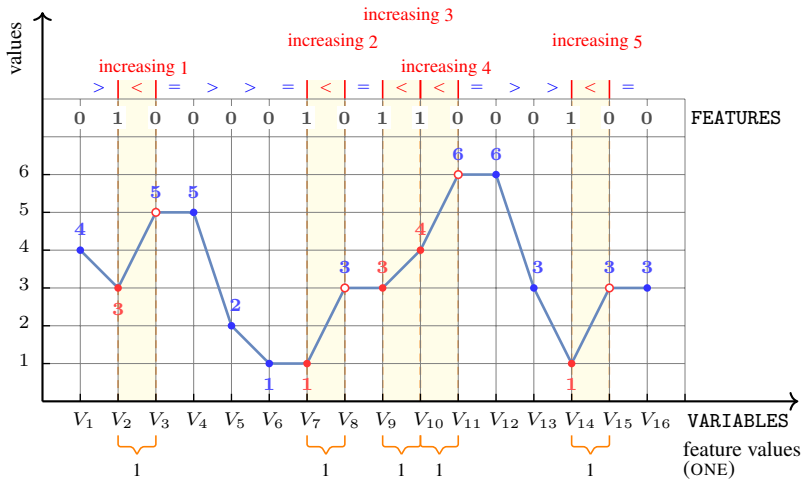
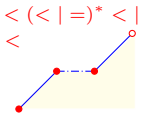


Figure 4.477: Illustrating the INDEX_INCREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint INDEX_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_SEQUENCE.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.478 provides an example where the INDEX_INCREASING_SEQUENCE $([4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3],[0,1,0,0,0,0,1,0,0,0,0,0,0,1,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

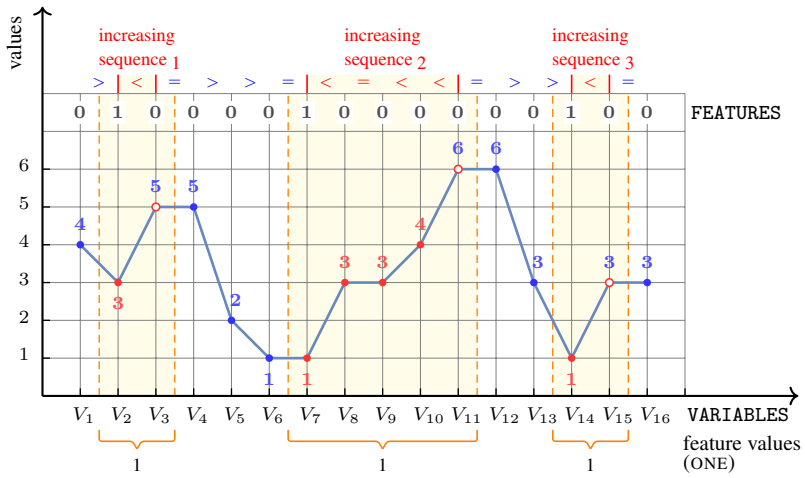


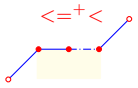
Figure 4.478: Illustrating the INDEX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint INDEX_INCREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_TERRACE.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.479 provides an example where the INDEX_INCREASING_TERRACE $([1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4],[0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

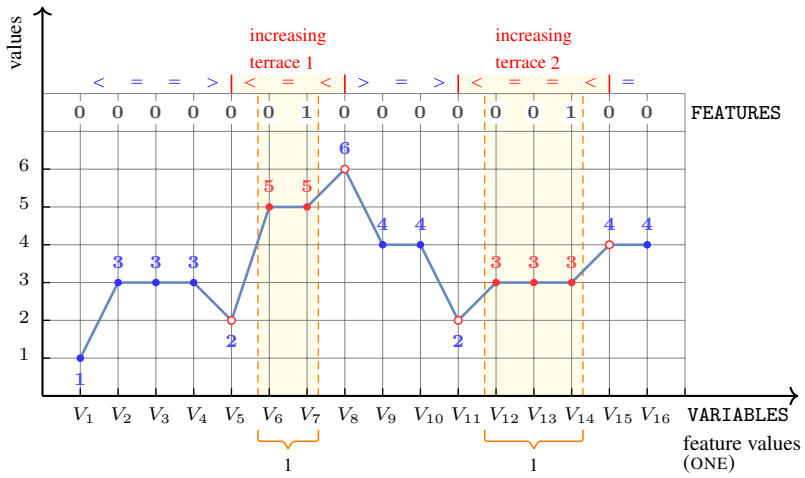


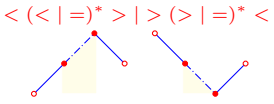
Figure 4.479: Illustrating the INDEX_INCREASING_TERRACE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint INDEX_INFLEXION(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INFLEXION is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INFLEXION.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.480 provides an example where the INDEX_INFLEXION $([1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4], [0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

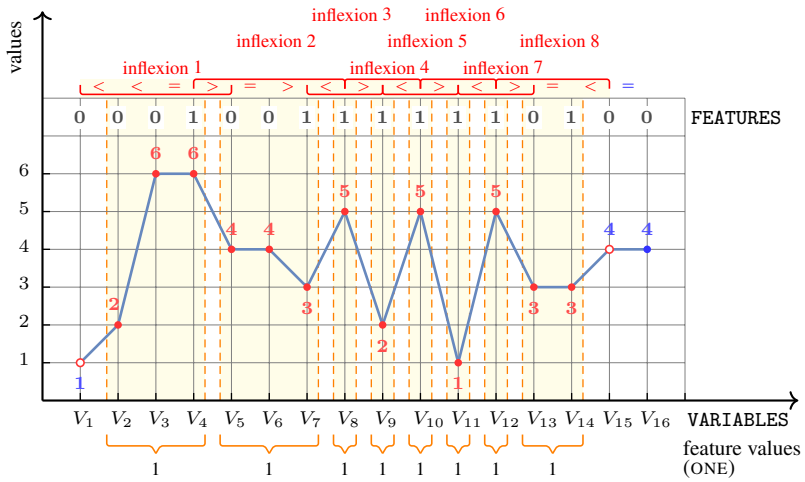


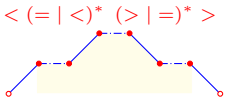
Figure 4.480: Illustrating the INDEX_INFLEXION constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

Constraint INDEX_PEAK(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PEAK is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PEAK.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.481 provides an example where the INDEX_PEAK $([7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1],[0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

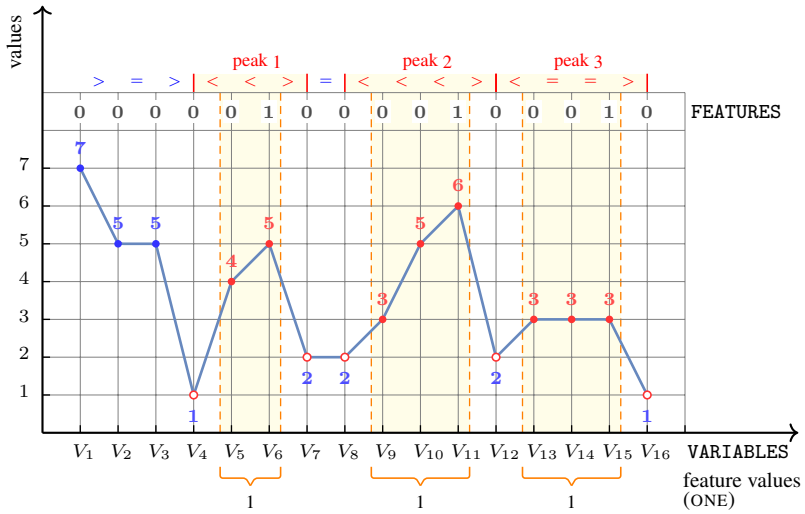


Figure 4.481: Illustrating the INDEX_PEAK constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_PLAIN

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint INDEX_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLAIN.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.482 provides an example where the INDEX_PLAIN $([2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3],[0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

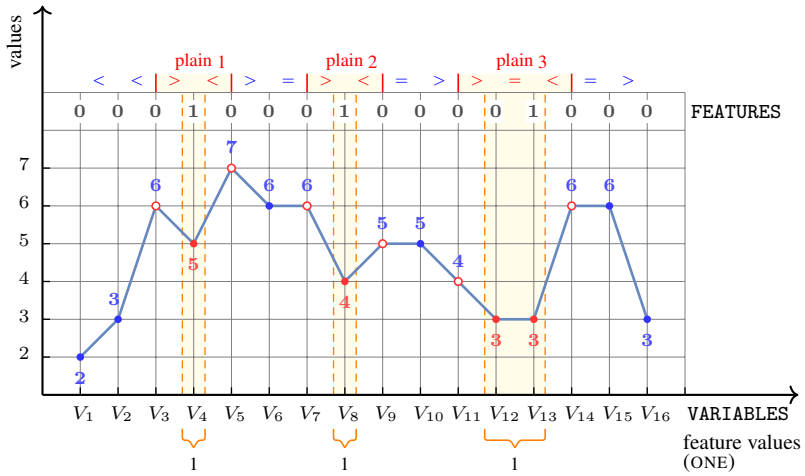


Figure 4.482: Illustrating the INDEX_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLATEAU pattern.

Constraint INDEX_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLATEAU.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.483 provides an example where the INDEX_PLATEAU $([7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5],[0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

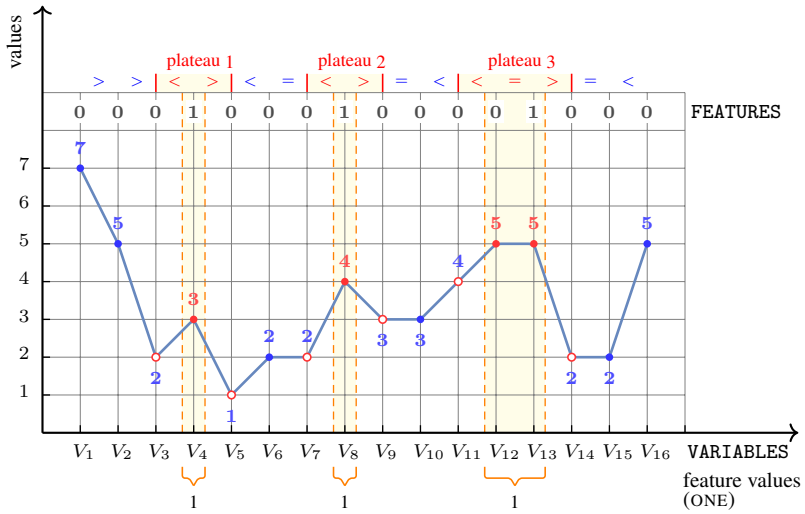


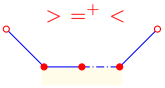
Figure 4.483: Illustrating the INDEX_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint INDEX_PROPER_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLAIN.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.484 provides an example where the INDEX_PROPER_PLAIN $([2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5],[0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

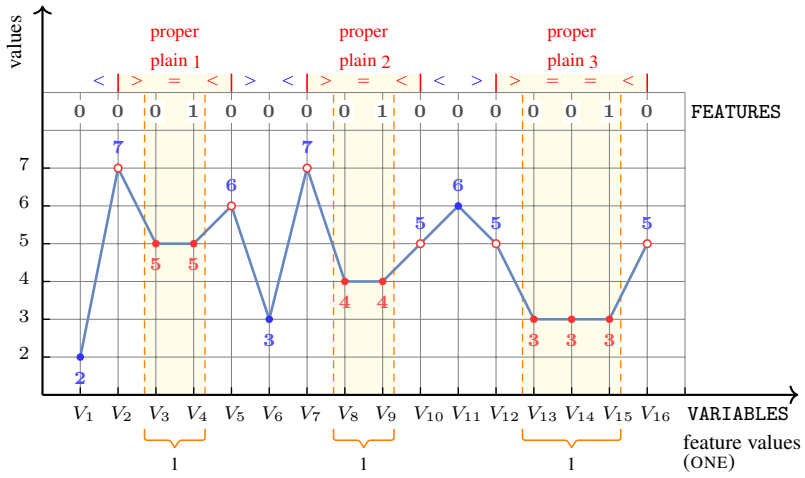


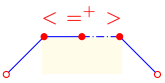
Figure 4.484: Illustrating the INDEX_PROPER_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint INDEX_PROPER_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLATEAU.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.485 provides an example where the INDEX_PROPER_PLATEAU $([7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3],[0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

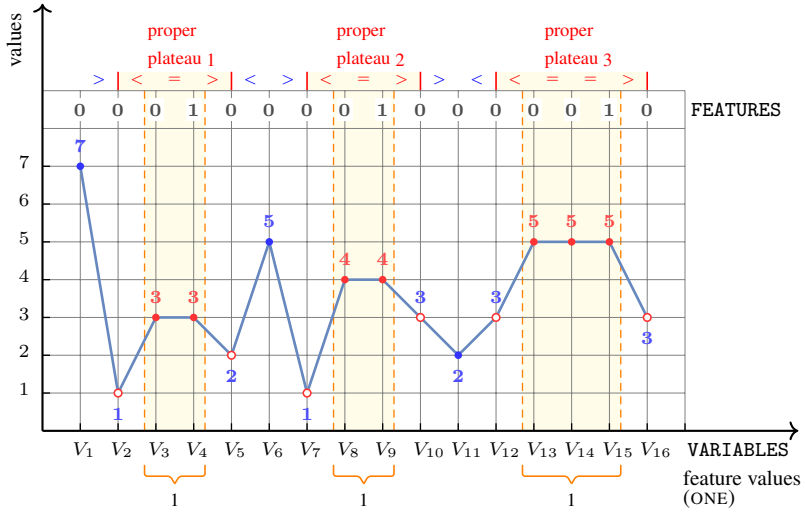


Figure 4.485: Illustrating the INDEX_PROPER_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint INDEX_STEADY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.486 provides an example where the INDEX_STEADY $([1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6], [1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0], 0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

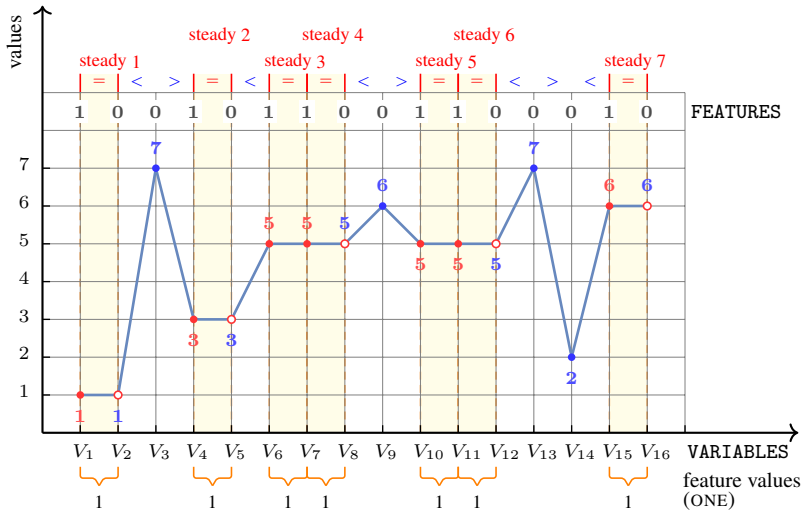


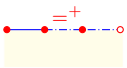
Figure 4.486: Illustrating the INDEX_STEADY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint INDEX_STEADY_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY_SEQUENCE.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.487 provides an example where the INDEX_STEADY_SEQUENCE $([3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1],[0,1,0,0,1,0,0,0,1,0,1,0,0,0,1,0],0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

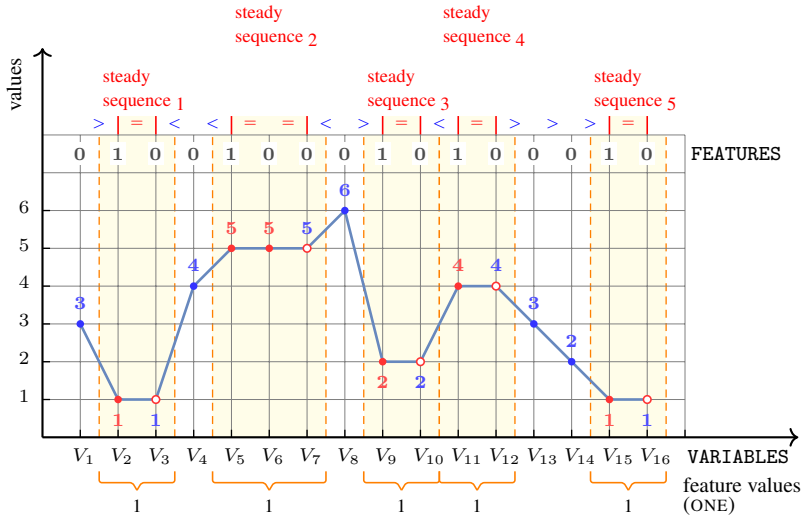


Figure 4.487: Illustrating the INDEX_STEADY_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint INDEX_STRICTLY_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_DECREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.488 provides an example where the INDEX_STRICTLY_DECREASING_SEQUENCE ($[4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3]$, $[0,0,1,0,0,0,0,0,0,0,1,0,0,0,$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

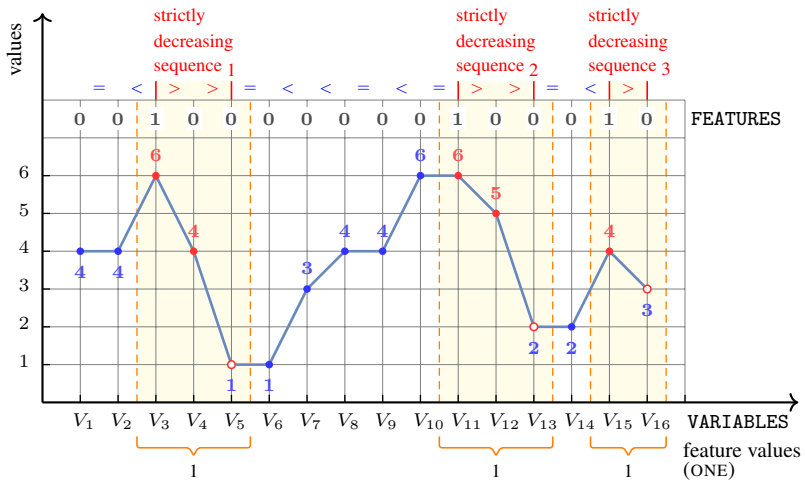


Figure 4.488: Illustrating the INDEX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint INDEX_STRICTLY_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_INCREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.489 provides an example where the INDEX_STRICTLY_INCREASING_SEQUENCE $([4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3],[0,1,0,0,0,0,1,0,0,0,0,0,0,1,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

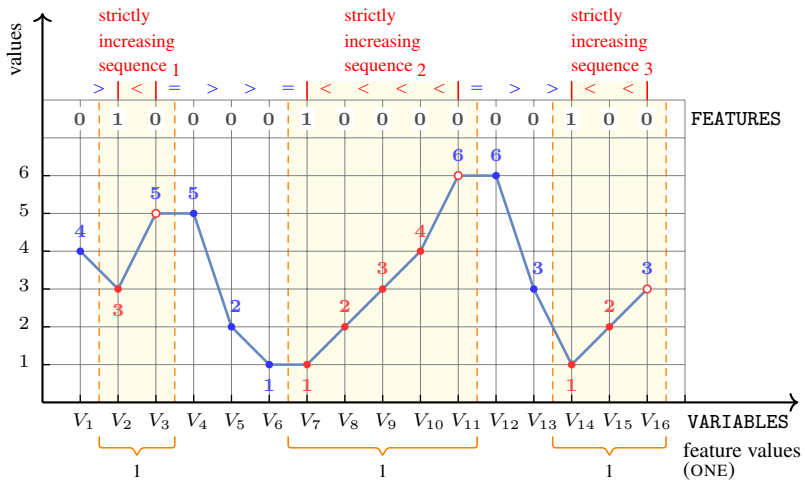


Figure 4.489: Illustrating the INDEX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

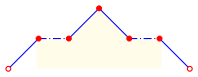
Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_SUMMIT

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

(< | < (= | <)* <)(> | > (= | >)* >)



Origin Based on the SUMMIT pattern.

Constraint INDEX_SUMMIT(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of SUMMIT is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of SUMMIT.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.490 provides an example where the INDEX_SUMMIT $([7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1], [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

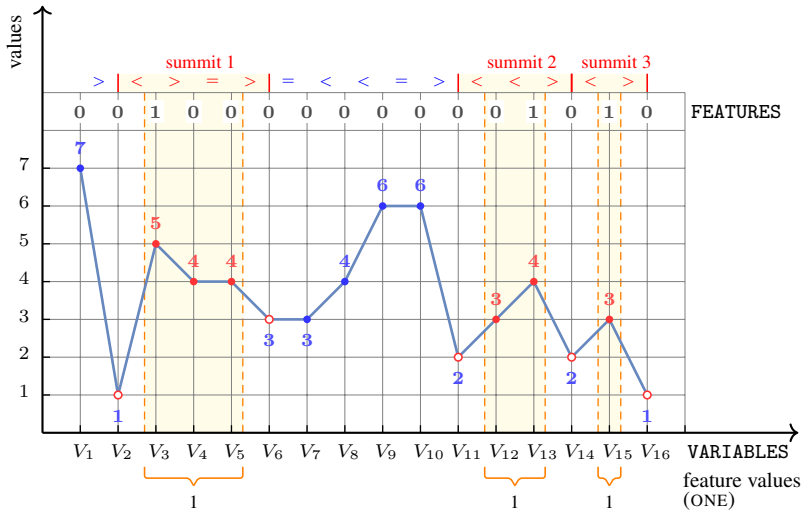


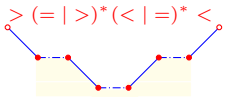
Figure 4.490: Illustrating the INDEX_SUMMIT constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

Constraint INDEX_VALLEY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of VALLEY is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of VALLEY.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.491 provides an example where the INDEX_VALLEY $([1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7],[0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

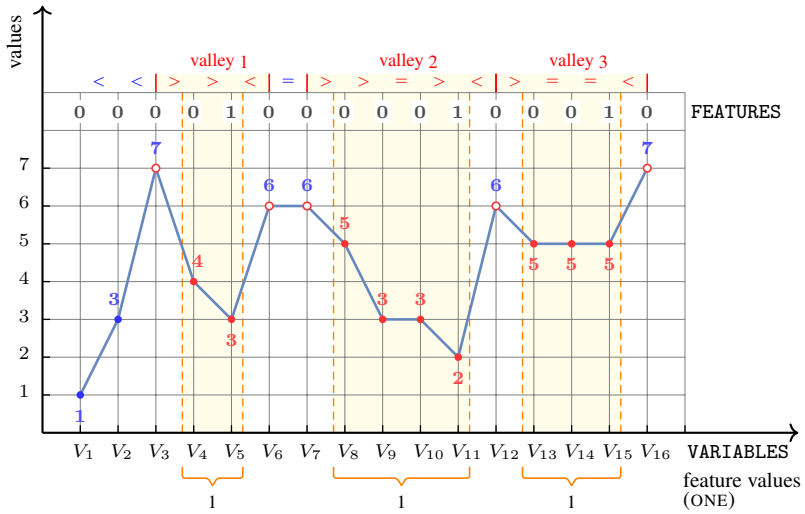


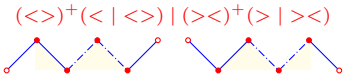
Figure 4.491: Illustrating the INDEX_VALLEY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

INDEX_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint INDEX_ZIGZAG(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ 1
DEFAULT = 0

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of ZIGZAG is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of ZIGZAG.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
The feature ONE, called INDEX in the name of the constraint returns 1.

Example Figure 4.492 provides an example where the INDEX_ZIGZAG $([4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1],[0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

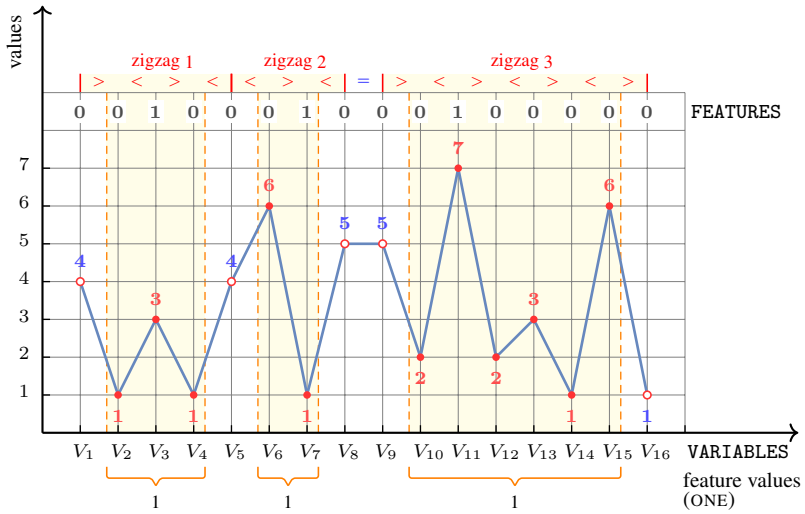
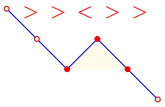


Figure 4.492: Illustrating the INDEX_ZIGZAG constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint MAX_BUMP_ON_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 5 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 2
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 2 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of BUMP_ON_DECREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of BUMP_ON_DECREASING_SEQUENCE.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example Figure 4.493 provides an example where the MAX_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

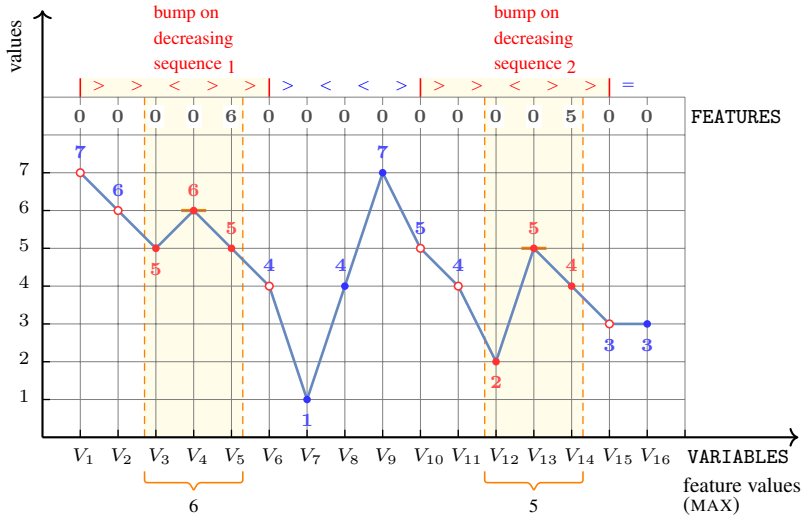


Figure 4.493: Illustrating the MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint MAX_DECREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.494 provides an example where the MAX_DECREASING $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 4, 0, 0, 0, 0, 6, 0, 4, 3, 0, 0, 0, 6, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

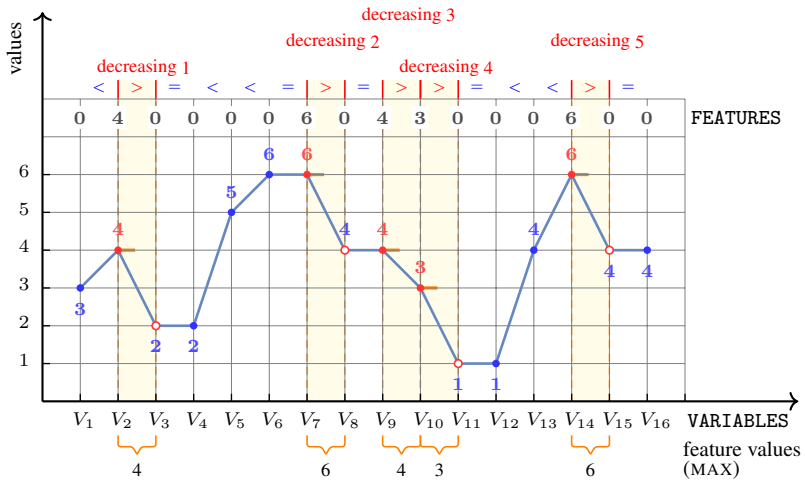
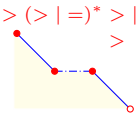


Figure 4.494: Illustrating the MAX_DECREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint MAX_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_SEQUENCE.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.495 provides an example where the MAX_DECREASING_SEQUENCE $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 4, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 6, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

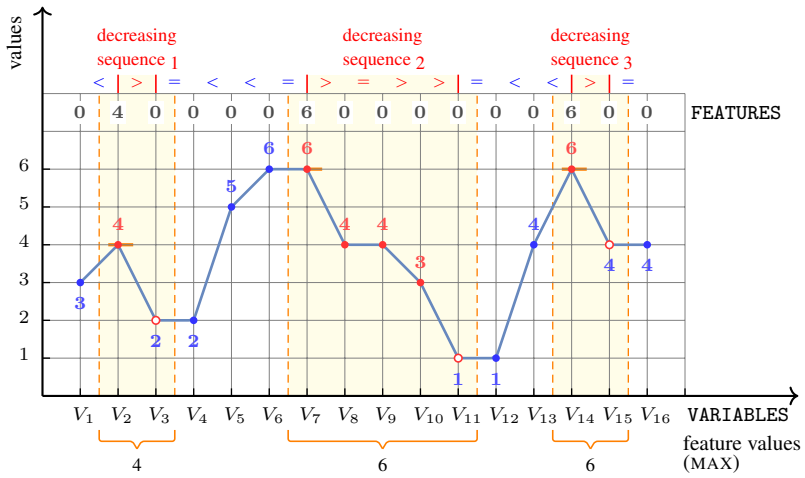
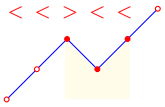


Figure 4.495: Illustrating the MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint MAX_DIP_ON_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 5 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 2
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 2 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DIP_ON_INCREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DIP_ON_INCREASING_SEQUENCE.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example Figure 4.496 provides an example where the MAX_DIP_ON_INCREASING_SEQUENCE $([1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4], [0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

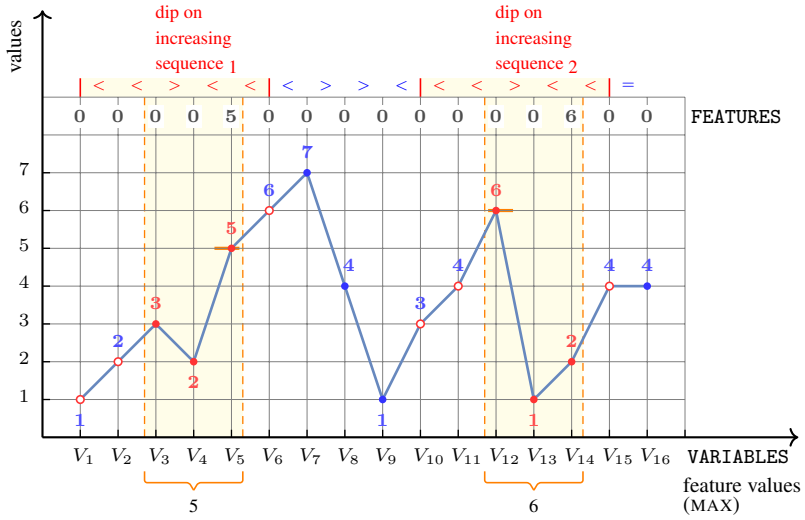


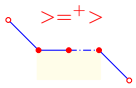
Figure 4.496: Illustrating the MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

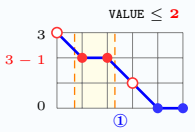
Constraint MAX_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(4, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3\rangle)$

Figure 4.497 provides an example where the MAX_HEIGHT_DECREASING_TERRACE $(4, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Arg. properties Functional dependency: VALUE determined by VARIABLES.

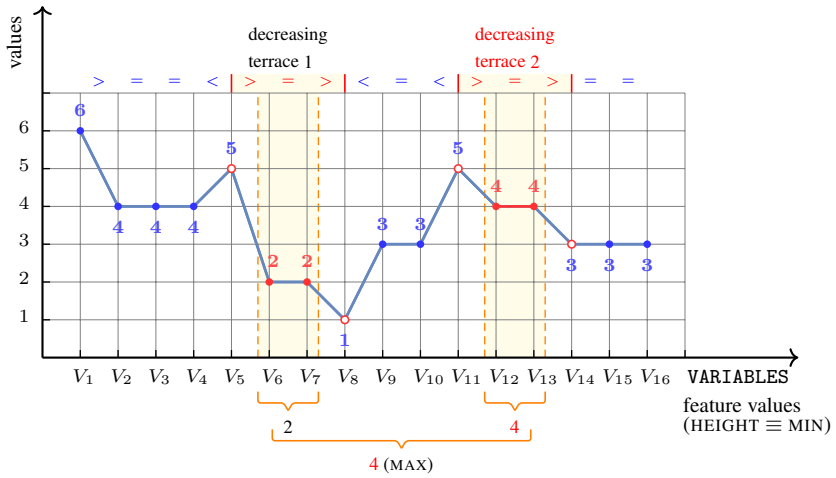


Figure 4.497: Illustrating the MAX_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.498 and 4.499 respectively depict the automaton associated with the constraint MAX_HEIGHT_DECREASING_TERRACE and its simplified form.

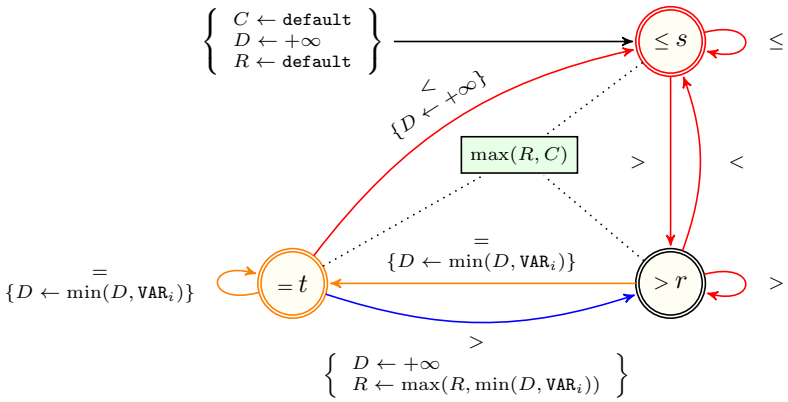


Figure 4.498: Automaton for the MAX_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where default is $-\infty$

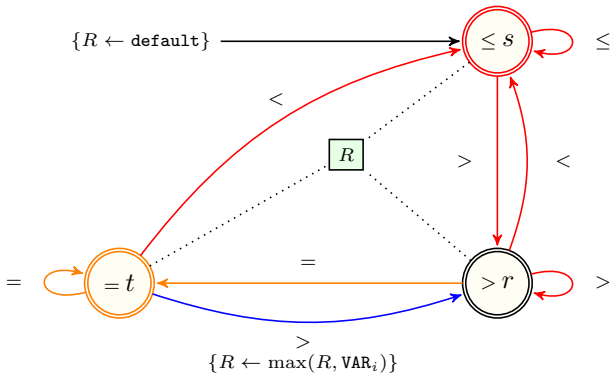


Figure 4.499: Simplified automaton for the MAX_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_TERRACE pattern where default is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.1: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MAX_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

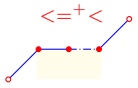
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\texttt{VAR}_{i+1}$ C
t	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.2: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MAX_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

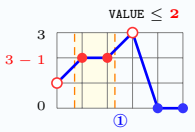
Constraint MAX_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.500 provides an example where the MAX_HEIGHT_INCREASING_TERRACE $(5, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

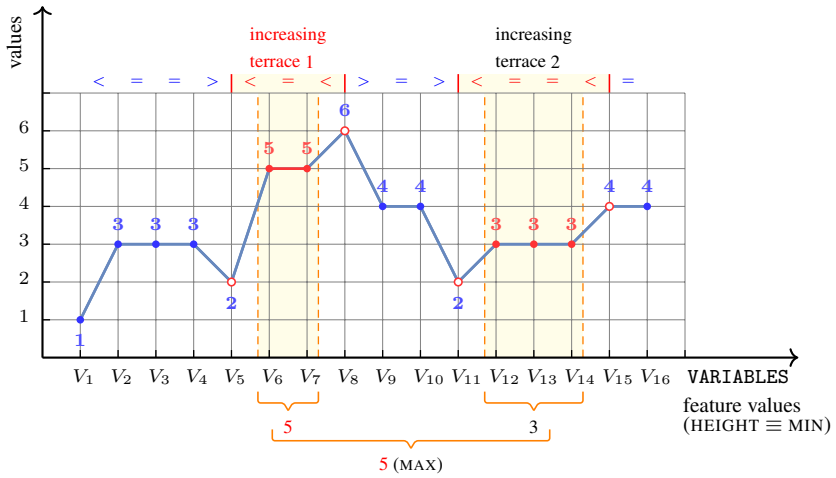


Figure 4.500: Illustrating the MAX_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.501 and 4.502 respectively depict the automaton associated with the constraint MAX_HEIGHT_INCREASING_TERRACE and its simplified form.

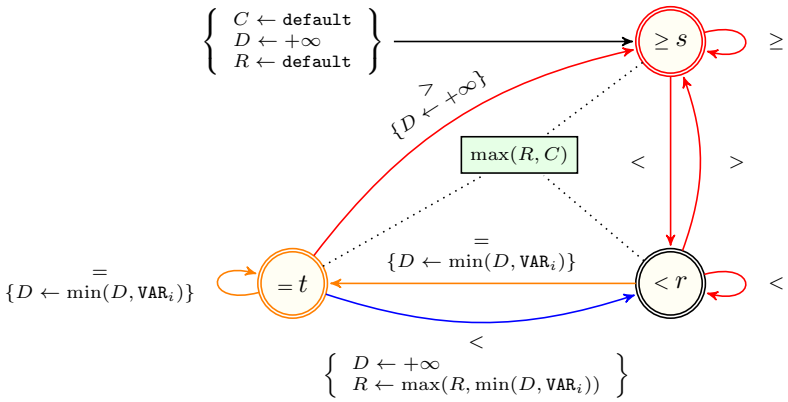


Figure 4.501: Automaton for the MAX_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $-\infty$

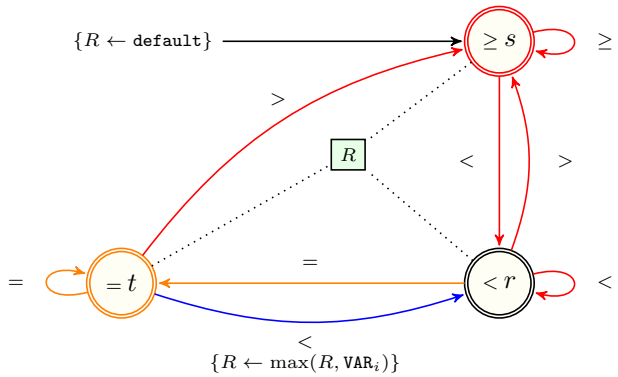


Figure 4.502: Simplified automaton for the MAX_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.3: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MAX_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern , the feature MIN , and the aggregator max ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\texttt{VAR}_{i+1}$ C
t	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.4: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MAX_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern , the feature MIN , and the aggregator max ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

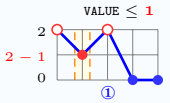
Constraint MAX_HEIGHT_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3 \rangle)$

Figure 4.503 provides an example where the MAX_HEIGHT_PLAIN $(5, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

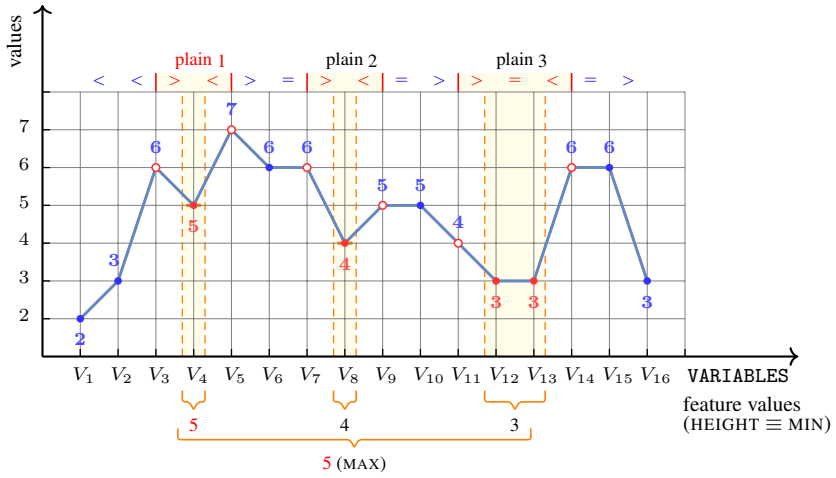


Figure 4.503: Illustrating the MAX_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figures 4.504 and 4.505 respectively depict the automaton associated with the constraint MAX_HEIGHT_PLAIN and its simplified form.

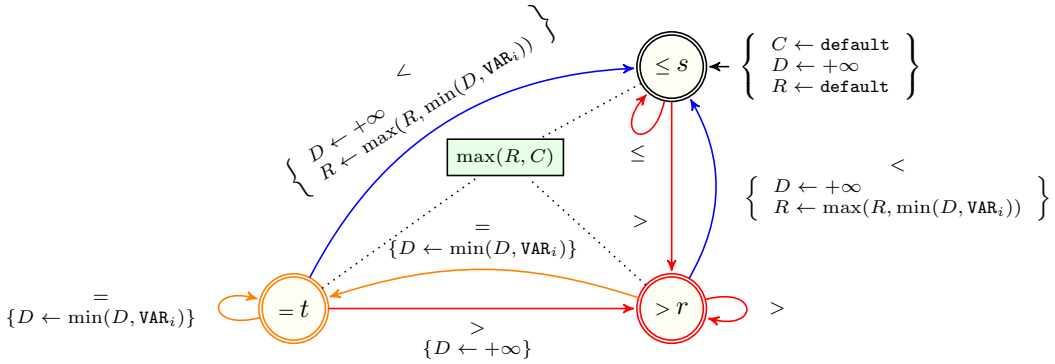


Figure 4.504: Automaton for the MAX_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is $-\infty$

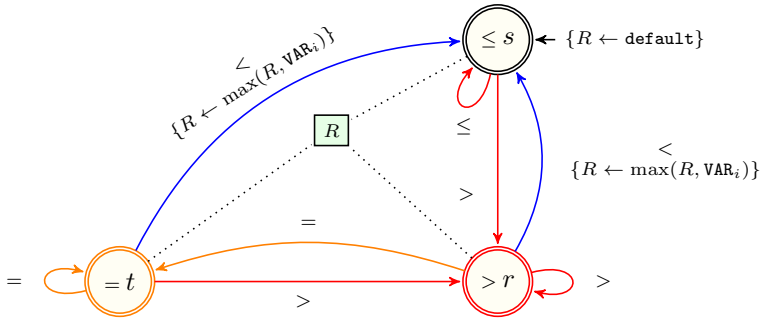


Figure 4.505: Simplified automaton for the MAX_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLAIN pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.5: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MAX_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C
t	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.6: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MAX_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

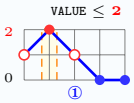
Constraint MAX_HEIGHT_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle)$

Figure 4.506 provides an example where the MAX_HEIGHT_PLATEAU $(5, [7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

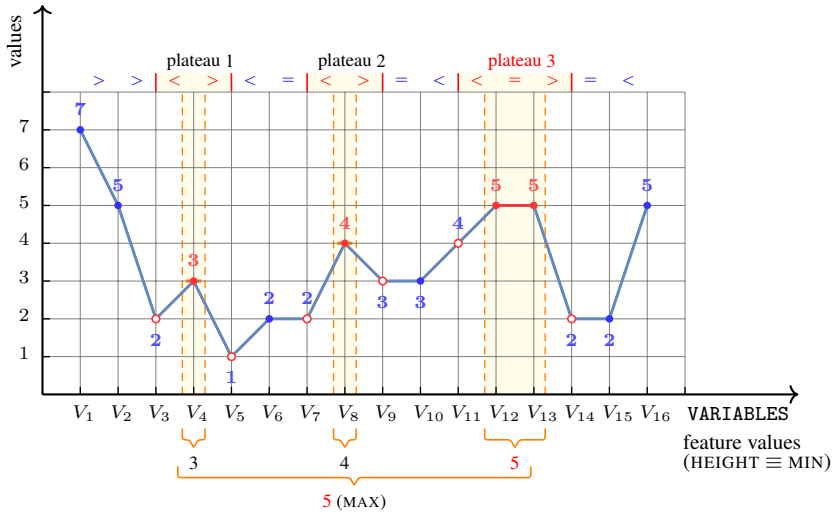


Figure 4.506: Illustrating the MAX_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figures 4.507 and 4.508 respectively depict the automaton associated with the constraint MAX_HEIGHT_PLATEAU and its simplified form.

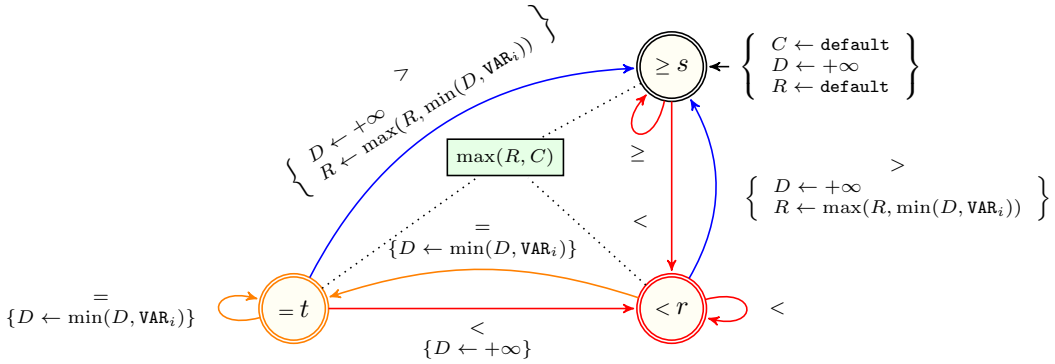


Figure 4.507: Automaton for the MAX_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is $-\infty$

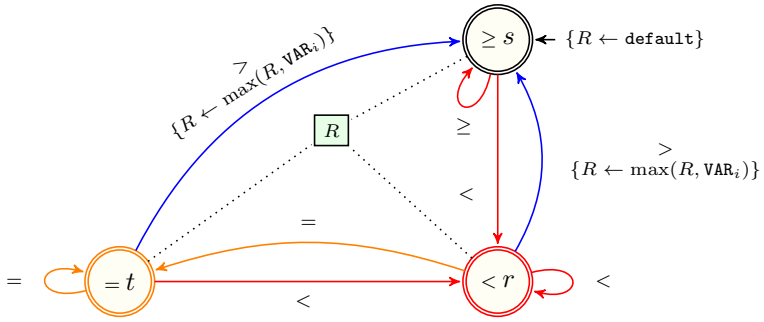


Figure 4.508: Simplified automaton for the MAX_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLATEAU pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

Table 4.7: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MAX_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

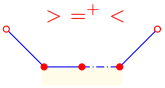
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C
t	$-\infty$	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.8: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MAX_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_PROPER_PLAIN

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLAIN pattern.

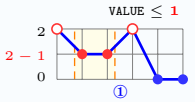
Constraint MAX_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5\rangle)$

Figure 4.509 provides an example where the MAX_HEIGHT_PROPER_PLAIN $(5, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5])$ constraint holds.

Typical
|VARIABLES| > 3
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

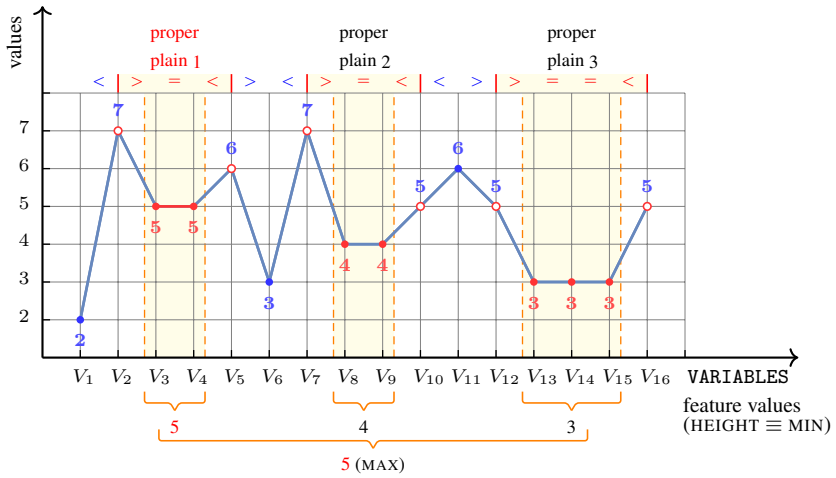


Figure 4.509: Illustrating the MAX_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.510 and 4.511 respectively depict the automaton associated with the constraint MAX_HEIGHT_PROPER_PLAIN and its simplified form.

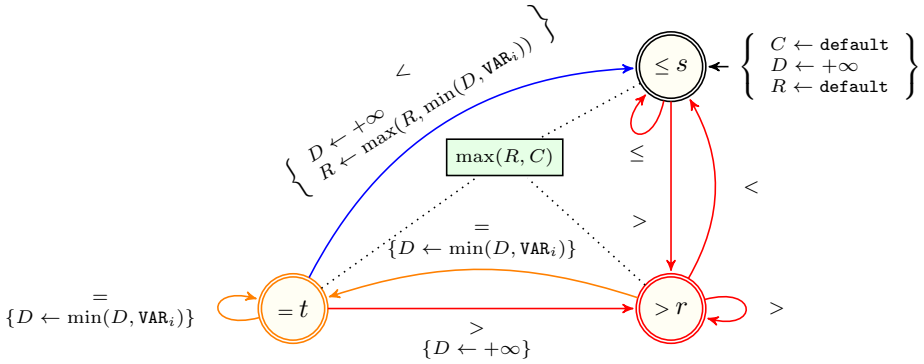


Figure 4.510: Automaton for the MAX_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where default is $-\infty$

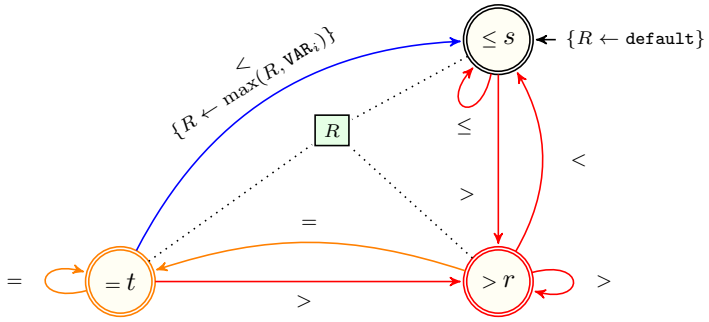


Figure 4.511: Simplified automaton for the MAX_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLAIN pattern where default is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.9: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MAX_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

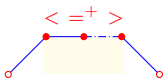
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\texttt{VAR}_{i+1}$ C
t	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.10: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MAX_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_PROPER_PLATEAU

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the PROPER_PLATEAU pattern.

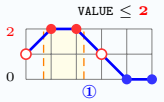
Constraint MAX_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3\rangle)$

Figure 4.512 provides an example where the MAX_HEIGHT_PROPER_PLATEAU $(5, [7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

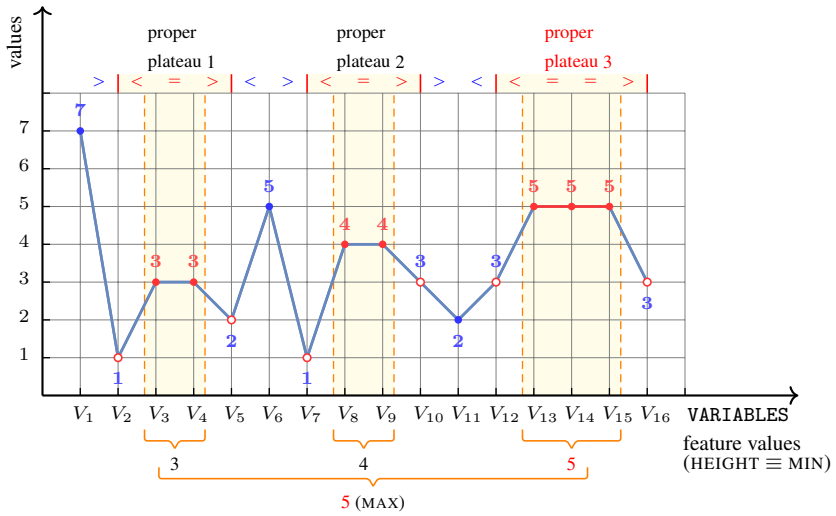


Figure 4.512: Illustrating the MAX_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.513 and 4.514 respectively depict the automaton associated with the constraint MAX_HEIGHT_PROPER_PLATEAU and its simplified form.

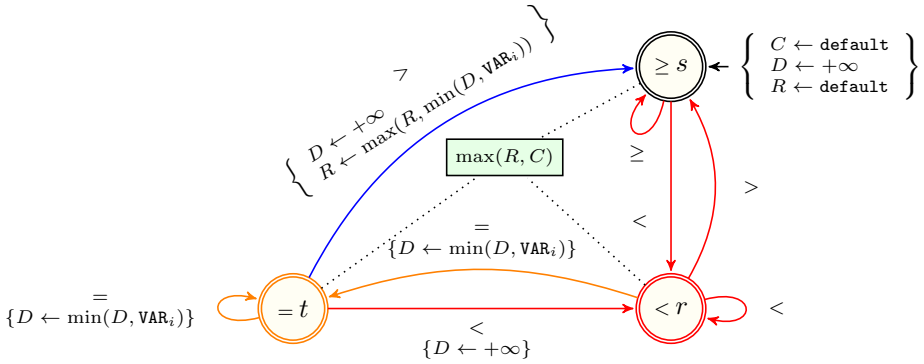


Figure 4.513: Automaton for the MAX_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where default is $-\infty$

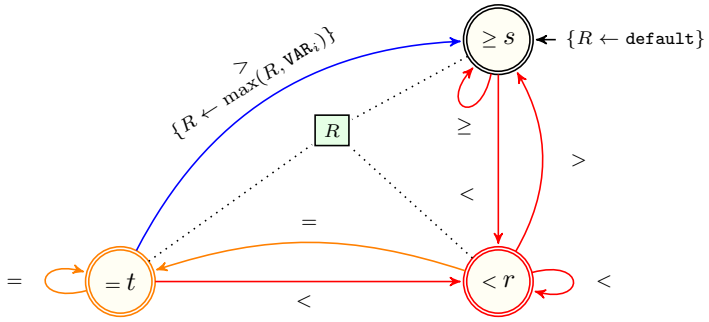


Figure 4.514: Simplified automaton for the MAX_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLATEAU pattern where default is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.11: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MAX_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\texttt{VAR}_{i+1}$ C
t	$-\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.12: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MAX_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

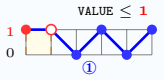
Constraint MAX_HEIGHT_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the maximum of all minimum values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(6, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6\rangle)$

Figure 4.515 provides an example where the MAX_HEIGHT_STEADY $(6, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

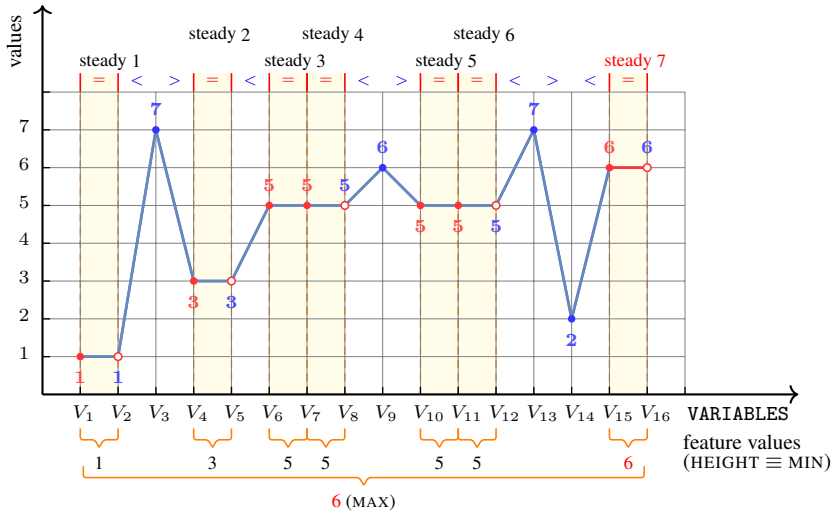


Figure 4.515: Illustrating the MAX_HEIGHT_STEADY constraint of the **Example** slot

Automaton

Figures 4.516 and 4.517 respectively depict the automaton associated with the constraint MAX_HEIGHT_STEADY and its simplified form.

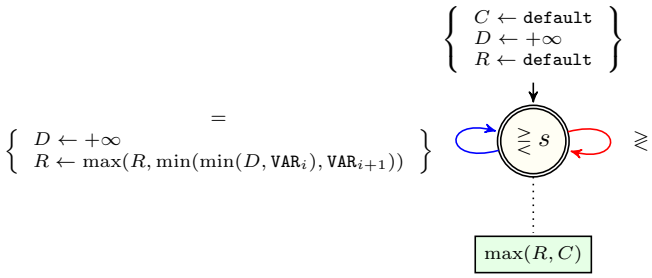


Figure 4.516: Automaton for the MAX_HEIGHT_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is $-\infty$

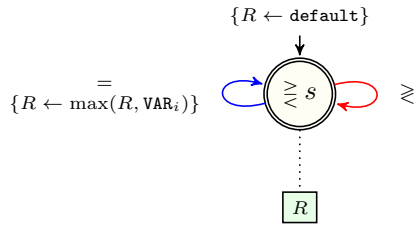


Figure 4.517: Simplified automaton for the MAX_HEIGHT_STEADY constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.13: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the MAX_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

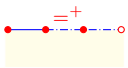
	s
s	$-\infty$

Table 4.14: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the MAX_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

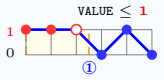
Constraint MAX_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the maximum of all minimum values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(5, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle)$

Figure 4.518 provides an example where the MAX_HEIGHT_STEADY_SEQUENCE $(5, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

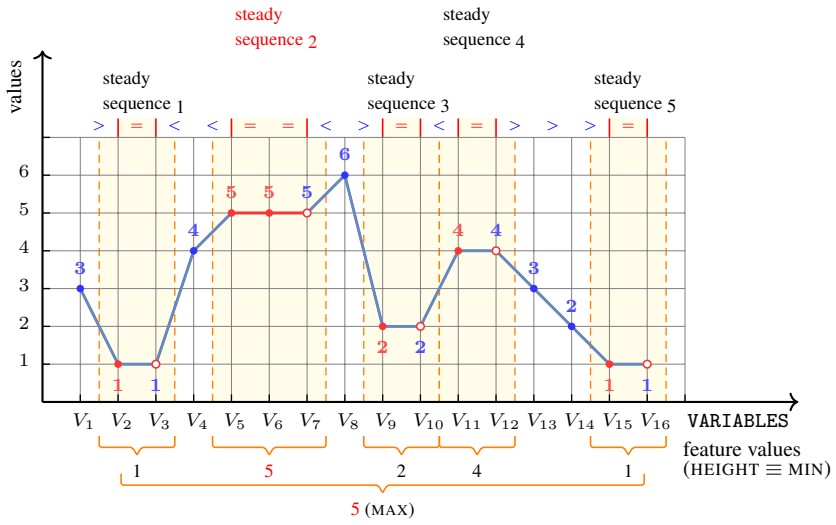


Figure 4.518: Illustrating the MAX_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.519 and 4.520 respectively depict the automaton associated with the constraint MAX_HEIGHT_STEADY_SEQUENCE and its simplified form.

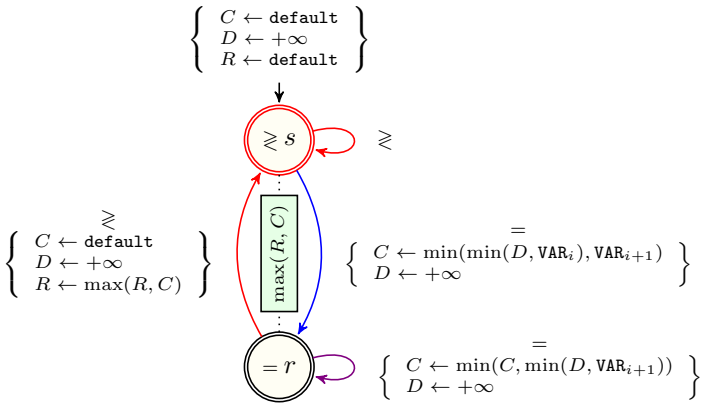


Figure 4.519: Automaton for the MAX_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $-\infty$

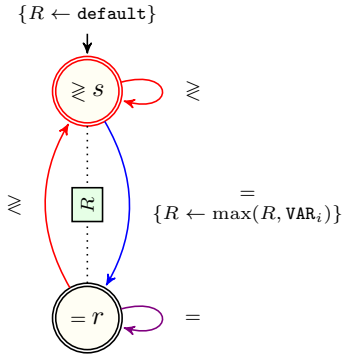


Figure 4.520: Simplified automaton for the MAX_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.15: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MAX_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$ M

Table 4.16: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MAX_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint MAX_INCREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.521 provides an example where the MAX_INCREASING $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 5, 0, 0, 0, 0, 3, 0, 4, 6, 0, 0, 0, 3, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

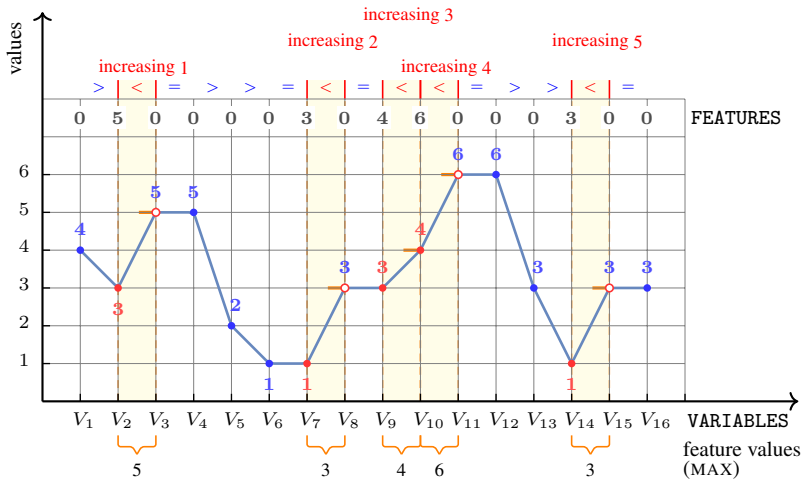
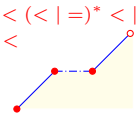


Figure 4.521: Illustrating the MAX_INCREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint MAX_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_SEQUENCE.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.522 provides an example where the MAX_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 5, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

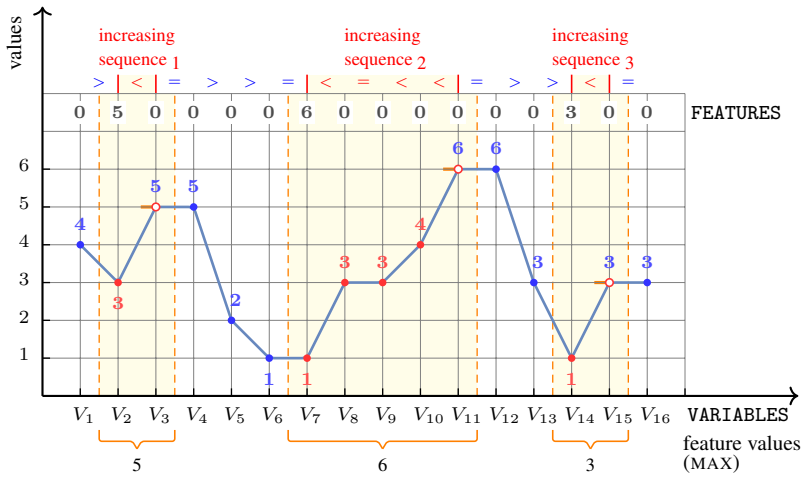
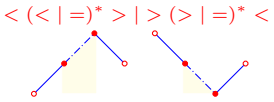


Figure 4.522: Illustrating the MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_INFLEXION



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INFLEXION pattern.

Constraint MAX_INFLEXION(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INFLEXION is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INFLEXION.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example Figure 4.523 provides an example where the MAX_INFLEXION $([1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4],[0,0,0,6,0,0,4,5,2,5,1,5,0,3,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

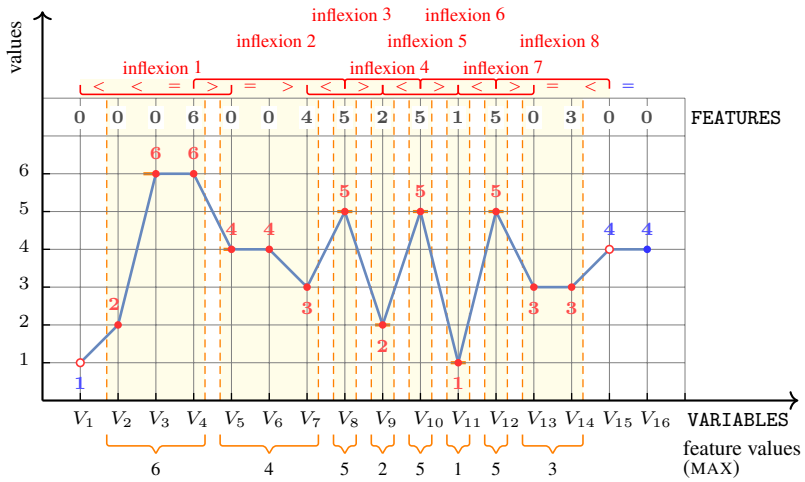
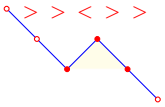


Figure 4.523: Illustrating the MAX_INFLEXION constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MAX_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

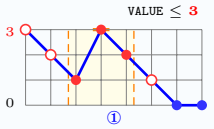
Constraint MAX_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 2$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$. An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(6, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.524 provides an example where the MAX_MAX_BUMP_ON_DECREASING_SEQUENCE $(6, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

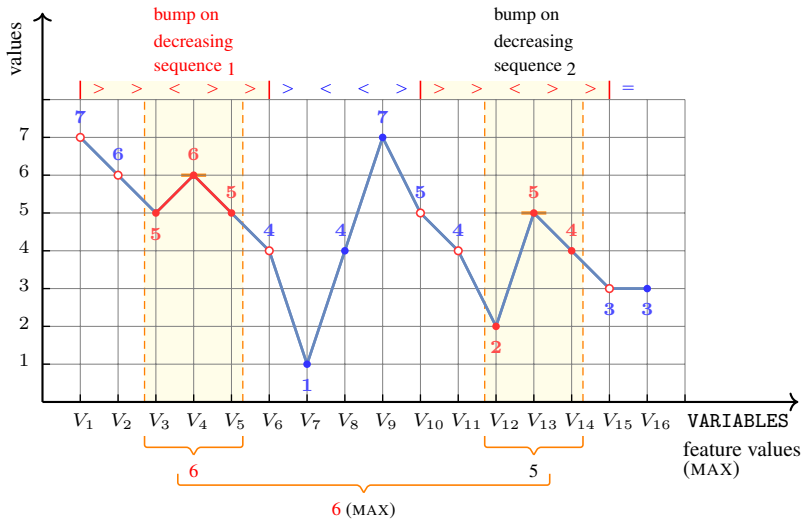


Figure 4.524: Illustrating the MAX_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.525 and 4.526 respectively depict the automaton associated with the constraint MAX_MAX_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

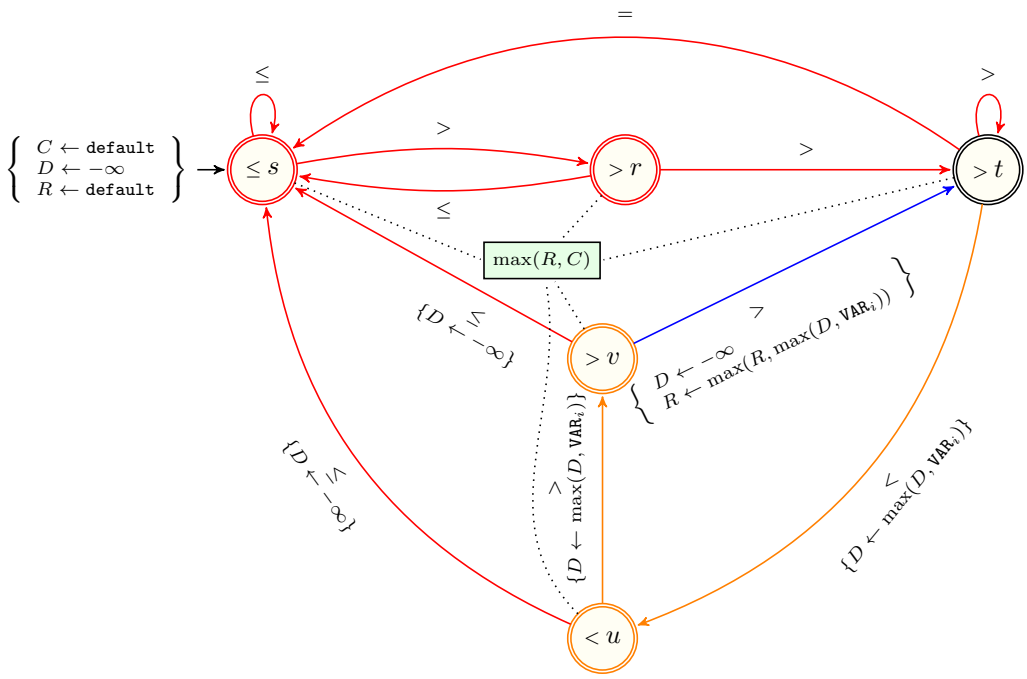


Figure 4.525: Automaton for the MAX_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$

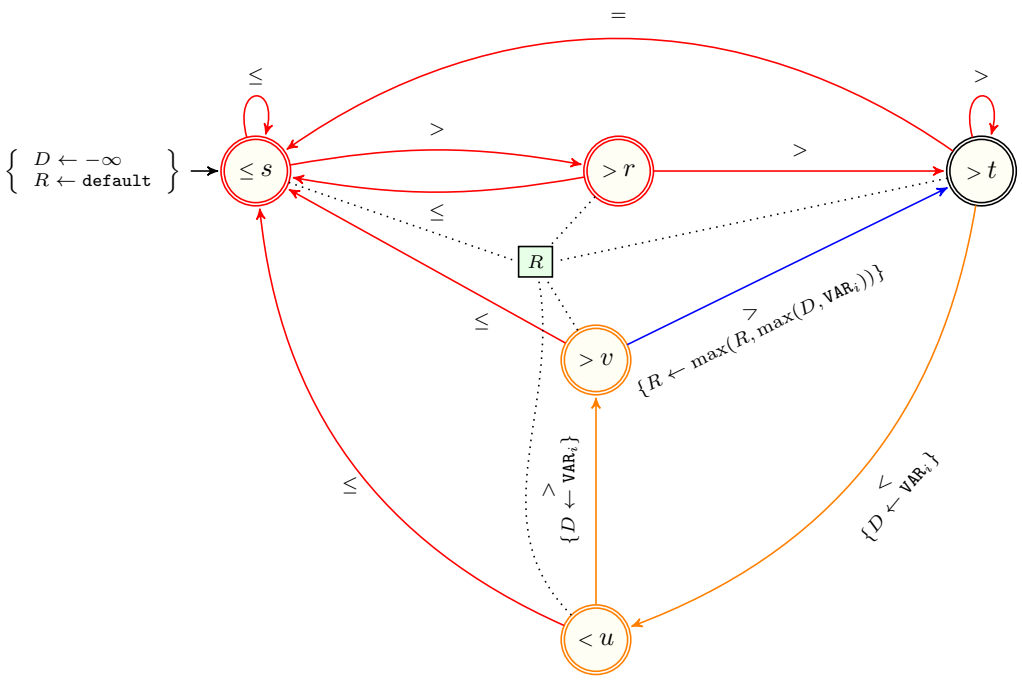


Figure 4.526: Simplified automaton for the MAX_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MAX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

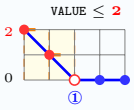
Constraint MAX_MAX_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.527 provides an example where the MAX_MAX_DECREASING $(6, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

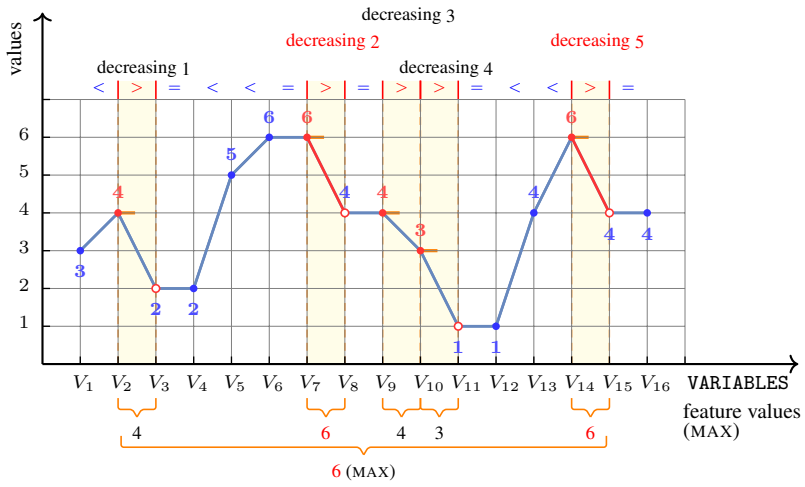


Figure 4.527: Illustrating the MAX_MAX_DECREASING constraint of the **Example** slot

Automaton

Figures 4.528 and 4.529 respectively depict the automaton associated with the constraint MAX_MAX_DECREASING and its simplified form.

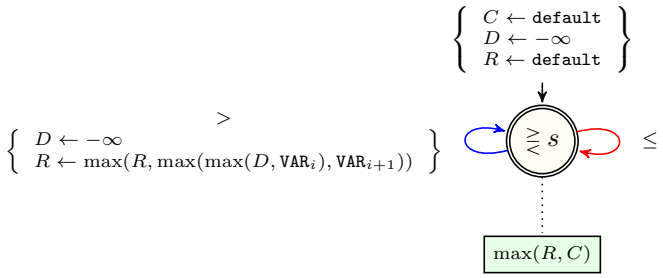


Figure 4.528: Automaton for the MAX_MAX_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $-\infty$

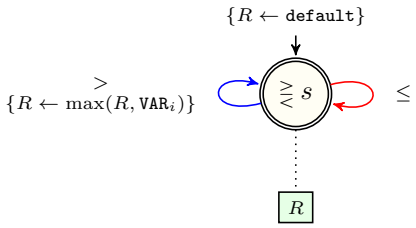


Figure 4.529: Simplified automaton for the MAX_MAX_DECREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

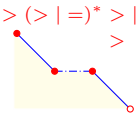
Table 4.17: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MAX_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.18: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MAX_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

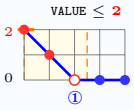
Constraint MAX_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.530 provides an example where the MAX_MAX_DECREASING_SEQUENCE $(6, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

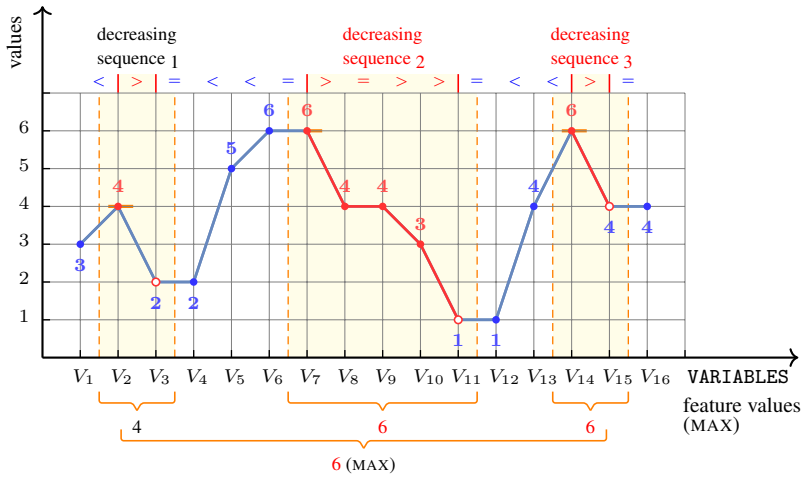


Figure 4.530: Illustrating the MAX_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.531 and 4.532 respectively depict the automaton associated with the constraint MAX_MAX_DECREASING_SEQUENCE and its simplified form.

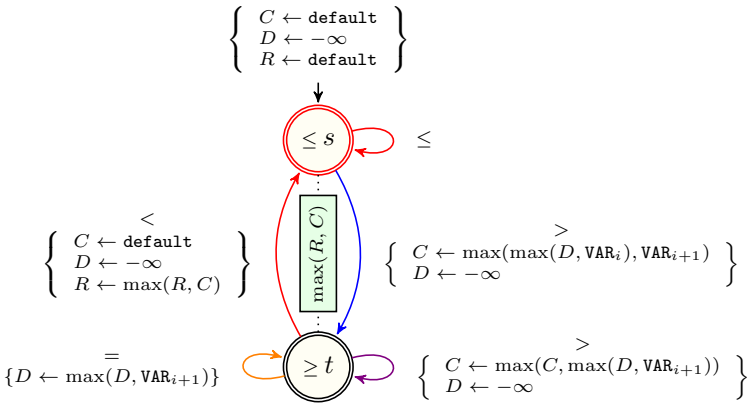


Figure 4.531: Automaton for the MAX_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $-\infty$

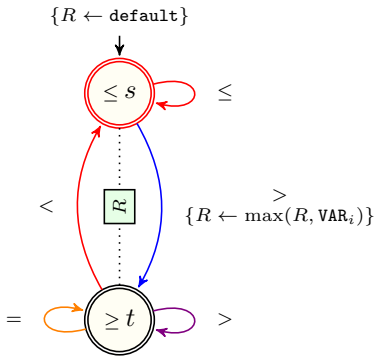


Figure 4.532: Simplified automaton for the MAX_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

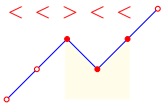
Table 4.19: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MAX_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	$-\infty$	$-\infty$
t	$-\infty$	$-\infty$ M

Table 4.20: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the MAX_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

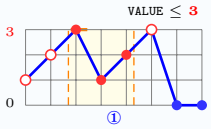
Constraint MAX_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 2$
$\texttt{VALUE} \leq \texttt{maxv}①$
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(6, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.533 provides an example where the MAX_MAX_DIP_ON_INCREASING_SEQUENCE $(6, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

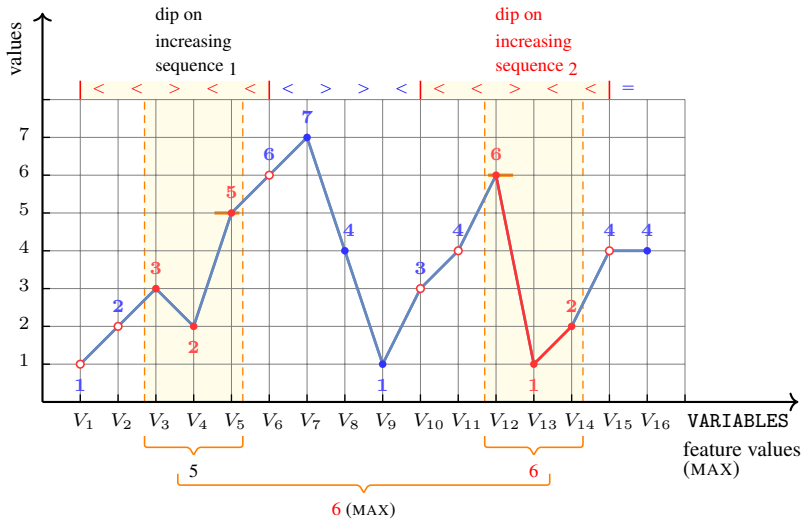


Figure 4.533: Illustrating the MAX_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.534 and 4.535 respectively depict the automaton associated with the constraint MAX_MAX_DIP_ON_INCREASING_SEQUENCE and its simplified form.

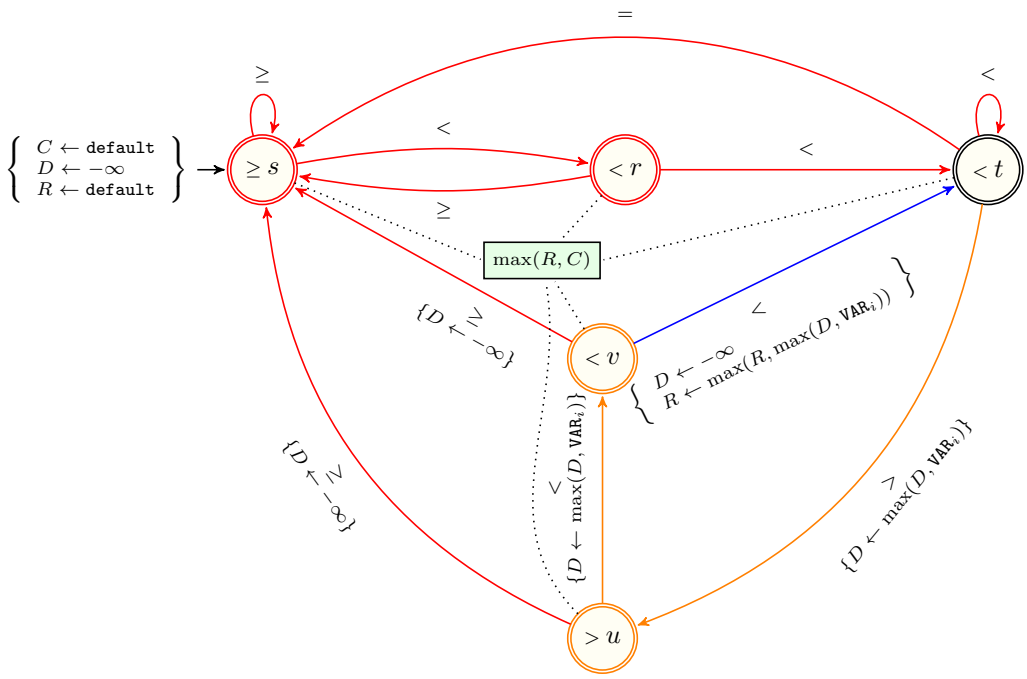


Figure 4.534: Automaton for the MAX_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$

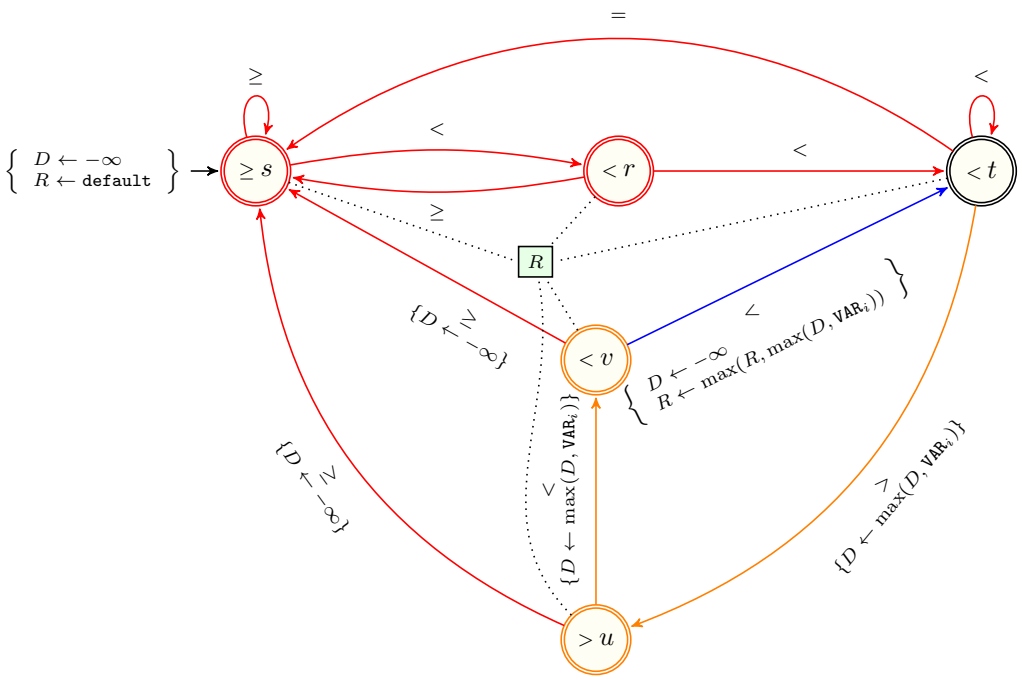


Figure 4.535: Simplified automaton for the MAX_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

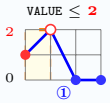
Constraint MAX_MAX_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all maximum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.536 provides an example where the MAX_MAX_INCREASING $(6, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

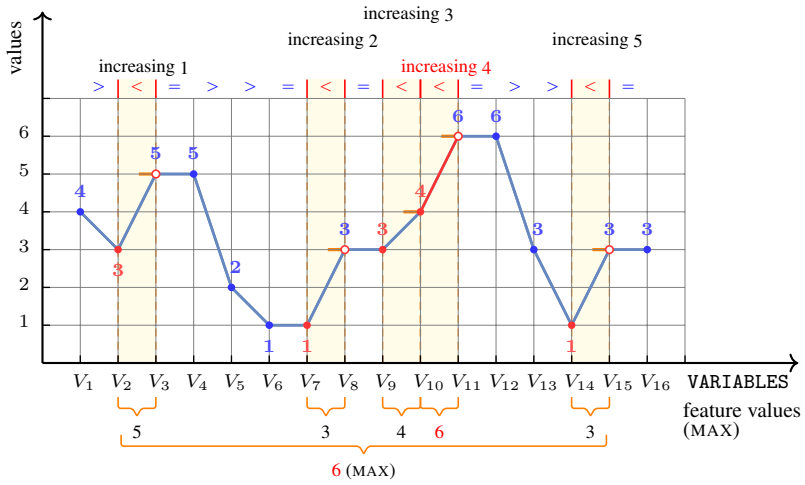


Figure 4.536: Illustrating the MAX_MAX_INCREASING constraint of the **Example** slot

Automaton

Figures 4.537 and 4.538 respectively depict the automaton associated with the constraint MAX_MAX_INCREASING and its simplified form.

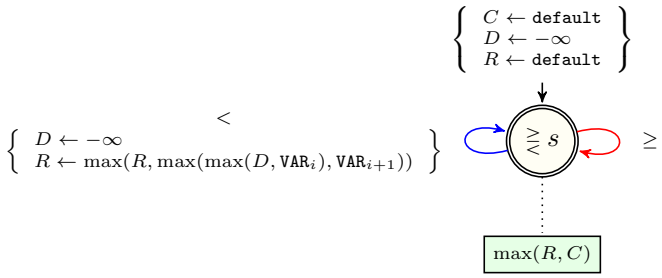


Figure 4.537: Automaton for the MAX_MAX_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $-\infty$

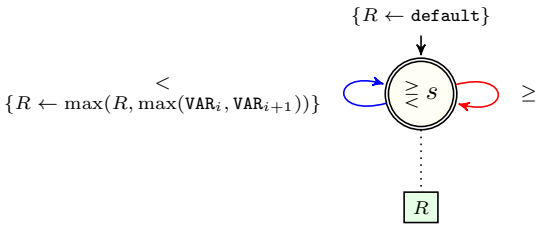


Figure 4.538: Simplified automaton for the MAX_MAX_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

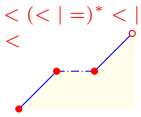
Table 4.21: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MAX_MAX_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.22: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MAX_MAX_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

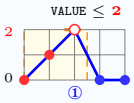
Constraint MAX_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.539 provides an example where the MAX_MAX_INCREASING_SEQUENCE $(6, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

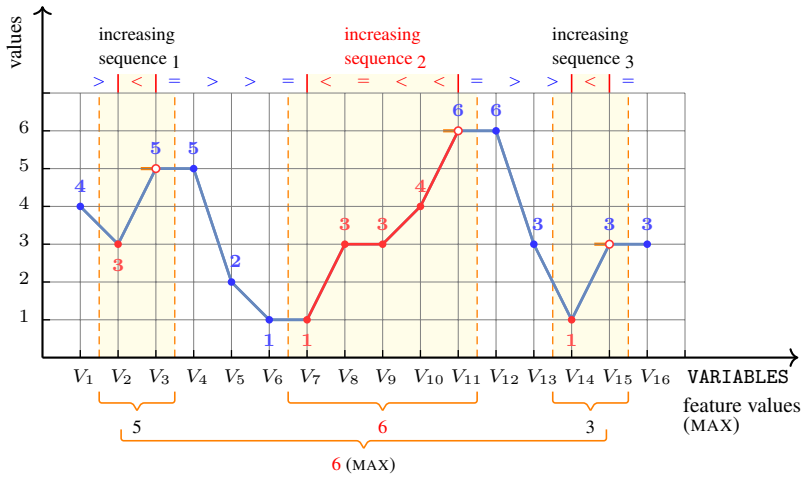


Figure 4.539: Illustrating the MAX_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.540 and 4.541 respectively depict the automaton associated with the constraint MAX_MAX_INCREASING_SEQUENCE and its simplified form.

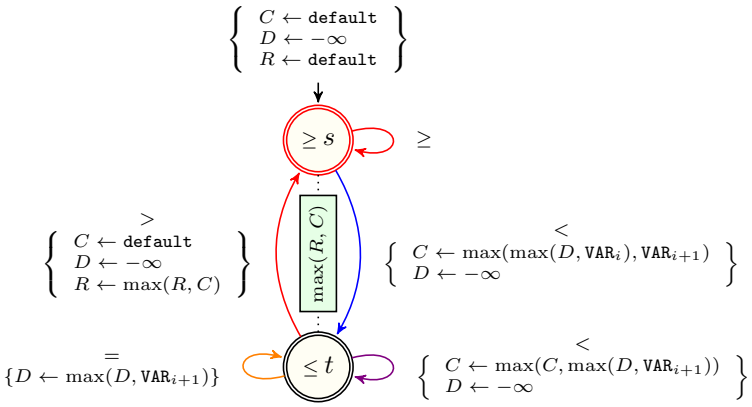


Figure 4.540: Automaton for the MAX_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $-\infty$

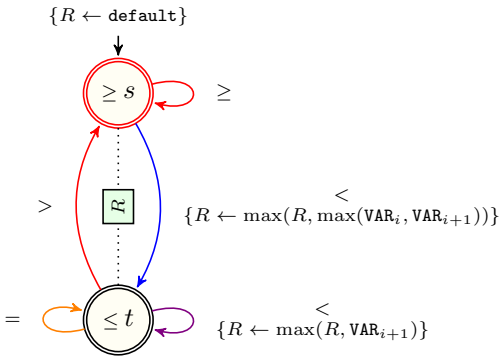


Figure 4.541: Simplified automaton for the MAX_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.23: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MAX_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

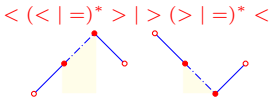
	s	t
s	$-\infty$	$-\infty$
t	$-\infty$	$-\infty$ M

Table 4.24: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the MAX_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

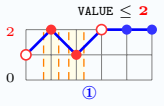
Constraint MAX_MAX_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(6, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.542 provides an example where the MAX_MAX_INFLEXION $(6, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

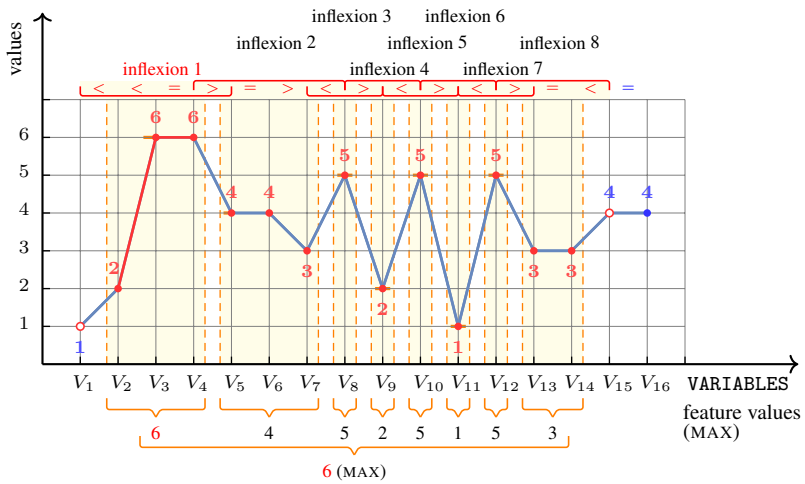


Figure 4.542: Illustrating the MAX_MAX_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.543 and 4.544 respectively depict the automaton associated with the constraint MAX_MAX_INFLEXION and its simplified form.

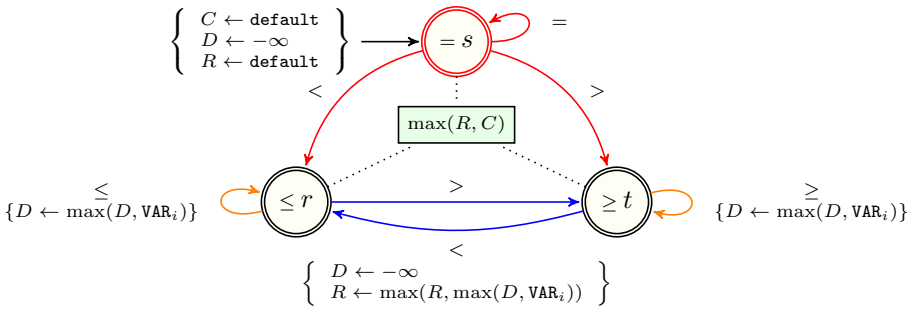


Figure 4.543: Automaton for the MAX_MAX_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

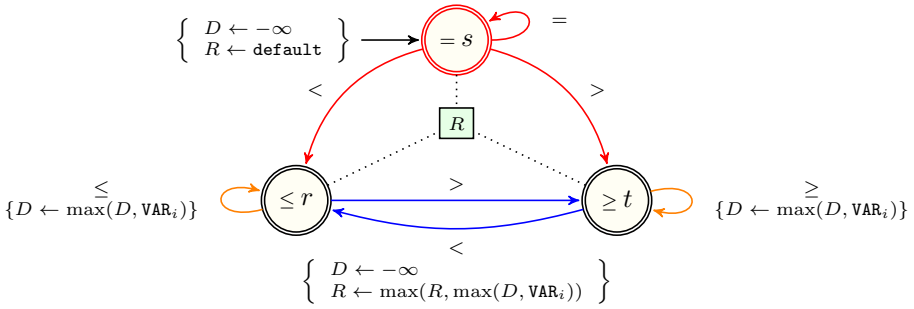
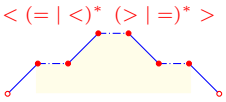


Figure 4.544: Simplified automaton for the MAX_MAX_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MAX_PEAK



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PEAK pattern.

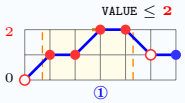
Constraint MAX_MAX_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all maximum values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 1$ to index j.

Example $(6, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.545 provides an example where the MAX_MAX_PEAK $(6, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

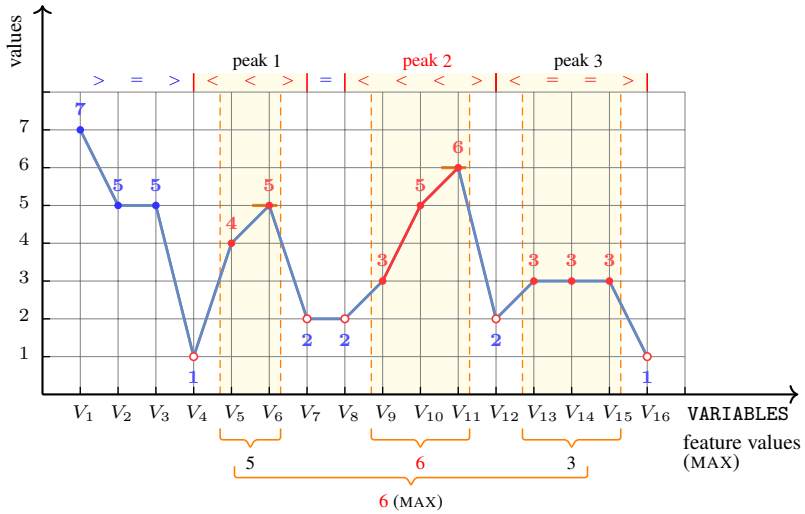


Figure 4.545: Illustrating the MAX_MAX_PEAK constraint of the **Example** slot

Automaton

Figures 4.546 and 4.547 respectively depict the automaton associated with the constraint MAX_MAX_PEAK and its simplified form.

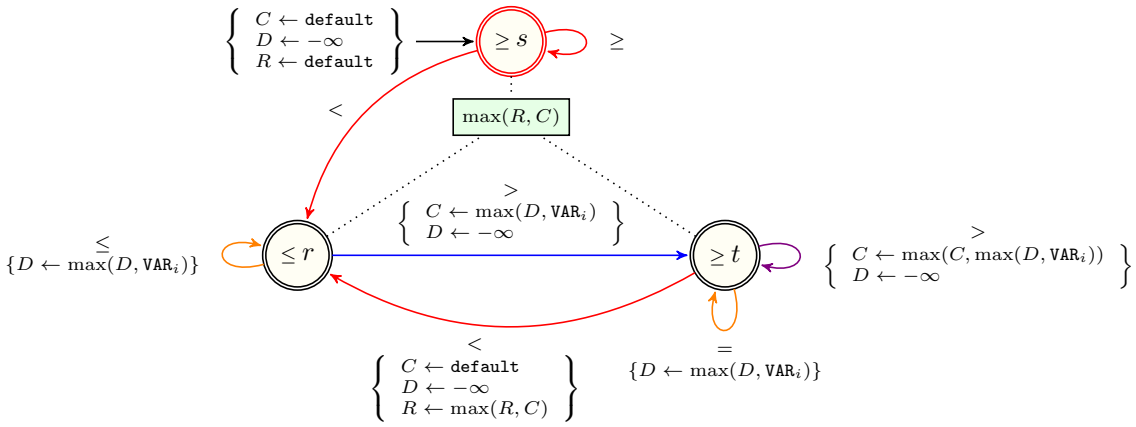


Figure 4.546: Automaton for the MAX_MAX_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is $-\infty$

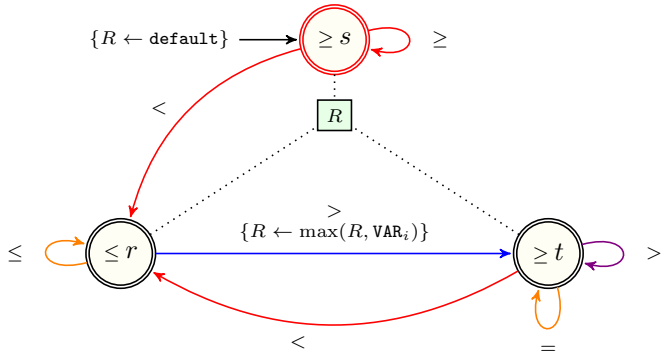


Figure 4.547: Simplified automaton for the MAX_MAX_PEAK constraint obtained by applying decoration Table 3.39 to the seed transducer of the PEAK pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.25: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MAX_MAX_PEAK constraint defined as the composition of the PEAK pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\texttt{VAR}_{i+1}$ C	$-\infty$ R
t	$-\infty$	$-\infty$ L	$-\infty$

Table 4.26: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the MAX_MAX_PEAK constraint defined as the composition of the PEAK pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

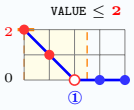
Constraint MAX_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$. An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3\rangle)$

Figure 4.548 provides an example where the MAX_MAX_STRICTLY_DECREASING_SEQUENCE $(6, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

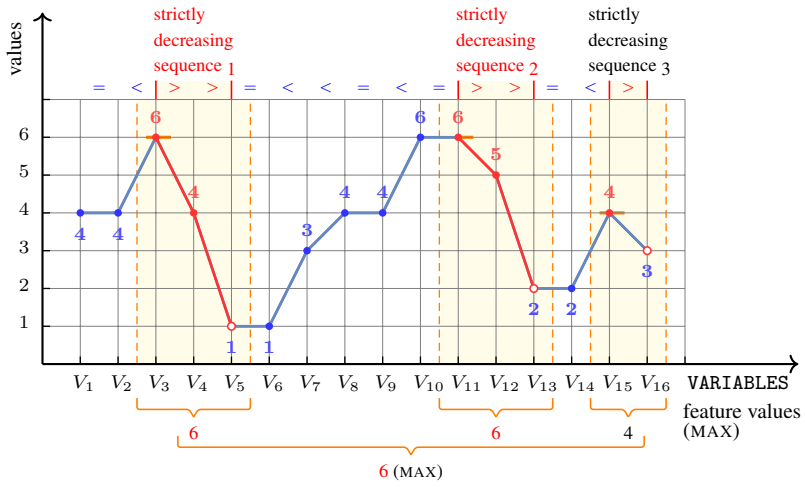


Figure 4.548: Illustrating the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.549 and 4.550 respectively depict the automaton associated with the constraint MAX_MAX_STRICTLY_DECREASING_SEQUENCE and its simplified form.

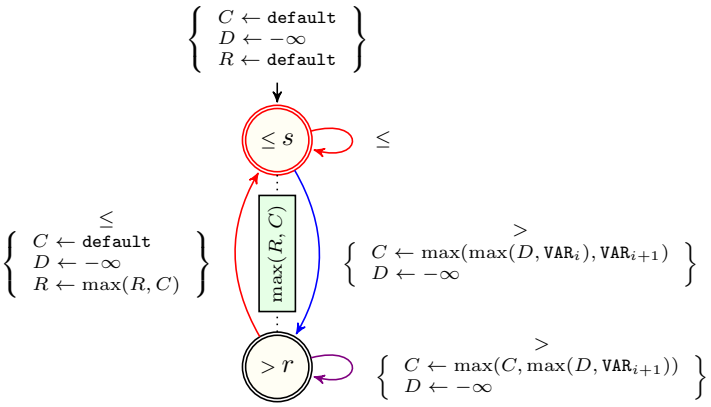


Figure 4.549: Automaton for the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$

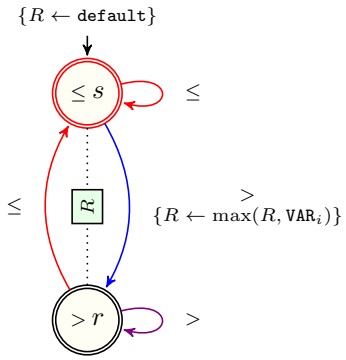


Figure 4.550: Simplified automaton for the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.27: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$ M

Table 4.28: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

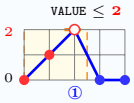
Constraint MAX_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all maximum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.551 provides an example where the MAX_MAX_STRICTLY_INCREASING_SEQUENCE $(6, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

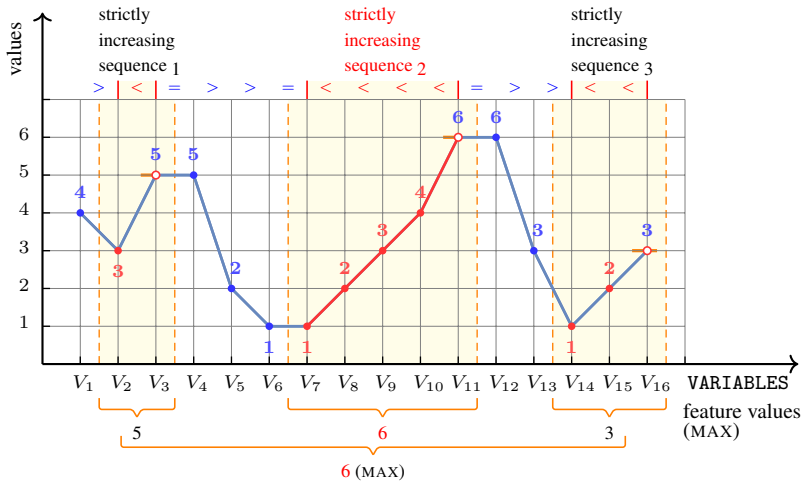


Figure 4.551: Illustrating the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.552 and 4.553 respectively depict the automaton associated with the constraint MAX_MAX_STRICTLY_INCREASING_SEQUENCE and its simplified form.

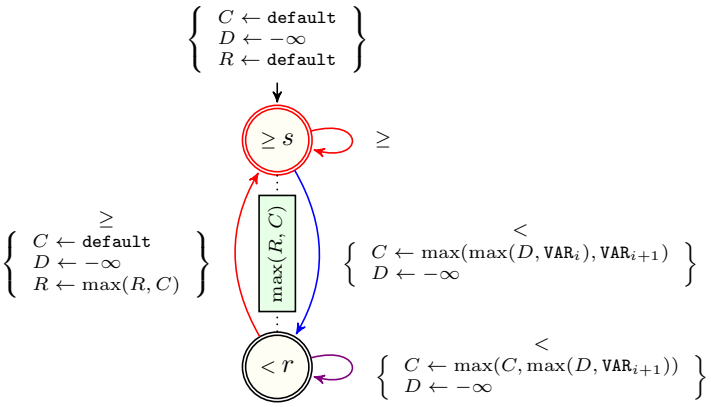


Figure 4.552: Automaton for the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$

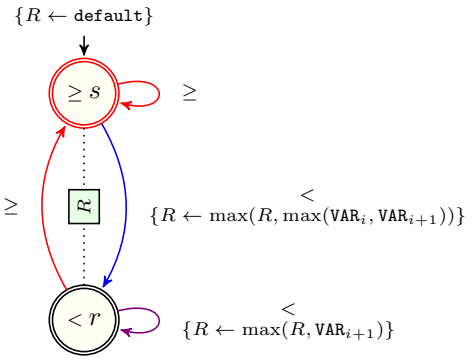


Figure 4.553: Simplified automaton for the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.29: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

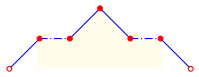
	s	r
s	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$ M

Table 4.30: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the SUMMIT pattern.

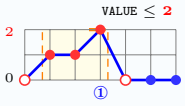
Constraint MAX_MAX_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all maximum values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(5, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.554 provides an example where the MAX_MAX_SUMMIT $(5, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

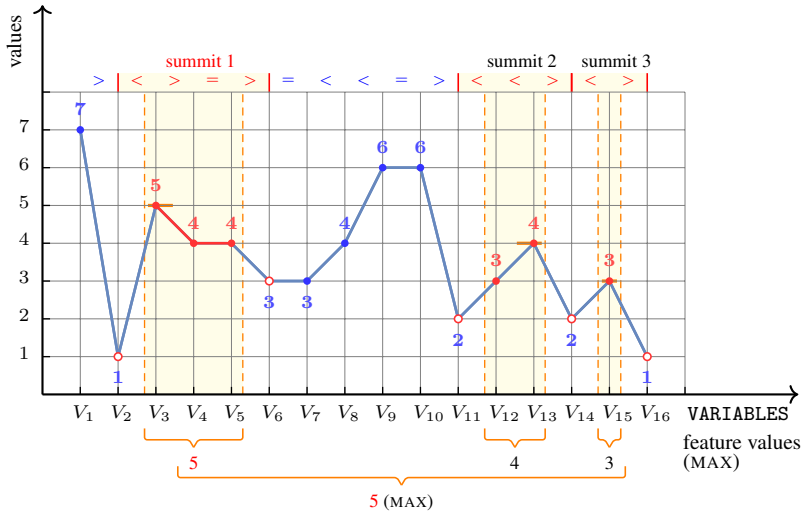


Figure 4.554: Illustrating the MAX_MAX_SUMMIT constraint of the **Example** slot

Automaton Figures 4.555 and 4.556 respectively depict the automaton associated with the constraint MAX_MAX_SUMMIT and its simplified form.

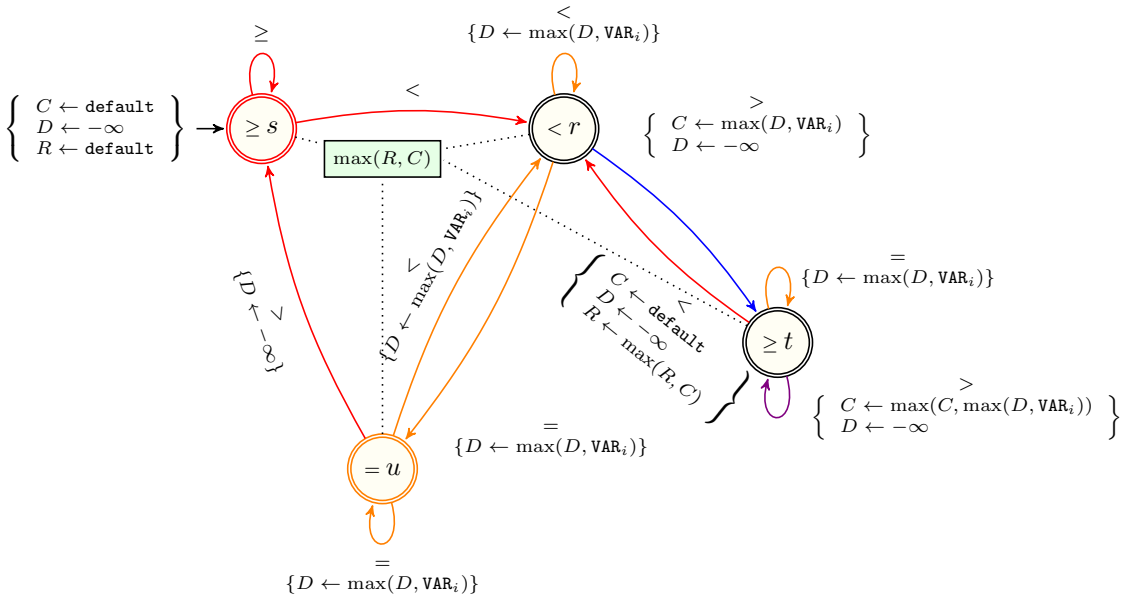


Figure 4.555: Automaton for the MAX_MAX_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is $-\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.31: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MAX_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

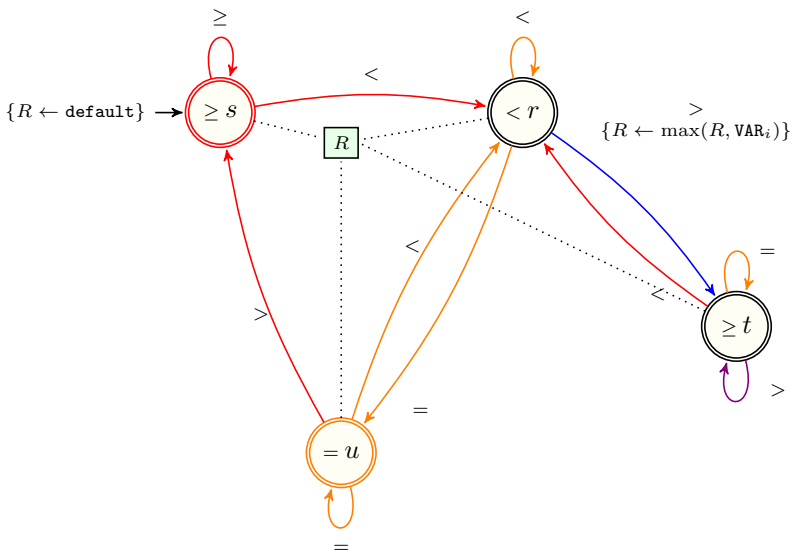


Figure 4.556: Simplified automaton for the MAX_MAX_SUMMIT constraint obtained by applying decoration Table 3.39 to the seed transducer of the SUMMIT pattern where `default` is $-\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

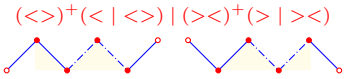
	s	r	t	u
s	$-\infty$	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\texttt{VAR}_{i+1}$ C	$-\infty$ R	$-\infty$
t	$-\infty$	$-\infty$ L	$-\infty$	$-\infty$ L
u	$-\infty$	$-\infty$	$-\infty$ R	$-\infty$

Table 4.32: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the MAX_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MAX_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint MAX_MAX_ZIGZAG(VALUE, VARIABLES)

Arguments

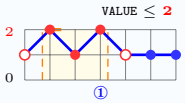
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

sv ≤ 3 ∨ rv ≤ 1 ⇒ VALUE = −∞
VALUE = −∞ ∨ VALUE ≥ minv + 1
VALUE ≤ maxv①
required(VARIABLES, var)

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all maximum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value −∞.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '(<>)+(< | <>) | (><)+(> | ><)'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 1$ to index j.

Example $(7, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1\rangle)$

Figure 4.557 provides an example where the MAX_MAX_ZIGZAG $(7, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical |VARIABLES| > 3
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

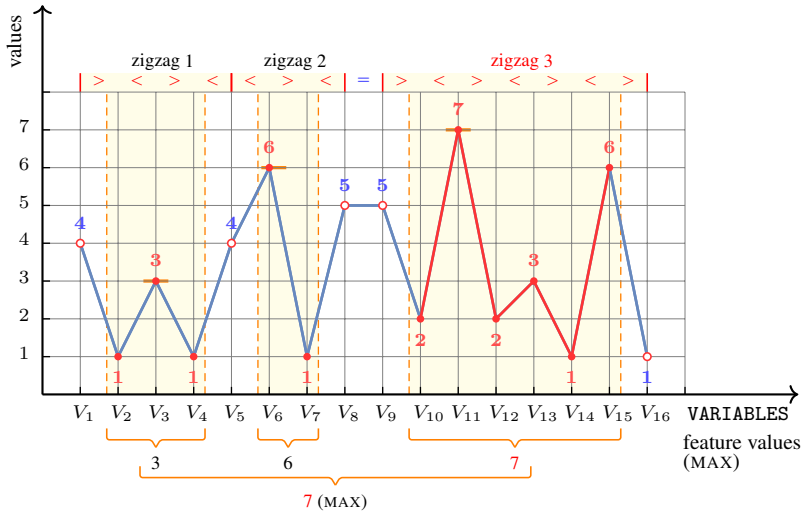


Figure 4.557: Illustrating the MAX_MAX_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.558 and 4.559 respectively depict the automaton associated with the constraint MAX_MAX_ZIGZAG and its simplified form.

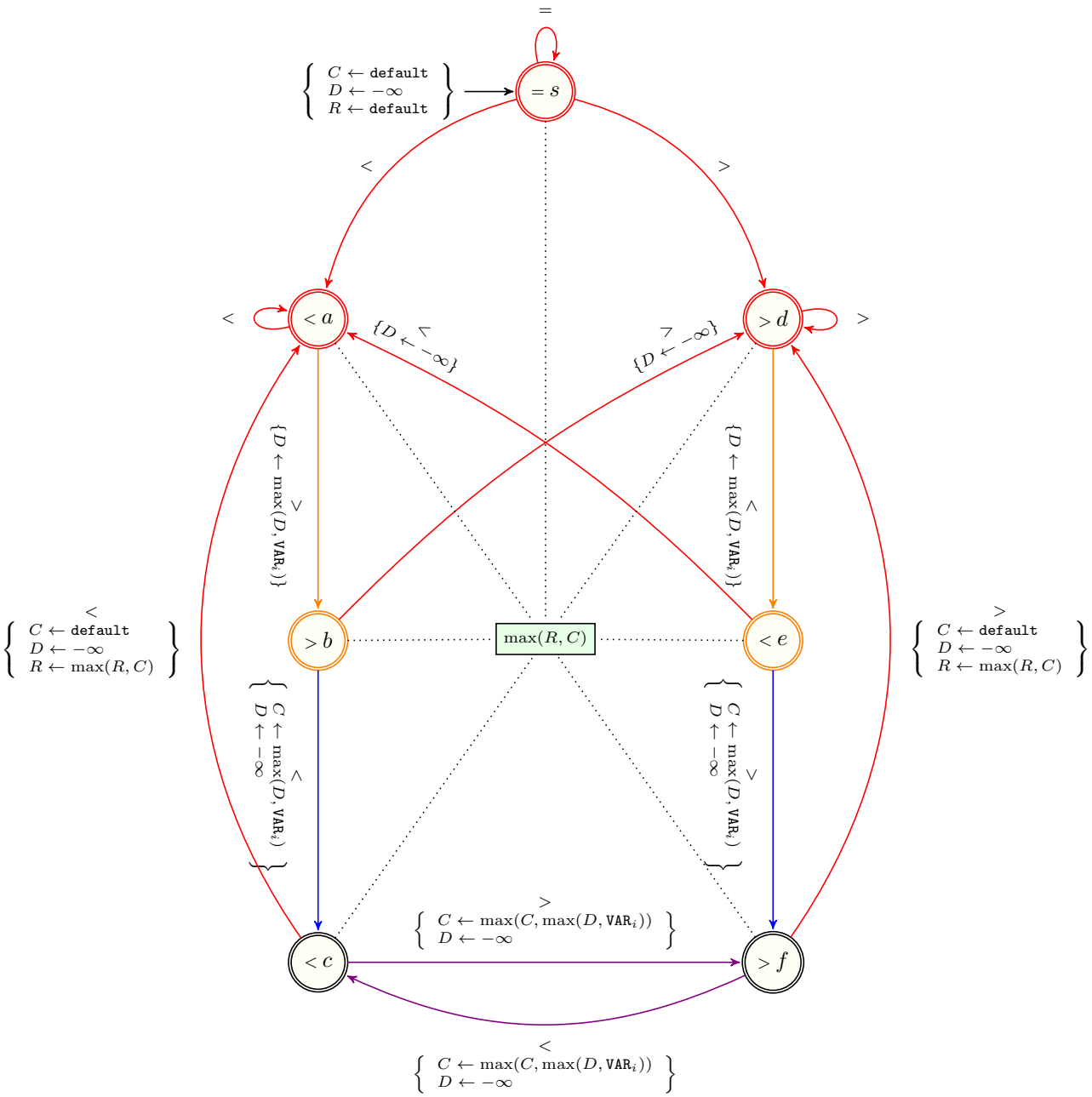


Figure 4.558: Automaton for the MAX_MAX_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

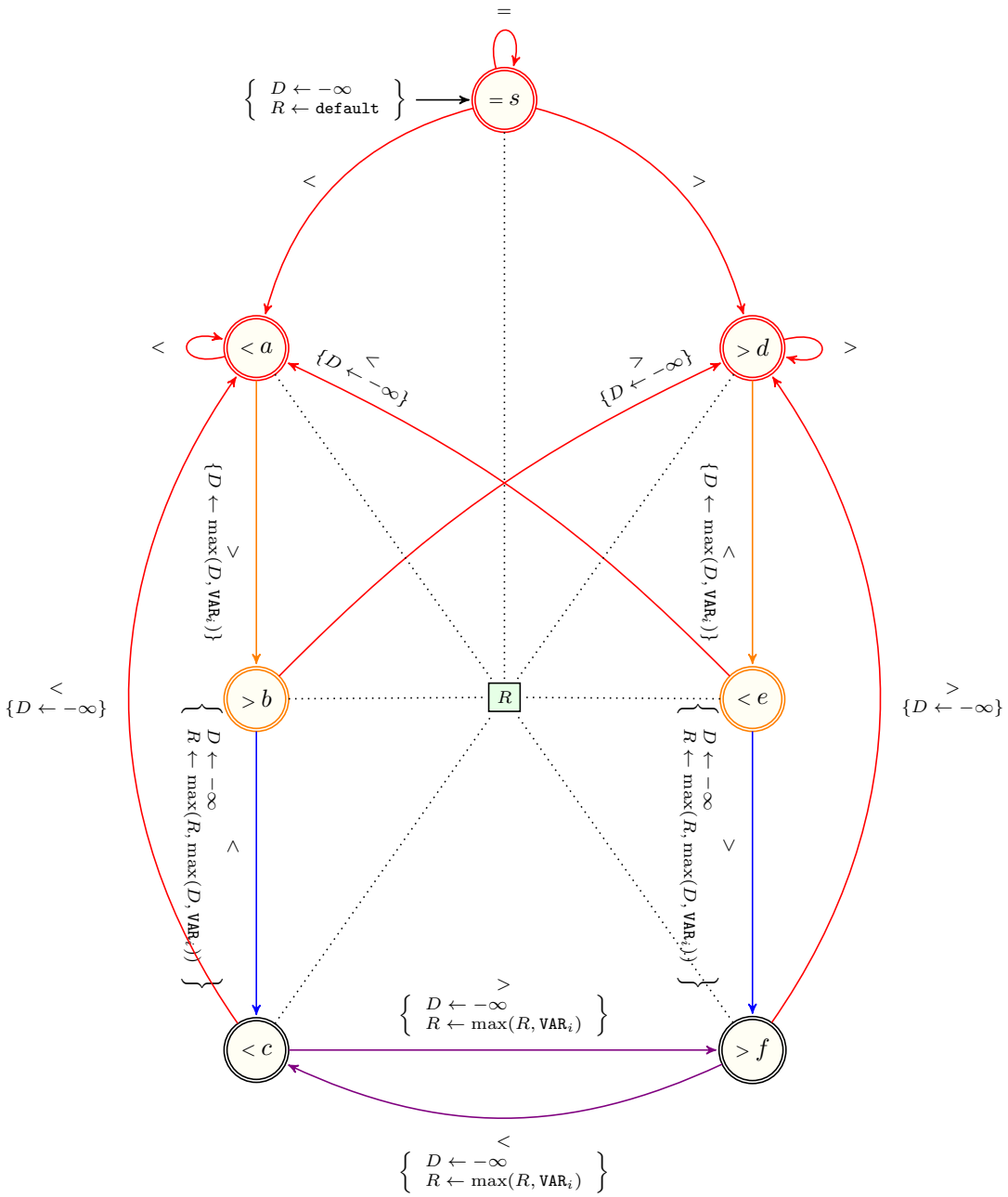


Figure 4.559: Simplified automaton for the MAX_MAX_ZIGZAG constraint obtained by applying decoration Table 3.29 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; $R_i - R_{i-1} \geq 0$ is a linear invariant.

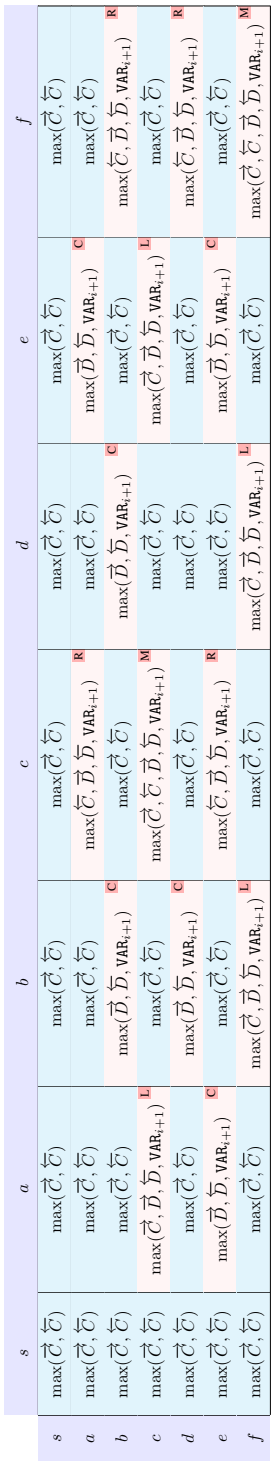


Table 4.33: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MAX_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MAX, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

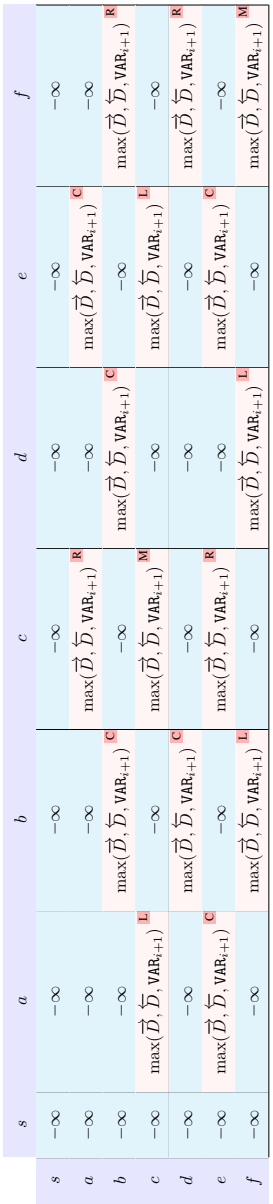
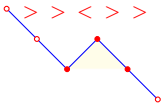


Table 4.34: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MAX_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern , the feature MAX , and the aggregator max ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

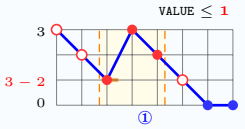
Constraint MAX_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 2$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$. An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(5, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.560 provides an example where the MAX_MIN_BUMP_ON_DECREASING_SEQUENCE $(5, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: VALUE determined by VARIABLES.

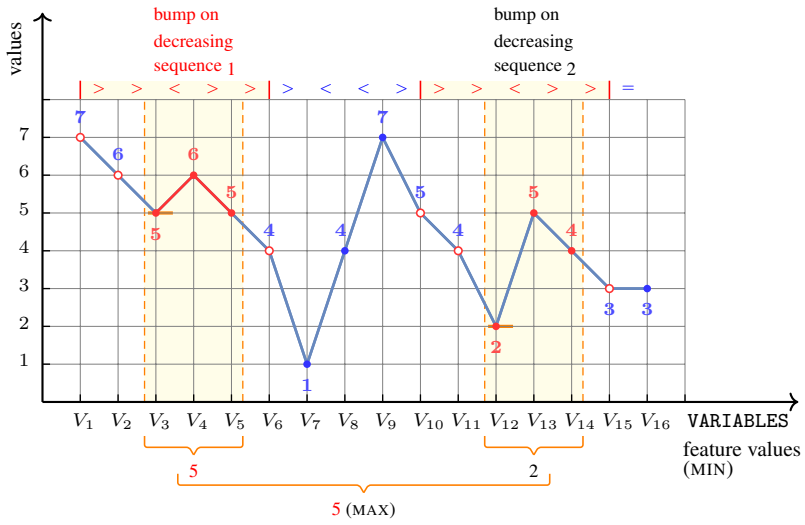


Figure 4.560: Illustrating the MAX_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.561 and 4.562 respectively depict the automaton associated with the constraint MAX_MIN_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

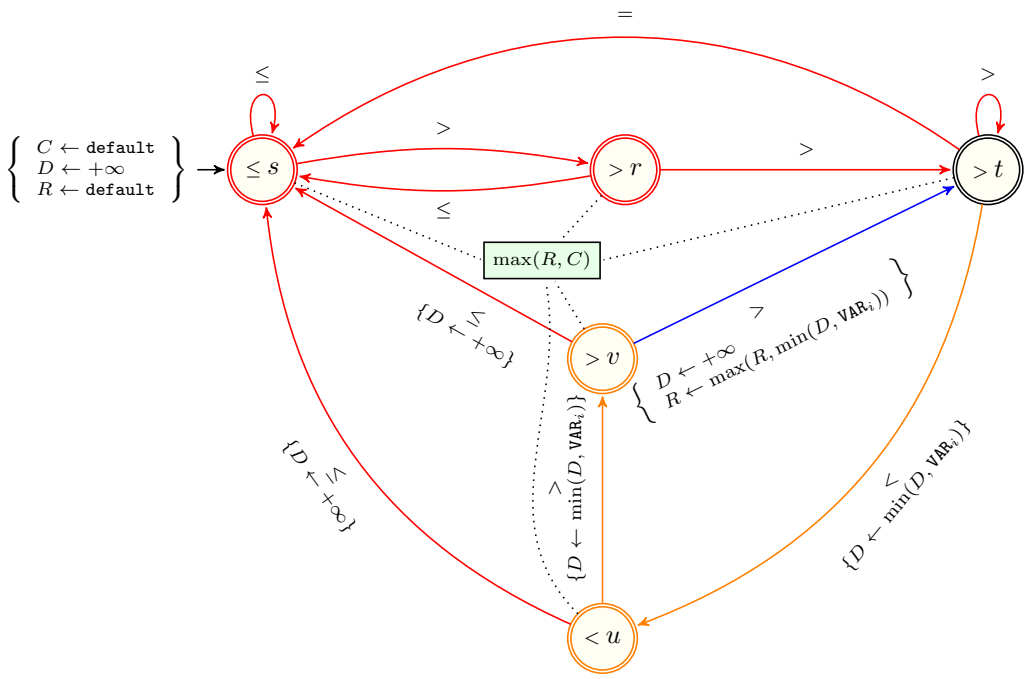


Figure 4.561: Automaton for the MAX_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$

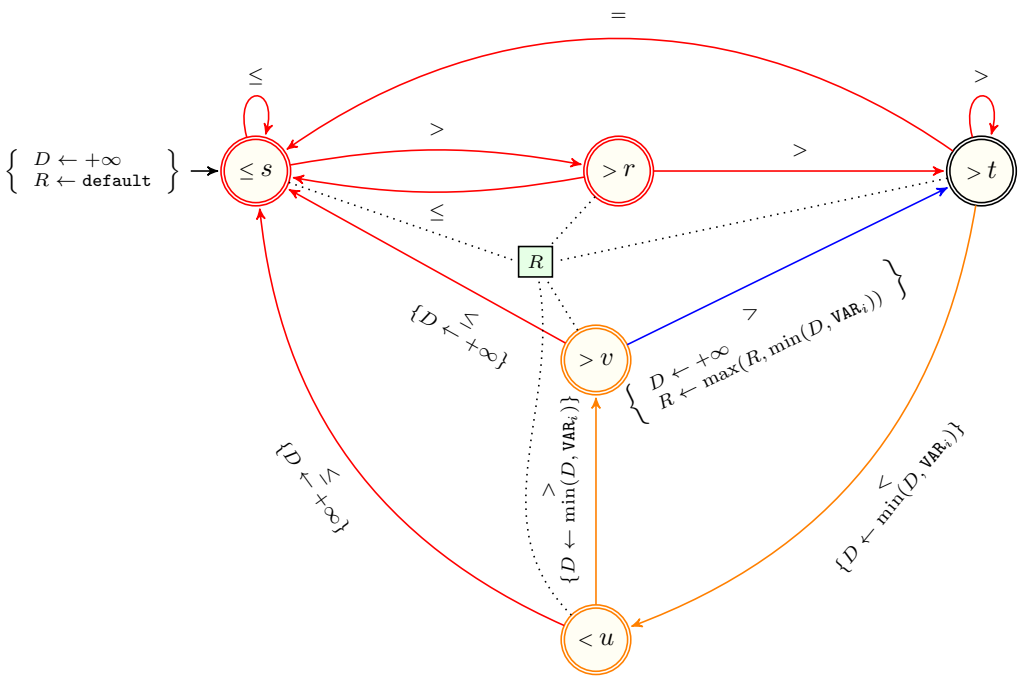


Figure 4.562: Simplified automaton for the MAX_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

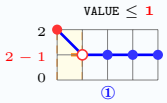
Constraint MAX_MIN_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

sv ≤ 1 ∨ rv ≤ 1 ⇒ VALUE = −∞
VALUE = −∞ ∨ VALUE ≥ minv
VALUE ≤ maxv − 1①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(4, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle)$

Figure 4.563 provides an example where the MAX_MIN_DECREASING $(4, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4])$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

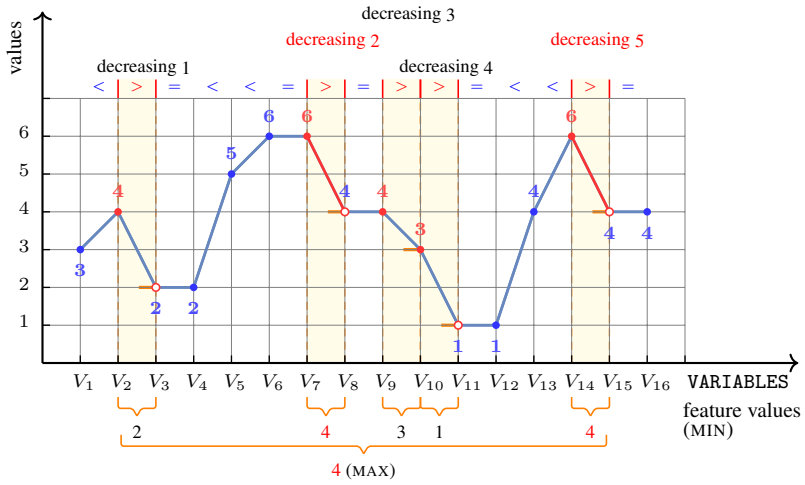


Figure 4.563: Illustrating the MAX_MIN_DECREASING constraint of the **Example** slot

Automaton

Figures 4.564 and 4.565 respectively depict the automaton associated with the constraint MAX_MIN_DECREASING and its simplified form.

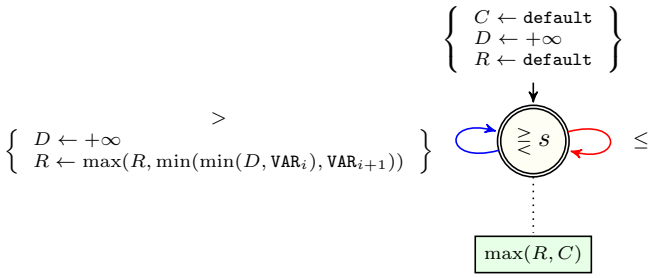


Figure 4.564: Automaton for the MAX_MIN_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $-\infty$

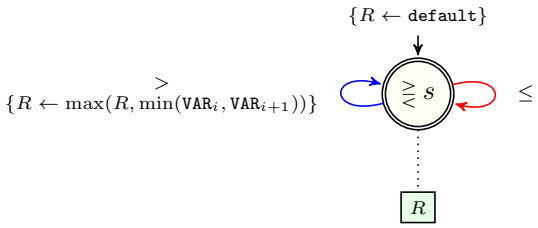


Figure 4.565: Simplified automaton for the MAX_MIN_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

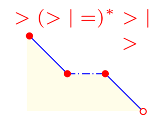
Table 4.35: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MAX_MIN_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.36: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MAX_MIN_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

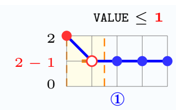
Constraint MAX_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv}$
$\mathtt{VALUE} \leq \mathtt{maxv} - 1$ ①
required(VARIABLES, var)
where
maxv = maxval(VARIABLES.var)
minv = minval(VARIABLES.var)
sv = |VARIABLES|
rv = range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$(4, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.566 provides an example where the MAX_MIN_DECREASING_SEQUENCE $(4, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

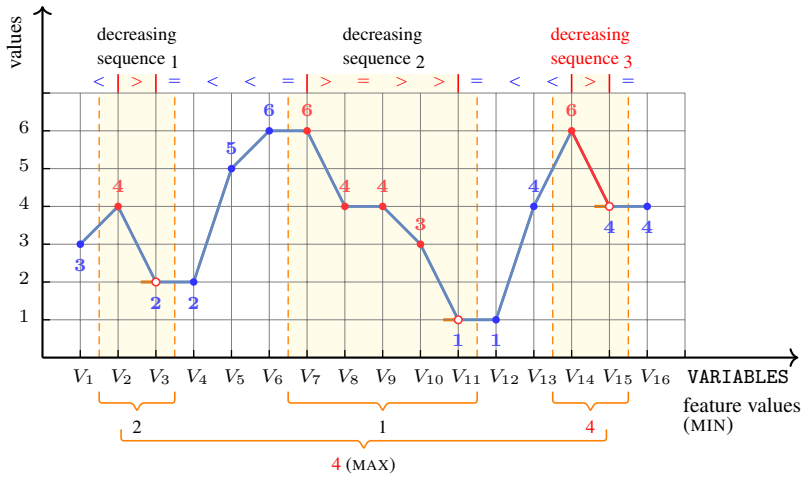


Figure 4.566: Illustrating the MAX_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.567 and 4.568 respectively depict the automaton associated with the constraint MAX_MIN_DECREASING_SEQUENCE and its simplified form.

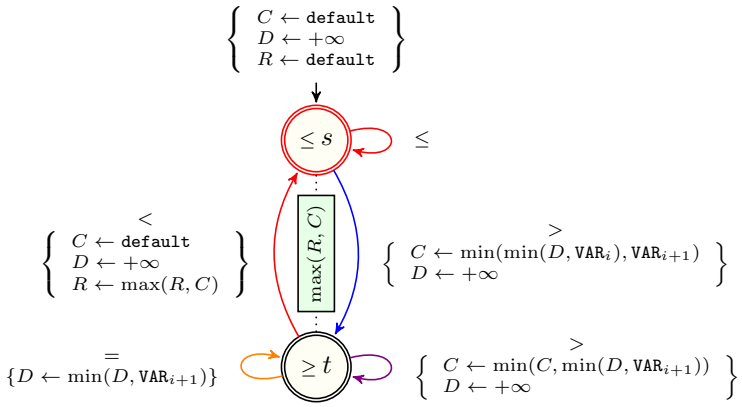


Figure 4.567: Automaton for the MAX_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $-\infty$

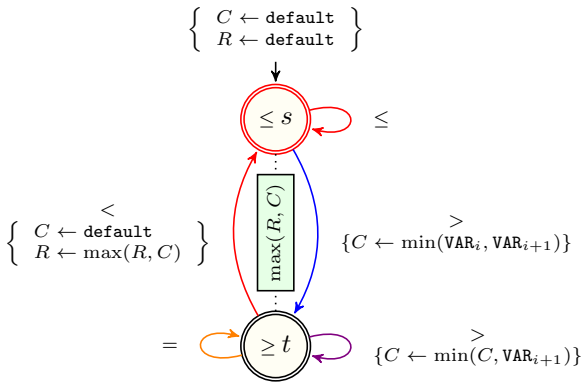


Figure 4.568: Simplified automaton for the MAX_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

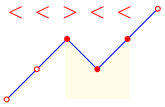
Table 4.37: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MAX_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	$-\infty$	$-\infty$
t	$\overrightarrow{C}$	$-\infty$ M

Table 4.38: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the MAX_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

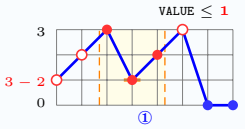
Constraint MAX_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 2$①
required(VARIABLES, var)
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(2, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.569 provides an example where the MAX_MIN_DIP_ON_INCREASING_SEQUENCE $(2, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

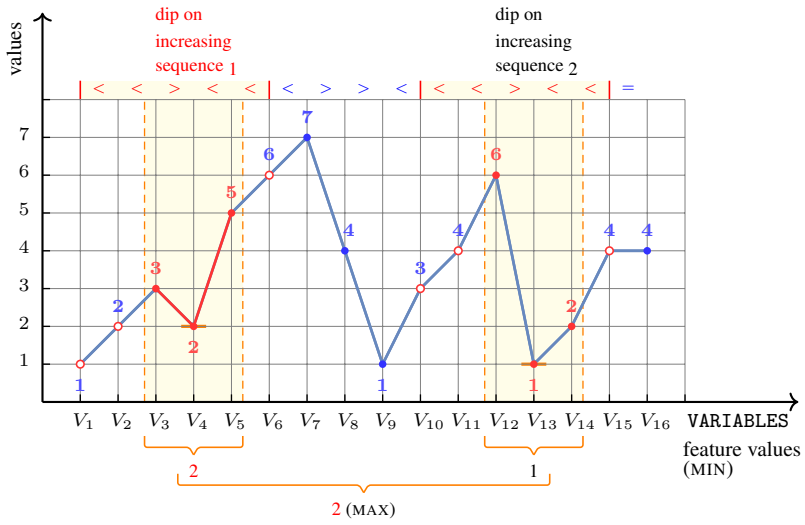


Figure 4.569: Illustrating the MAX_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.570 and 4.571 respectively depict the automaton associated with the constraint MAX_MIN_DIP_ON_INCREASING_SEQUENCE and its simplified form.

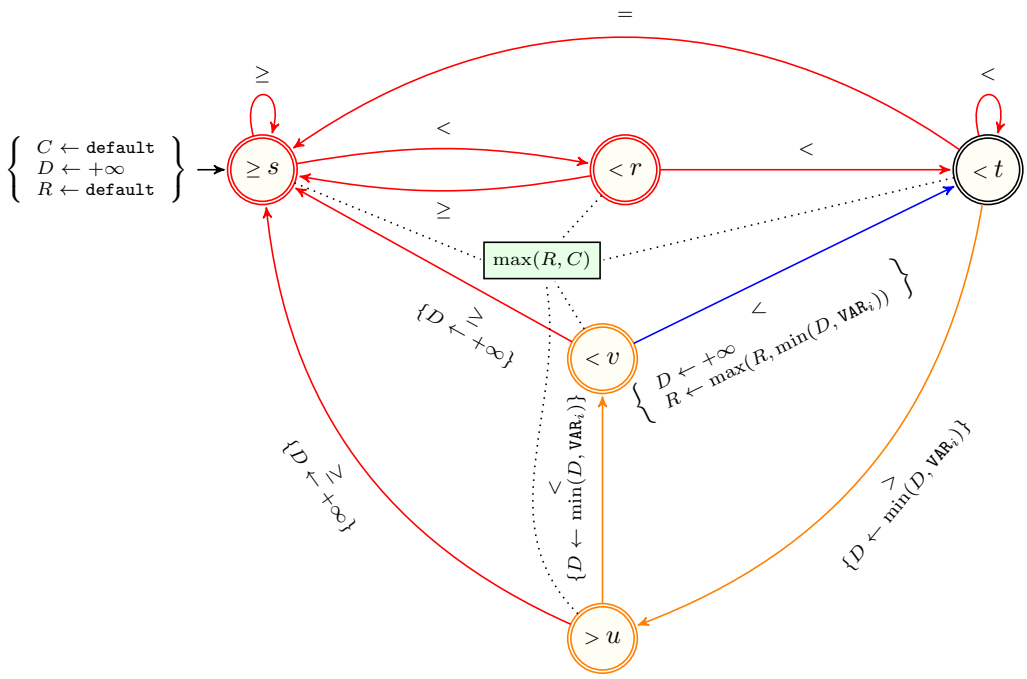


Figure 4.570: Automaton for the MAX_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$

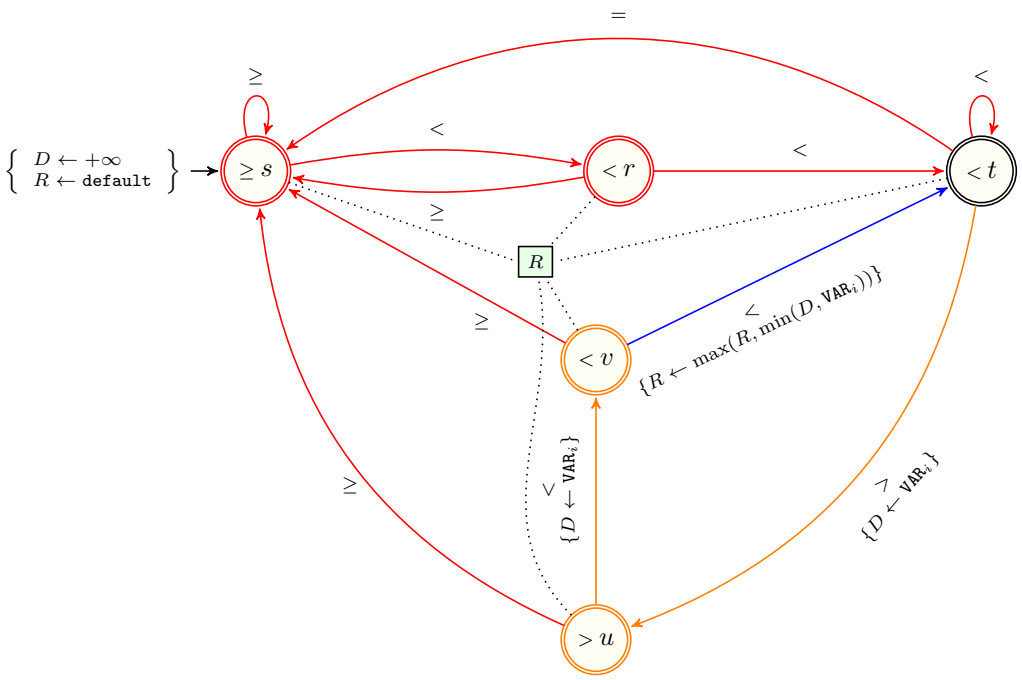
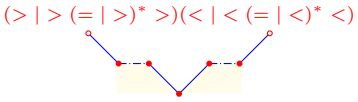


Figure 4.571: Simplified automaton for the MAX_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MIN_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint MAX_MIN_GORGE(VALUE, VARIABLES)

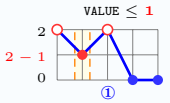
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)

where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximum of all minimum values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.572 provides an example where the MAX_MIN_GORGE $(5, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

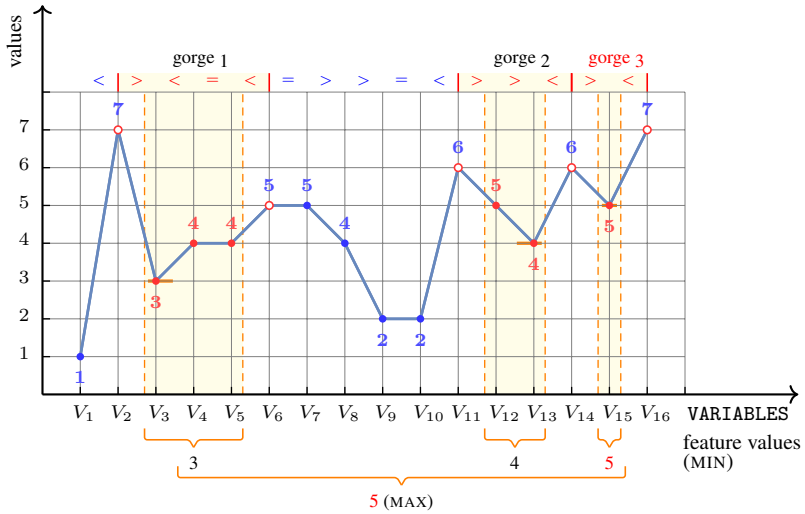


Figure 4.572: Illustrating the MAX_MIN_GORGE constraint of the **Example** slot

Automaton Figures 4.573 and 4.574 respectively depict the automaton associated with the constraint MAX_MIN_GORGE and its simplified form.

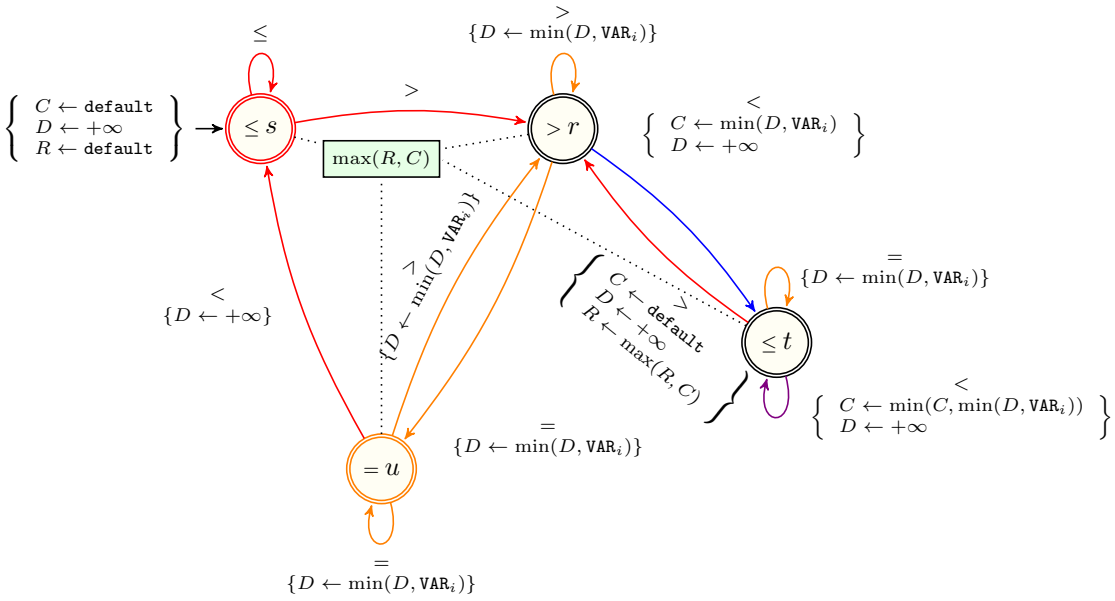


Figure 4.573: Automaton for the MAX_MIN_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is $-\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

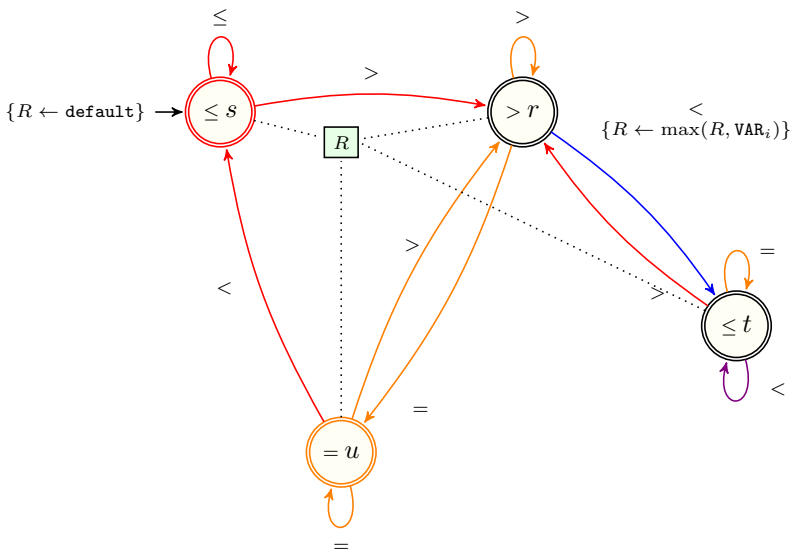


Figure 4.574: Simplified automaton for the MAX_MIN_GORGE constraint obtained by applying decoration Table 3.39 to the seed transducer of the GORGE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.39: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MAX_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	$-\infty$	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\texttt{VAR}_{i+1}$ C	$-\infty$ R	$-\infty$
t	$-\infty$	$-\infty$ L	$-\infty$	$-\infty$ L
u	$-\infty$	$-\infty$	$-\infty$ R	$-\infty$

Table 4.40: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the MAX_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

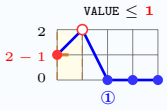
Constraint MAX_MIN_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(4, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.575 provides an example where the MAX_MIN_INCREASING $(4, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

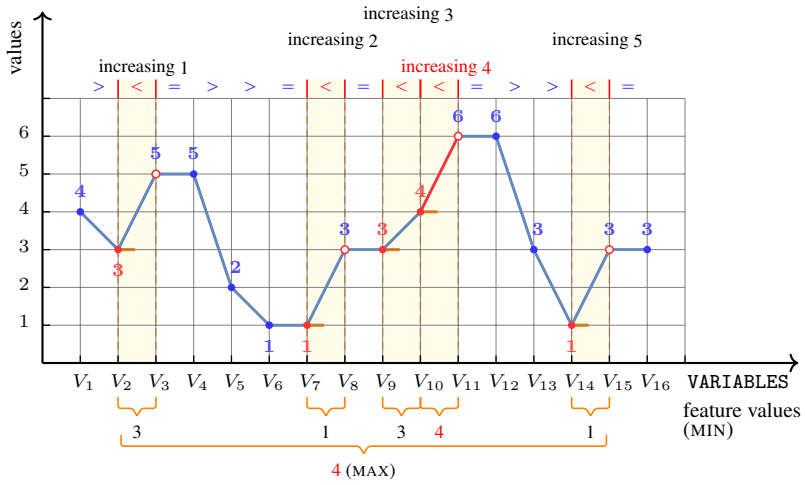


Figure 4.575: Illustrating the MAX_MIN_INCREASING constraint of the **Example** slot

Automaton

Figures 4.576 and 4.577 respectively depict the automaton associated with the constraint MAX_MIN_INCREASING and its simplified form.

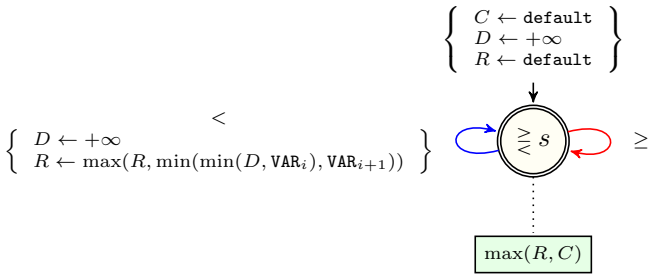


Figure 4.576: Automaton for the MAX_MIN_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $-\infty$

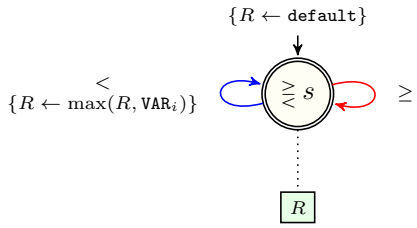


Figure 4.577: Simplified automaton for the MAX_MIN_INCREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

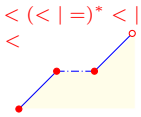
Table 4.41: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MAX_MIN_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.42: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MAX_MIN_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING_SEQUENCE pattern.

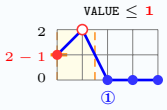
Constraint MAX_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.578 provides an example where the MAX_MIN_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

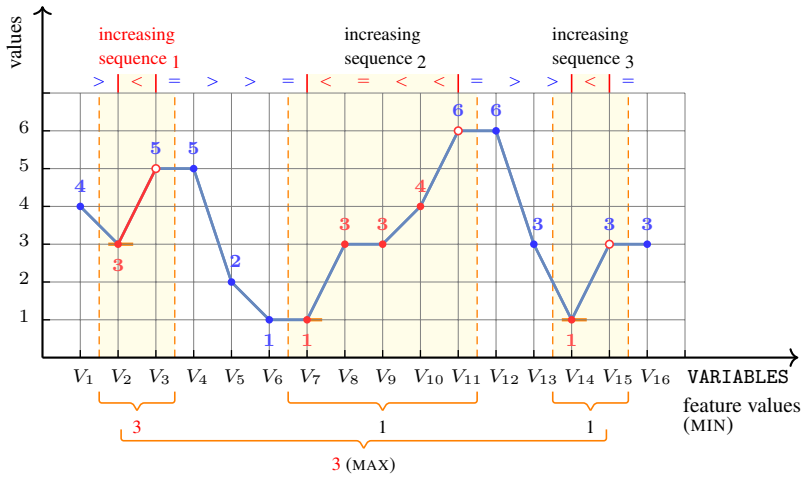


Figure 4.578: Illustrating the MAX_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.579 and 4.580 respectively depict the automaton associated with the constraint MAX_MIN_INCREASING_SEQUENCE and its simplified form.

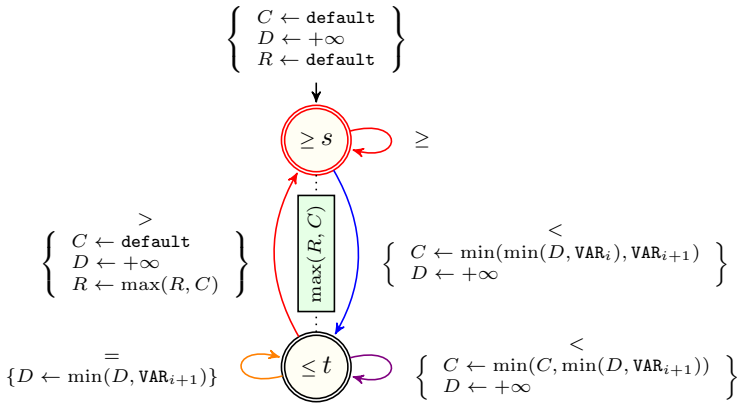


Figure 4.579: Automaton for the MAX_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $-\infty$

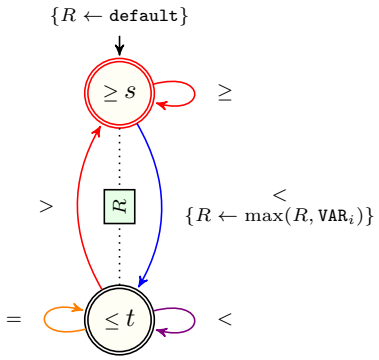


Figure 4.580: Simplified automaton for the MAX_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

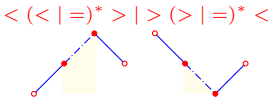
Table 4.43: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MAX_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	$-\infty$	$\overleftarrow{C}$
t	$-\infty$	$-\infty$ M

Table 4.44: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the MAX_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INFLEXION pattern.

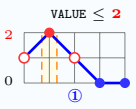
Constraint MAX_MIN_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.581 provides an example where the MAX_MIN_INFLEXION $(5, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

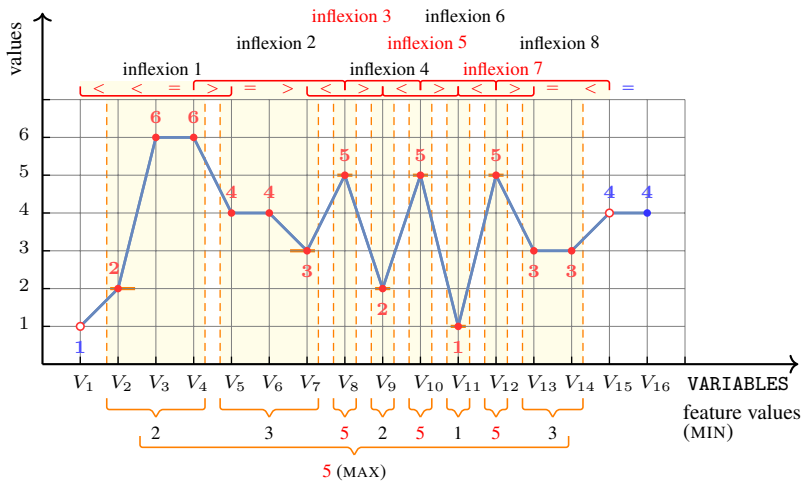


Figure 4.581: Illustrating the MAX_MIN_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.582 and 4.583 respectively depict the automaton associated with the constraint MAX_MIN_INFLEXION and its simplified form.

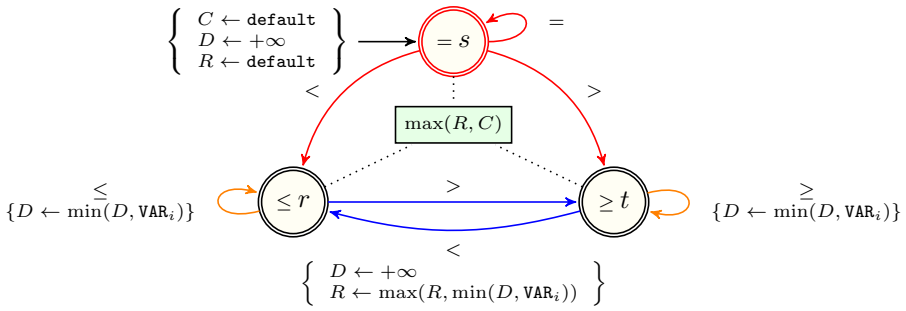


Figure 4.582: Automaton for the MAX_MIN_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

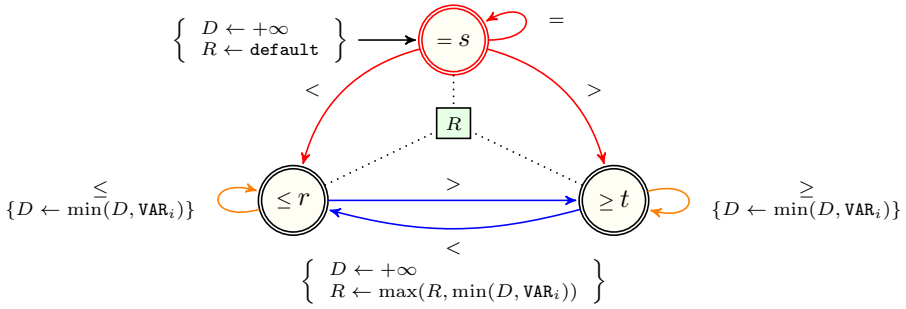


Figure 4.583: Simplified automaton for the MAX_MIN_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

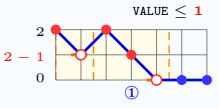
Constraint MAX_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$. An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(3, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3\rangle)$

Figure 4.584 provides an example where the MAX_MIN_STRICTLY_DECREASING_SEQUENCE $(3, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

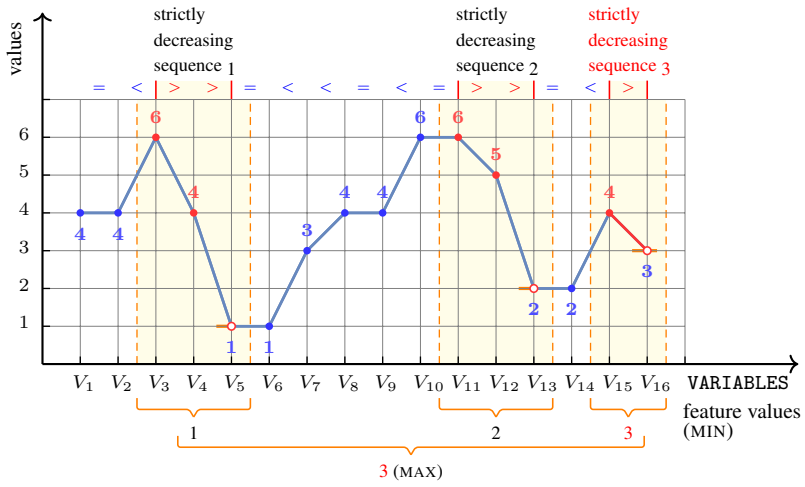


Figure 4.584: Illustrating the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.585 and 4.586 respectively depict the automaton associated with the constraint MAX_MIN_STRICTLY_DECREASING_SEQUENCE and its simplified form.

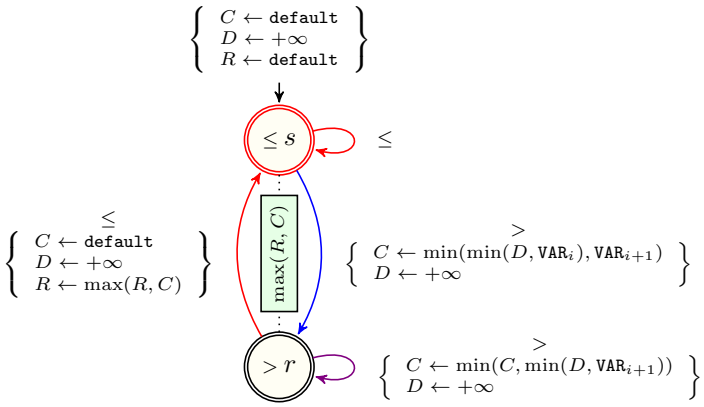


Figure 4.585: Automaton for the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$

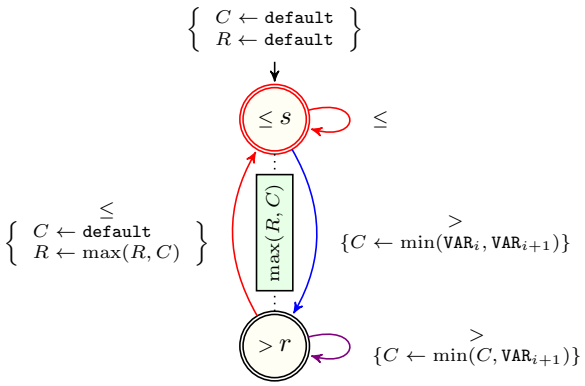


Figure 4.586: Simplified automaton for the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.45: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$-\infty$	$-\infty$
r	$\overrightarrow{C}$	$-\infty$ M

Table 4.46: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

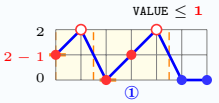
Constraint MAX_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3\rangle)$

Figure 4.587 provides an example where the MAX_MIN_STRICTLY_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

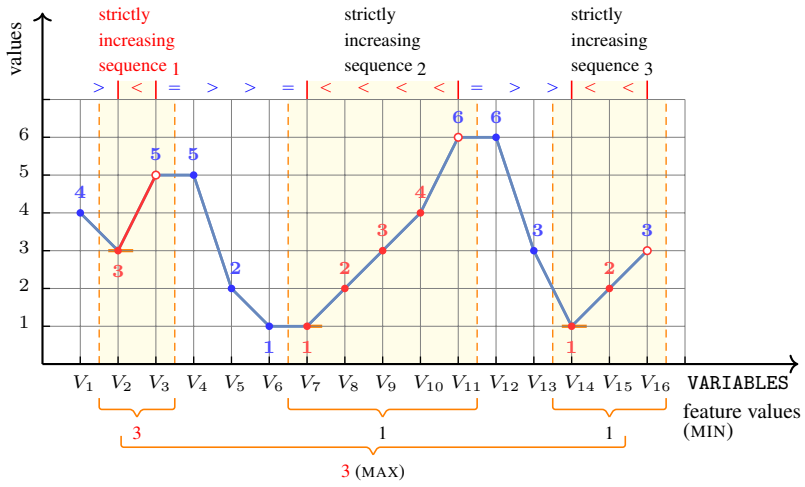


Figure 4.587: Illustrating the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.588 and 4.589 respectively depict the automaton associated with the constraint MAX_MIN_STRICTLY_INCREASING_SEQUENCE and its simplified form.

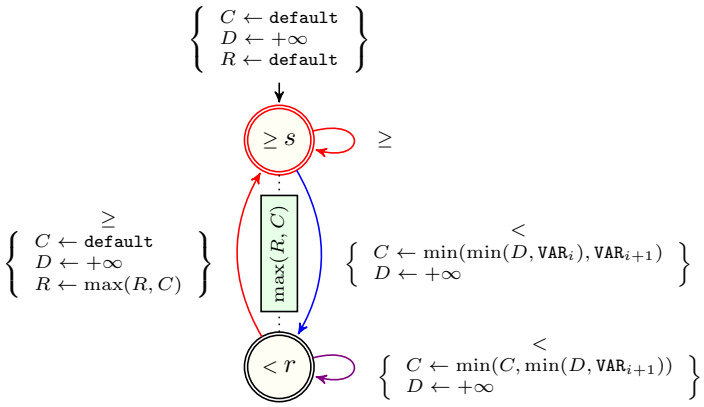


Figure 4.588: Automaton for the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$

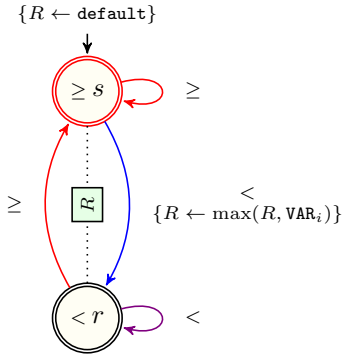


Figure 4.589: Simplified automaton for the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.47: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

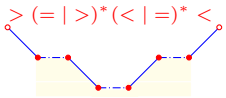
	s	r
s	$-\infty$	$\overleftarrow{C}$
r	$-\infty$	$-\infty$ M

Table 4.48: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

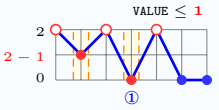
Constraint MAX_MIN_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(5, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.590 provides an example where the MAX_MIN_VALLEY $(5, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

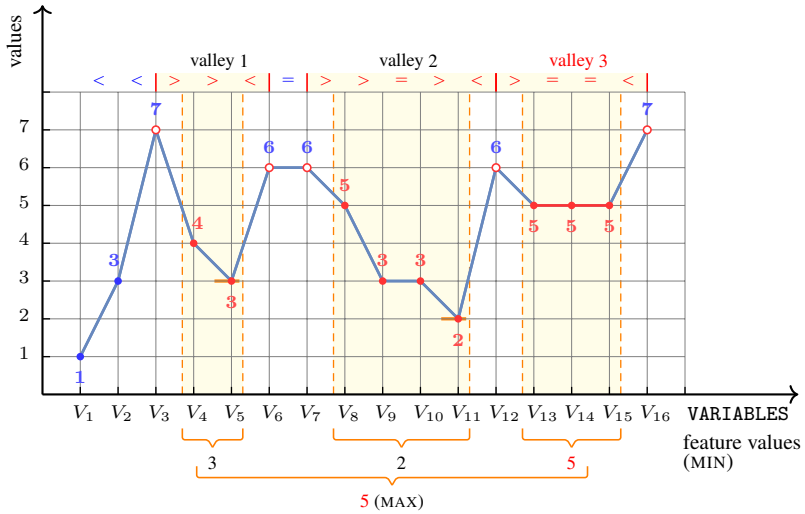


Figure 4.590: Illustrating the MAX_MIN_VALLEY constraint of the **Example** slot

Automaton Figures 4.591 and 4.592 respectively depict the automaton associated with the constraint MAX_MIN_VALLEY and its simplified form.

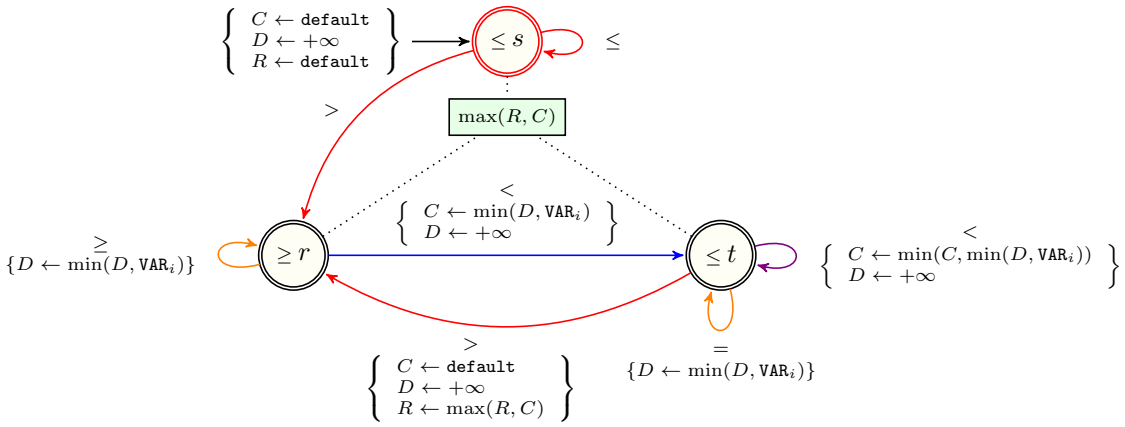


Figure 4.591: Automaton for the MAX_MIN_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is $-\infty$

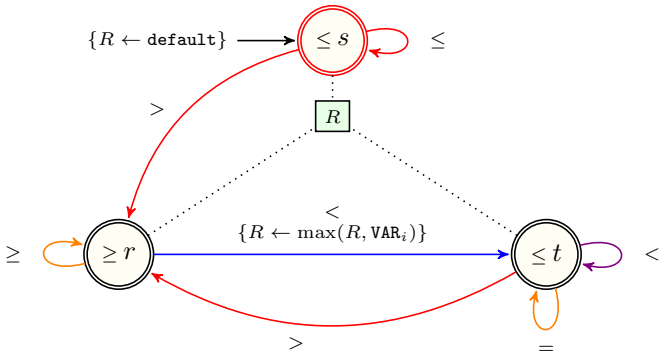


Figure 4.592: Simplified automaton for the MAX_MIN_VALLEY constraint obtained by applying decoration Table 3.39 to the seed transducer of the VALLEY pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.49: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MAX_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

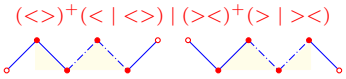
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\texttt{VAR}_{i+1}$ C	$-\infty$ R
t	$-\infty$	$-\infty$ L	$-\infty$

Table 4.50: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the MAX_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_MIN_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint MAX_MIN_ZIGZAG(VALUE, VARIABLES)

Arguments

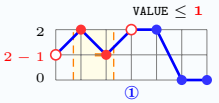
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

sv ≤ 3 ∨ rv ≤ 1 ⇒ VALUE = −∞
VALUE = −∞ ∨ VALUE ≥ minv
VALUE ≤ maxv − 1①
required(VARIABLES, var)

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximum of all minimum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value −∞.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '(<>)+(< | <>) | (><)+(> | ><)'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 1$ to index j.

Example $(1, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1\rangle)$

Figure 4.593 provides an example where the MAX_MIN_ZIGZAG $(1, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical
|VARIABLES| > 3
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

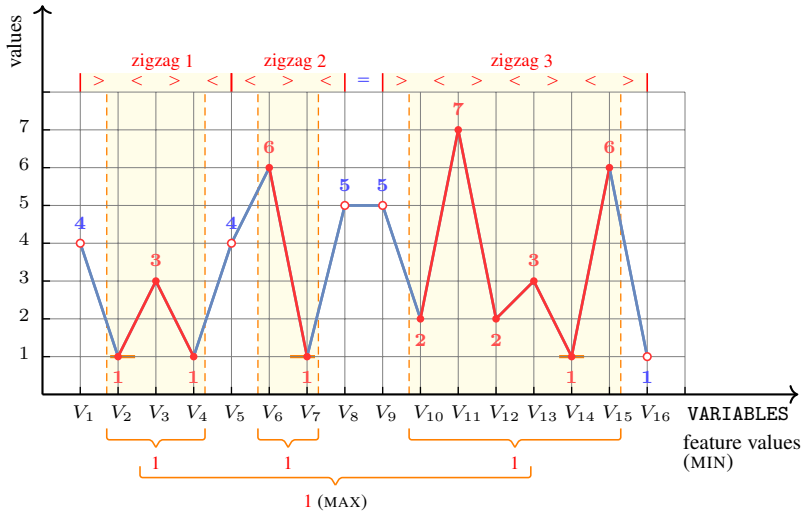


Figure 4.593: Illustrating the MAX_MIN_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.594 and 4.595 respectively depict the automaton associated with the constraint MAX_MIN_ZIGZAG and its simplified form.

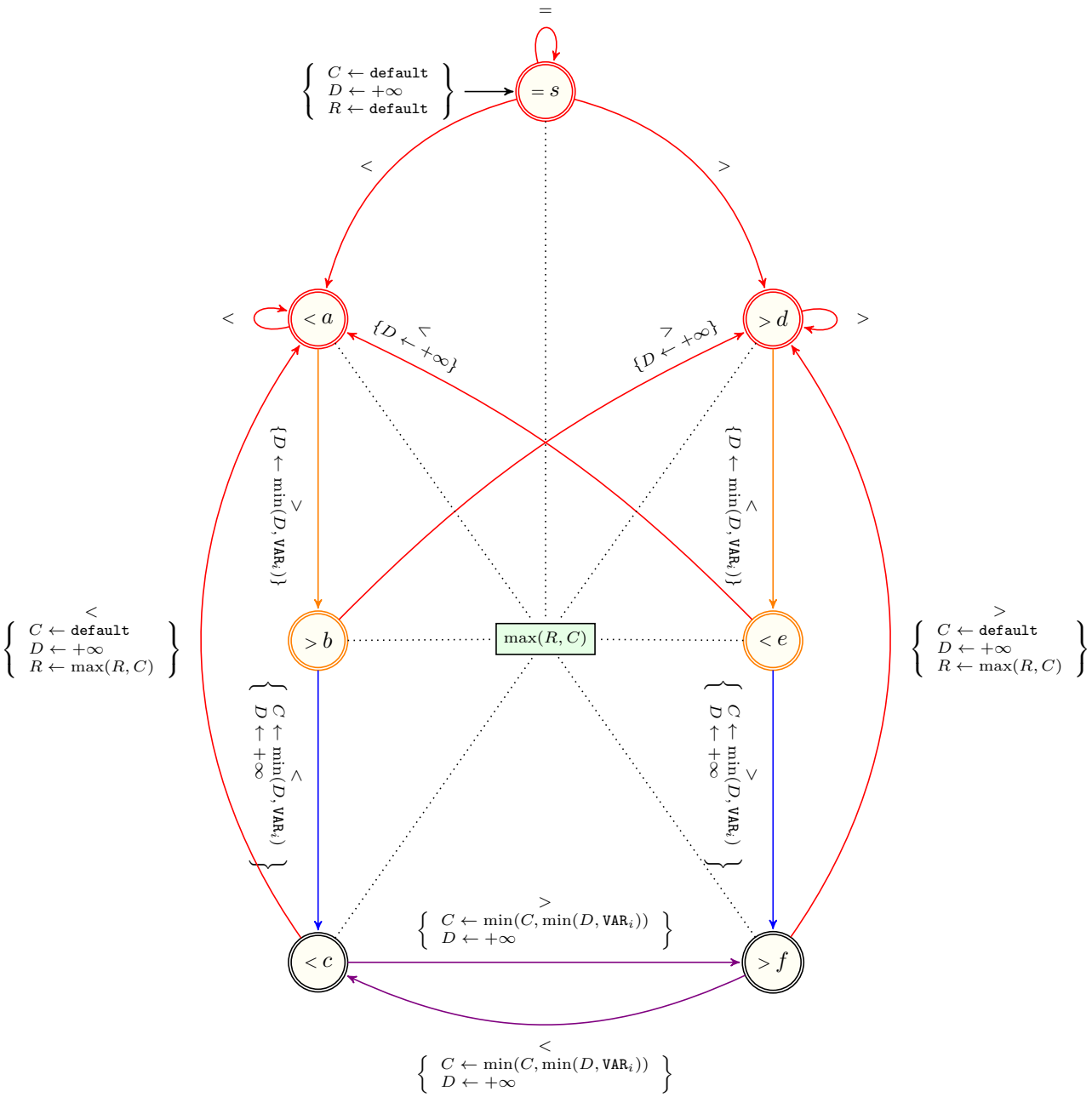


Figure 4.594: Automaton for the MAX_MIN_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

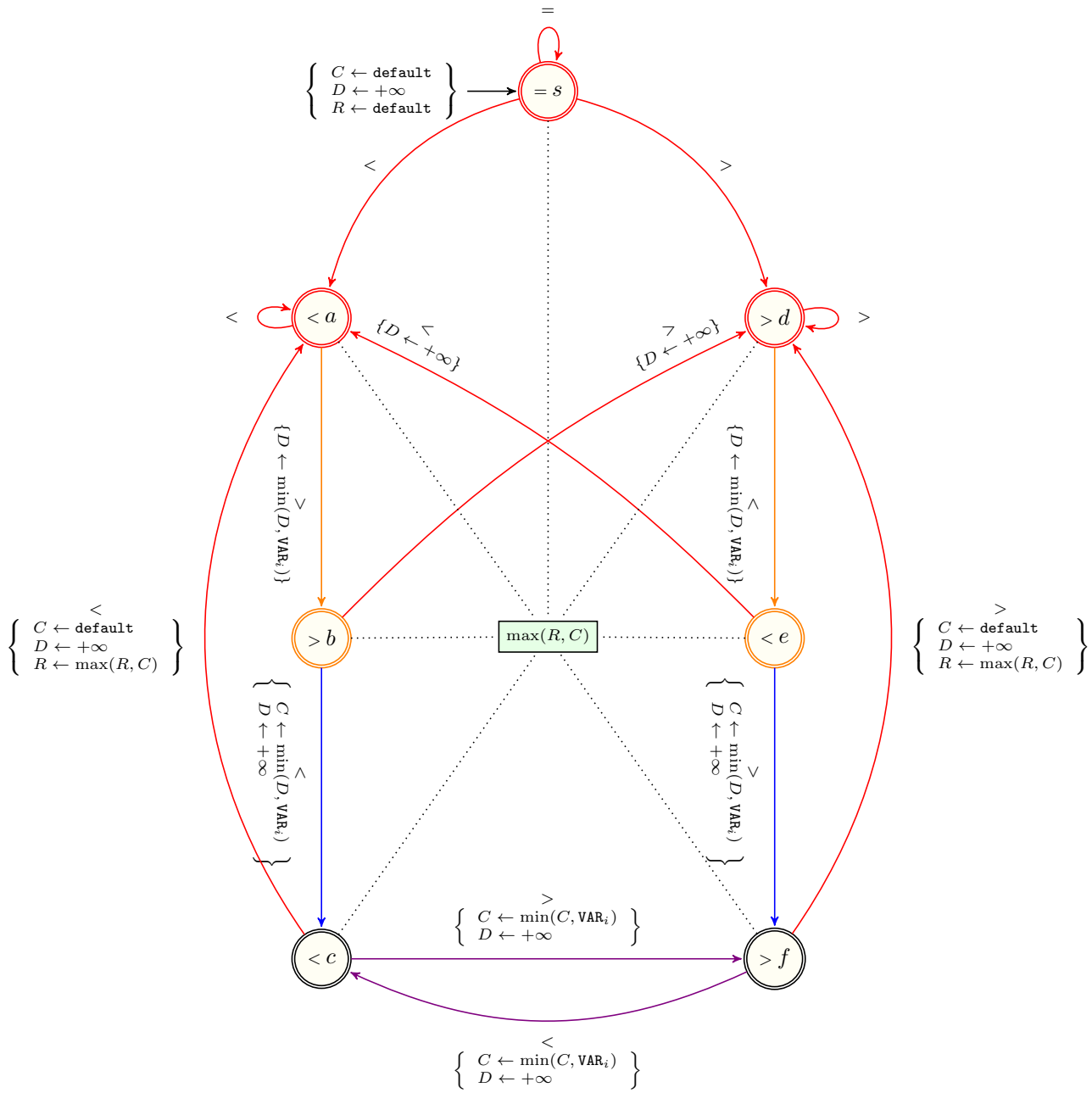
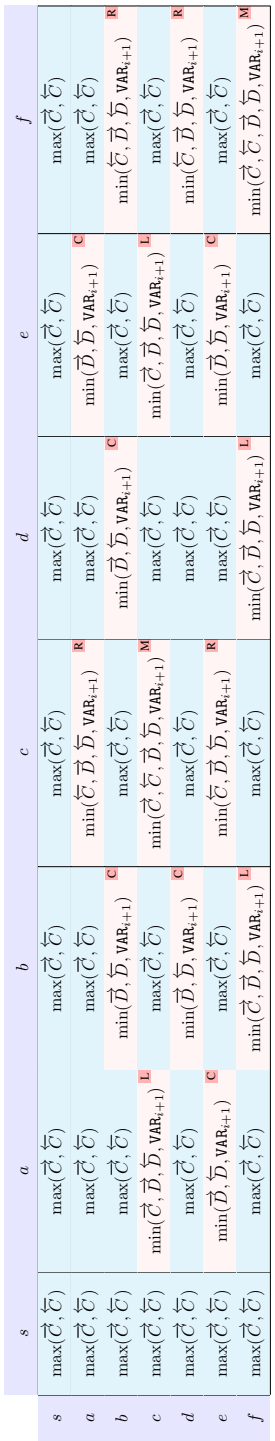


Figure 4.595: Simplified automaton for the MAX_MIN_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.; $R_i - R_{i-1} \geq 0$ is a linear invariant.



	s	a	b	c	d	e	f
s	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$
a	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\min(\overleftarrow{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ R	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$
b	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$	$\min(\overleftarrow{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ R
c	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ L	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ M	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ L	$\max(\vec{C},\overleftarrow{C})$
d	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\min(\overleftarrow{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ R	$\min(\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ R
e	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$	$\min(\overleftarrow{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ R	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ C	$\max(\vec{C},\overleftarrow{C})$
f	$\max(\vec{C},\overleftarrow{C})$	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ L	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ L	$\max(\vec{C},\overleftarrow{C})$	$\min(\vec{C},\vec{C},\vec{D},\overleftarrow{D},\texttt{VAR}_{i+1})$ M

Table 4.51: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MAX_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

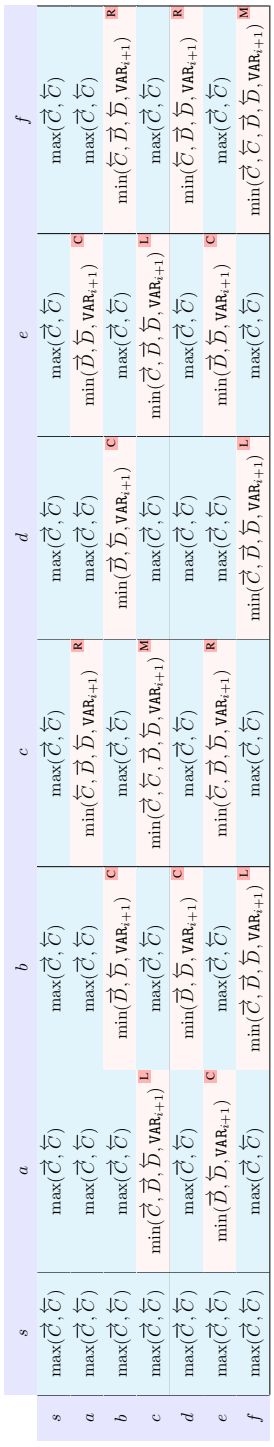
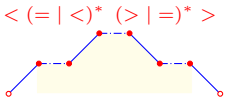


Table 4.52: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MAX_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MIN, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MAX_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

Constraint MAX_PEAK(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PEAK is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PEAK.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example Figure 4.596 provides an example where the MAX_PEAK $([7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1],[0,0,0,0,0,5,0,0,0,0,6,0,0,0,3,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

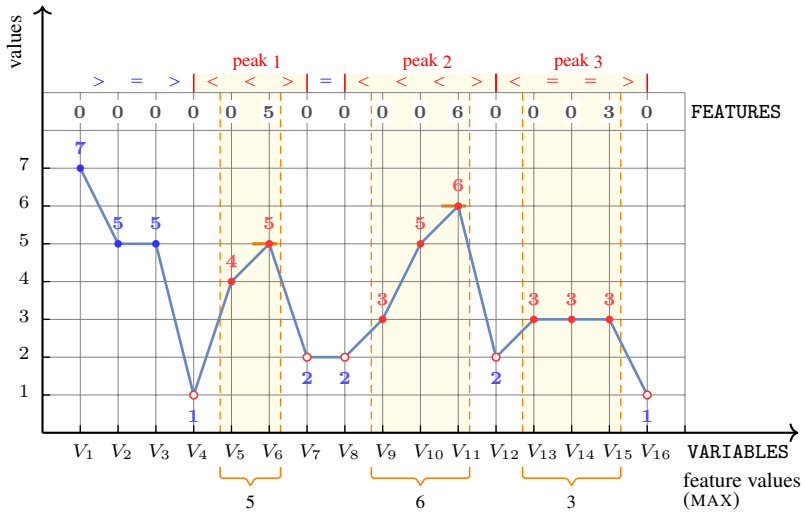


Figure 4.596: Illustrating the MAX_PEAK constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MAX_RANGE_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

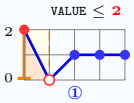
Constraint MAX_RANGE_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{rv} - 1$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(2, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.597 provides an example where the MAX_RANGE_DECREASING $(2, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

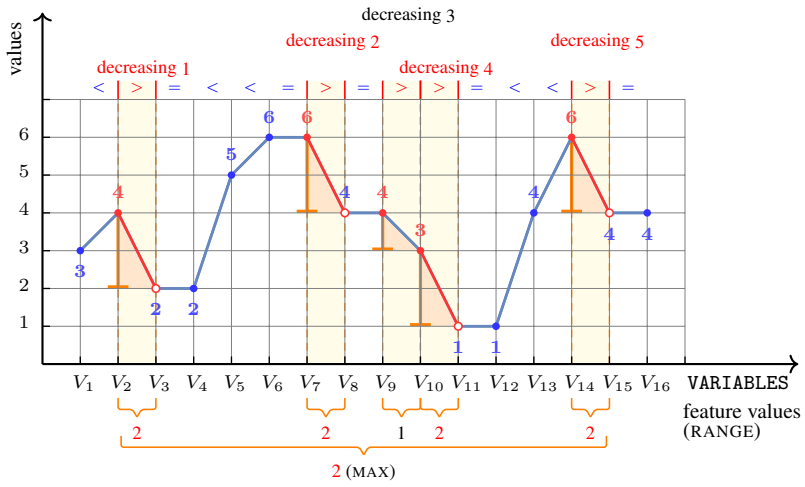


Figure 4.597: Illustrating the MAX_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figures 4.598 and 4.599 respectively depict the automaton associated with the constraint MAX_RANGE_DECREASING and its simplified form.

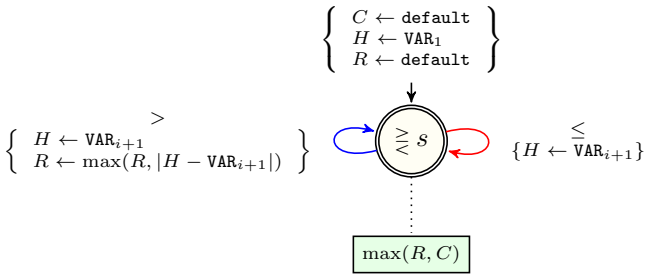


Figure 4.598: Automaton for the MAX_RANGE_DECREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING pattern where `default` is 0

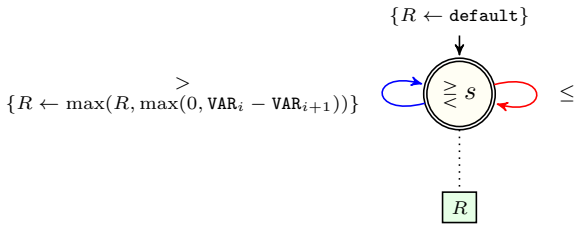
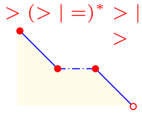


Figure 4.599: Simplified automaton for the MAX_RANGE_DECREASING constraint obtained by applying decoration Table 3.46 to the seed transducer of the DECREASING pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $R_i + \texttt{VAR}_{i-1} - \texttt{VAR}_{i-2} \geq 0$ are linear invariants.

AGGREGATOR ↑ FEATURE ↑ PATTERN ↑

MAX_RANGE_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

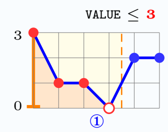
Constraint MAX_RANGE_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE : dvar
VARIABLES : collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = 0$
$\mathtt{VALUE} \geq 0$
$\mathtt{VALUE} \leq \mathtt{rv} - 1$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
sv = |VARIABLES|



Purpose VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(5, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.600 provides an example where the MAX_RANGE_DECREASING_SEQUENCE $(5, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

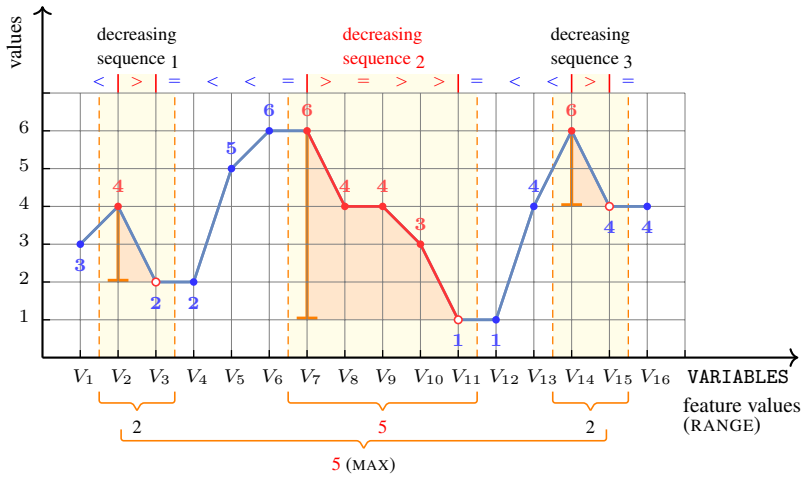


Figure 4.600: Illustrating the MAX_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.601 and 4.602 respectively depict the automaton associated with the constraint MAX_RANGE_DECREASING_SEQUENCE and its simplified form.

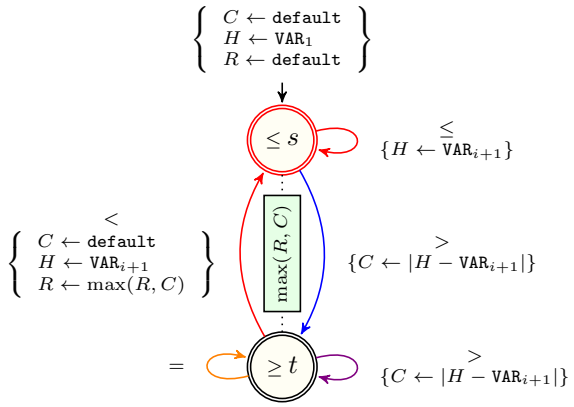


Figure 4.601: Automaton for the MAX_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

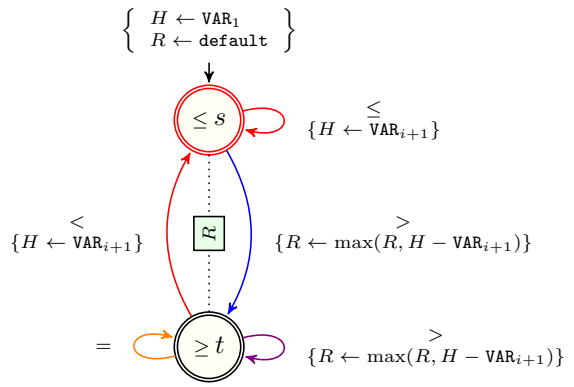


Figure 4.602: Simplified automaton for the MAX_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.44 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_RANGE_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

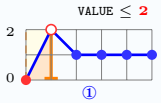
Constraint MAX_RANGE_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{rv} - 1$①
required(VARIABLES, var)
where
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(2, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.603 provides an example where the MAX_RANGE_INCREASING $(2, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

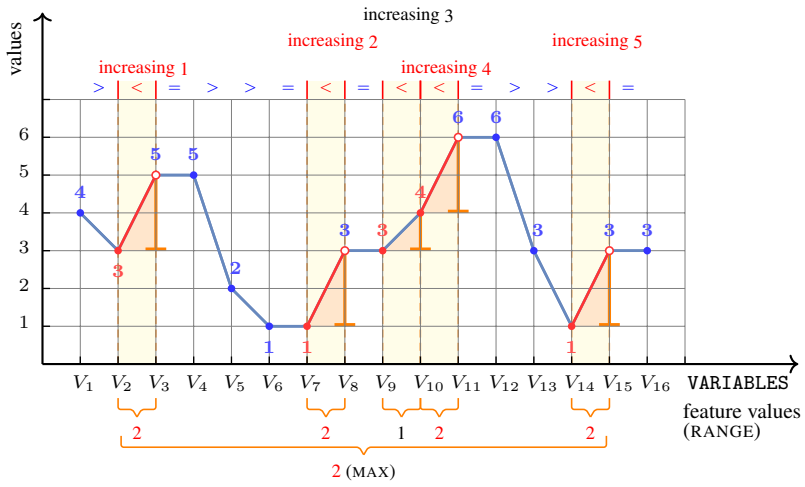


Figure 4.603: Illustrating the MAX_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figures 4.604 and 4.605 respectively depict the automaton associated with the constraint MAX_RANGE_INCREASING and its simplified form.

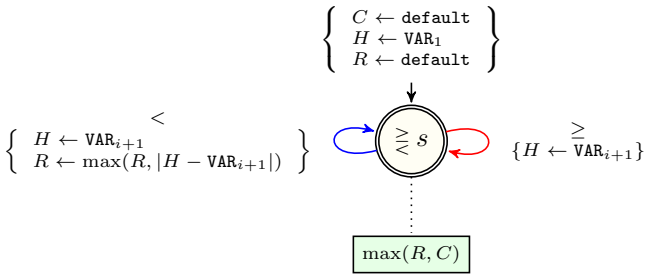


Figure 4.604: Automaton for the MAX_RANGE_INCREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING pattern where `default` is 0

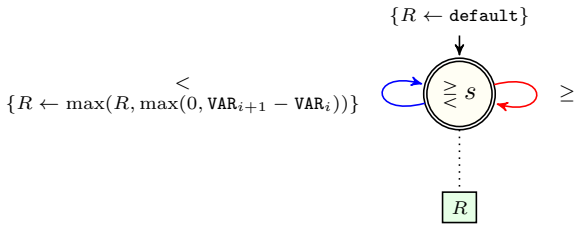
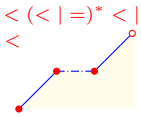


Figure 4.605: Simplified automaton for the MAX_RANGE_INCREASING constraint obtained by applying decoration Table 3.47 to the seed transducer of the INCREASING pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $R_i - \texttt{VAR}_{i-1} + \texttt{VAR}_{i-2} \geq 0$ are linear invariants.

AGGREGATOR FEATURE PATTERN

MAX_RANGE_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

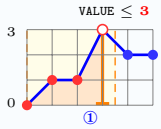
Constraint MAX_RANGE_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{rv} - 1$①
required(VARIABLES, var)
where
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.606 provides an example where the MAX_RANGE_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

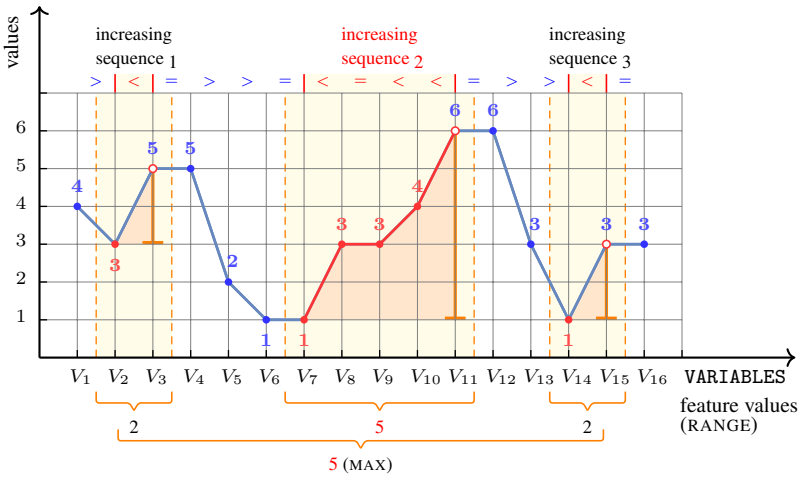


Figure 4.606: Illustrating the MAX_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.607 and 4.608 respectively depict the automaton associated with the constraint MAX_RANGE_INCREASING_SEQUENCE and its simplified form.

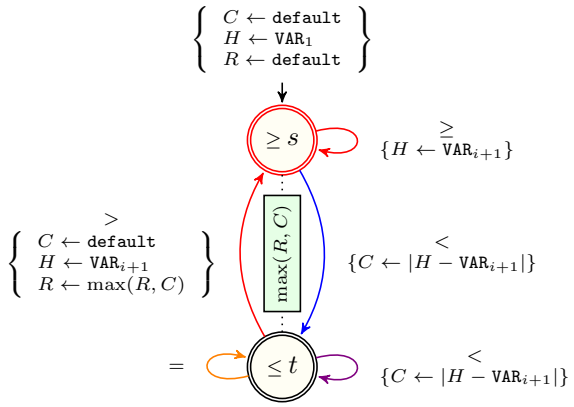


Figure 4.607: Automaton for the MAX_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

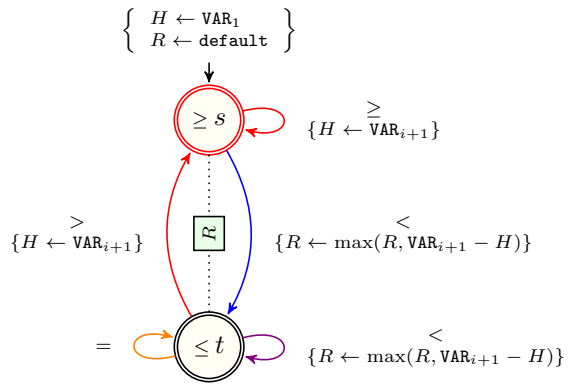


Figure 4.608: Simplified automaton for the MAX_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.45 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

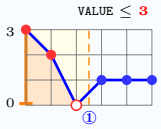
Constraint MAX_RANGE_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{rv} - 1$①
required(VARIABLES, var)
where
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(5, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.609 provides an example where the MAX_RANGE_STRICTLY_DECREASING_SEQUENCE $(5, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

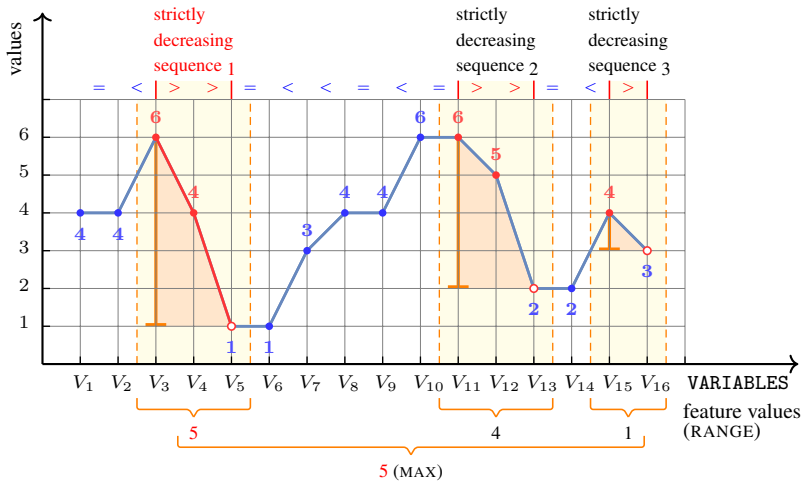


Figure 4.609: Illustrating the MAX_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.610 and 4.611 respectively depict the automaton associated with the constraint MAX_RANGE_STRICTLY_DECREASING_SEQUENCE and its simplified form.

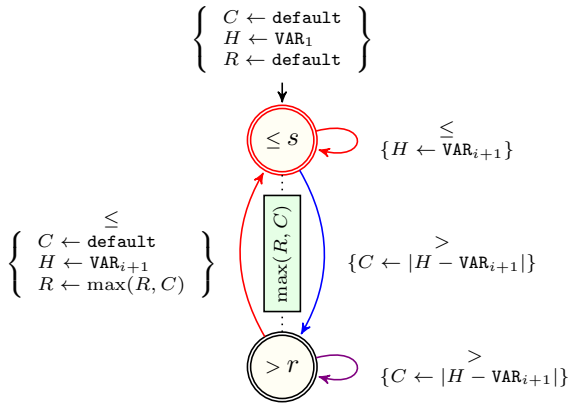


Figure 4.610: Automaton for the MAX_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

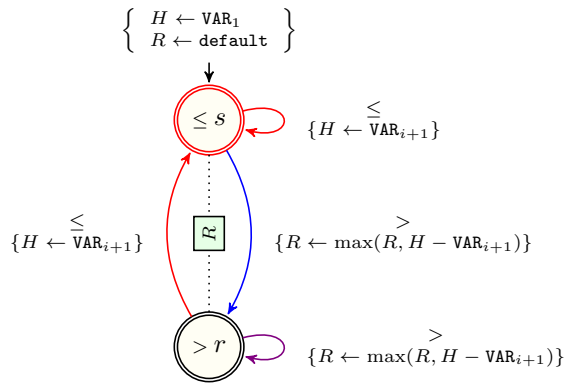


Figure 4.611: Simplified automaton for the MAX_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.44 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

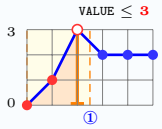
Constraint MAX_RANGE_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{rv} - 1$①
required(VARIABLES, var)
where
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

VALUE is the maximum value of the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3\rangle)$

Figure 4.612 provides an example where the MAX_RANGE_STRICTLY_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

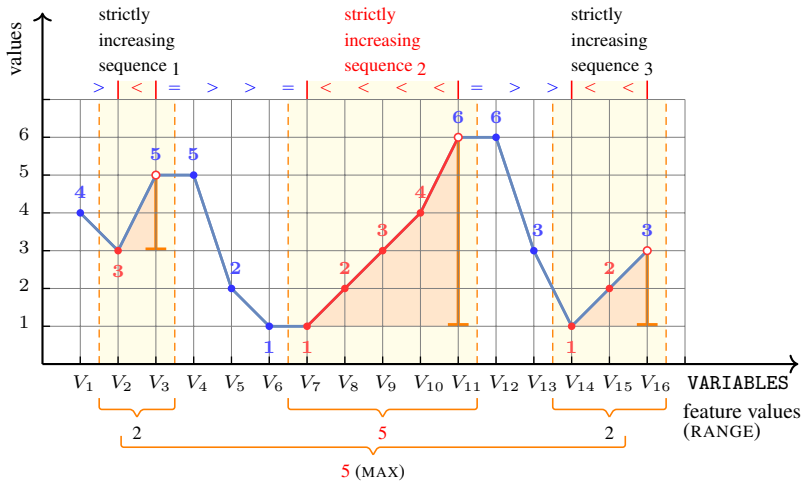


Figure 4.612: Illustrating the MAX_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.613 and 4.614 respectively depict the automaton associated with the constraint MAX_RANGE_STRICTLY_INCREASING_SEQUENCE and its simplified form.

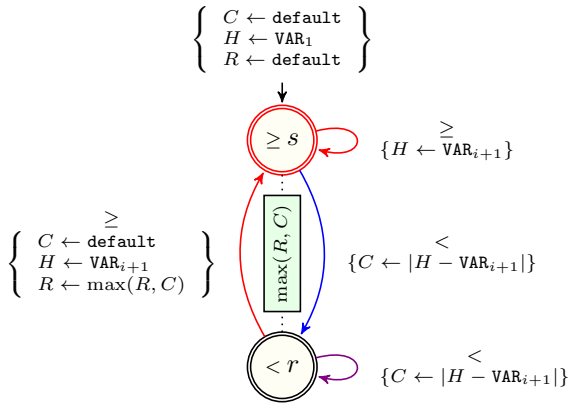


Figure 4.613: Automaton for the MAX_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

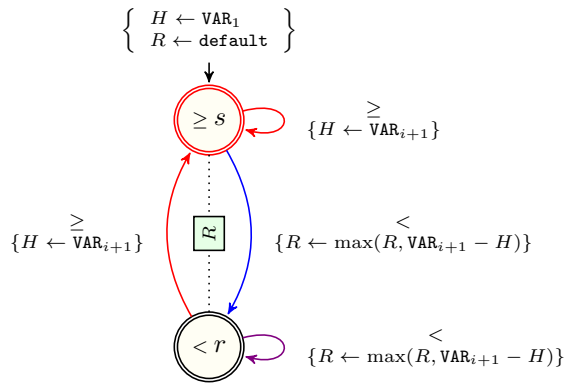


Figure 4.614: Simplified automaton for the MAX_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.45 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ is a linear invariant.

FEATURE PATTERN

MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint MAX_STRICTLY_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_DECREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.615 provides an example where the MAX_STRICTLY_DECREASING_SEQUENCE $([4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3], [0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 4, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

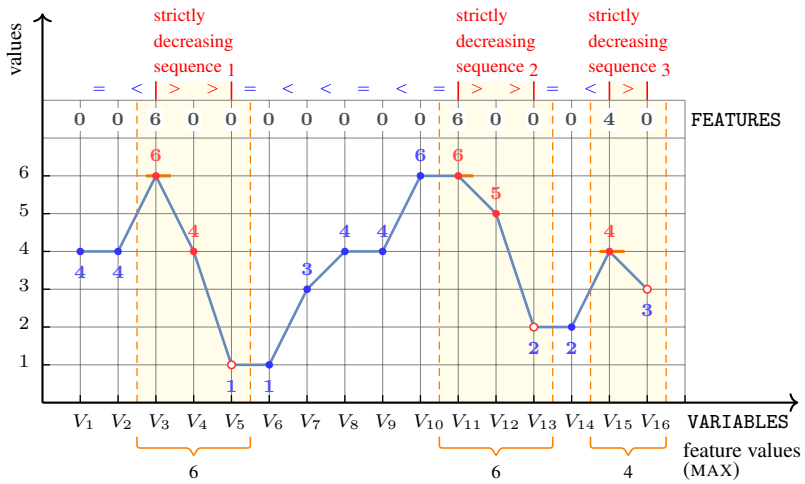


Figure 4.615: Illustrating the MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint MAX_STRICTLY_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES[i] is the default value DEFAULT; otherwise FEATURES[i] gives the feature value of the corresponding occurrence of STRICTLY_INCREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example Figure 4.616 provides an example where the MAX_STRICTLY_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3], [0, 5, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 3, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

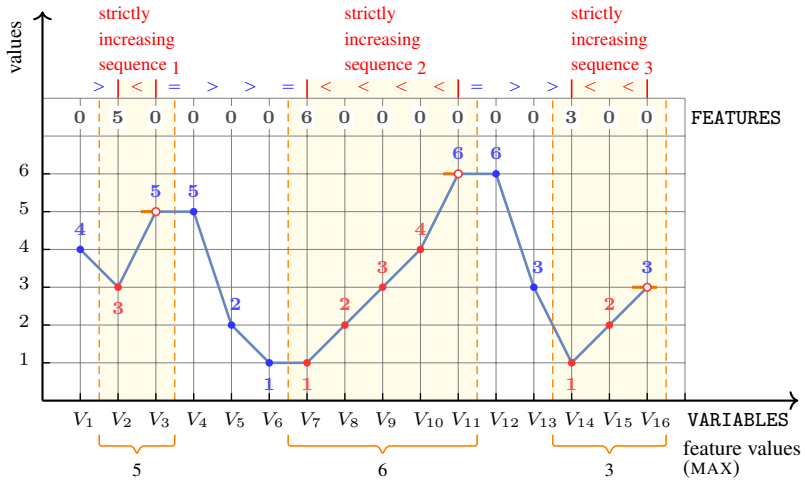


Figure 4.616: Illustrating the MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

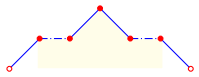
Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MAX_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the SUMMIT pattern.

Constraint MAX_SUMMIT(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of SUMMIT is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of SUMMIT.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example Figure 4.617 provides an example where the MAX_SUMMIT $([7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1],[0,0,5,0,0,0,0,0,0,0,0,0,0,4,0,3,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

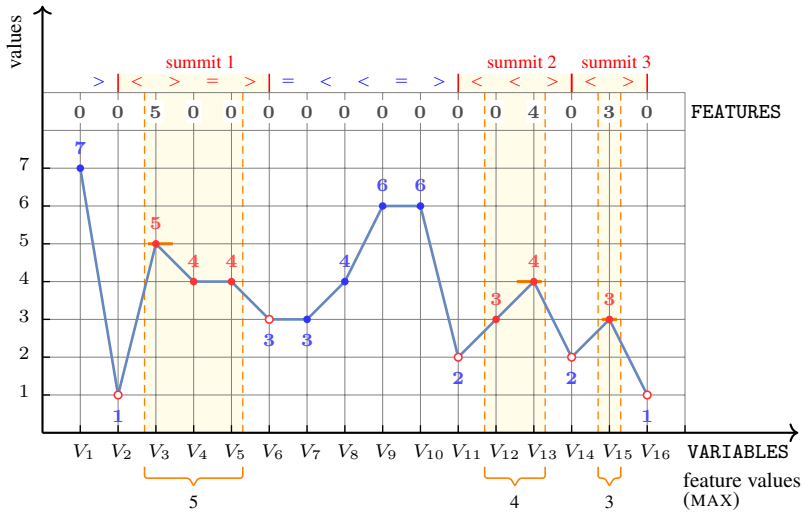
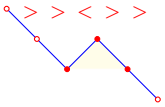


Figure 4.617: Illustrating the MAX_SUMMIT constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MAX_SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint MAX_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

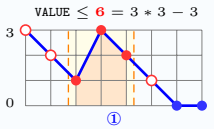
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 3 * \texttt{minv} + 3$
$\texttt{VALUE} \leq 3 * \texttt{maxv} - 3$①
among(n1, VARIABLES[3, sv − 1], ⟨maxv − 2, maxv − 1, maxv⟩)
$\texttt{n1} \geq \texttt{VALUE} - 3 - \max(\texttt{maxv} - 3, 0)$
among(n2, VARIABLES[3, sv − 1], ⟨minv, minv + 1, minv + 2⟩)
$\texttt{n2} \geq \min(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
required(VARIABLES, var)

where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximal surface of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.

An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.

Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(16, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.618 provides an example where the MAX_SURF_BUMP_ON_DECREASING_SEQUENCE $(16, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

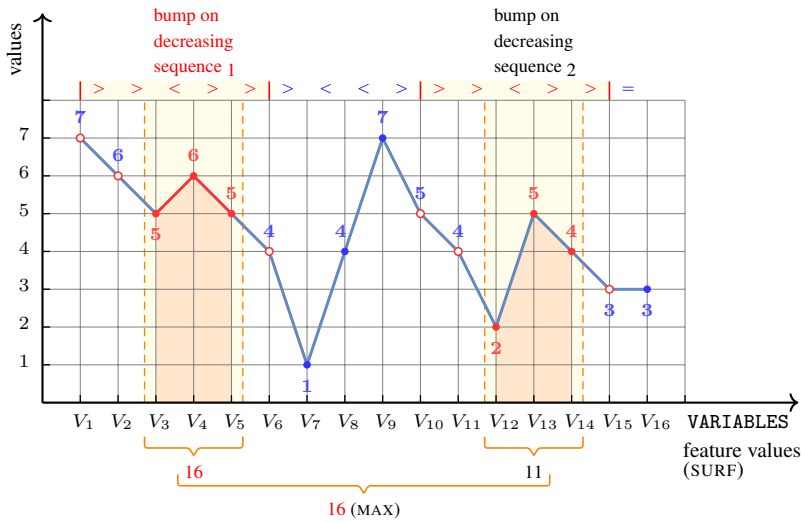


Figure 4.618: Illustrating the MAX_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.619 and 4.620 respectively depict the automaton associated with the constraint MAX_SURF_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

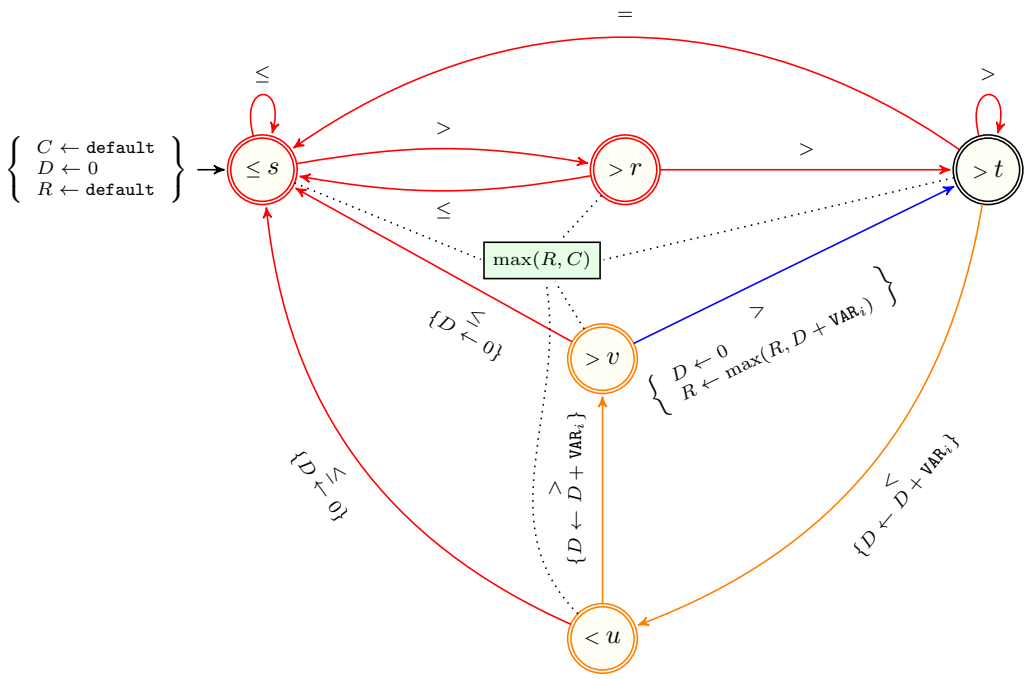


Figure 4.619: Automaton for the MAX_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$

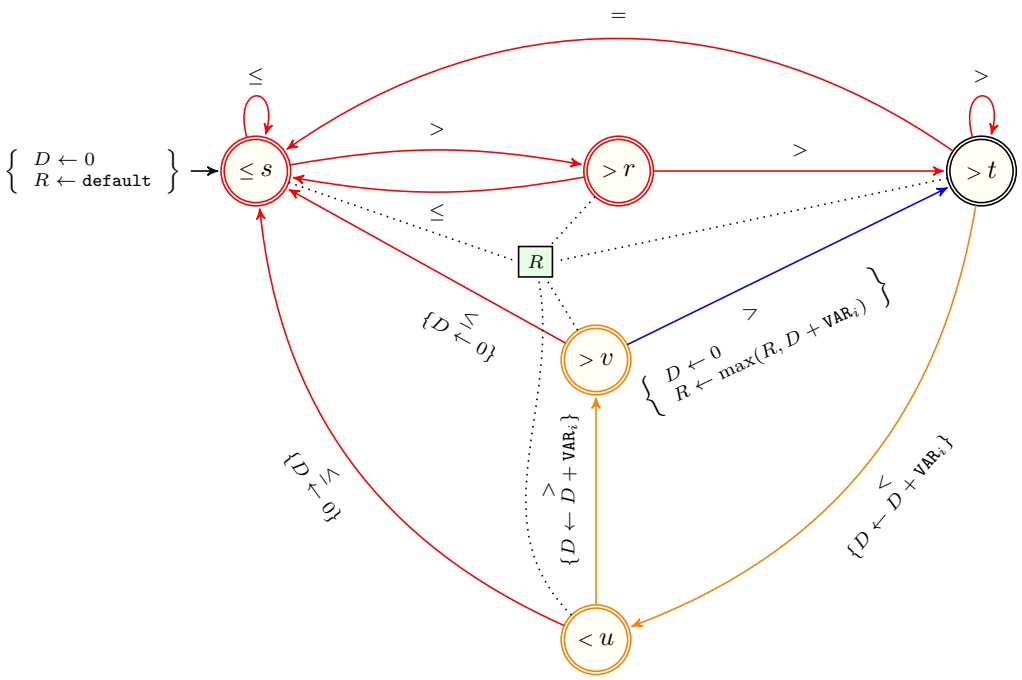


Figure 4.620: Simplified automaton for the MAX_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_SURF_DECREASING

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

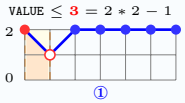
Constraint MAX_SURF_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the maximal surface of occurrences of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(10, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.621 provides an example where the MAX_SURF_DECREASING $(10, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

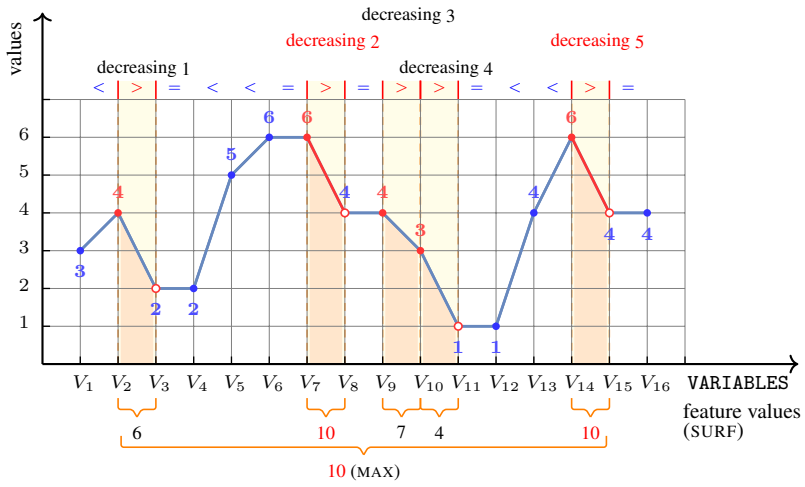


Figure 4.621: Illustrating the MAX_SURF_DECREASING constraint of the **Example** slot

Automaton

Figures 4.622 and 4.623 respectively depict the automaton associated with the constraint MAX_SURF_DECREASING and its simplified form.

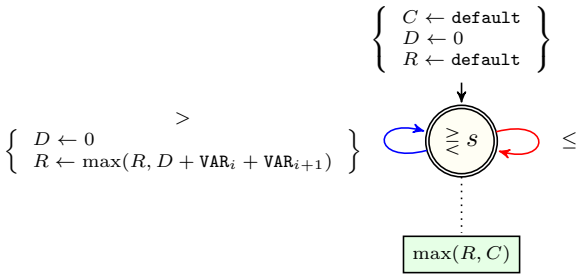


Figure 4.622: Automaton for the MAX_SURF_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $-\infty$

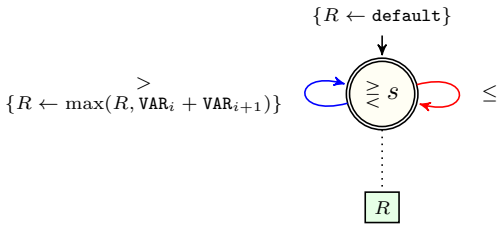


Figure 4.623: Simplified automaton for the MAX_SURF_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

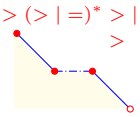
Table 4.53: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MAX_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.54: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MAX_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint MAX_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES)

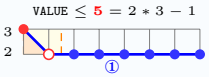
Arguments

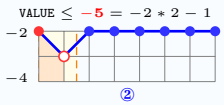
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

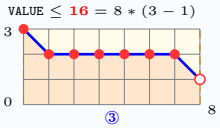
Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv} + 1, \texttt{sv} * (\texttt{minv} + 1))$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv} - 1$②$, \texttt{sv} * (\texttt{maxv} - 1)$③$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, 2 * \texttt{maxv} - 3)$
$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{sv} * (\texttt{maxv} - 2) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \texttt{min}(0, 2 * \texttt{minv} + 3) - \texttt{VALUE}$
$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$







Purpose

VALUE is the maximal surface of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(18, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.624 provides an example where the MAX_SURF_DECREASING_SEQUENCE $(18, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

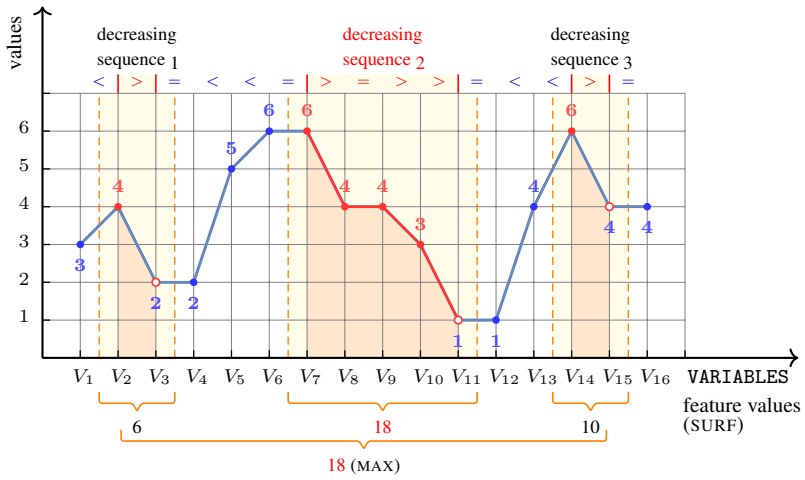


Figure 4.624: Illustrating the MAX_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton Figure 4.625 depicts the automaton associated with the constraint MAX_SURF_DECREASING_SEQUENCE.

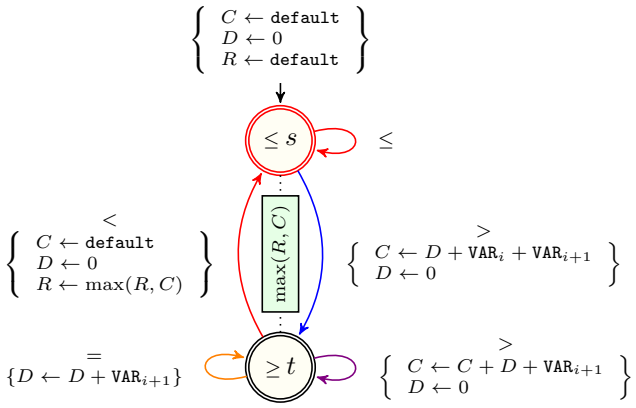


Figure 4.625: Automaton for the MAX_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

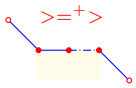
	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.55: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MAX_SURF_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

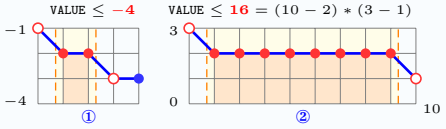
Constraint MAX_SURF_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1), (\texttt{sv} - 2) * (\texttt{minv} + 1))$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1)①, (\texttt{sv} - 2) * (\texttt{maxv} - 1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(8, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3 \rangle)$

Figure 4.626 provides an example where the MAX_SURF_DECREASING_TERRACE $(8, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

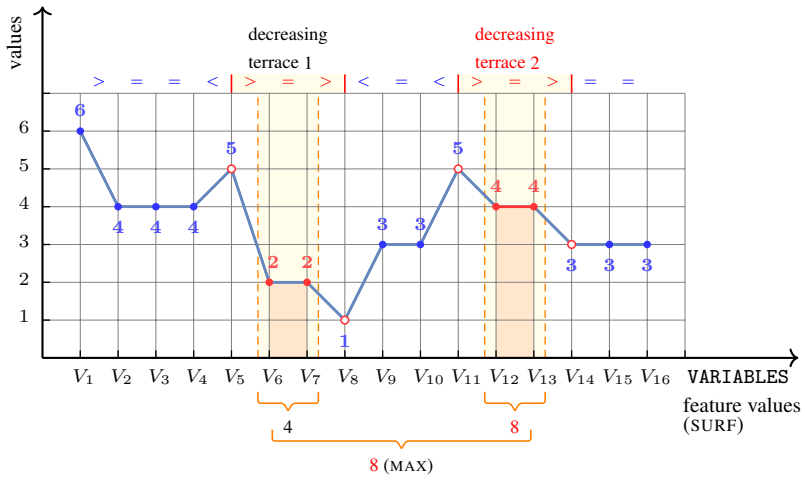


Figure 4.626: Illustrating the MAX_SURF_DECREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.627 and 4.628 respectively depict the automaton associated with the constraint MAX_SURF_DECREASING_TERRACE and its simplified form.

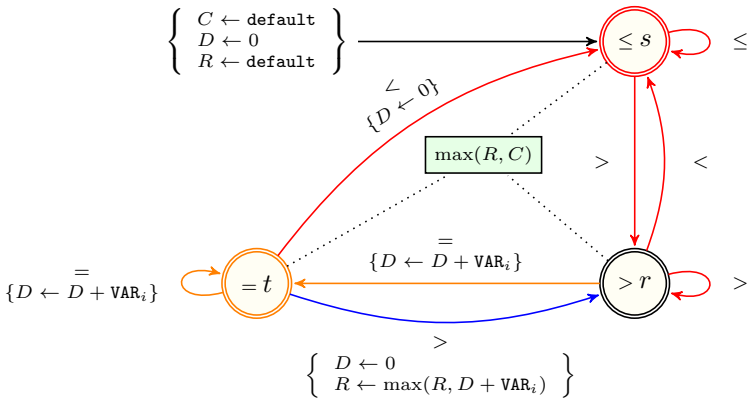


Figure 4.627: Automaton for the MAX_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $-\infty$

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.56: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MAX_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

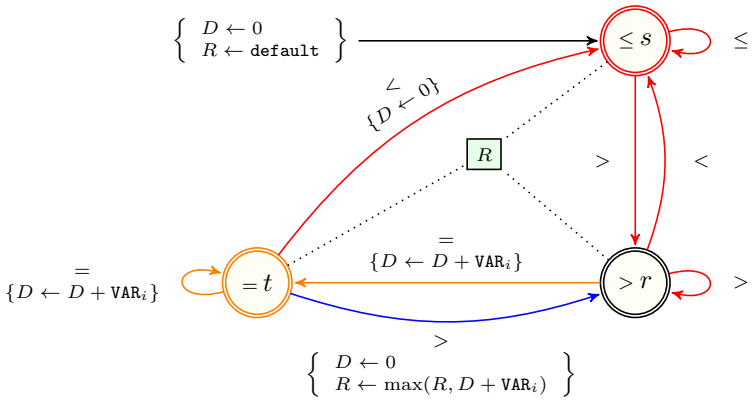


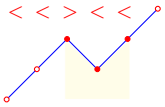
Figure 4.628: Simplified automaton for the MAX_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.57: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MAX_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

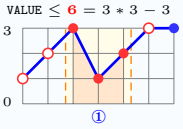
Constraint MAX_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 3 * \texttt{minv} + 3$
$\texttt{VALUE} \leq 3 * \texttt{maxv} - 3$①
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - 3 - \max(\texttt{maxv} - 3, 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{n2} \geq \min(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(10, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle)$

Figure 4.629 provides an example where the MAX_SURF_DIP_ON_INCREASING_SEQUENCE $(10, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

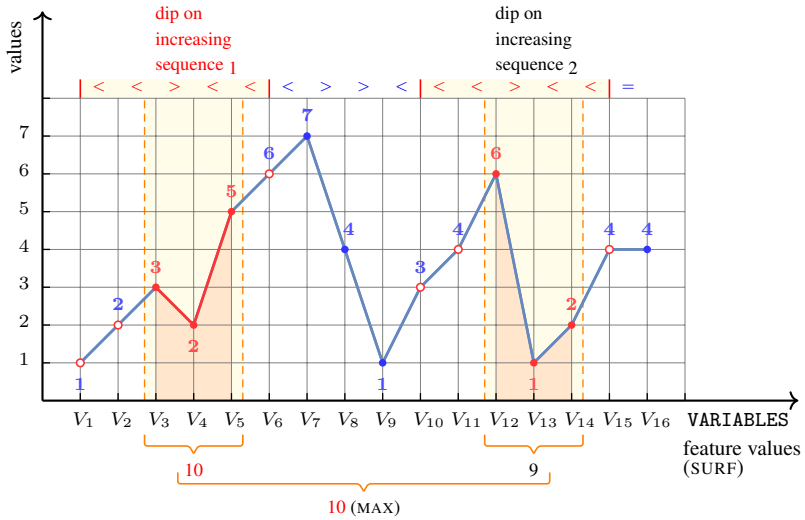


Figure 4.629: Illustrating the MAX_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.630 and 4.631 respectively depict the automaton associated with the constraint MAX_SURF_DIP_ON_INCREASING_SEQUENCE and its simplified form.

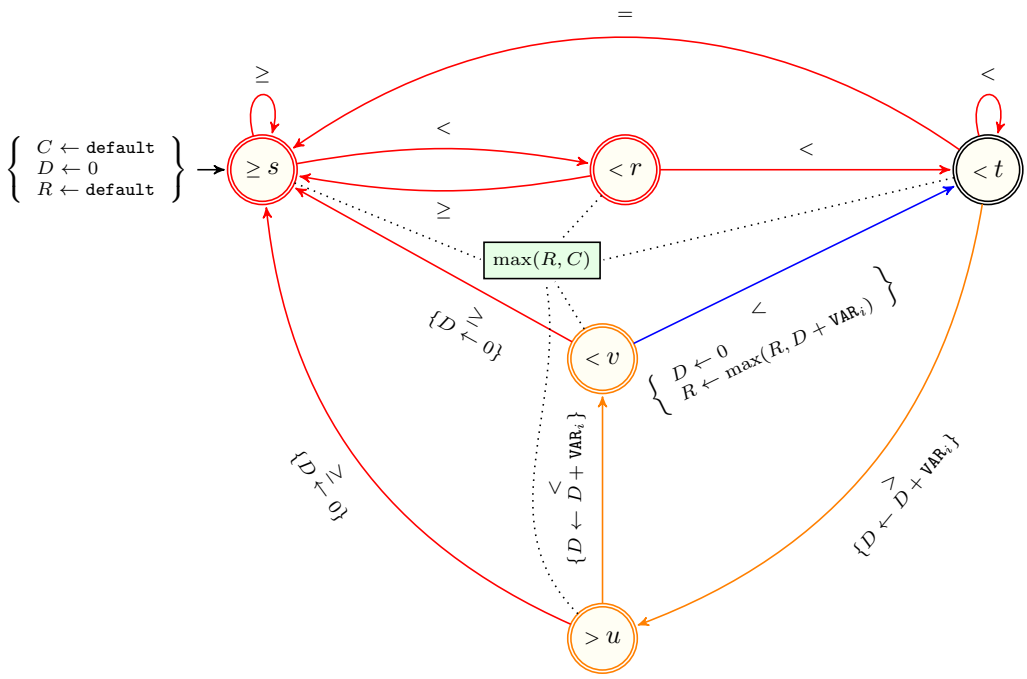


Figure 4.630: Automaton for the MAX_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$

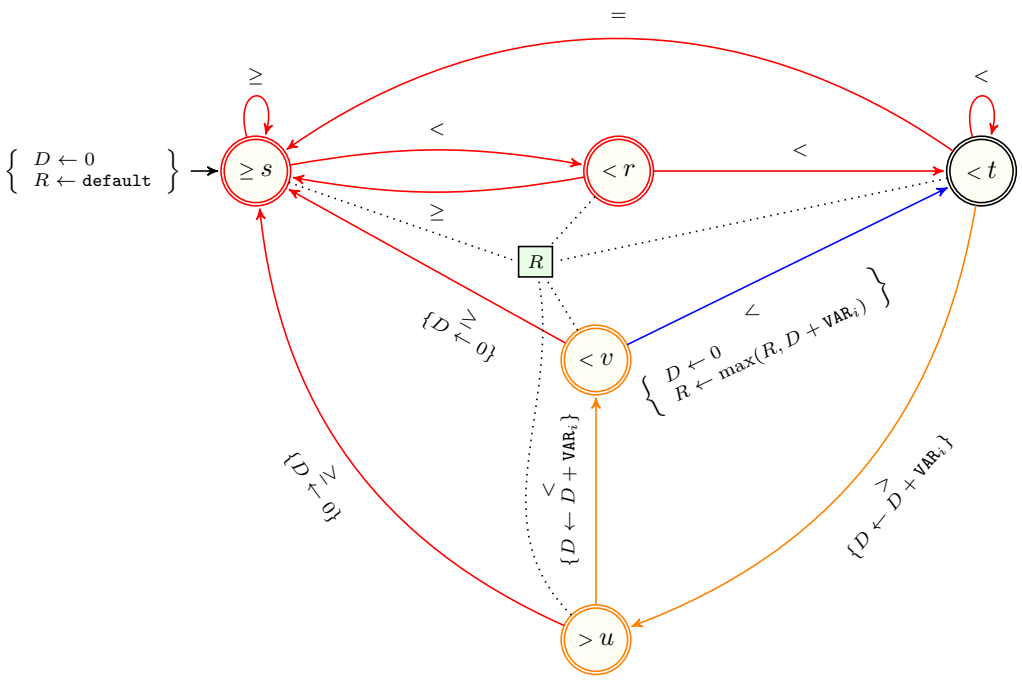
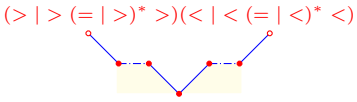


Figure 4.631: Simplified automaton for the MAX_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_SURF_GORGE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint MAX_SURF_GORGE(VALUE, VARIABLES)

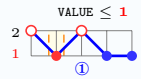
Arguments

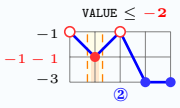
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

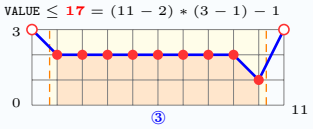
Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}, (\texttt{sv}-2) * (\texttt{minv}+1) - 1)$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{maxv} - 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1$②$, (\texttt{sv}-2) * (\texttt{maxv}-1) - 1$③$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - (\texttt{sv}-2) * (\texttt{maxv}-2)$
$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{maxv}-2)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}, \texttt{minv}+1 \rangle)$
$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq (\texttt{sv}-2) * (\texttt{minv}+2) - 1 - \texttt{VALUE}$
$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{minv}+1) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$







Purpose

VALUE is the maximal surface of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(11, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.632 provides an example where the MAX_SURF_GORGE $(11, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

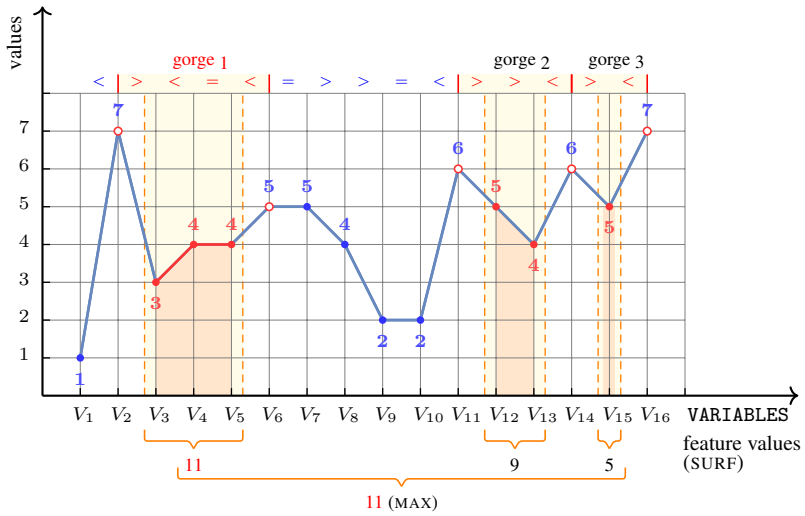


Figure 4.632: Illustrating the MAX_SURF_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.633 depicts the automaton associated with the constraint MAX_SURF_GORGE.

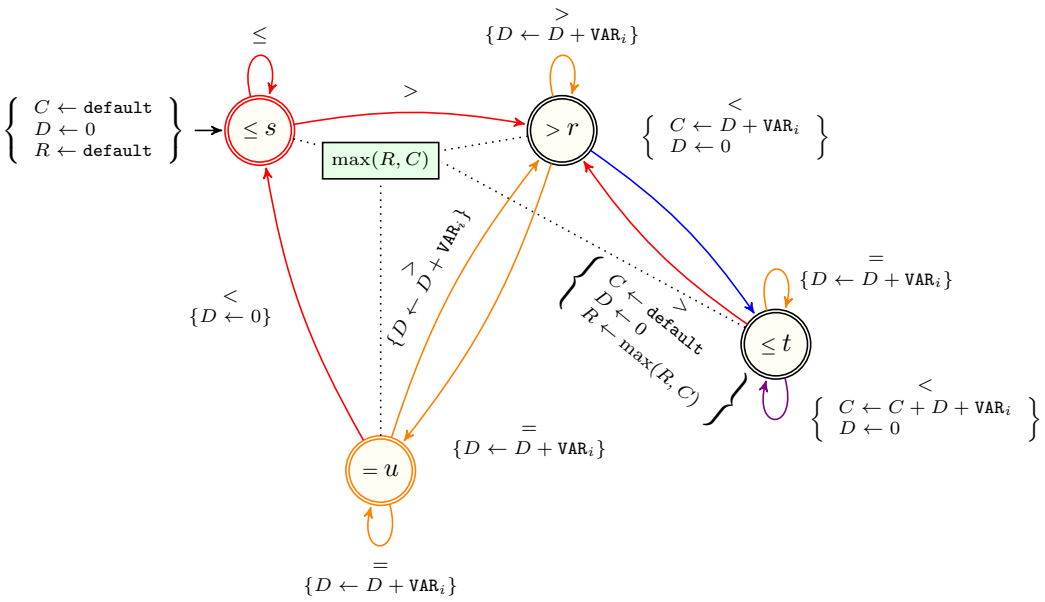


Figure 4.633: Automaton for the MAX_SURF_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is $-\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.58: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MAX_SURF_GORGE constraint defined as the composition of the GORGE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

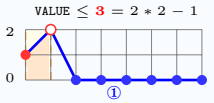
Constraint MAX_SURF_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the maximal surface of occurrences of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(10, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.634 provides an example where the MAX_SURF_INCREASING $(10, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

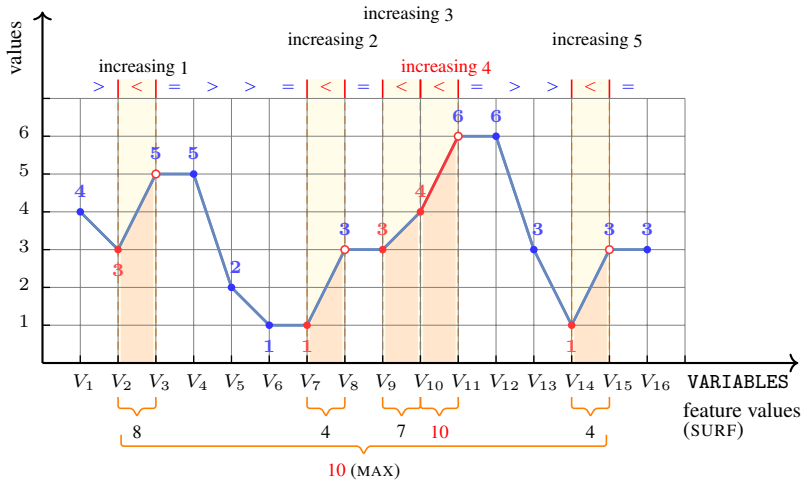


Figure 4.634: Illustrating the MAX_SURF_INCREASING constraint of the **Example** slot

Automaton

Figures 4.635 and 4.636 respectively depict the automaton associated with the constraint MAX_SURF_INCREASING and its simplified form.

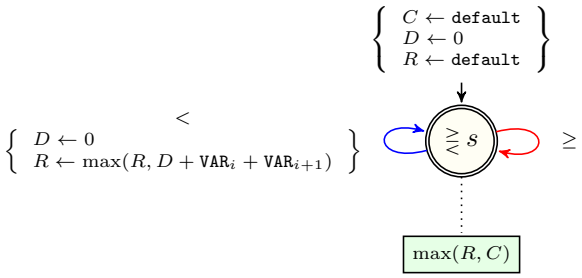


Figure 4.635: Automaton for the MAX_SURF_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $-\infty$

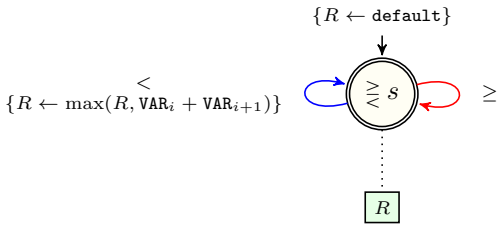


Figure 4.636: Simplified automaton for the MAX_SURF_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

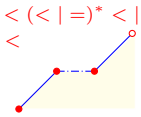
Table 4.59: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MAX_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$-\infty$

Table 4.60: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MAX_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint MAX_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES)

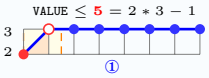
Arguments

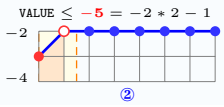
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

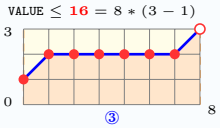
Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv} + 1, \texttt{sv} * (\texttt{minv} + 1))$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv} - 1$②$, \texttt{sv} * (\texttt{maxv} - 1)$③$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, 2 * \texttt{maxv} - 3)$
$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{sv} * (\texttt{maxv} - 2) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \texttt{min}(0, 2 * \texttt{minv} + 3) - \texttt{VALUE}$
$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$







Purpose

VALUE is the maximal surface of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(17, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.637 provides an example where the MAX_SURF_INCREASING_SEQUENCE $(17, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

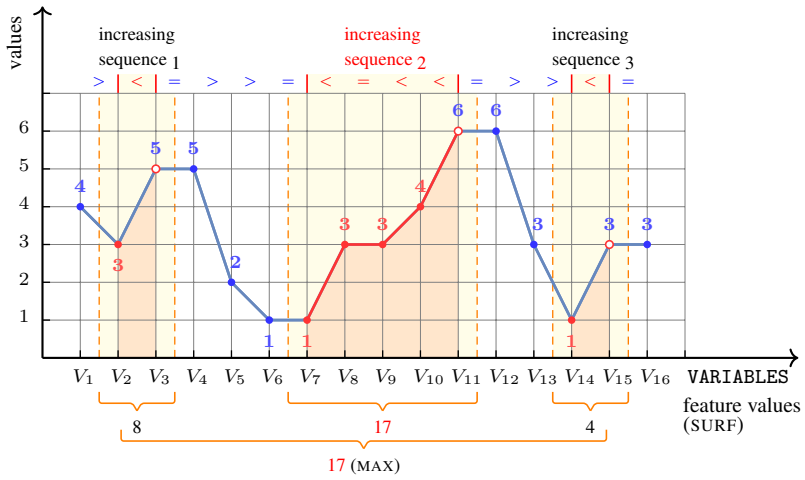


Figure 4.637: Illustrating the MAX_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton Figure 4.638 depicts the automaton associated with the constraint MAX_SURF_INCREASING_SEQUENCE.

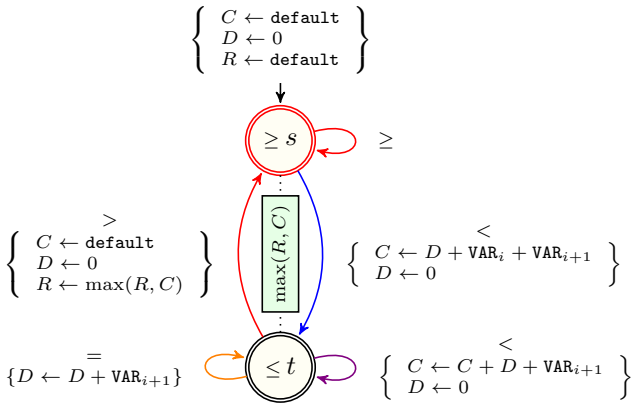


Figure 4.638: Automaton for the MAX_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

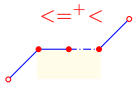
	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.61: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MAX_SURF_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

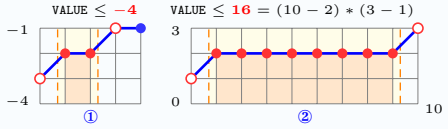
Constraint MAX_SURF_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv}+1), (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv}-1)①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the maximal surface of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(10, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4 \rangle)$

Figure 4.639 provides an example where the MAX_SURF_INCREASING_TERRACE $(10, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

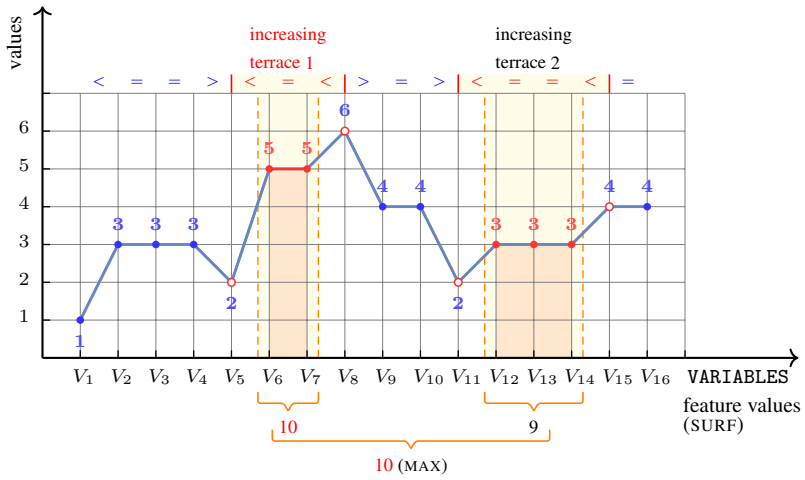


Figure 4.639: Illustrating the MAX_SURF_INCREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.640 and 4.641 respectively depict the automaton associated with the constraint MAX_SURF_INCREASING_TERRACE and its simplified form.

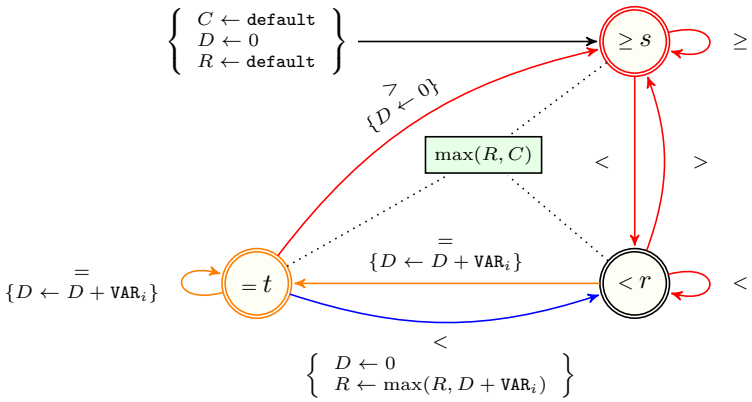


Figure 4.640: Automaton for the MAX_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $-\infty$

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.62: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MAX_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

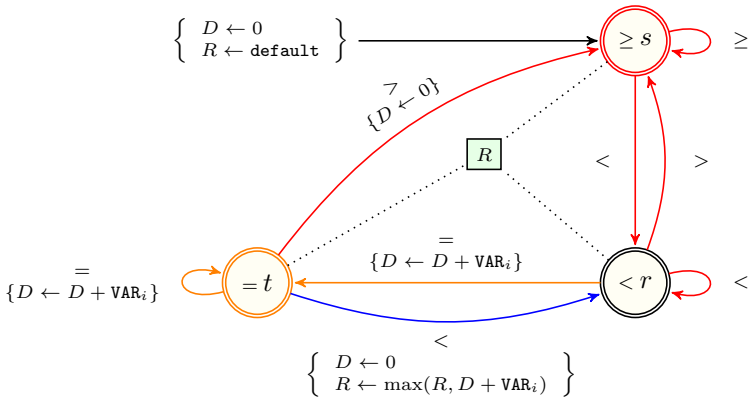


Figure 4.641: Simplified automaton for the MAX_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

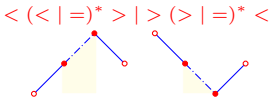
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.63: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MAX_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

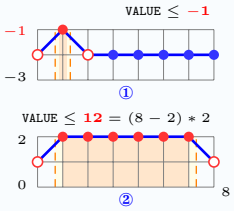
Constraint MAX_SURF_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \min(\mathtt{minv}, (\mathtt{sv}-2) * \mathtt{minv})$
$\mathtt{VALUE} \leq \max(\mathtt{maxv}①, (\mathtt{sv}-2) * \mathtt{maxv}②)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv} \rangle)$
$\mathtt{n1} \geq \mathtt{VALUE} - \max(0, (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv} \rangle)$
$\mathtt{n2} \geq \min(0, (\mathtt{sv}-2) * (\mathtt{minv}+1)) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(14, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.642 provides an example where the MAX_SURF_INFLEXION $(14, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\mathtt{VARIABLES}| > 2$
$\mathtt{range}(\mathtt{VARIABLES.var}) > 1$

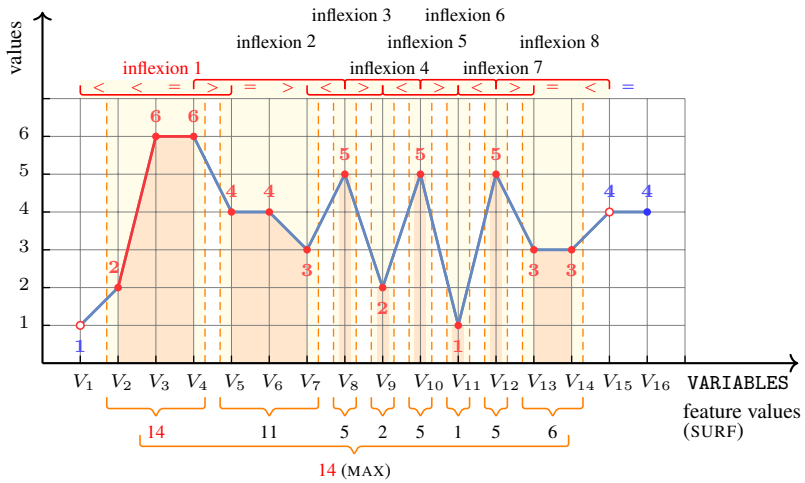


Figure 4.642: Illustrating the MAX_SURF_INFLEXION constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.643 and 4.644 respectively depict the automaton associated with the constraint MAX_SURF_INFLEXION and its simplified form.

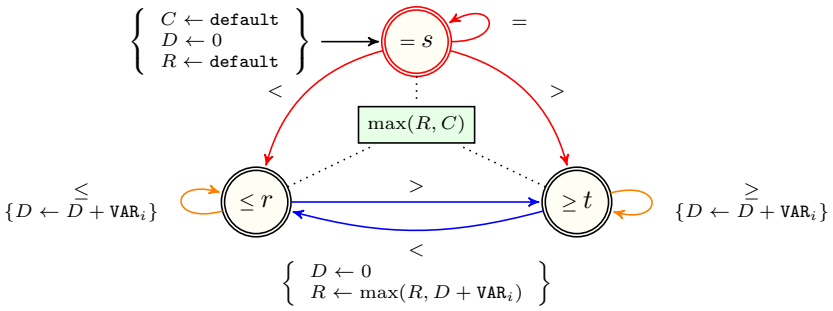


Figure 4.643: Automaton for the MAX_SURF_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

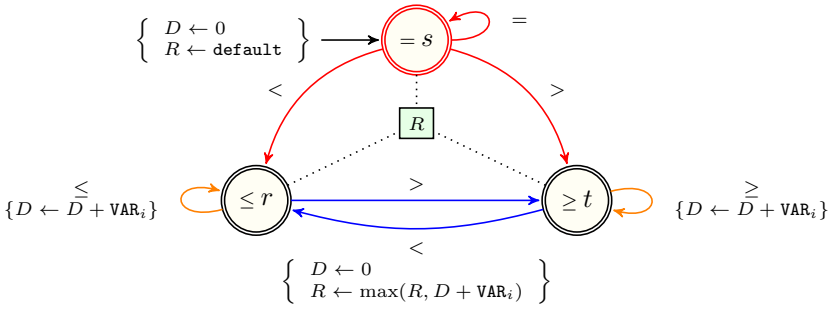
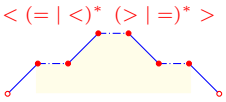


Figure 4.644: Simplified automaton for the MAX_SURF_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is $-\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MAX_SURF_PEAK



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the PEAK pattern.

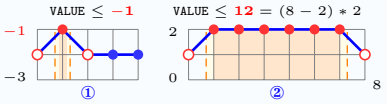
Constraint MAX_SURF_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}+1, (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \texttt{max}(\texttt{maxv①}, (\texttt{sv}-2) * \texttt{maxv②})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(14, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.645 provides an example where the MAX_SURF_PEAK $(14, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

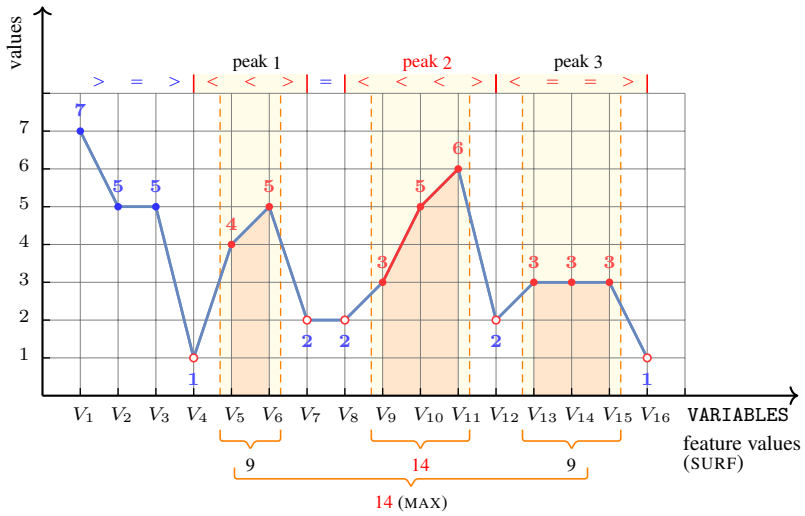


Figure 4.645: Illustrating the MAX_SURF_PEAK constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.646 depicts the automaton associated with the constraint MAX_SURF_PEAK.

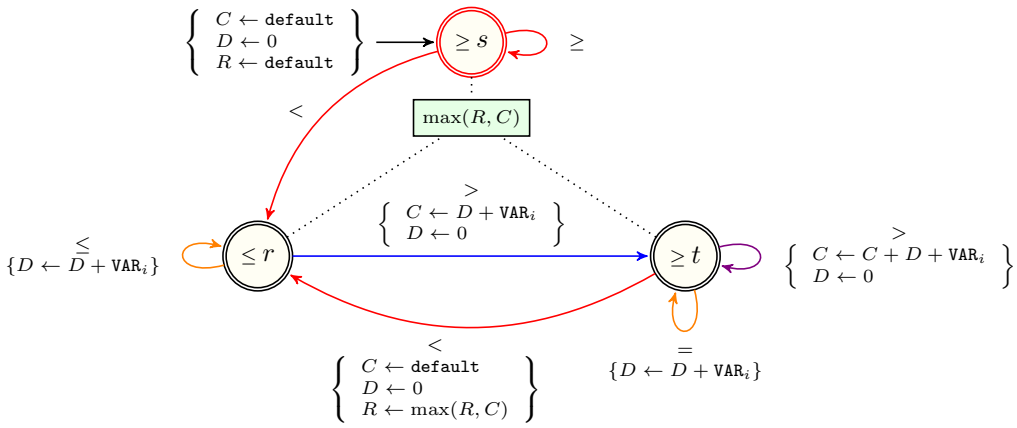


Figure 4.646: Automaton for the MAX_SURF_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where default is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.64: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MAX_SURF_PEAK constraint defined as the composition of the PEAK pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_PLAIN

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint MAX_SURF_PLAIN(VALUE, VARIABLES)

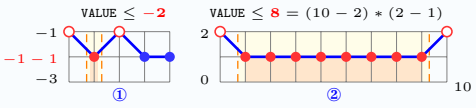
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}, (\texttt{sv}-2) * \texttt{minv})$
$\texttt{VALUE} \leq \texttt{max}(\texttt{maxv}-1①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(6, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3 \rangle)$

Figure 4.647 provides an example where the MAX_SURF_PLAIN $(6, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

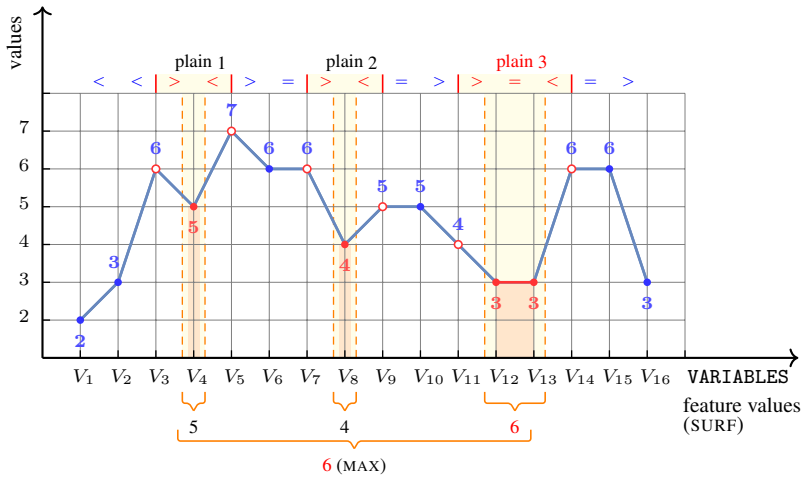


Figure 4.647: Illustrating the MAX_SURF_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.648 and 4.649 respectively depict the automaton associated with the constraint MAX_SURF_PLAIN and its simplified form.

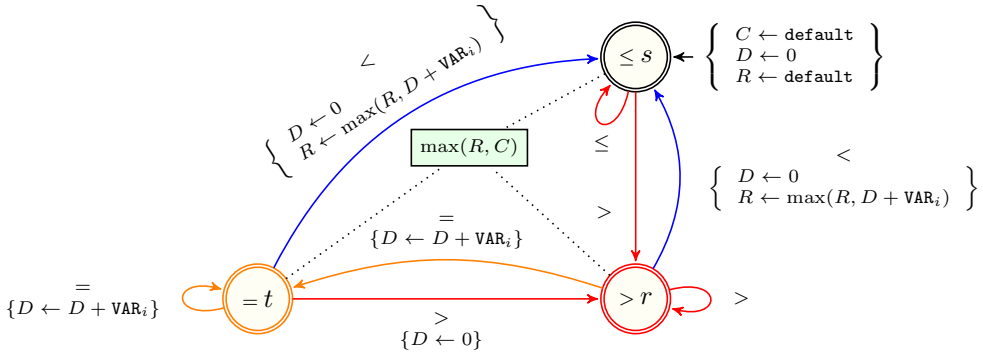


Figure 4.648: Automaton for the MAX_SURF_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is $-\infty$

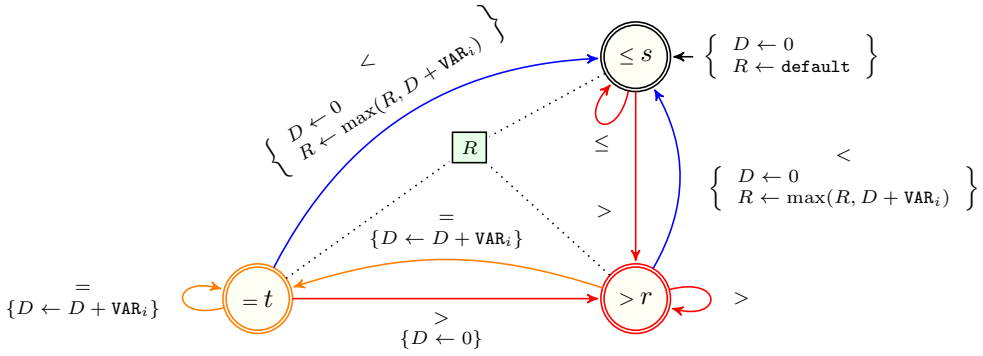


Figure 4.649: Simplified automaton for the MAX_SURF_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLAIN pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.65: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MAX_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.66: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MAX_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

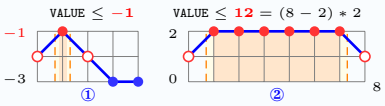
Constraint MAX_SURF_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(\texttt{minv}+1, (\texttt{sv}-2)*(\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(\texttt{maxv}①, (\texttt{sv}-2)*\texttt{maxv}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2)*(\texttt{maxv}-1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2)*(\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the maximal surface of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '<=* >'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(10, \langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5 \rangle)$

Figure 4.650 provides an example where the MAX_SURF_PLATEAU $(10, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

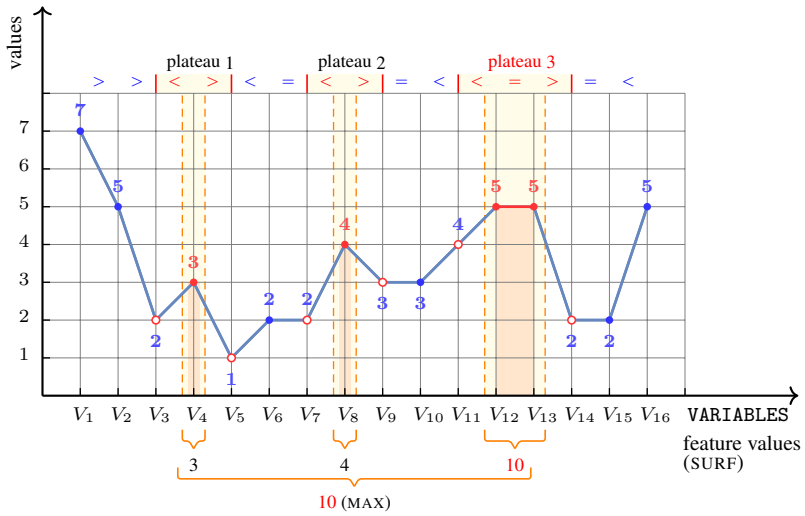


Figure 4.650: Illustrating the MAX_SURF_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.651 and 4.652 respectively depict the automaton associated with the constraint MAX_SURF_PLATEAU and its simplified form.

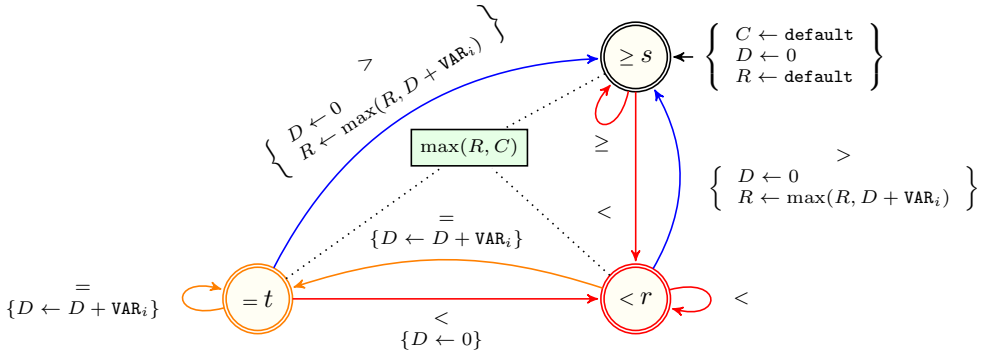


Figure 4.651: Automaton for the MAX_SURF_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is $-\infty$

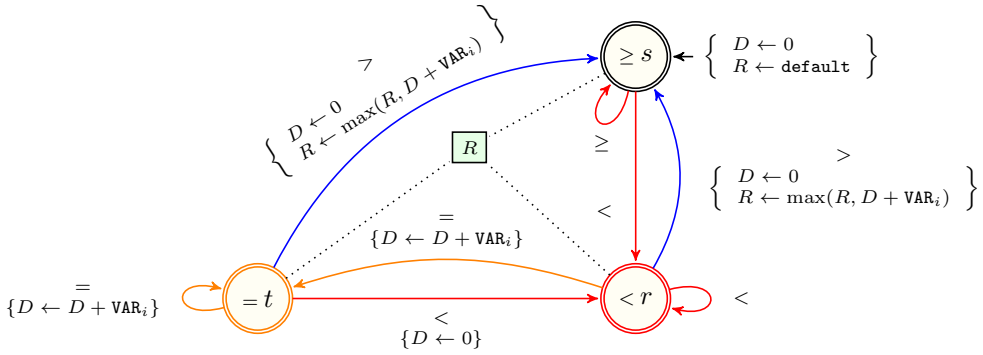


Figure 4.652: Simplified automaton for the MAX_SURF_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLATEAU pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.67: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MAX_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

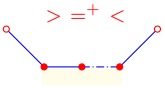
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.68: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MAX_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

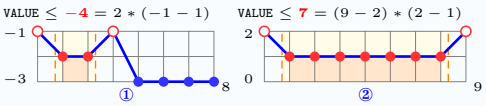
Constraint MAX_SURF_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * \texttt{minv}, (\texttt{sv} - 2) * \texttt{minv})$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1)①, (\texttt{sv} - 2) * (\texttt{maxv} - 1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the maximal surface of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(10, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5 \rangle)$

Figure 4.653 provides an example where the MAX_SURF_PROPER_PLAIN $(10, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

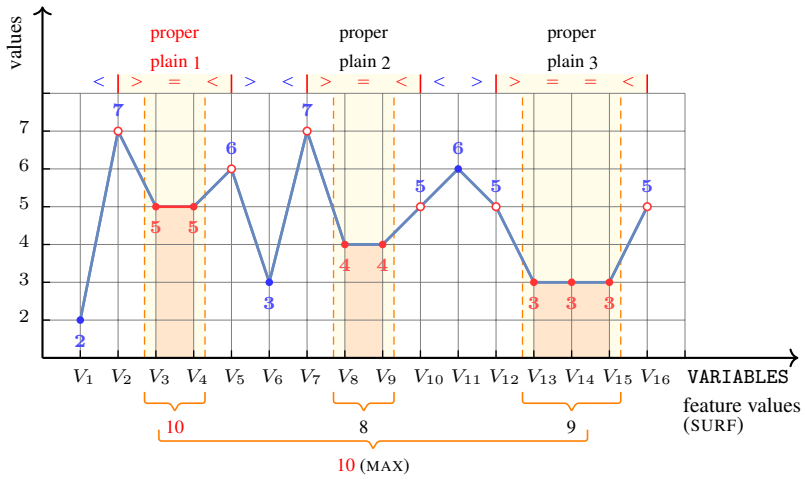


Figure 4.653: Illustrating the MAX_SURF_PROPER_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.654 and 4.655 respectively depict the automaton associated with the constraint MAX_SURF_PROPER_PLAIN and its simplified form.

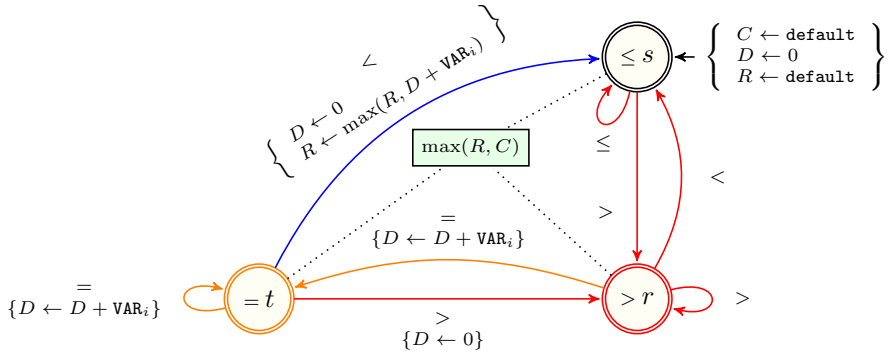


Figure 4.654: Automaton for the MAX_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is $-\infty$

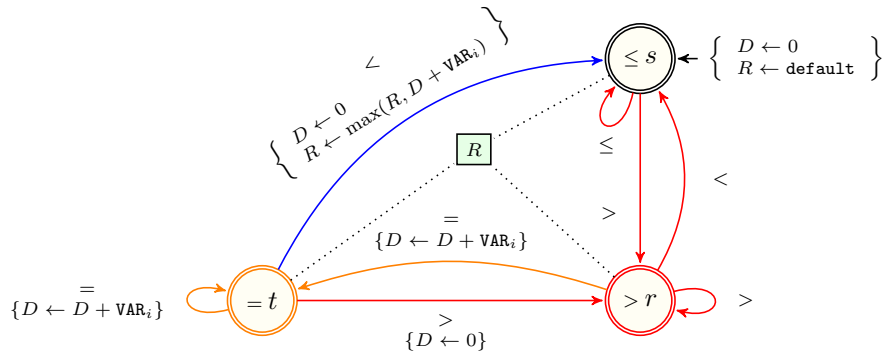


Figure 4.655: Simplified automaton for the MAX_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLAIN pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.69: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MAX_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

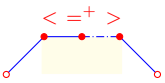
	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.70: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MAX_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

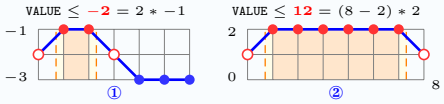
Constraint MAX_SURF_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv}+1), (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(2 * \texttt{maxv}①, (\texttt{sv}-2) * \texttt{maxv}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(15, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3 \rangle)$

Figure 4.656 provides an example where the MAX_SURF_PROPER_PLATEAU $(15, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

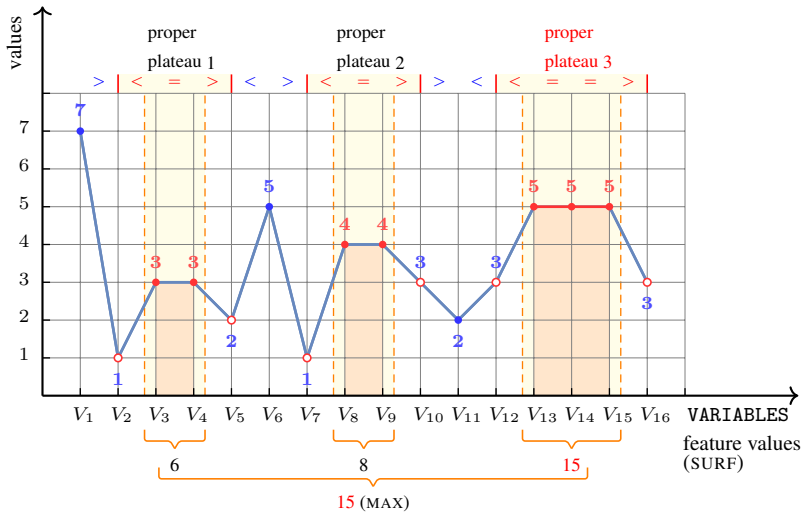


Figure 4.656: Illustrating the MAX_SURF_PROPER_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.657 and 4.658 respectively depict the automaton associated with the constraint MAX_SURF_PROPER_PLATEAU and its simplified form.

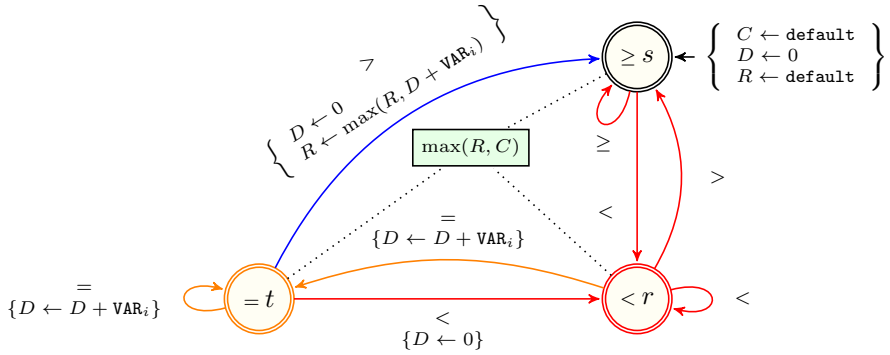


Figure 4.657: Automaton for the MAX_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $-\infty$

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.71: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MAX_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

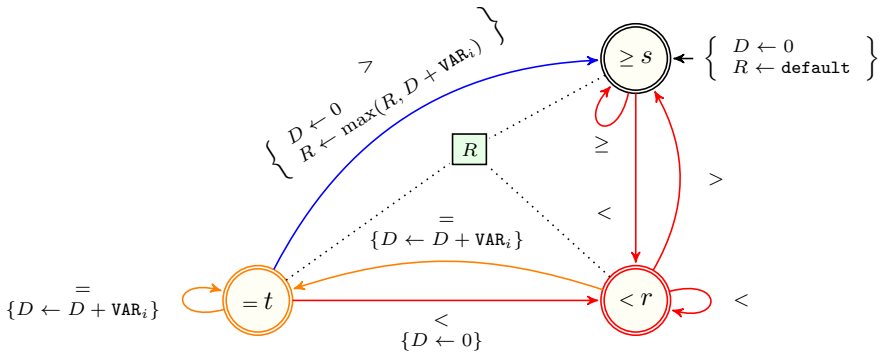


Figure 4.658: Simplified automaton for the MAX_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$-\infty$	$-\infty$	$-\infty$
r	$-\infty$	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$-\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.72: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MAX_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

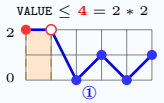
Constraint MAX_SURF_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv}$
$\texttt{VALUE} \leq 2 * \texttt{maxv}$①
among(n1, VARIABLES[1, sv], ⟨maxv⟩)
$\texttt{n1} \geq \texttt{VALUE} - 2 * (\texttt{maxv} - 1)$
among(n2, VARIABLES[1, sv], ⟨minv⟩)
$\texttt{n2} \geq 2 * (\texttt{minv} + 1) - \texttt{VALUE}$
required(VARIABLES, var)
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|



Purpose

VALUE is the maximal surface of occurrences of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(12, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6\rangle)$

Figure 4.659 provides an example where the MAX_SURF_STEADY $(12, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

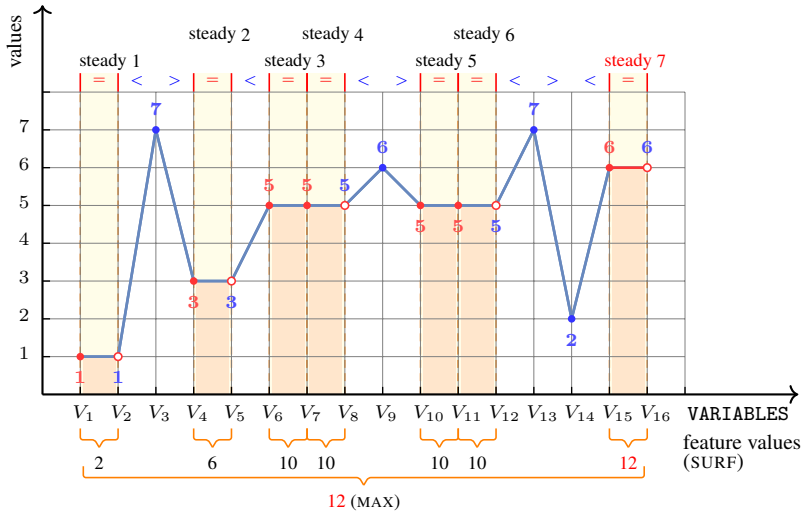


Figure 4.659: Illustrating the MAX_SURF_STEADY constraint of the **Example** slot

Automaton

Figures 4.660 and 4.661 respectively depict the automaton associated with the constraint MAX_SURF_STEADY and its simplified form.

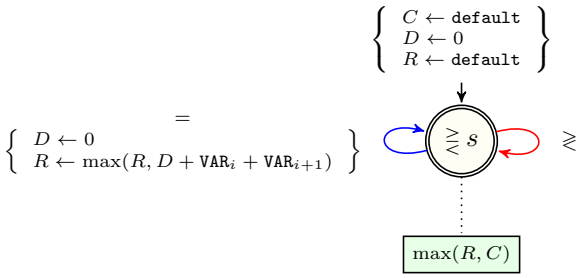


Figure 4.660: Automaton for the MAX_SURF_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is $-\infty$

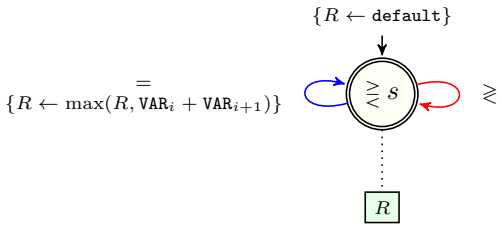


Figure 4.661: Simplified automaton for the MAX_SURF_STEADY constraint obtained by applying decoration Table 3.40 to the seed transducer of the STEADY pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.73: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the MAX_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

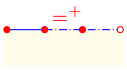
	s
s	$-\infty$

Table 4.74: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the MAX_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint MAX_SURF_STEADY_SEQUENCE(VALUE, VARIABLES)

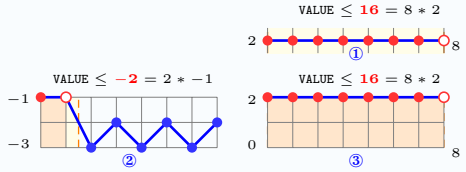
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{sv} * \texttt{minv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}, \texttt{sv} * \texttt{minv})$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \leq \texttt{sv} * \texttt{maxv}①$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}②, \texttt{sv} * \texttt{maxv}③)$
among(n1, VARIABLES[1, sv], ⟨maxv⟩)
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{sv} * (\texttt{maxv} - 1))$
among(n2, VARIABLES[2, sv], ⟨minv⟩)
$\texttt{n2} \geq \texttt{min}(0, \texttt{sv} * (\texttt{minv} + 1)) - \texttt{VALUE}$
required(VARIABLES, var)

where
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)
sv = |VARIABLES|
minv =minval(VARIABLES.var)



Purpose VALUE is the maximal surface of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(15, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.662 provides an example where the MAX_SURF_STEADY_SEQUENCE $(15, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

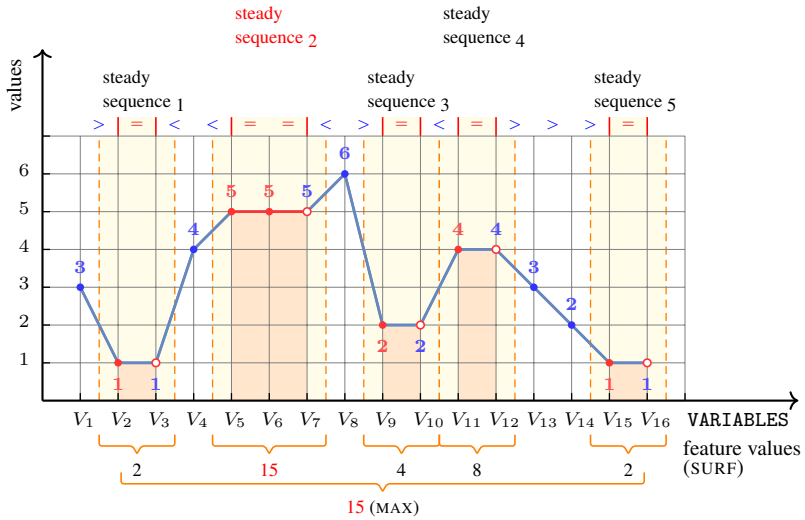


Figure 4.662: Illustrating the MAX_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.663 and 4.664 respectively depict the automaton associated with the constraint MAX_SURF_STEADY_SEQUENCE and its simplified form.

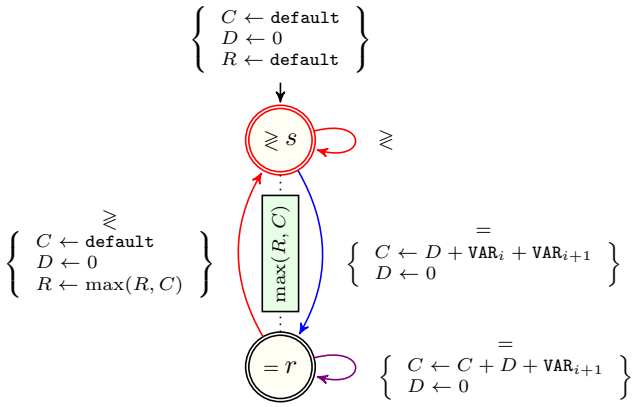


Figure 4.663: Automaton for the MAX_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $-\infty$

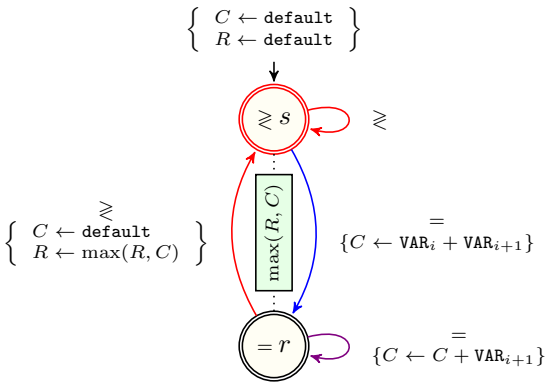


Figure 4.664: Simplified automaton for the MAX_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.75: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MAX_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \texttt{VAR}_{i+1}$ M

Table 4.76: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MAX_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint MAX_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

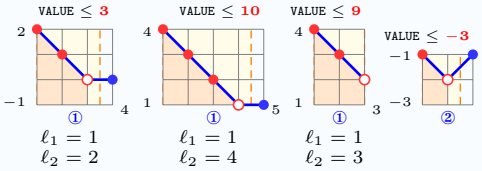
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$①
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$②
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(13, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.665 provides an example where the MAX_SURF_STRICTLY_DECREASING_SEQUENCE $(13, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

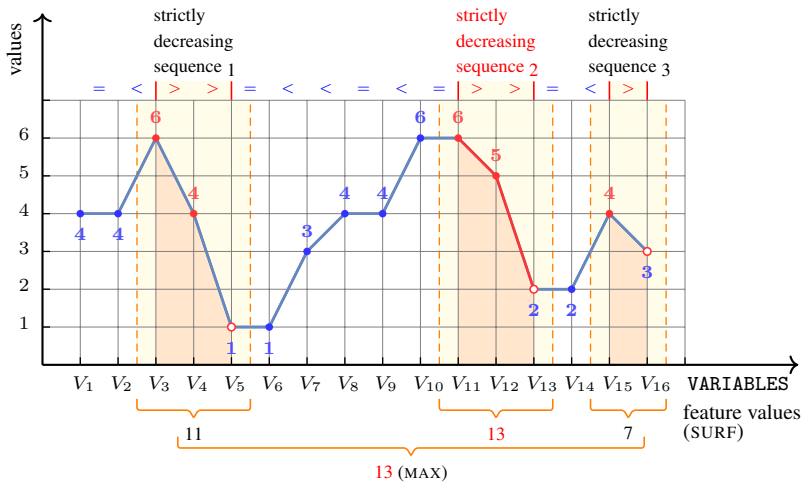


Figure 4.665: Illustrating the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.666 and 4.667 respectively depict the automaton associated with the constraint MAX_SURF_STRICTLY_DECREASING_SEQUENCE and its simplified form.

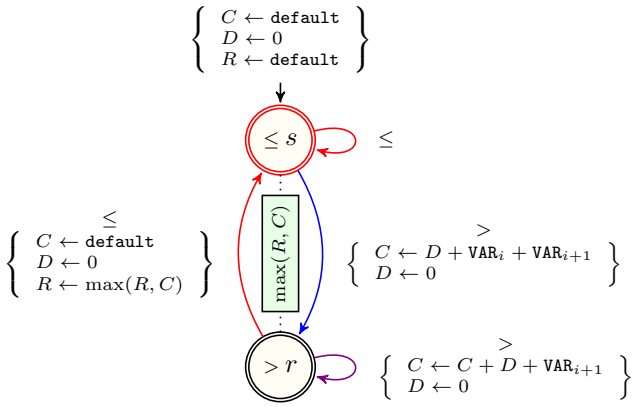


Figure 4.666: Automaton for the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$

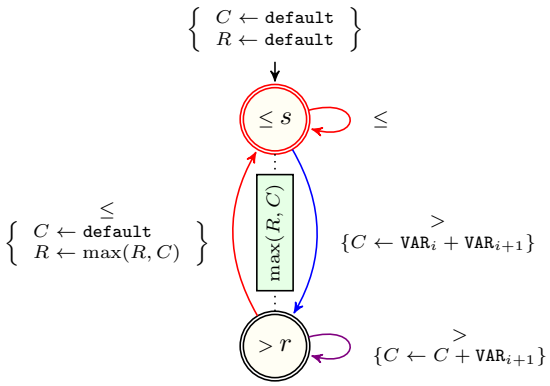


Figure 4.667: Simplified automaton for the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.77: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \texttt{VAR}_{i+1}$ M

Table 4.78: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint MAX_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

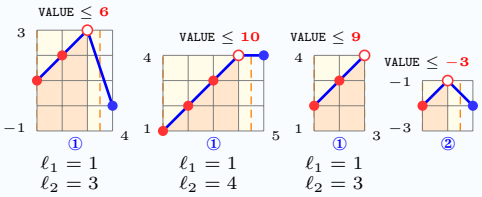
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$①
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$②
among(n1, VARIABLES[1, sv], ⟨maxv − 1, maxv⟩)
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
among(n2, VARIABLES[1, sv], ⟨minv, minv + 1⟩)
$\texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
required(VARIABLES, var)

where
maxv = maxval(VARIABLES.var)
rv = range(VARIABLES.var)
sv = |VARIABLES|
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
minv = minval(VARIABLES.var)



Purpose VALUE is the maximal surface of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(16, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.668 provides an example where the MAX_SURF_STRICTLY_INCREASING_SEQUENCE $(16, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

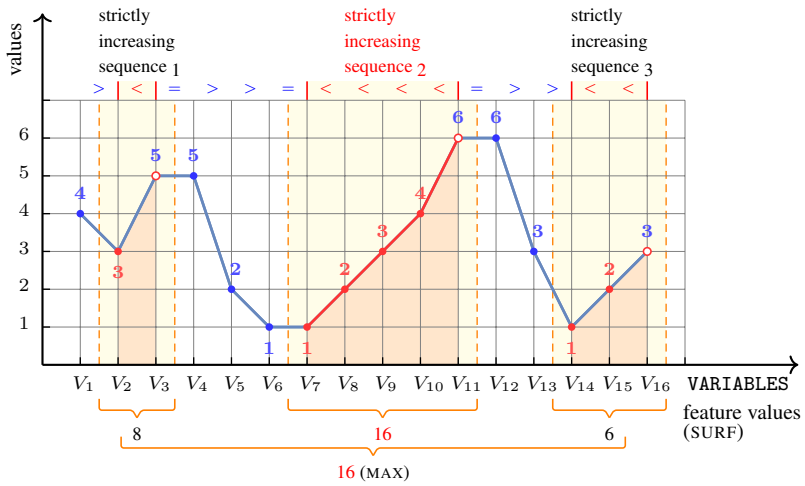


Figure 4.668: Illustrating the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.669 and 4.670 respectively depict the automaton associated with the constraint MAX_SURF_STRICTLY_INCREASING_SEQUENCE and its simplified form.

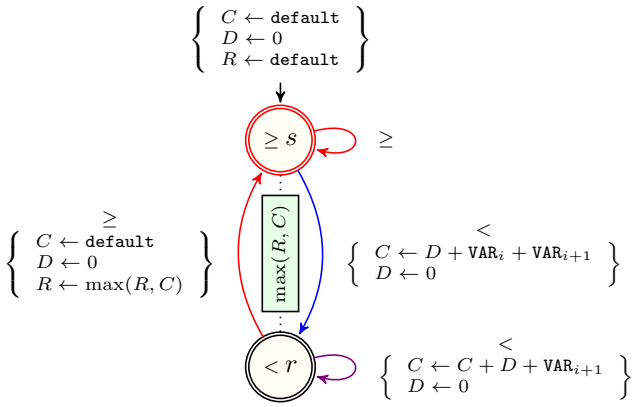


Figure 4.669: Automaton for the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$

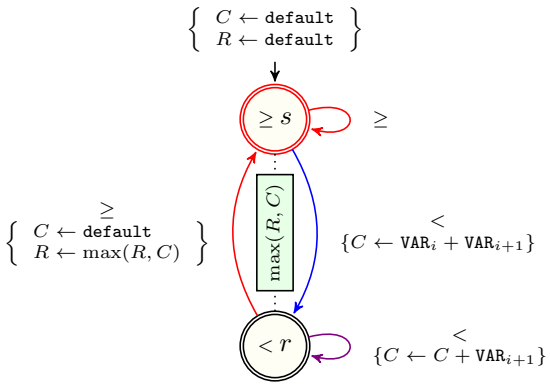


Figure 4.670: Simplified automaton for the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.79: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \texttt{VAR}_{i+1}$ M

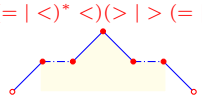
Table 4.80: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the SUMMIT pattern.

Constraint MAX_SURF_SUMMIT(VALUE, VARIABLES)

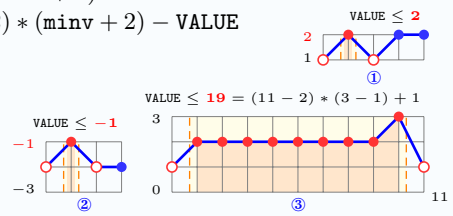
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{rv} \geq 3 \Rightarrow$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(\texttt{minv} + 1, (\texttt{sv} - 2) * (\texttt{minv} + 1) + 1)$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{maxv}$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(\texttt{maxv}$②$, (\texttt{sv} - 2) * (\texttt{maxv} - 1) + 1$③$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{maxv} - 1)$
$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - (\texttt{sv} - 2) * (\texttt{maxv} - 2) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \min(0, \texttt{minv} + 2) - \texttt{VALUE}$
$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq (\texttt{sv} - 2) * (\texttt{minv} + 2) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(13, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.671 provides an example where the MAX_SURF_SUMMIT $(13, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

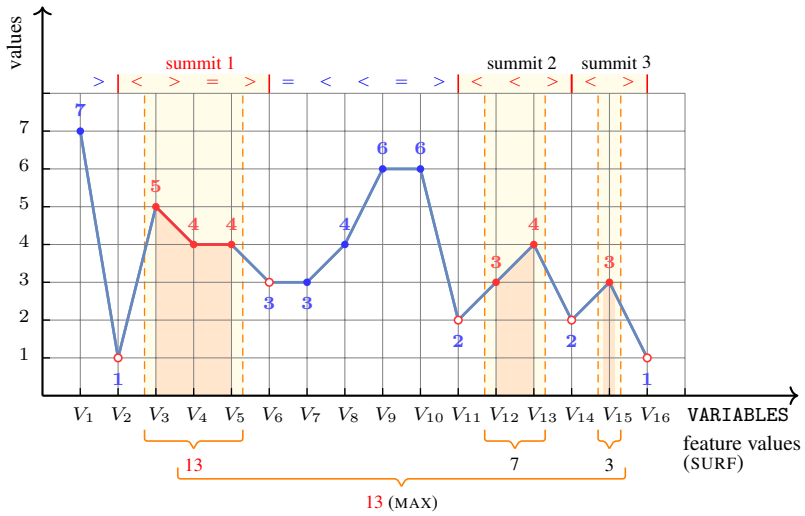


Figure 4.671: Illustrating the MAX_SURF_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.672 depicts the automaton associated with the constraint MAX_SURF_SUMMIT.

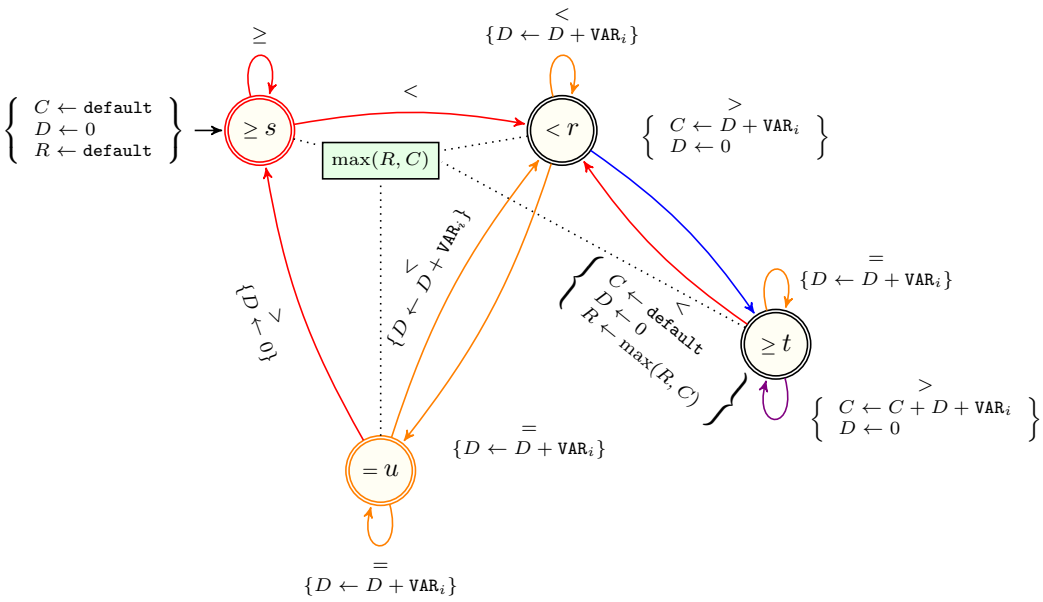


Figure 4.672: Automaton for the MAX_SURF_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is $-\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ is a linear invariant.

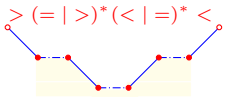
	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.81: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MAX_SURF_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

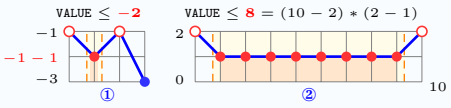
Constraint MAX_SURF_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(\texttt{minv}, (\texttt{sv}-2) * \texttt{minv})$
$\texttt{VALUE} \leq \max(\texttt{maxv}-1①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the maximal surface of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(15, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.673 provides an example where the MAX_SURF_VALLEY $(15, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

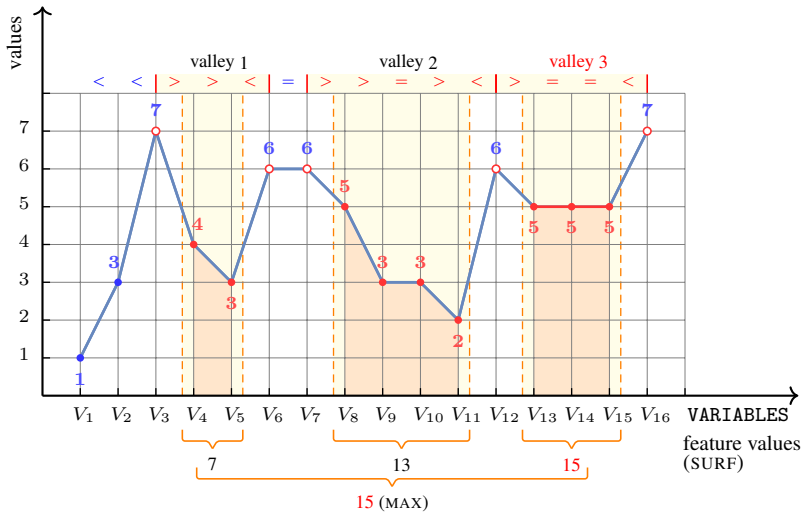


Figure 4.673: Illustrating the MAX_SURF_VALLEY constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.674 depicts the automaton associated with the constraint MAX_SURF_VALLEY.

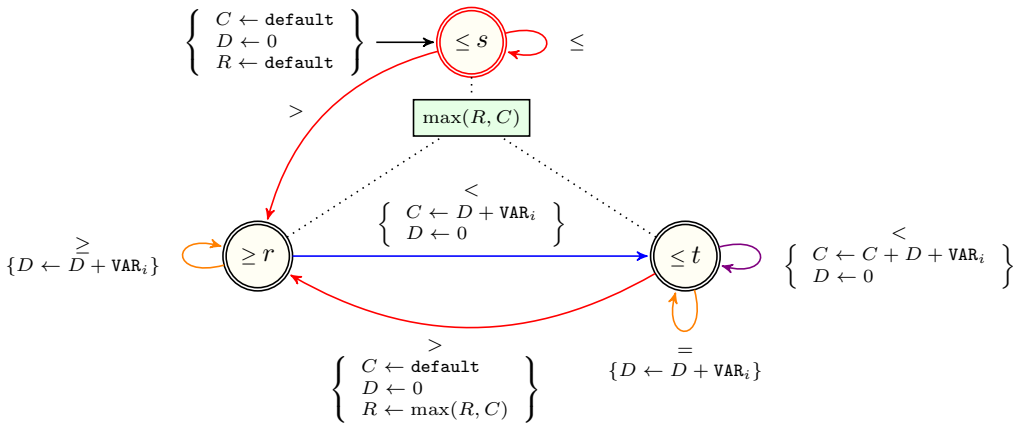


Figure 4.674: Automaton for the MAX_SURF_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is $-\infty$; $R_i - R_{i-1} \geq 0$ is a linear invariant.

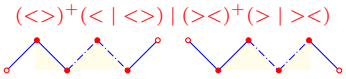
	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.82: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MAX_SURF_VALLEY constraint defined as the composition of the VALLEY pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_SURF_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint MAX_SURF_ZIGZAG(VALUE, VARIABLES)

Arguments

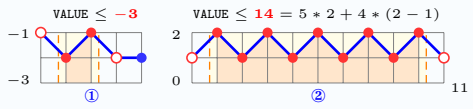
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\begin{array}{l}
\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty \\
\bigvee \left(\begin{array}{l} \mathtt{VALUE} = -\infty, \\ \mathtt{VALUE} \geq \mathtt{min} \left(\begin{array}{l} 2 * \mathtt{minv} + 1, \\ \lfloor (\mathtt{sv}-1)/2 \rfloor * \mathtt{minv} + \lfloor (\mathtt{sv}-2)/2 \rfloor * (\mathtt{minv}+1) \end{array} \right) \end{array} \right) \\
\mathtt{VALUE} \leq \mathtt{max} \left(\begin{array}{l} 2 * \mathtt{maxv} - 1①, \\ \lfloor (\mathtt{sv}-1)/2 \rfloor * \mathtt{maxv} + \lfloor (\mathtt{sv}-2)/2 \rfloor * (\mathtt{maxv}-1)② \end{array} \right) \\
\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1, \mathtt{maxv} \rangle) \\
\mathtt{n1} \geq \mathtt{VALUE} - \lfloor (\mathtt{sv}-1)/2 \rfloor - \mathtt{max}(0, (\mathtt{sv}-2) * (\mathtt{maxv}-2)) \\
\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}, \mathtt{minv}+1 \rangle) \\
\mathtt{n2} \geq \mathtt{min}(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \lfloor (\mathtt{sv}-1)/2 \rfloor - \mathtt{VALUE} \\
\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})
\end{array}$$

where

$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose VALUE is the maximal surface of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $-\infty$.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(21, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.675 provides an example where the MAX_SURF_ZIGZAG $(21, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

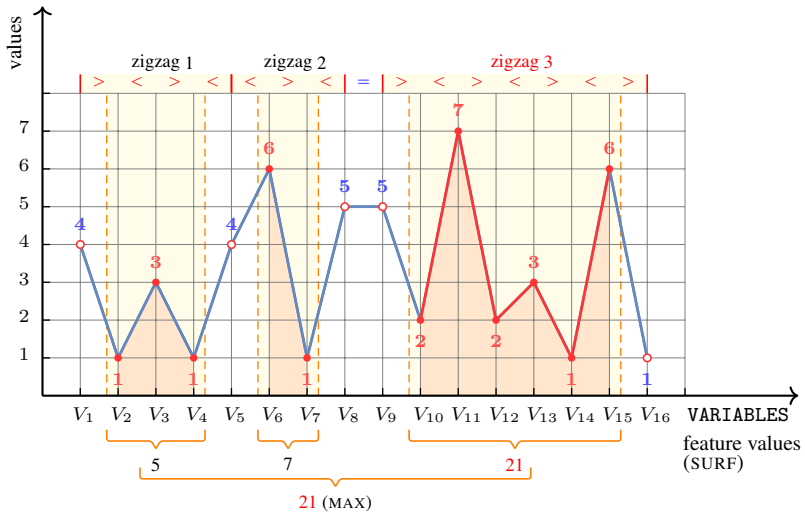


Figure 4.675: Illustrating the MAX_SURF_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.676 and 4.677 respectively depict the automaton associated with the constraint MAX_SURF_ZIGZAG and its simplified form.

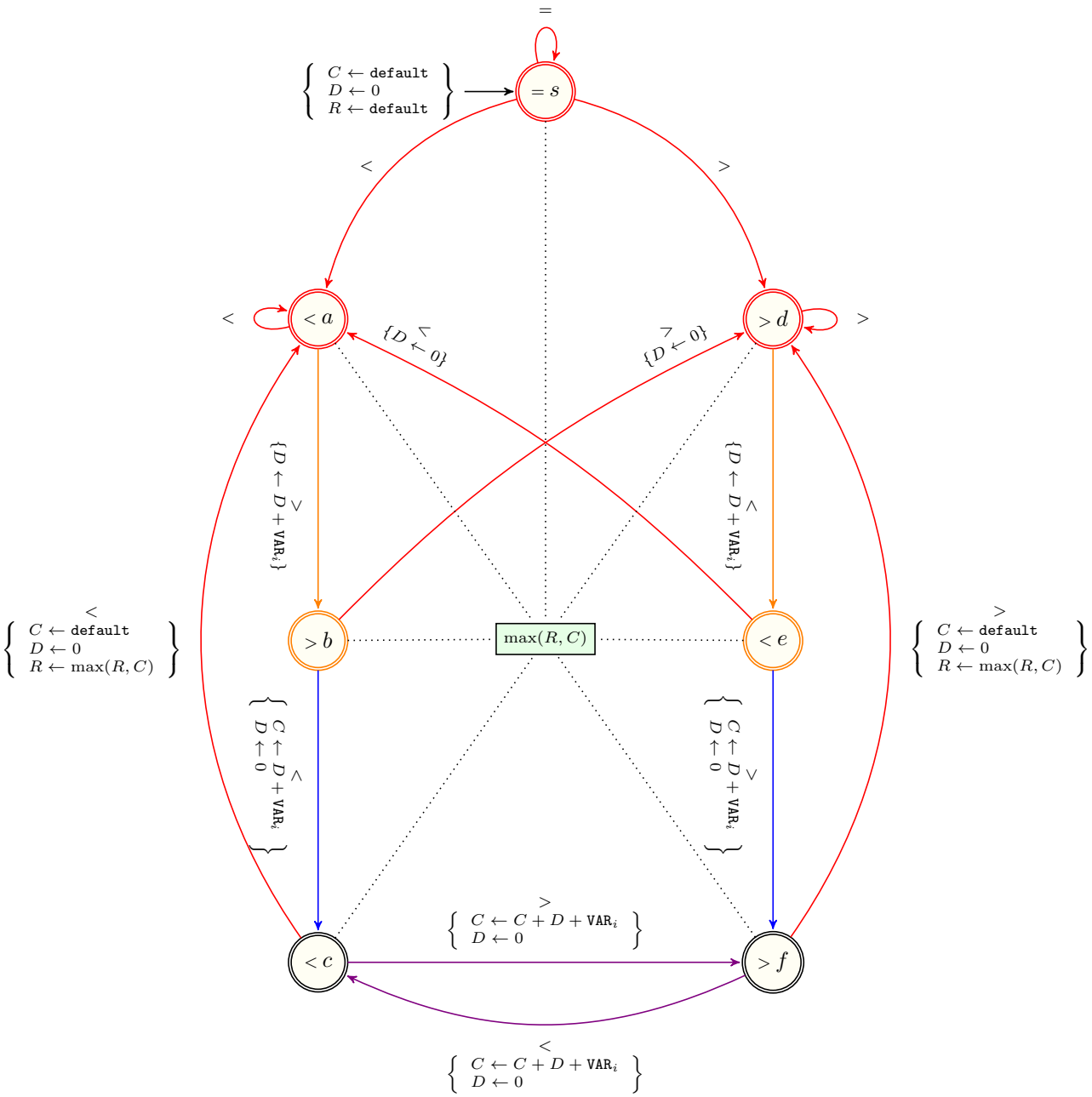


Figure 4.676: Automaton for the MAX_SURF_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

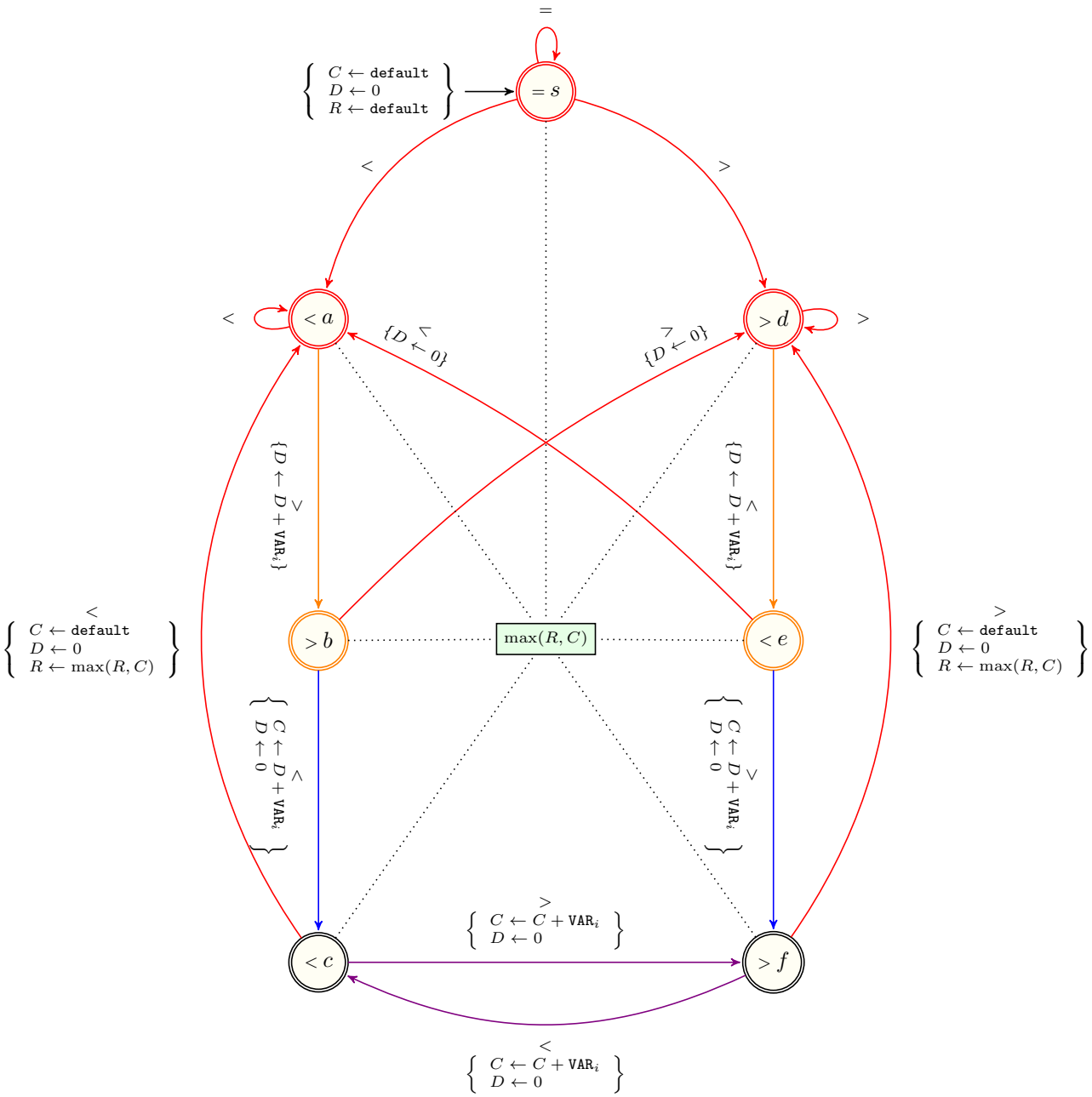


Figure 4.677: Simplified automaton for the MAX_SURF_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is $-\infty$; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.; $R_i - R_{i-1} \geq 0$ is a linear invariant.

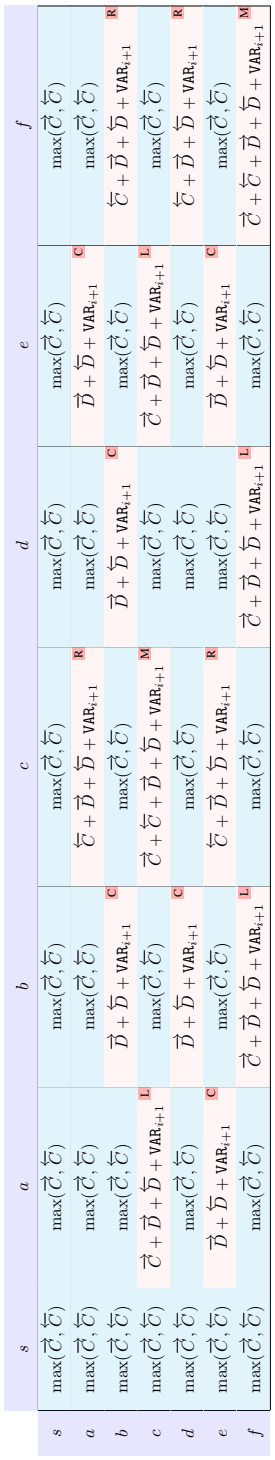


Table 4.83: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MAX_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature SURF, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

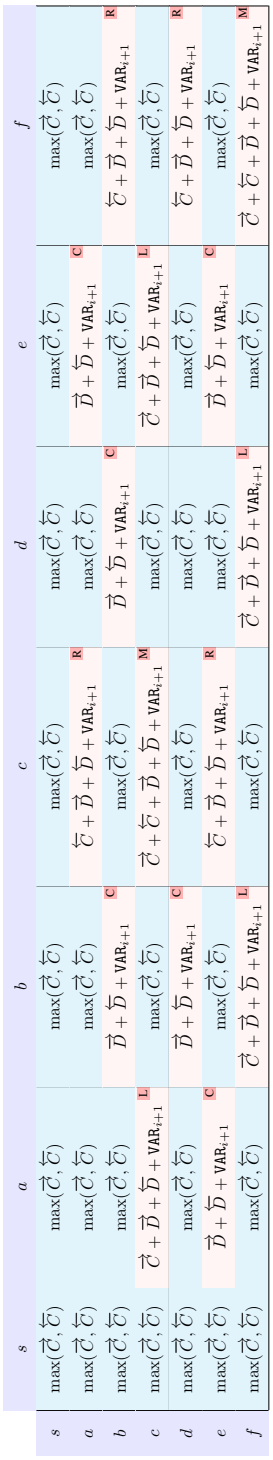
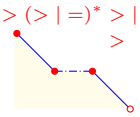


Table 4.84: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MAX_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern , the feature SURF , and the aggregator max ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

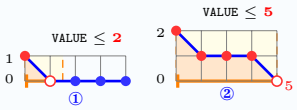
Constraint MAX_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(5, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.678 provides an example where the MAX_WIDTH_DECREASING_SEQUENCE $(5, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

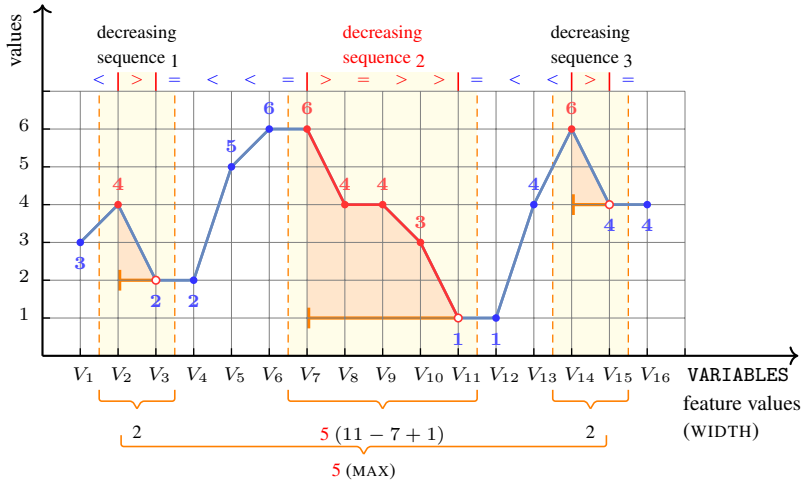


Figure 4.678: Illustrating the MAX_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.679 and 4.680 respectively depict the automaton associated with the constraint MAX_WIDTH_DECREASING_SEQUENCE and its simplified form.

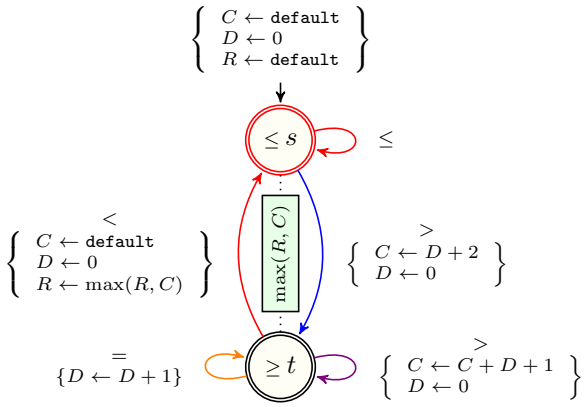


Figure 4.679: Automaton for the MAX_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.85: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MAX_WIDTH_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

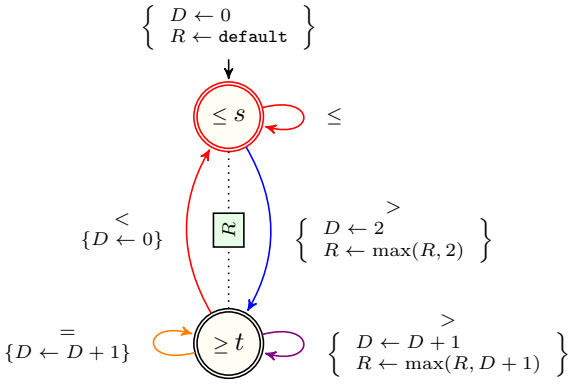


Figure 4.680: Simplified automaton for the MAX_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.31 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

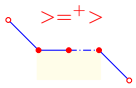
	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.86: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the MAX_WIDTH_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

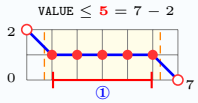
Constraint MAX_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3\rangle)$

Figure 4.681 provides an example where the MAX_WIDTH_DECREASING_TERRACE $(2, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

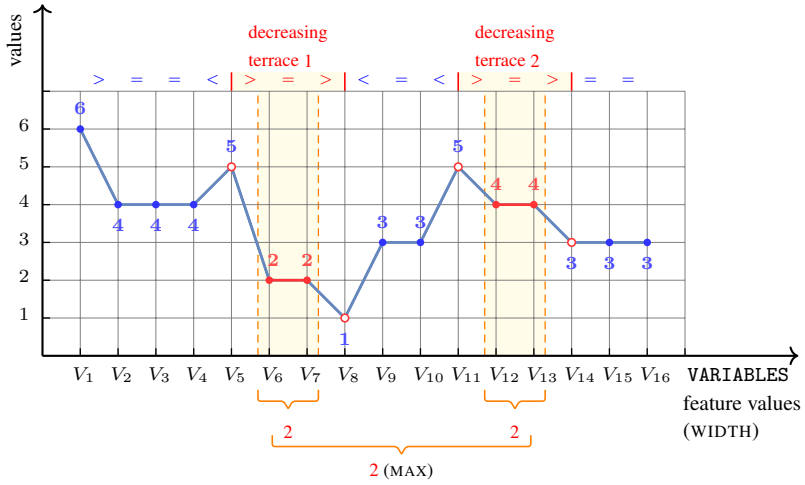


Figure 4.681: Illustrating the MAX_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.682 and 4.683 respectively depict the automaton associated with the constraint MAX_WIDTH_DECREASING_TERRACE and its simplified form.

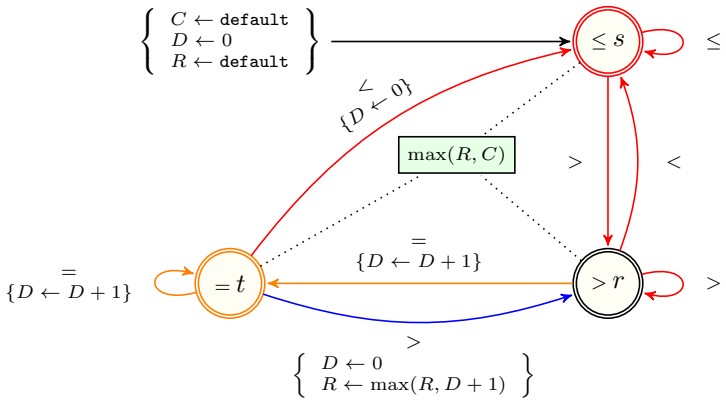


Figure 4.682: Automaton for the MAX_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.87: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MAX_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

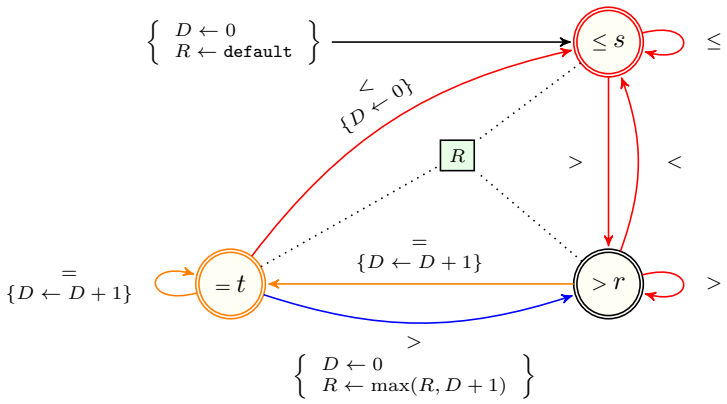


Figure 4.683: Simplified automaton for the MAX_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

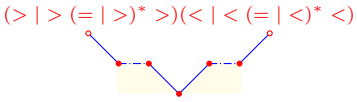
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.88: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MAX_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

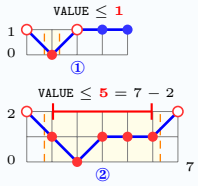
Constraint MAX_WIDTH_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.684 provides an example where the MAX_WIDTH_GORGE $(3, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

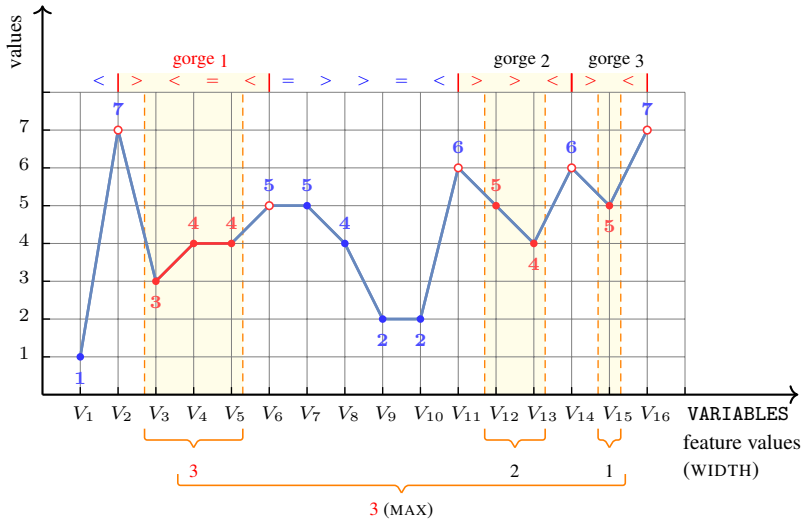


Figure 4.684: Illustrating the MAX_WIDTH_GORGE constraint of the **Example** slot

Automaton Figures 4.685 and 4.686 respectively depict the automaton associated with the constraint MAX_WIDTH_GORGE and its simplified form.

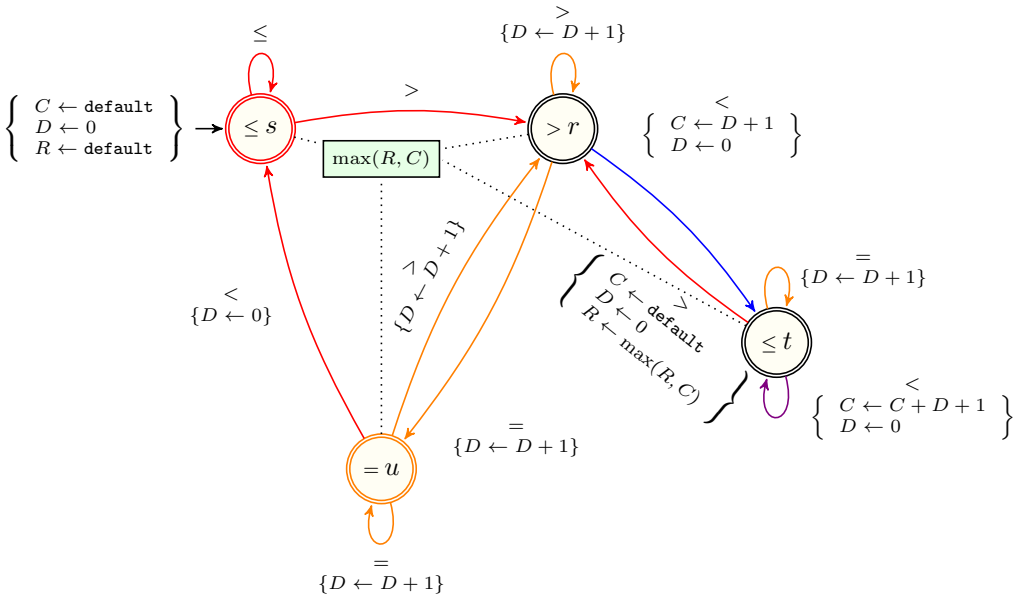


Figure 4.685: Automaton for the MAX_WIDTH_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

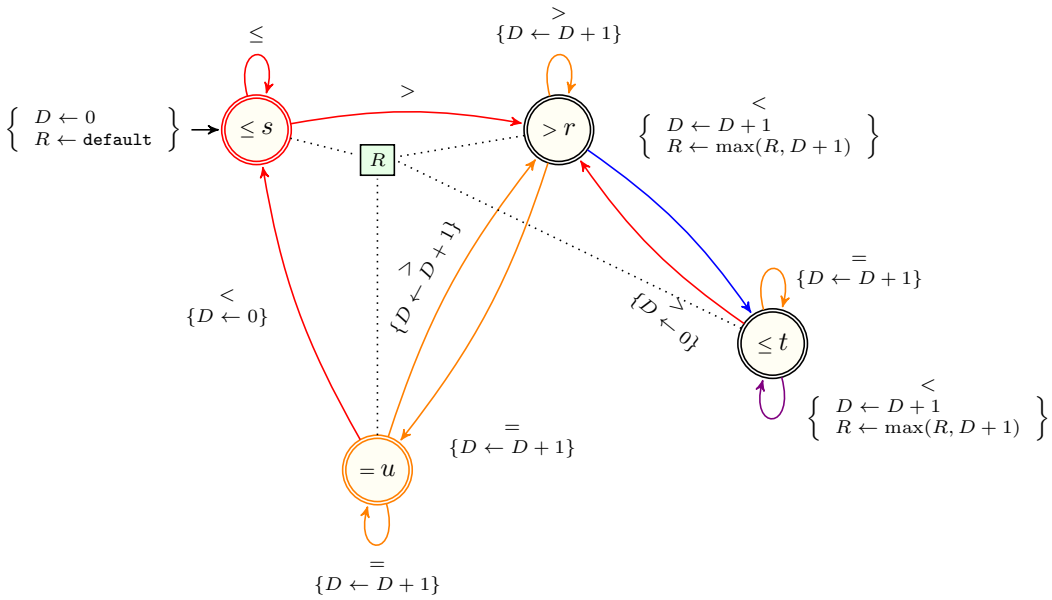


Figure 4.686: Simplified automaton for the MAX_WIDTH_GORGE constraint obtained by applying decoration Table 3.27 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

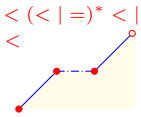
Table 4.89: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MAX_WIDTH_GORGE constraint defined as the composition of the GORGE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0

Table 4.90: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the MAX_WIDTH_GORGE constraint defined as the composition of the GORGE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

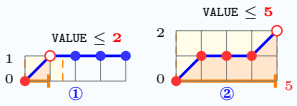
Constraint MAX_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.687 provides an example where the MAX_WIDTH_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

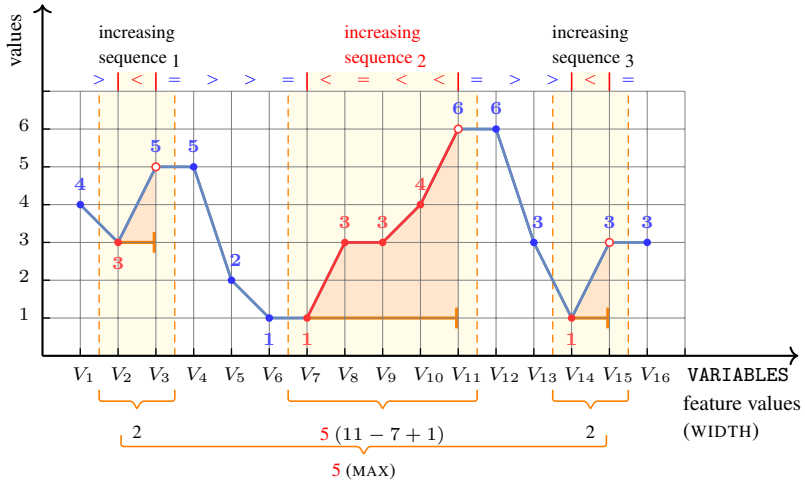


Figure 4.687: Illustrating the MAX_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.688 and 4.689 respectively depict the automaton associated with the constraint MAX_WIDTH_INCREASING_SEQUENCE and its simplified form.

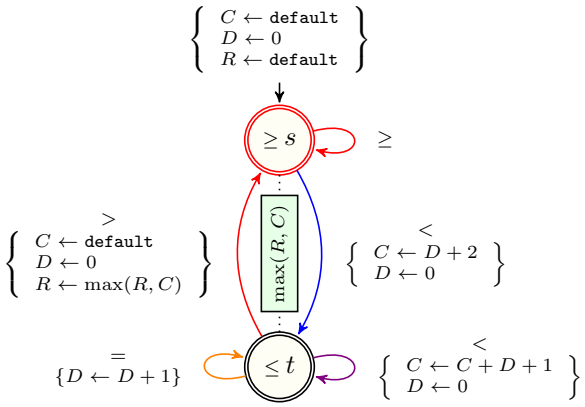


Figure 4.688: Automaton for the MAX_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.91: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MAX_WIDTH_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern , the feature WIDTH , and the aggregator max ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

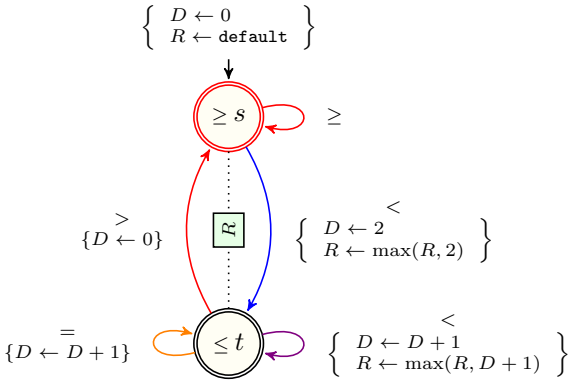


Figure 4.689: Simplified automaton for the MAX_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.31 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

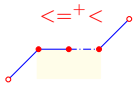
	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.92: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the MAX_WIDTH_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

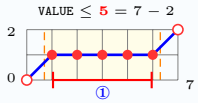
Constraint MAX_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4\rangle)$

Figure 4.690 provides an example where the MAX_WIDTH_INCREASING_TERRACE $(3, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

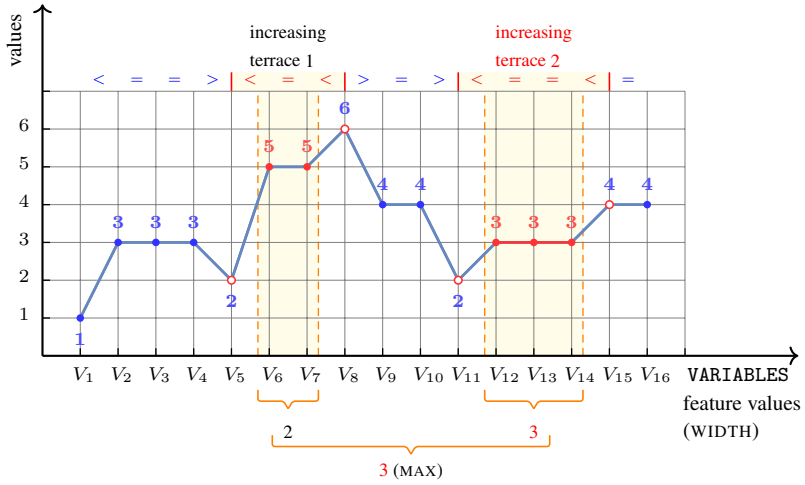


Figure 4.690: Illustrating the MAX_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.691 and 4.692 respectively depict the automaton associated with the constraint MAX_WIDTH_INCREASING_TERRACE and its simplified form.

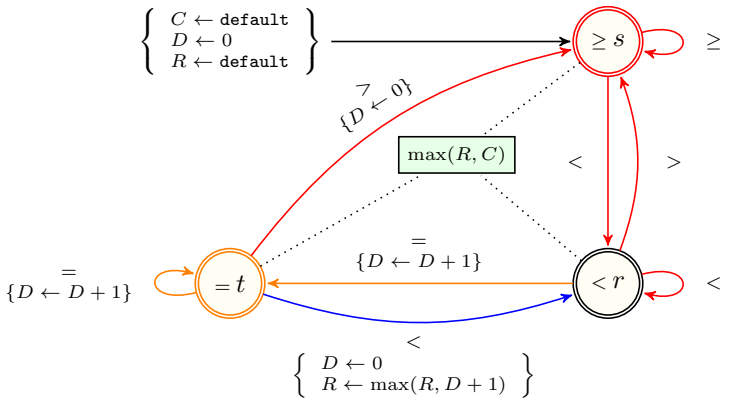


Figure 4.691: Automaton for the MAX_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.93: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MAX_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

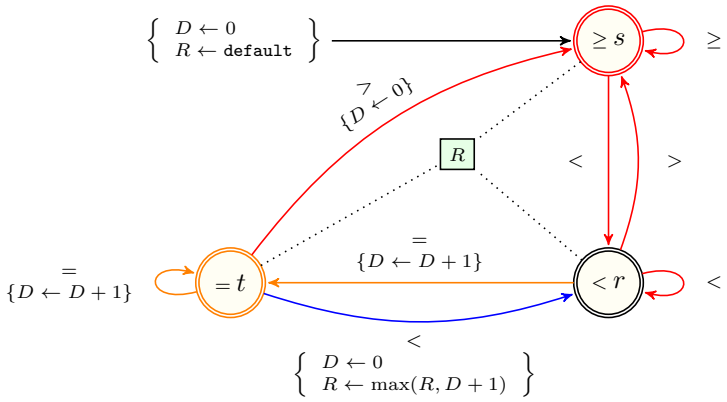


Figure 4.692: Simplified automaton for the MAX_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

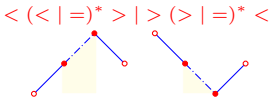
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.94: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MAX_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

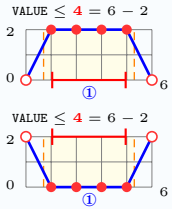
Constraint MAX_WIDTH_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal width of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.693 provides an example where the MAX_WIDTH_INFLEXION $(3, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

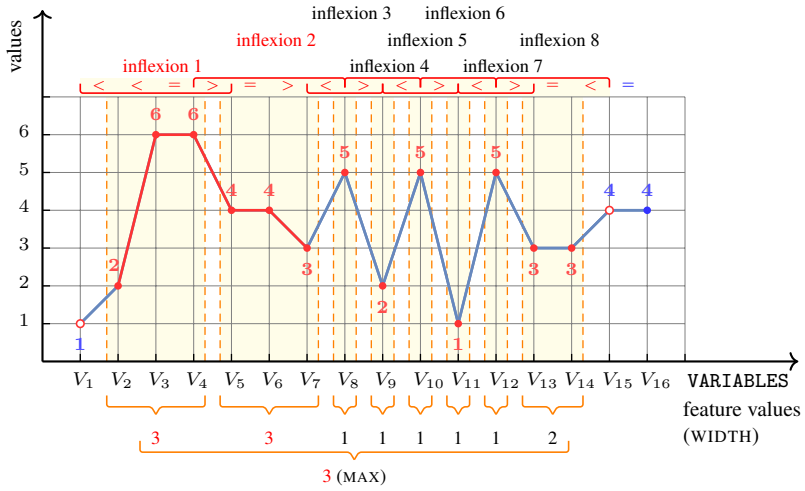


Figure 4.693: Illustrating the MAX_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.694 and 4.695 respectively depict the automaton associated with the constraint MAX_WIDTH_INFLEXION and its simplified form.

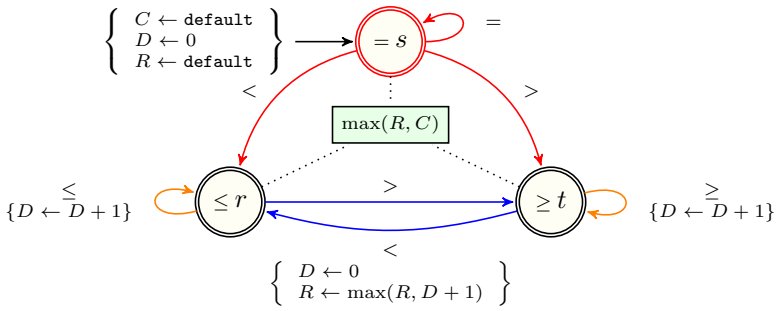


Figure 4.694: Automaton for the MAX_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

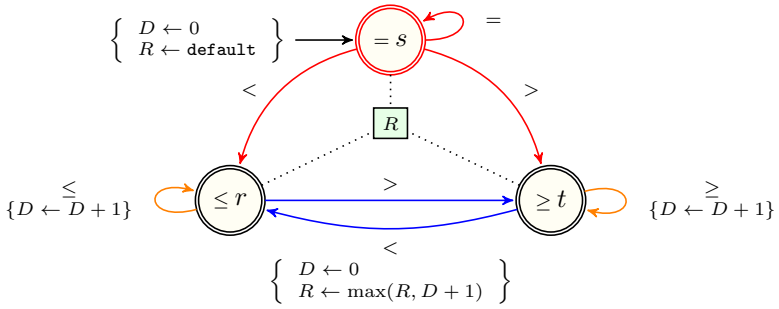
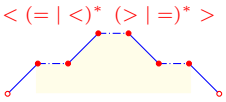


Figure 4.695: Simplified automaton for the MAX_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_PEAK



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PEAK pattern.

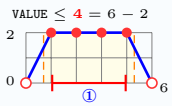
Constraint MAX_WIDTH_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the maximal width of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.696 provides an example where the MAX_WIDTH_PEAK $(3, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

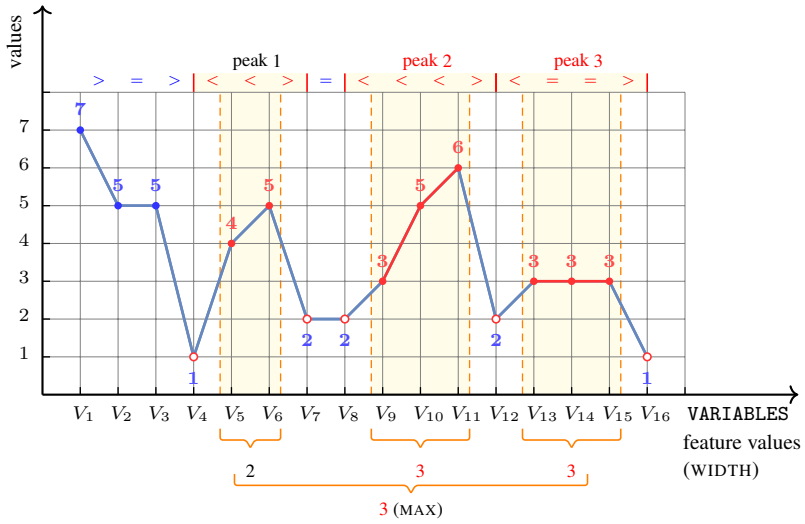


Figure 4.696: Illustrating the MAX_WIDTH_PEAK constraint of the **Example** slot

Automaton

Figures 4.697 and 4.698 respectively depict the automaton associated with the constraint MAX_WIDTH_PEAK and its simplified form.

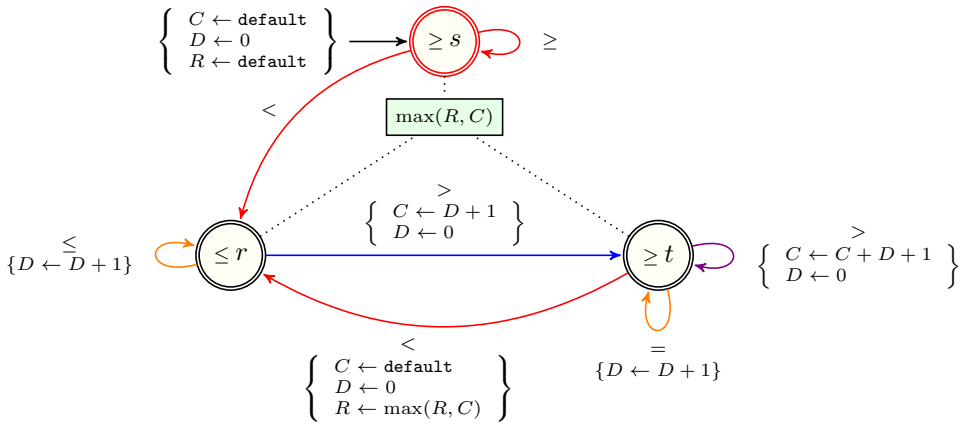


Figure 4.697: Automaton for the MAX_WIDTH_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is 0

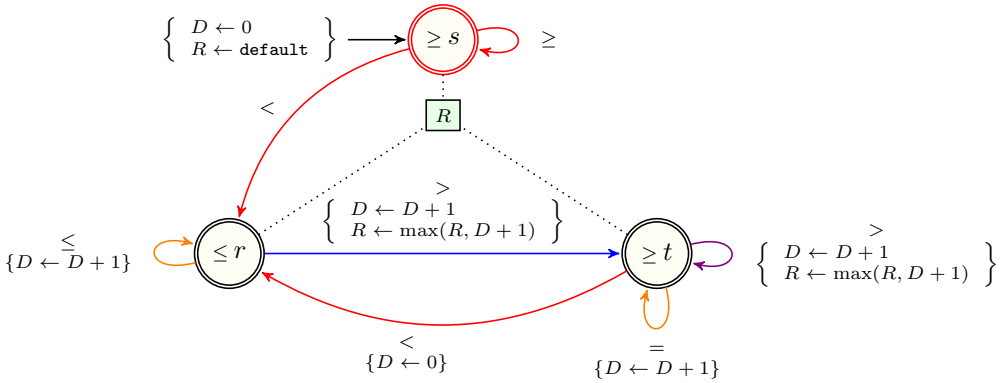


Figure 4.698: Simplified automaton for the MAX_WIDTH_PEAK constraint obtained by applying decoration Table 3.27 to the seed transducer of the PEAK pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.95: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MAX_WIDTH_PEAK constraint defined as the composition of the PEAK pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0

Table 4.96: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the MAX_WIDTH_PEAK constraint defined as the composition of the PEAK pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

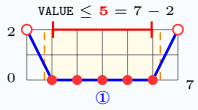
Constraint MAX_WIDTH_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3\rangle)$

Figure 4.699 provides an example where the MAX_WIDTH_PLAIN $(2, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

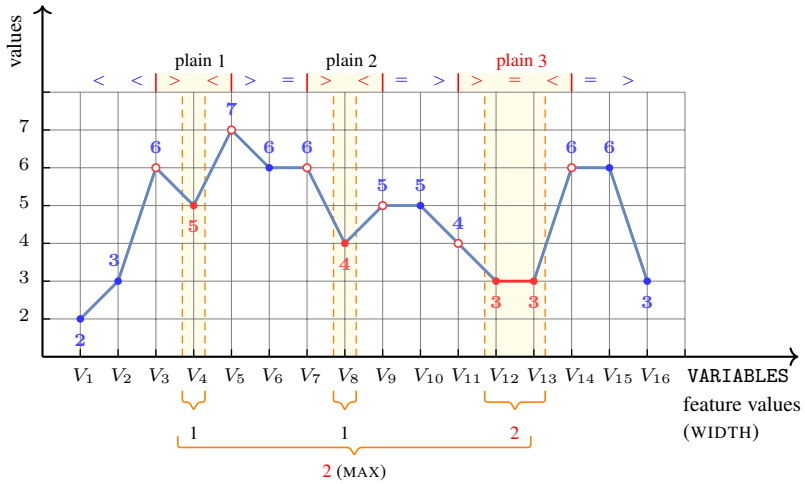


Figure 4.699: Illustrating the MAX_WIDTH_PLAIN constraint of the **Example** slot

Automaton Figures 4.700 and 4.701 respectively depict the automaton associated with the constraint MAX_WIDTH_PLAIN and its simplified form.

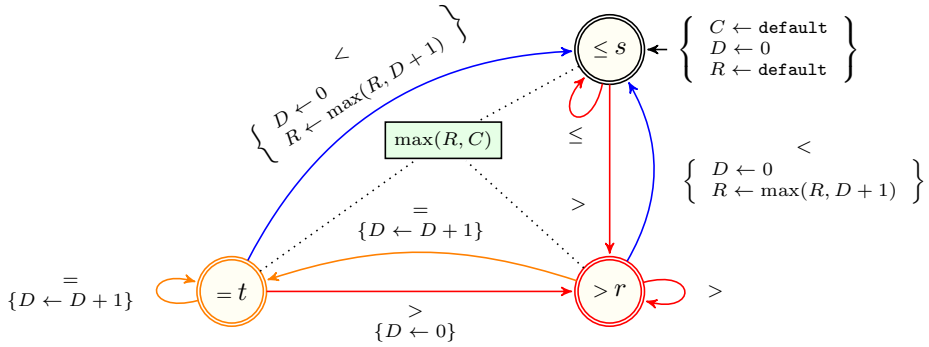


Figure 4.700: Automaton for the MAX_WIDTH_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is 0

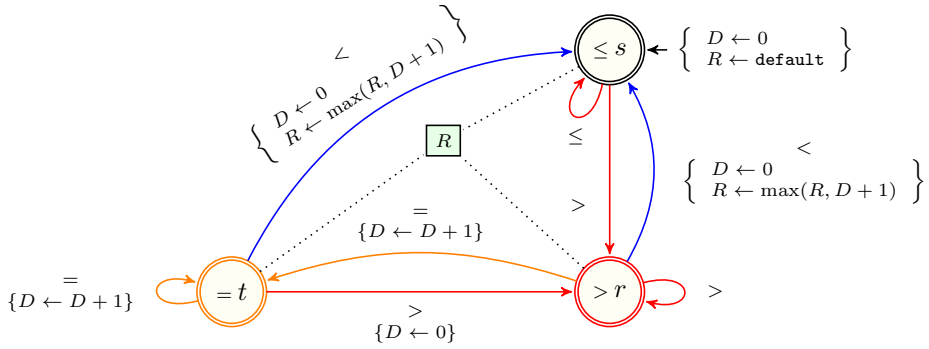


Figure 4.701: Simplified automaton for the MAX_WIDTH_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLAIN pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.97: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MAX_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.98: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MAX_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

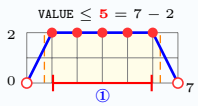
Constraint MAX_WIDTH_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(4, \langle 1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5 \rangle)$

Figure 4.702 provides an example where the MAX_WIDTH_PLATEAU $(4, [1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

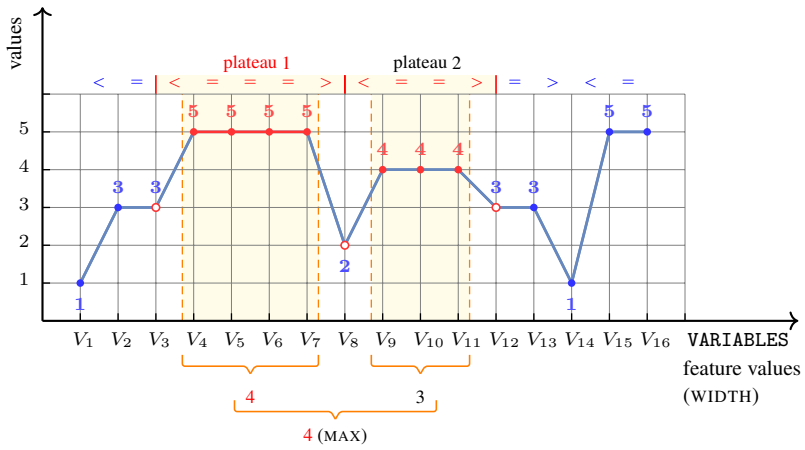


Figure 4.702: Illustrating the MAX_WIDTH_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.703 and 4.704 respectively depict the automaton associated with the constraint MAX_WIDTH_PLATEAU and its simplified form.

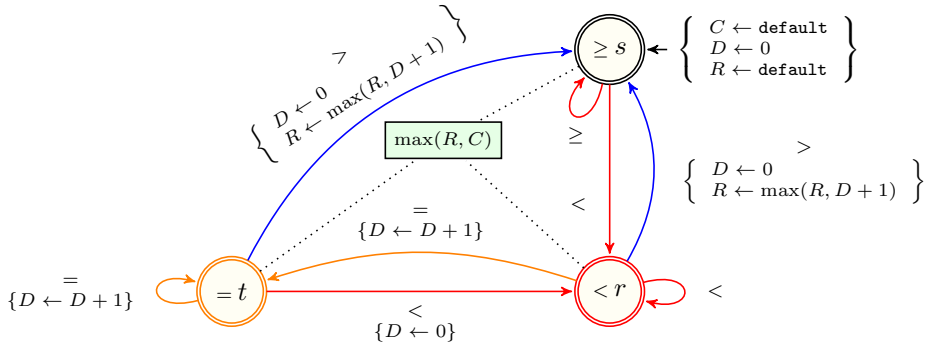


Figure 4.703: Automaton for the MAX_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is 0

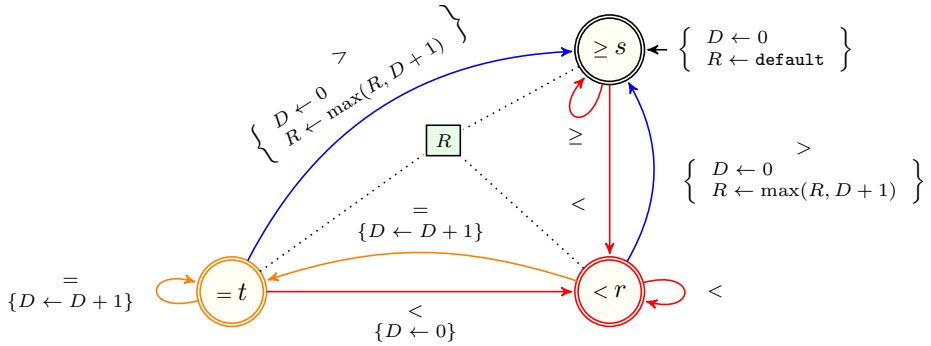


Figure 4.704: Simplified automaton for the MAX_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.99: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MAX_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

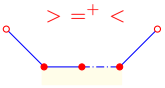
	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.100: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MAX_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

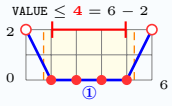
Constraint MAX_WIDTH_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5\rangle)$

Figure 4.705 provides an example where the MAX_WIDTH_PROPER_PLAIN $(3, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

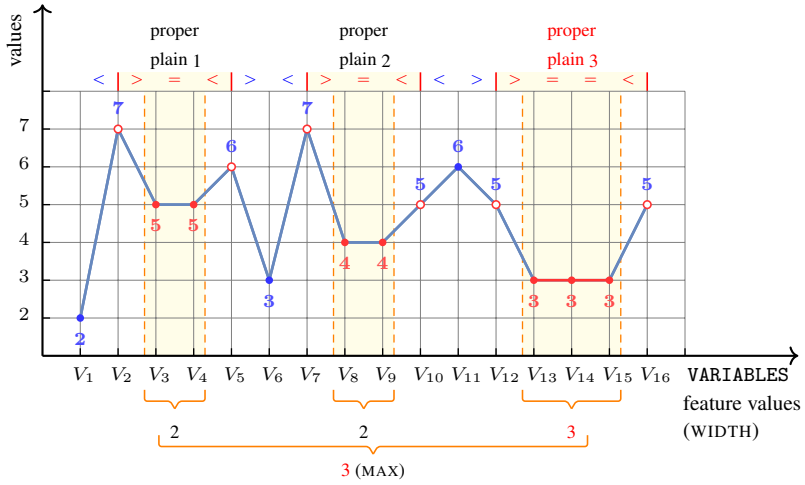


Figure 4.705: Illustrating the MAX_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.706 and 4.707 respectively depict the automaton associated with the constraint MAX_WIDTH_PROPER_PLAIN and its simplified form.

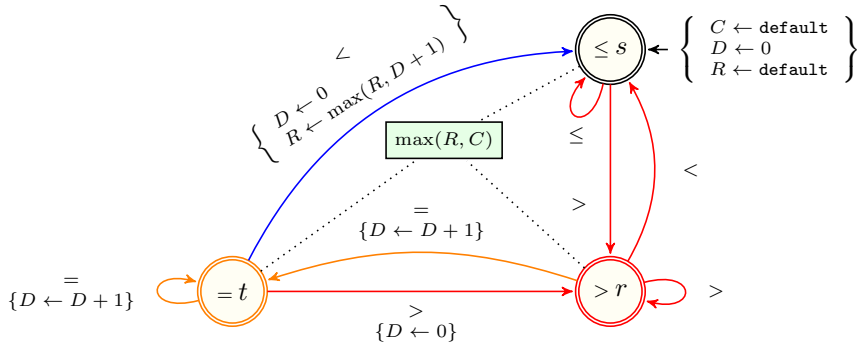


Figure 4.706: Automaton for the MAX_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

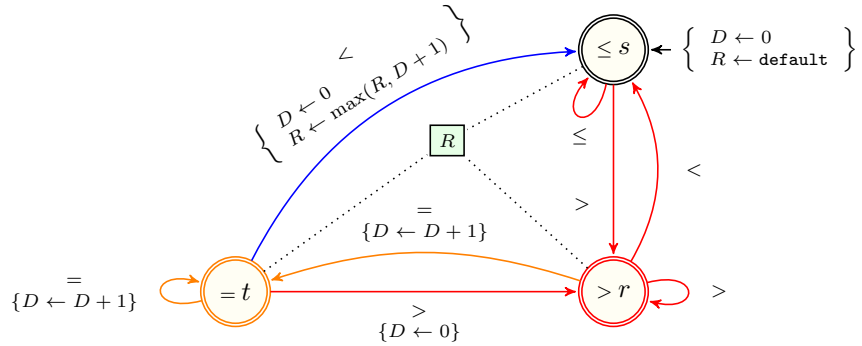


Figure 4.707: Simplified automaton for the MAX_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.101: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MAX_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

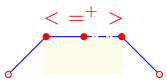
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.102: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MAX_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

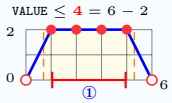
Constraint MAX_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3\rangle)$

Figure 4.708 provides an example where the MAX_WIDTH_PROPER_PLATEAU $(3, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

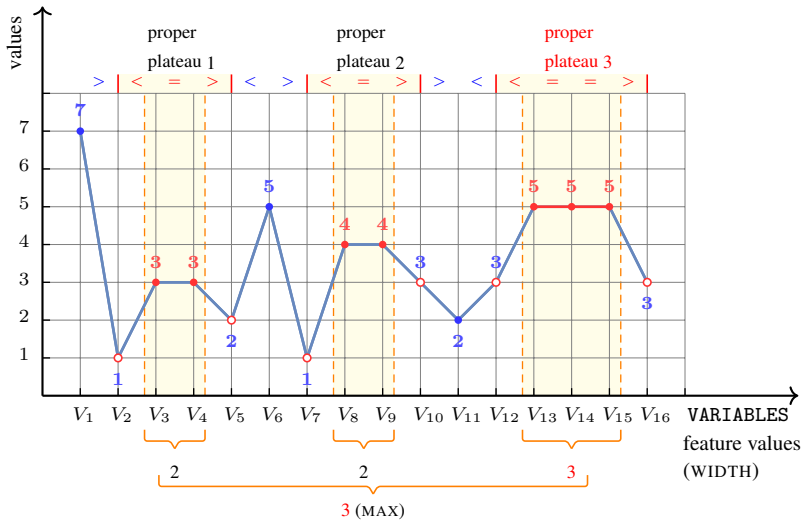


Figure 4.708: Illustrating the MAX_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.709 and 4.710 respectively depict the automaton associated with the constraint MAX_WIDTH_PROPER_PLATEAU and its simplified form.

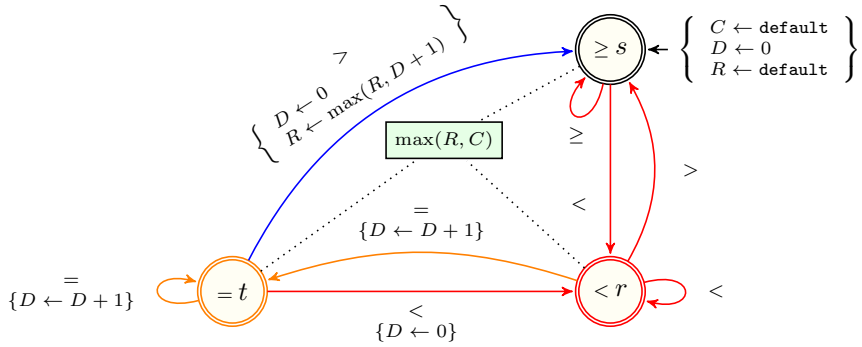


Figure 4.709: Automaton for the MAX_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

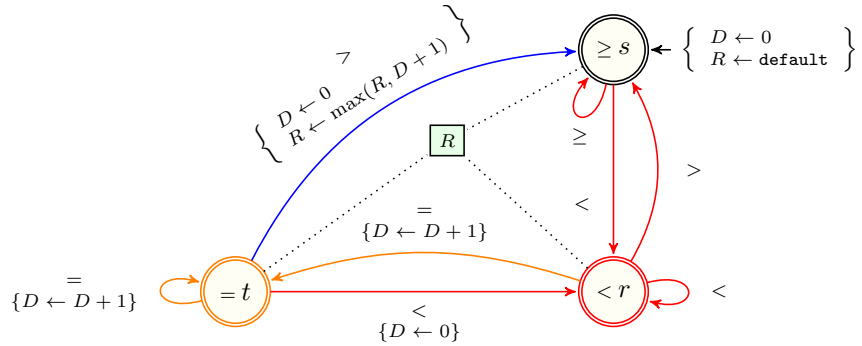


Figure 4.710: Simplified automaton for the MAX_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.103: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MAX_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

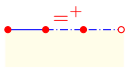
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.104: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MAX_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

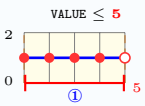
Constraint MAX_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq \texttt{sv}$
$\texttt{rv} \geq 2 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{sv}$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(3, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.711 provides an example where the MAX_WIDTH_STEADY_SEQUENCE $(3, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

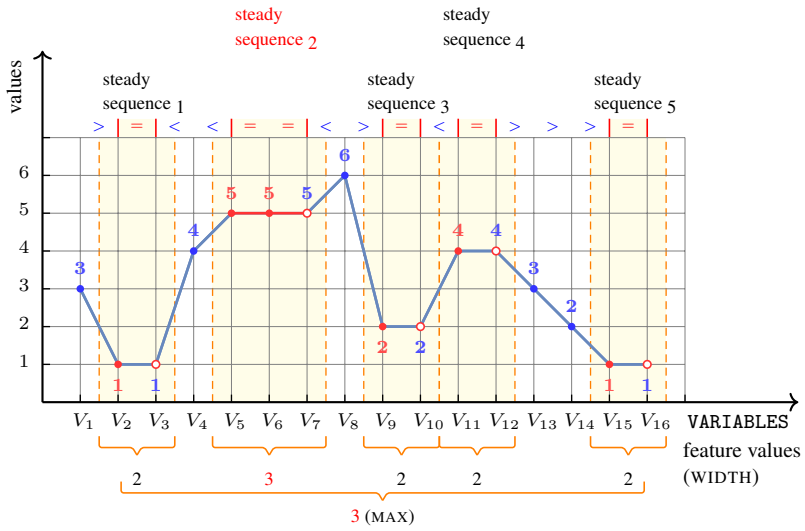


Figure 4.711: Illustrating the MAX_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.712 and 4.713 respectively depict the automaton associated with the constraint MAX_WIDTH_STEADY_SEQUENCE and its simplified form.

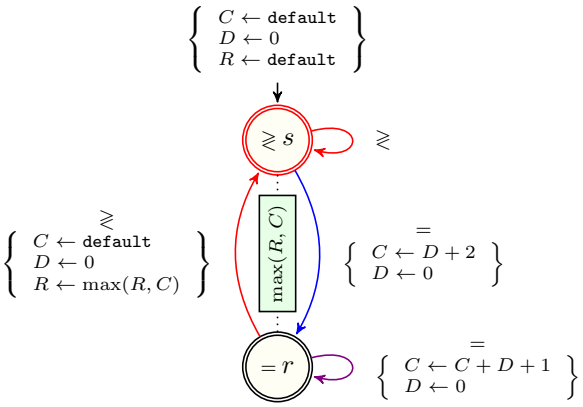


Figure 4.712: Automaton for the MAX_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.105: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MAX_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

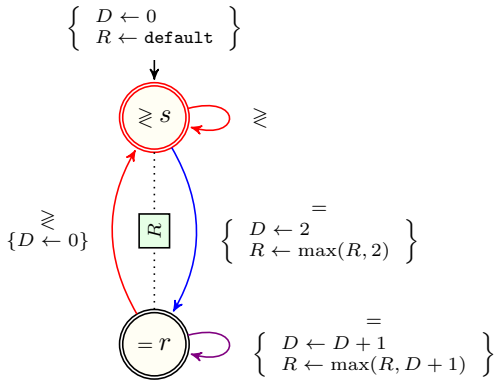


Figure 4.713: Simplified automaton for the MAX_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.31 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.106: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MAX_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

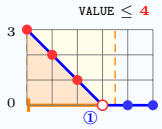
Constraint MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}[1])$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the maximal width of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^{+}$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(3, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.714 provides an example where the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE $(3, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

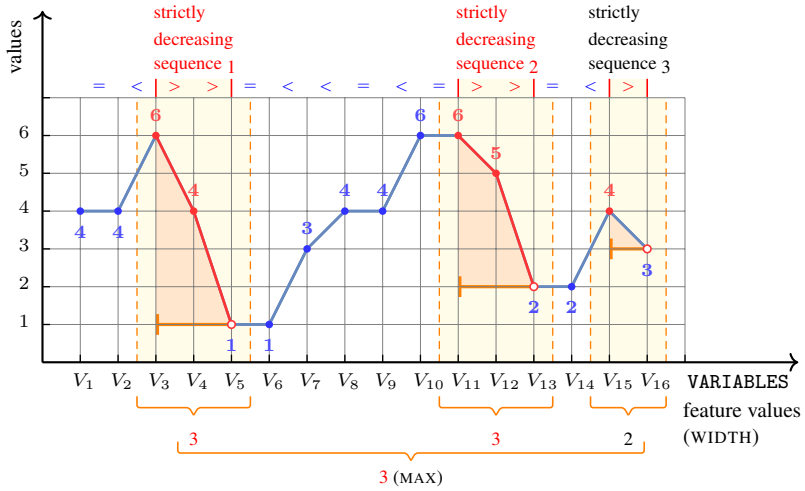


Figure 4.714: Illustrating the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.715 and 4.716 respectively depict the automaton associated with the constraint MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE and its simplified form.

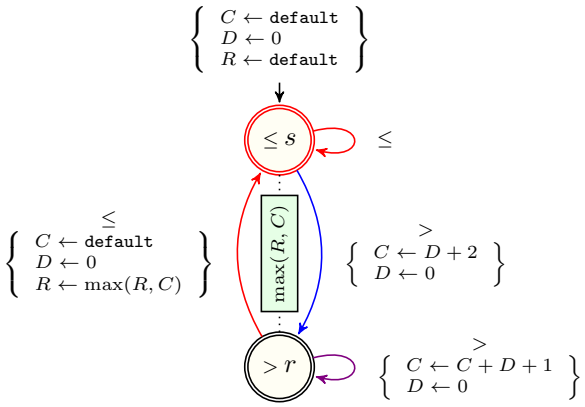


Figure 4.715: Automaton for the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.107: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

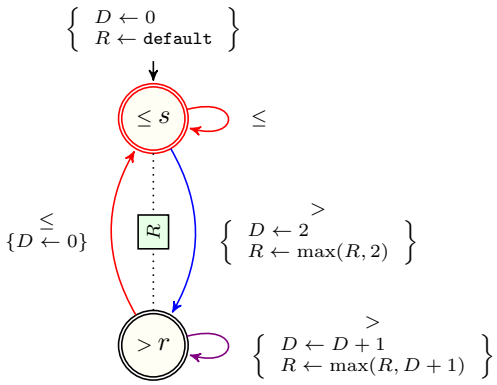


Figure 4.716: Simplified automaton for the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.31 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where default is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.108: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

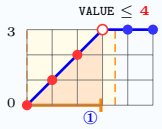
Constraint MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}[1])$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.717 provides an example where the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

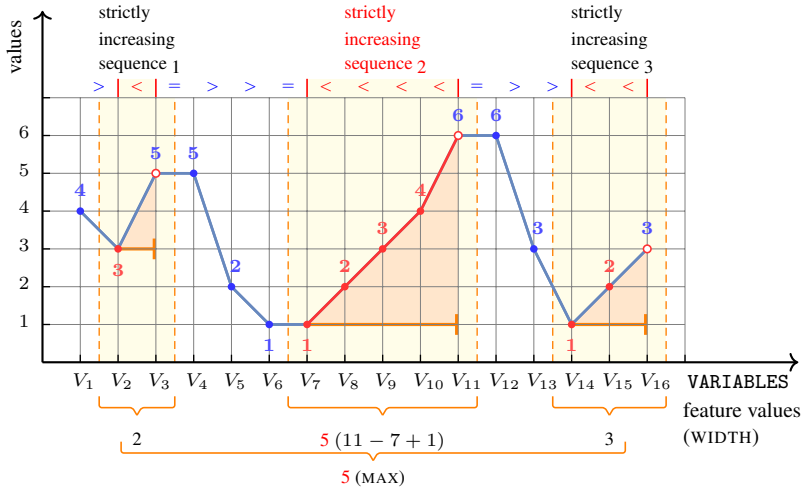


Figure 4.717: Illustrating the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.718 and 4.719 respectively depict the automaton associated with the constraint MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE and its simplified form.

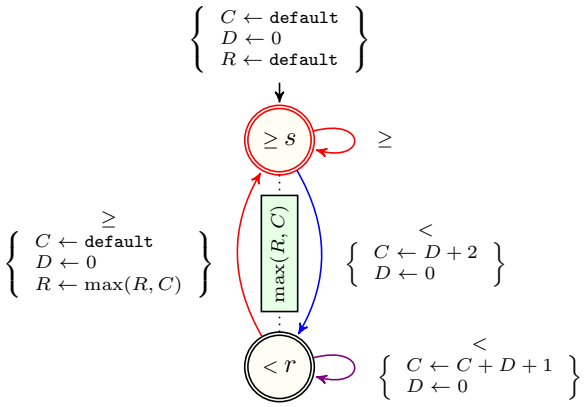


Figure 4.718: Automaton for the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.109: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

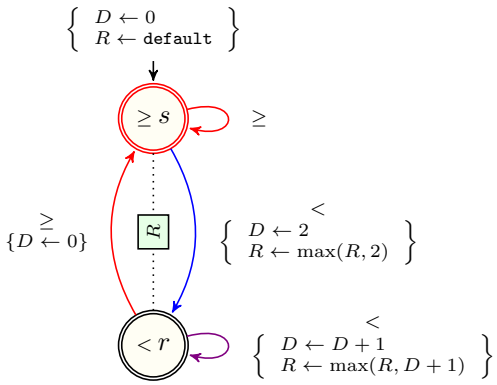


Figure 4.719: Simplified automaton for the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.31 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where default is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

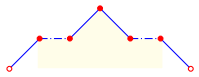
	s	r
s	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.110: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

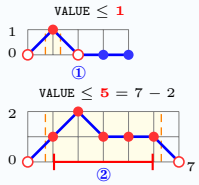
Constraint MAX_WIDTH_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.720 provides an example where the MAX_WIDTH_SUMMIT $(3, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

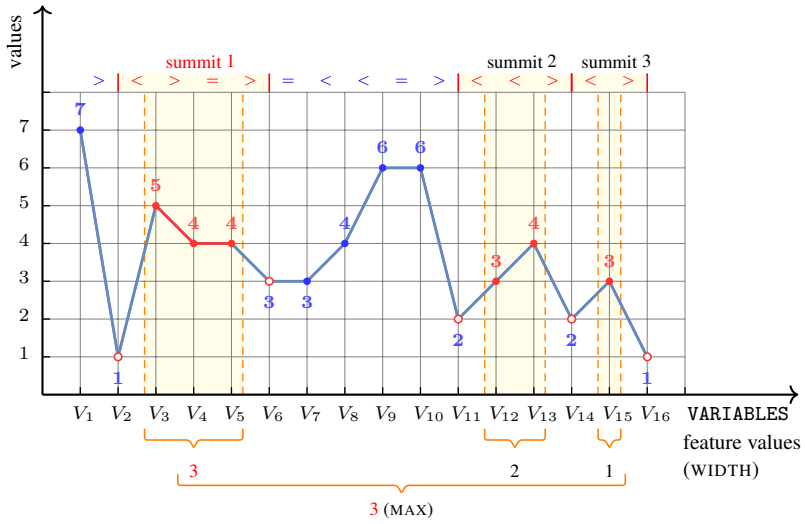


Figure 4.720: Illustrating the MAX_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figures 4.721 and 4.722 respectively depict the automaton associated with the constraint MAX_WIDTH_SUMMIT and its simplified form.

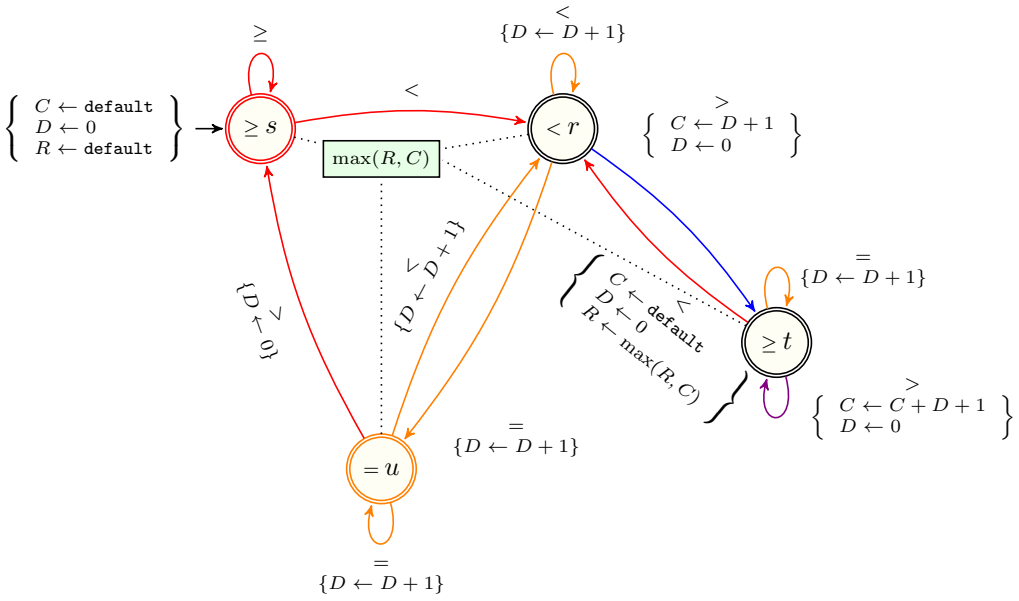


Figure 4.721: Automaton for the MAX_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

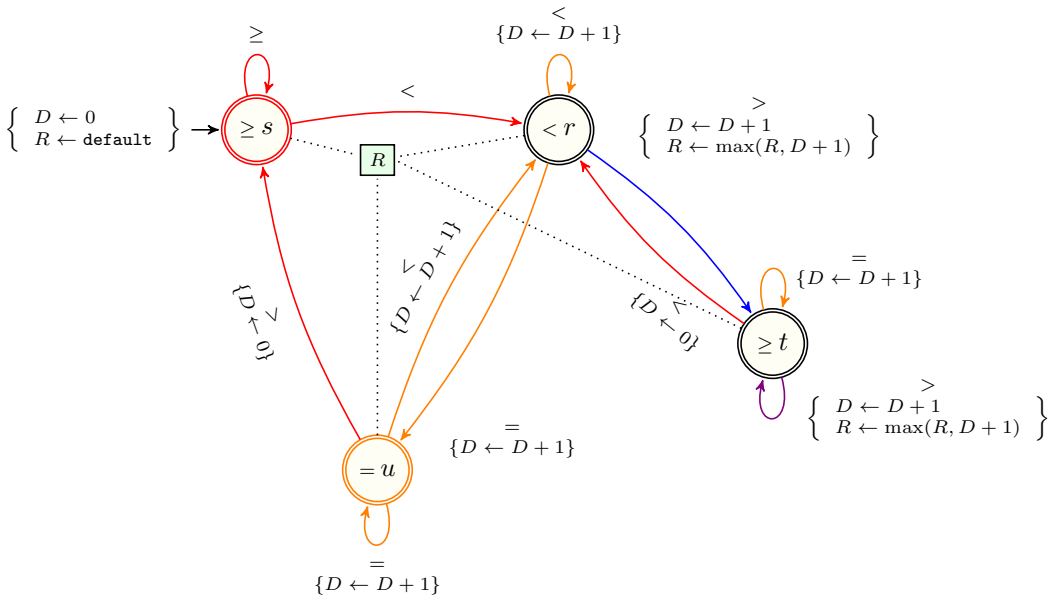


Figure 4.722: Simplified automaton for the MAX_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.27 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t	u
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.111: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MAX_WIDTH_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

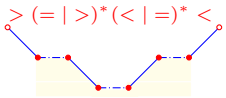
	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0

Table 4.112: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the MAX_WIDTH_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_VALLEY

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the VALLEY pattern.

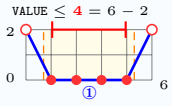
Constraint MAX_WIDTH_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(4, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.723 provides an example where the MAX_WIDTH_VALLEY $(4, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

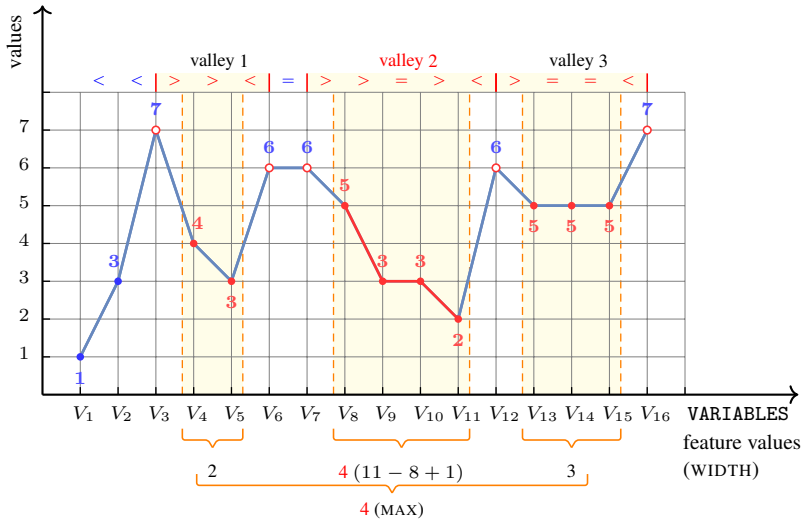


Figure 4.723: Illustrating the MAX_WIDTH_VALLEY constraint of the **Example** slot

Automaton

Figures 4.724 and 4.725 respectively depict the automaton associated with the constraint MAX_WIDTH_VALLEY and its simplified form.

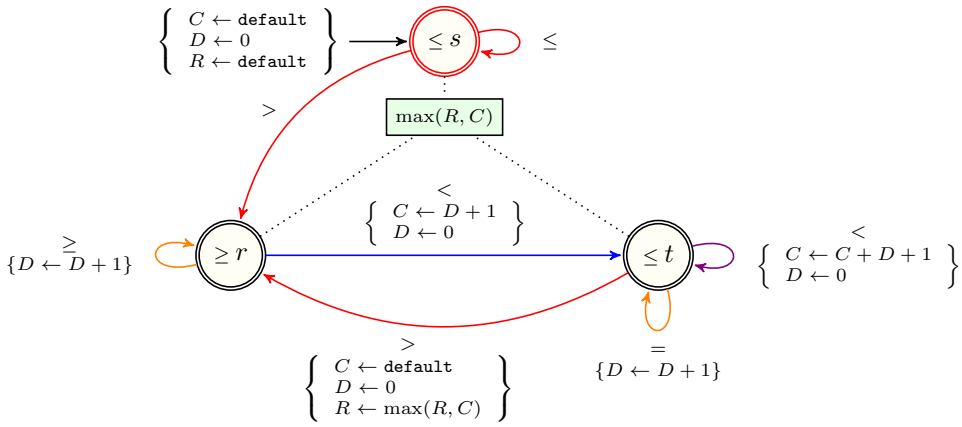


Figure 4.724: Automaton for the MAX_WIDTH_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is 0

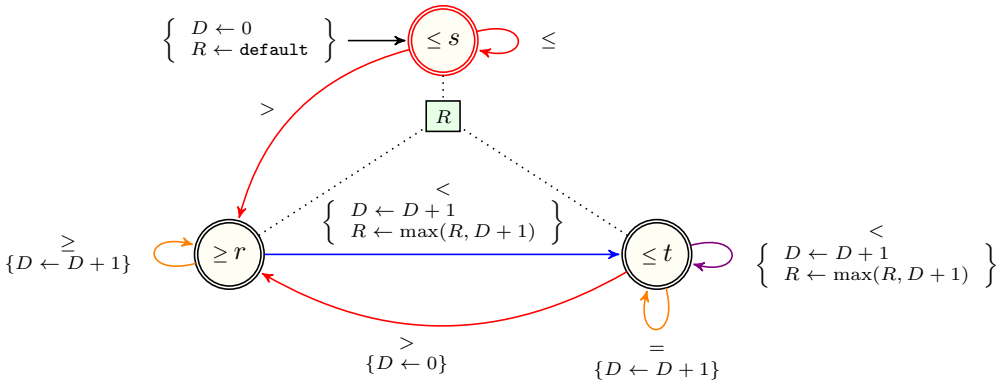


Figure 4.725: Simplified automaton for the MAX_WIDTH_VALLEY constraint obtained by applying decoration Table 3.27 to the seed transducer of the VALLEY pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_i \geq 0$ are linear invariants.

	s	r	t
s	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C})$
r	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\max(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\max(\overrightarrow{C}, \overleftarrow{C})$

Table 4.113: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MAX_WIDTH_VALLEY constraint defined as the composition of the VALLEY pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

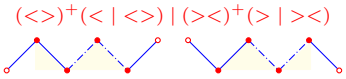
	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0

Table 4.114: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the MAX_WIDTH_VALLEY constraint defined as the composition of the VALLEY pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MAX_WIDTH_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

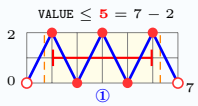
Constraint MAX_WIDTH_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the maximal width of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(6, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.726 provides an example where the MAX_WIDTH_ZIGZAG $(6, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

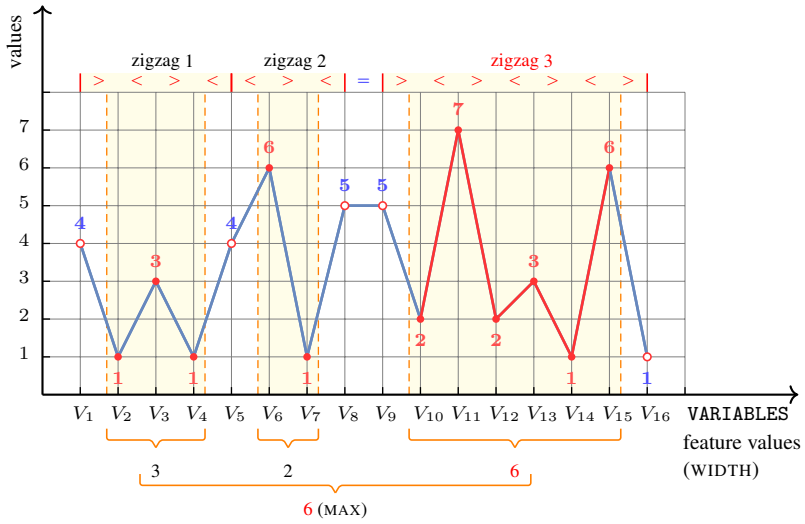


Figure 4.726: Illustrating the MAX_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.727 and 4.728 respectively depict the automaton associated with the constraint MAX_WIDTH_ZIGZAG and its simplified form.

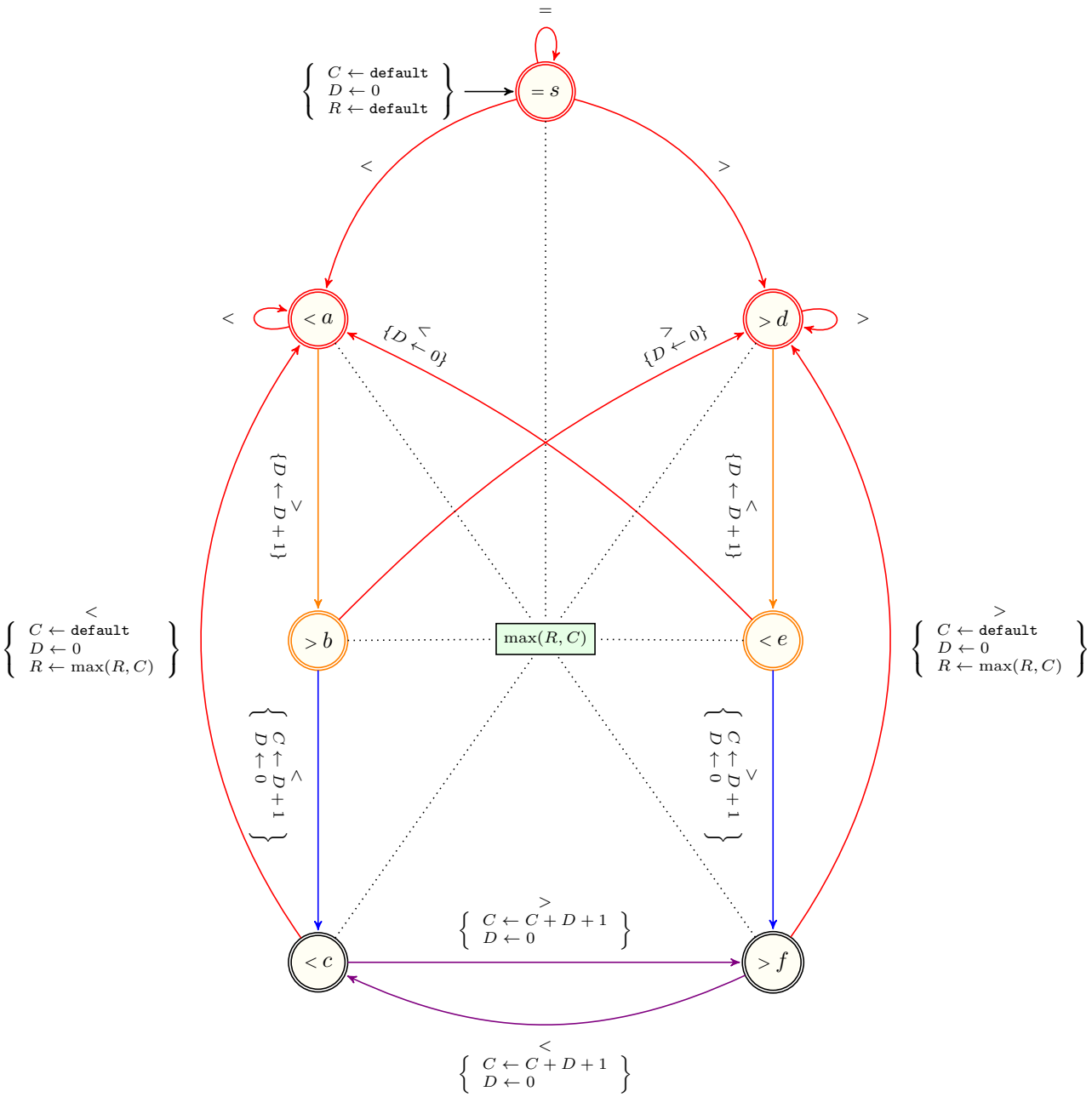


Figure 4.727: Automaton for the MAX_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

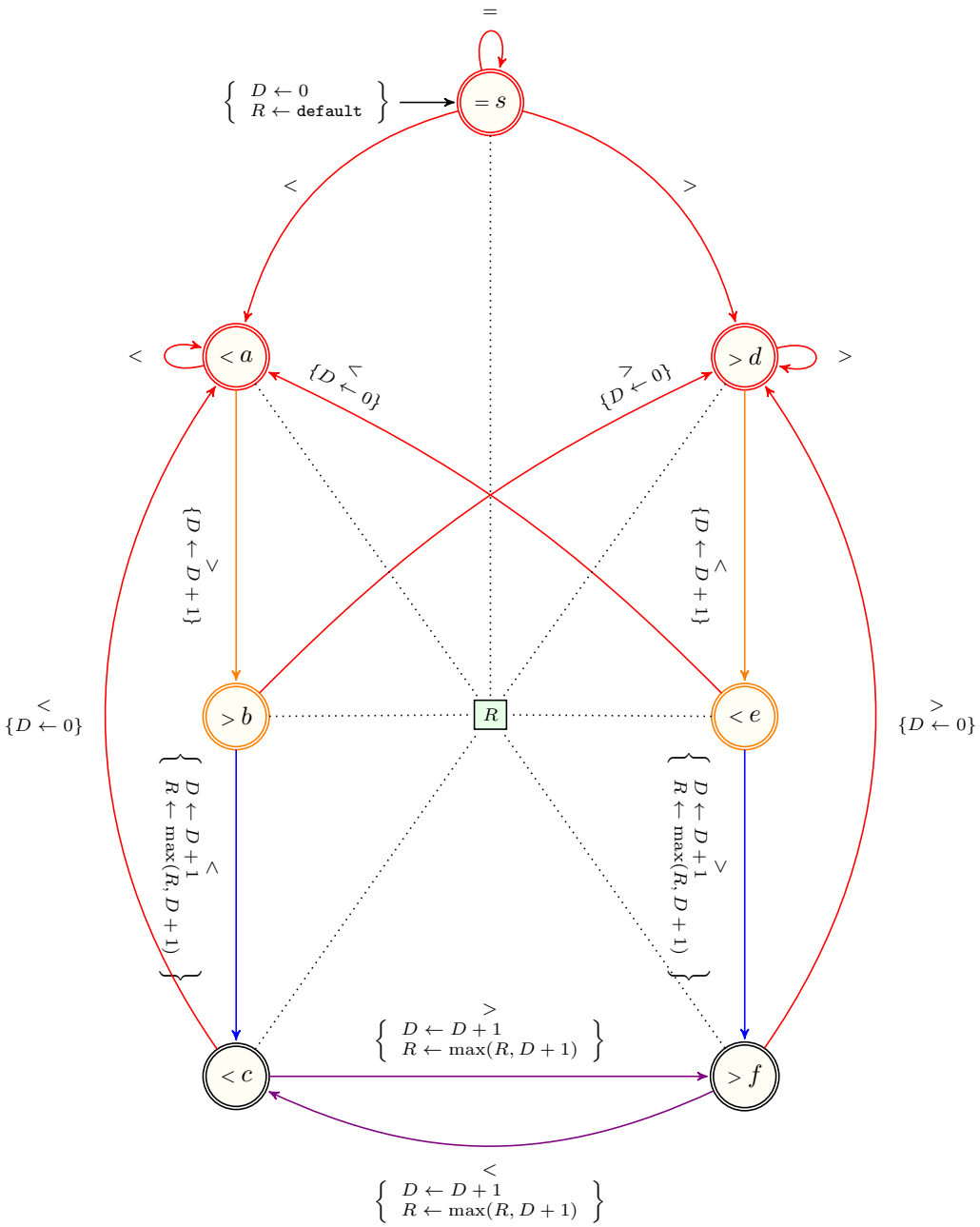


Figure 4.728: Simplified automaton for the MAX_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.27 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

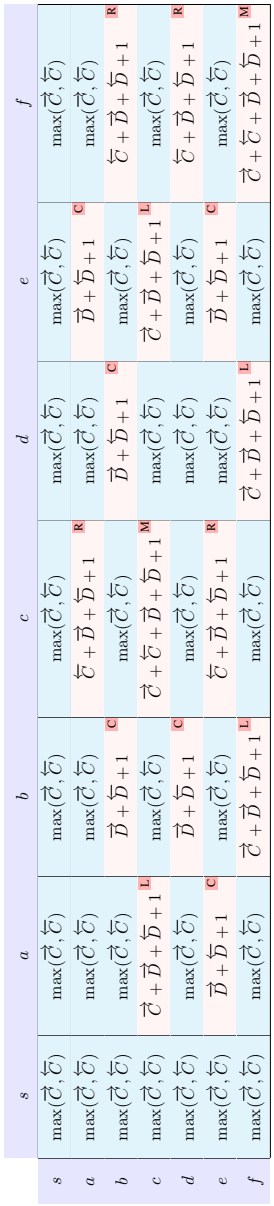


Table 4.115: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MAX_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

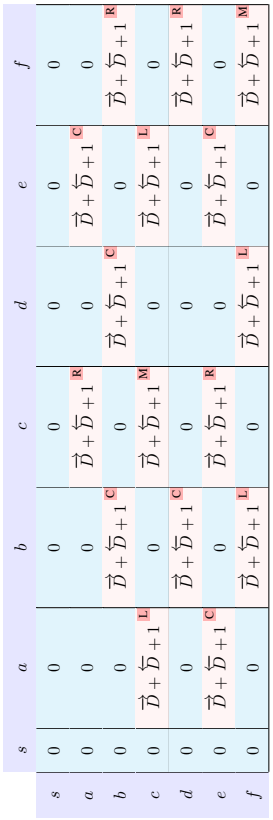
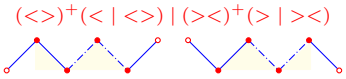


Table 4.116: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MAX_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator max; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MAX_ZIGZAG

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint MAX_ZIGZAG(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv + 1
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv + 1 ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of ZIGZAG is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of ZIGZAG.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example Figure 4.729 provides an example where the MAX_ZIGZAG $([4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1],[0,0,3,0,0,0,6,0,0,0,7,0,0,0,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

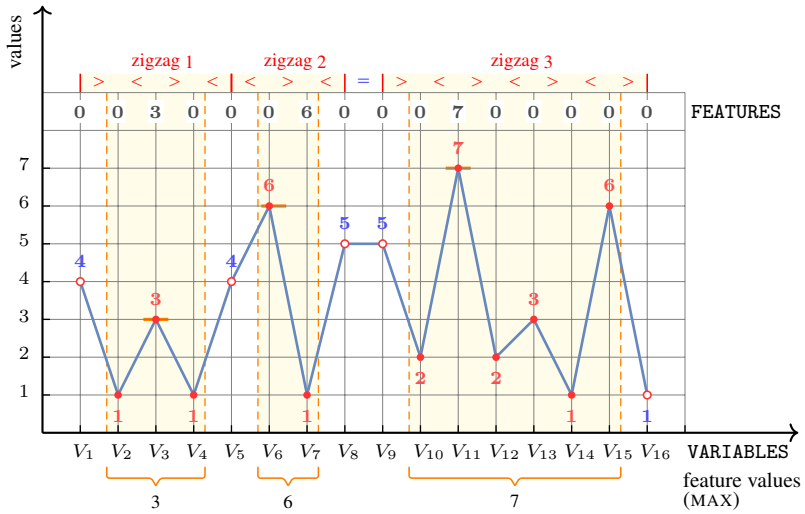
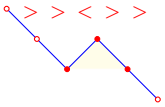


Figure 4.729: Illustrating the MAX_ZIGZAG constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint MIN_BUMP_ON_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 5 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 2
DEFAULT < minv ∨ DEFAULT > maxv − 2
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of BUMP_ON_DECREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of BUMP_ON_DECREASING_SEQUENCE.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example Figure 4.730 provides an example where the MIN_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

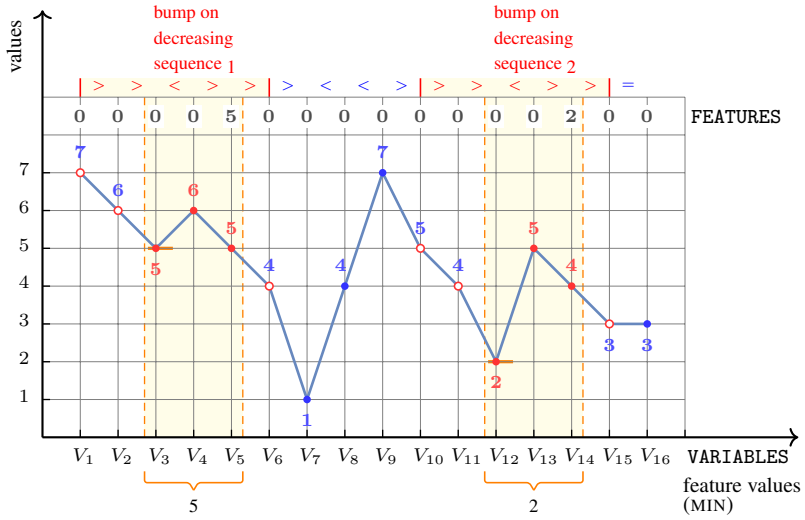


Figure 4.730: Illustrating the MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint MIN_DECREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.731 provides an example where the MIN_DECREASING $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 2, 0, 0, 0, 0, 4, 0, 3, 1, 0, 0, 0, 4, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

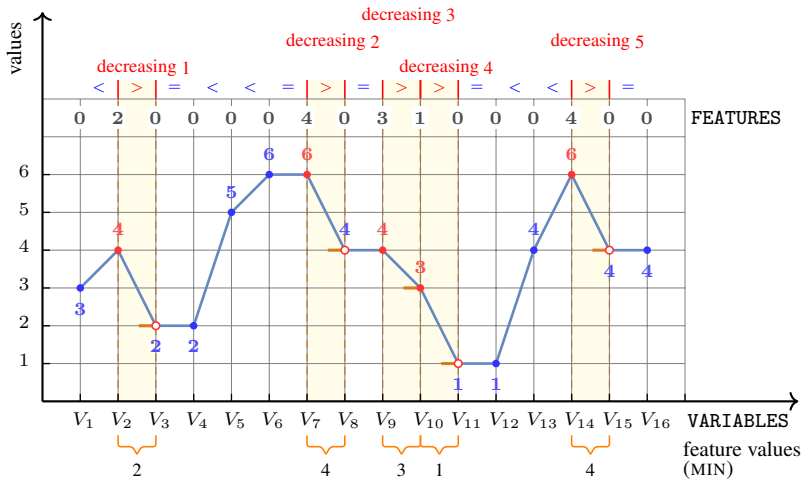
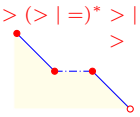


Figure 4.731: Illustrating the MIN_DECREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint MIN_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_SEQUENCE.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.732 provides an example where the MIN_DECREASING_SEQUENCE $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 4, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

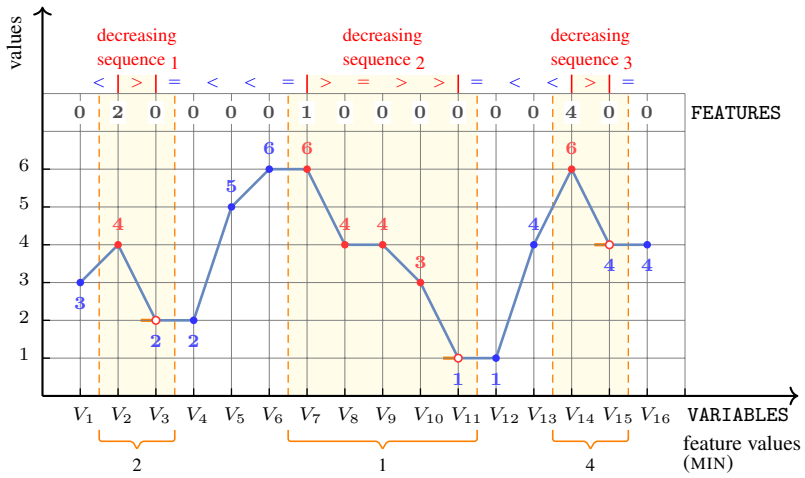
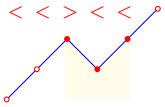


Figure 4.732: Illustrating the MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint MIN_DIP_ON_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 5 ∨ rv ≤ 2 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 2
DEFAULT < minv ∨ DEFAULT > maxv − 2
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DIP_ON_INCREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DIP_ON_INCREASING_SEQUENCE.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example Figure 4.733 provides an example where the MIN_DIP_ON_INCREASING_SEQUENCE $([1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

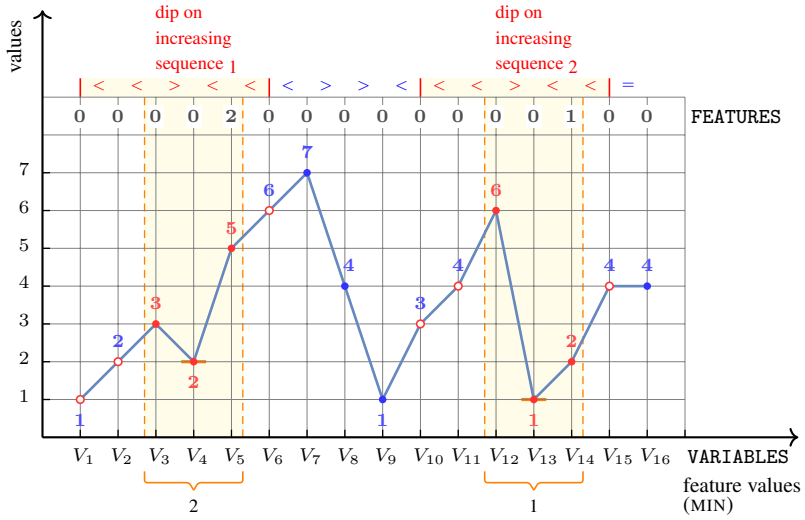


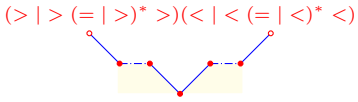
Figure 4.733: Illustrating the MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the GORGE pattern.

Constraint MIN_GORGE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of GORGE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of GORGE.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.734 provides an example where the MIN_GORGE $([1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7],[0,0,3,0,0,0,0,0,0,0,0,0,4,0,5,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

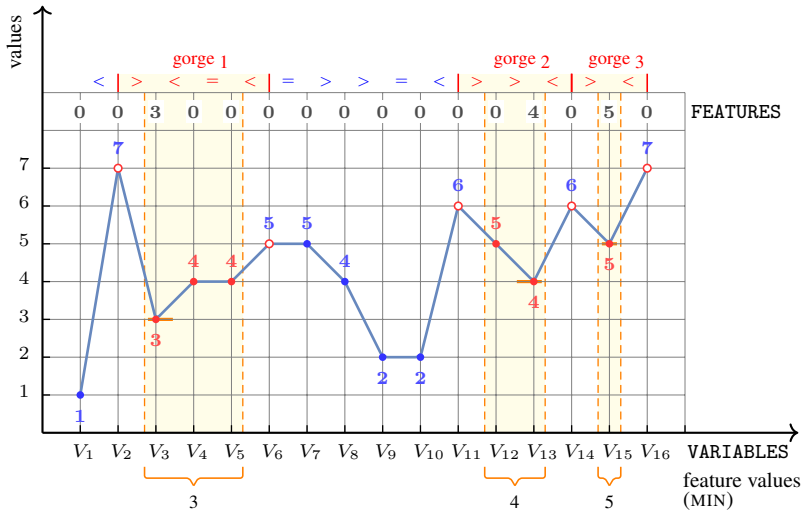


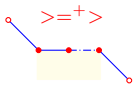
Figure 4.734: Illustrating the MIN_GORGE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

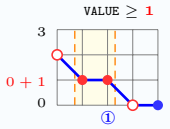
Constraint MIN_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(2, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3 \rangle)$

Figure 4.735 provides an example where the MIN_HEIGHT_DECREASING_TERRACE $(2, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Arg. properties Functional dependency: VALUE determined by VARIABLES.

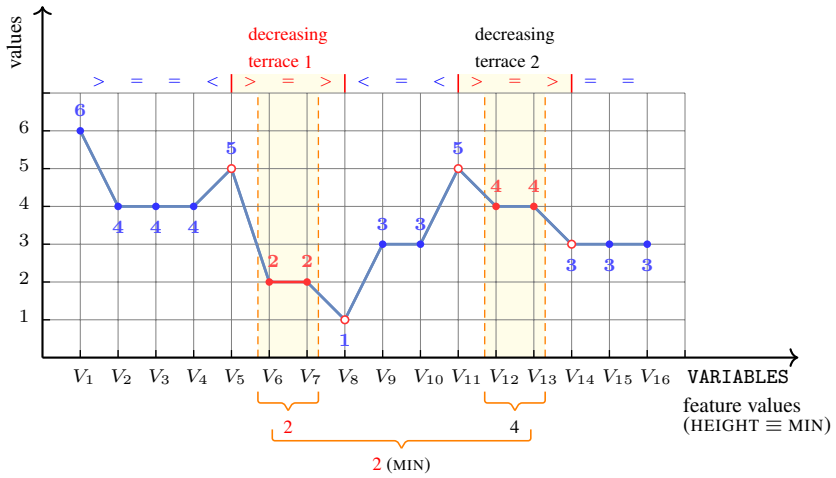


Figure 4.735: Illustrating the MIN_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.736 and 4.737 respectively depict the automaton associated with the constraint MIN_HEIGHT_DECREASING_TERRACE and its simplified form.

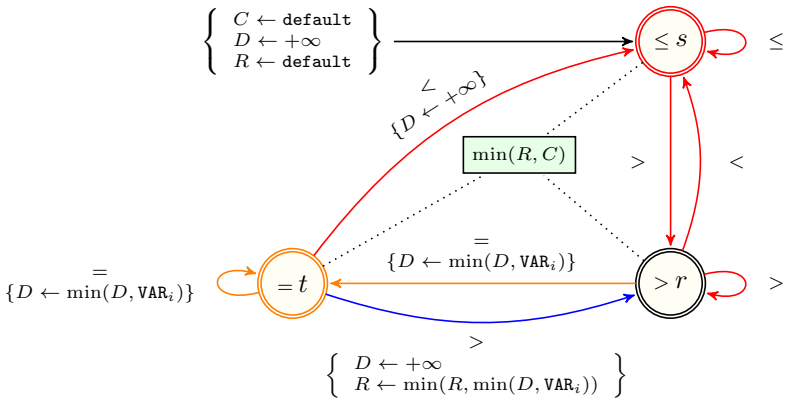


Figure 4.736: Automaton for the MIN_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where default is $+\infty$

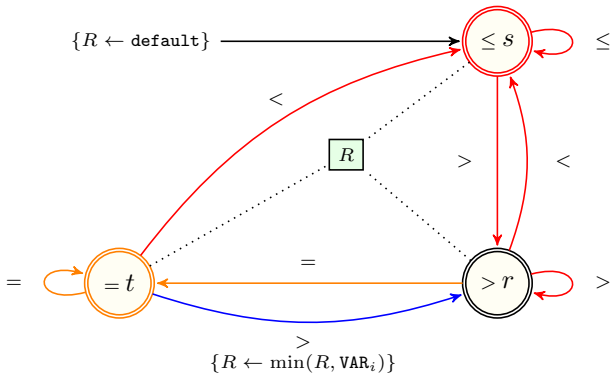


Figure 4.737: Simplified automaton for the MIN_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_TERRACE pattern where default is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

Table 4.117: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MIN_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

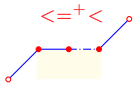
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\mathtt{VAR}_{i+1}$ C
t	$+\infty$	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.118: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MIN_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

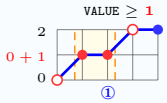
Constraint MIN_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.738 provides an example where the MIN_HEIGHT_INCREASING_TERRACE $(3, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

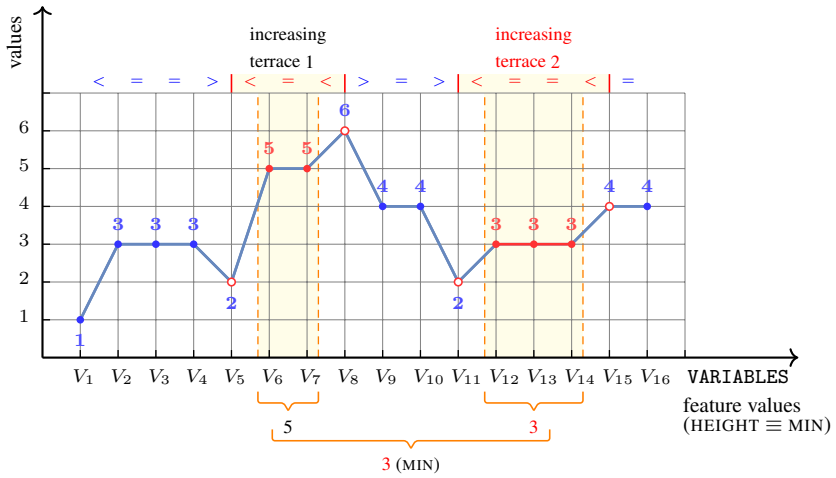


Figure 4.738: Illustrating the MIN_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.739 and 4.740 respectively depict the automaton associated with the constraint MIN_HEIGHT_INCREASING_TERRACE and its simplified form.

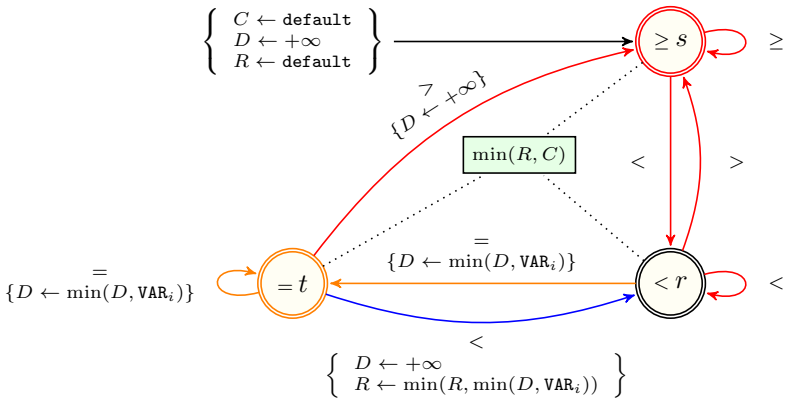


Figure 4.739: Automaton for the MIN_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where default is $+\infty$

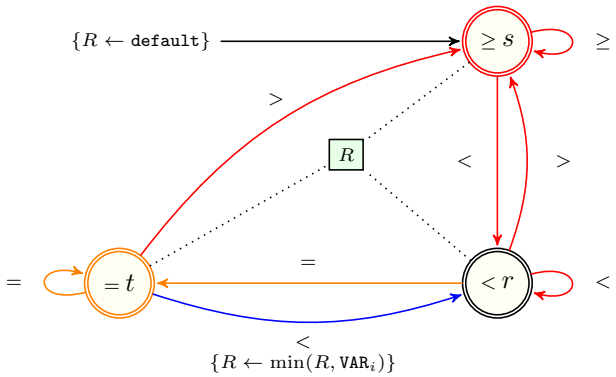


Figure 4.740: Simplified automaton for the MIN_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_TERRACE pattern where default is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

Table 4.119: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MIN_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern , the feature MIN , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\mathtt{VAR}_{i+1}$ C
t	$+\infty$	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.120: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MIN_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern , the feature MIN , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint MIN_HEIGHT_PLAIN(VALUE, VARIABLES)

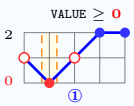
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)

where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3 \rangle)$

Figure 4.741 provides an example where the MIN_HEIGHT_PLAIN $(3, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

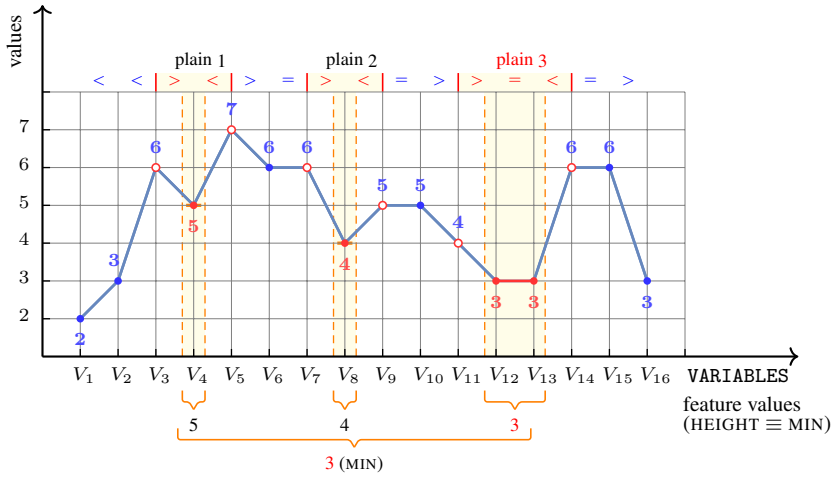


Figure 4.741: Illustrating the MIN_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figures 4.742 and 4.743 respectively depict the automaton associated with the constraint MIN_HEIGHT_PLAIN and its simplified form.

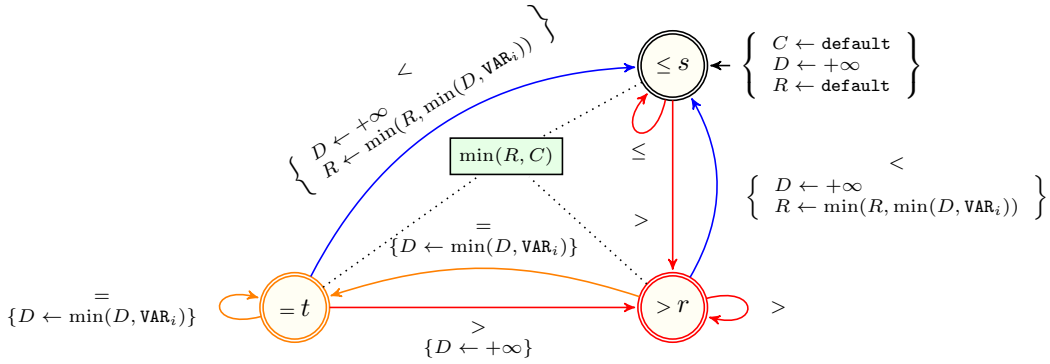


Figure 4.742: Automaton for the MIN_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is $+\infty$

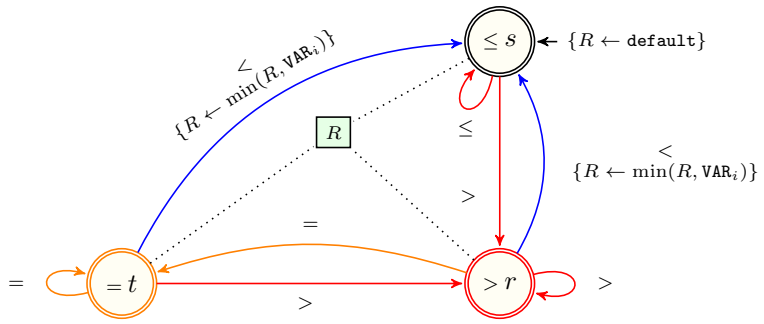


Figure 4.743: Simplified automaton for the MIN_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLAIN pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.121: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MIN_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C
t	$+\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.122: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MIN_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

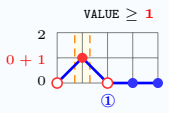
Constraint MIN_HEIGHT_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle)$

Figure 4.744 provides an example where the MIN_HEIGHT_PLATEAU $(3, [7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

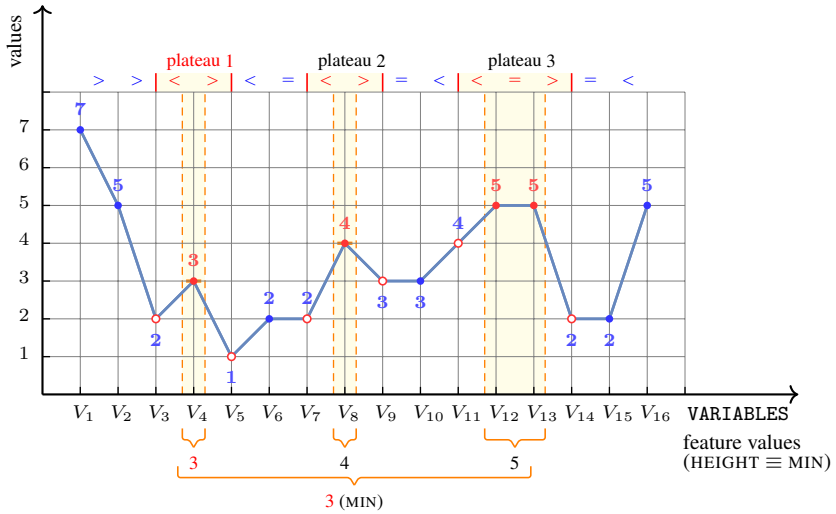


Figure 4.744: Illustrating the MIN_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figures 4.745 and 4.746 respectively depict the automaton associated with the constraint MIN_HEIGHT_PLATEAU and its simplified form.

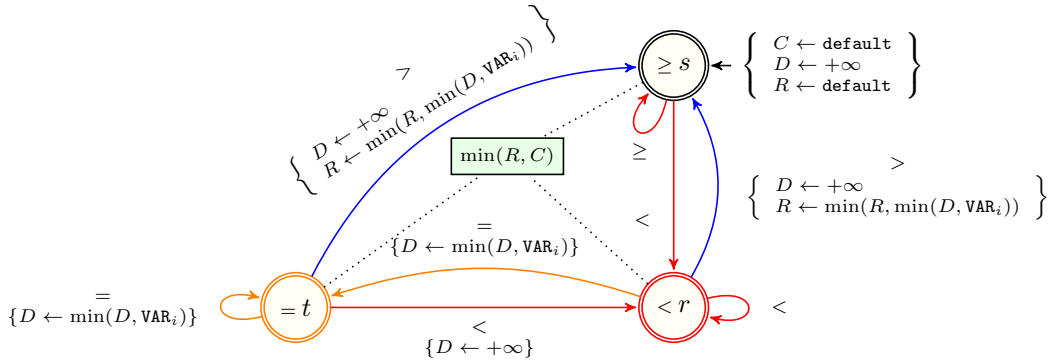


Figure 4.745: Automaton for the MIN_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is $+\infty$

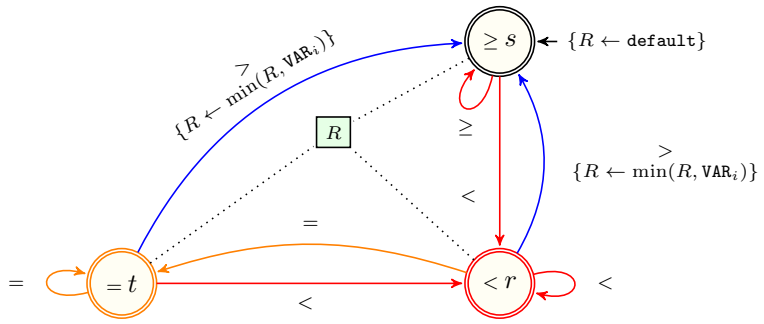


Figure 4.746: Simplified automaton for the MIN_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLATEAU pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.123: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MIN_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

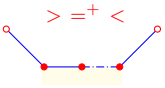
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C
t	$+\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.124: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MIN_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

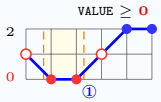
Constraint MIN_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
$\texttt{minv} = \text{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \text{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \text{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the minimum of all minimum values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5 \rangle)$

Figure 4.747 provides an example where the MIN_HEIGHT_PROPER_PLAIN $(3, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\text{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

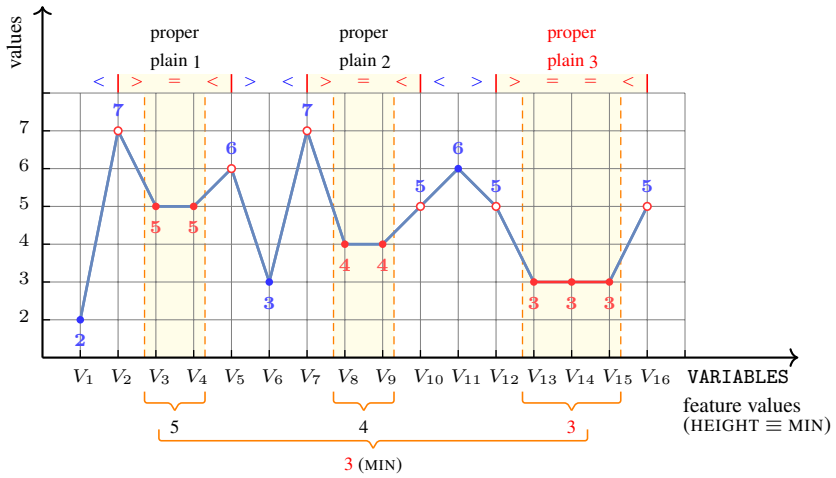


Figure 4.747: Illustrating the MIN_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.748 and 4.749 respectively depict the automaton associated with the constraint MIN_HEIGHT_PROPER_PLAIN and its simplified form.

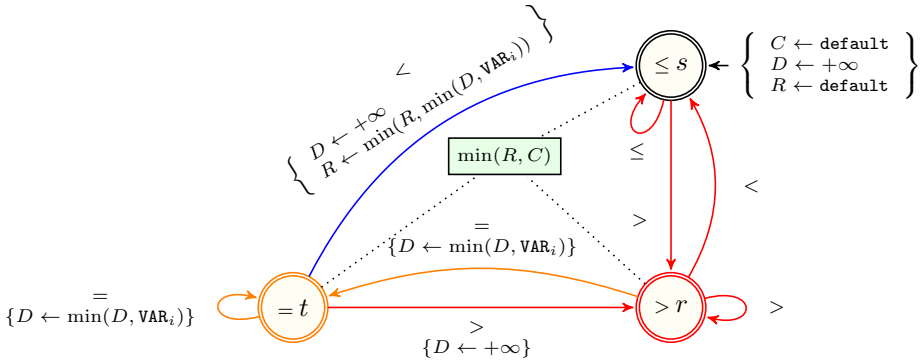


Figure 4.748: Automaton for the MIN_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is $+\infty$

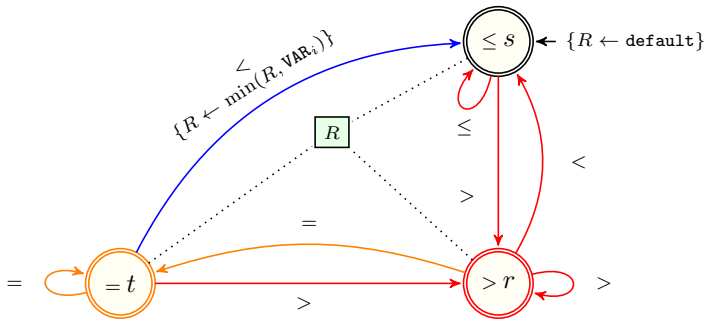


Figure 4.749: Simplified automaton for the MIN_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLAIN pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.125: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MIN_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

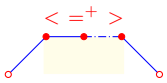
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\texttt{VAR}_{i+1}$ C
t	$+\infty$	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.126: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MIN_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

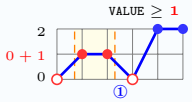
Constraint MIN_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3 \rangle)$

Figure 4.750 provides an example where the MIN_HEIGHT_PROPER_PLATEAU $(3, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

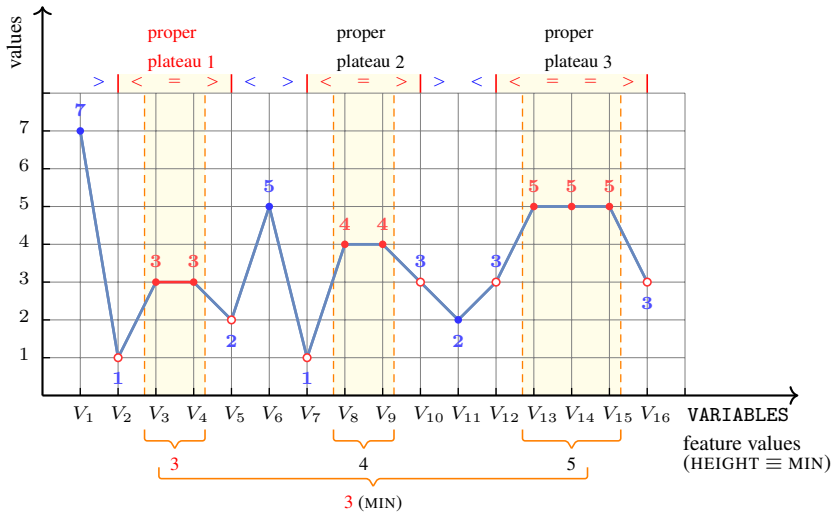


Figure 4.750: Illustrating the MIN_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.751 and 4.752 respectively depict the automaton associated with the constraint MIN_HEIGHT_PROPER_PLATEAU and its simplified form.

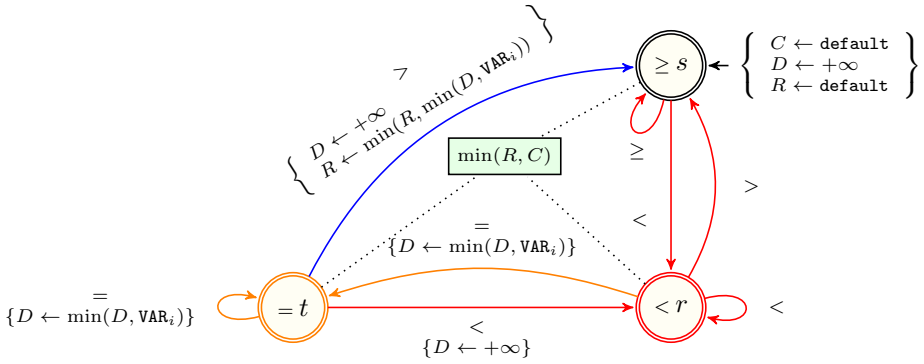


Figure 4.751: Automaton for the MIN_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where default is $+\infty$

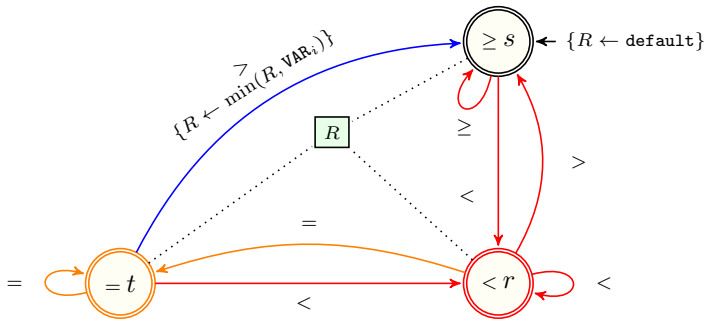


Figure 4.752: Simplified automaton for the MIN_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLATEAU pattern where default is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

Table 4.127: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MIN_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\mathtt{VAR}_{i+1}$ C
t	$+\infty$	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.128: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MIN_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

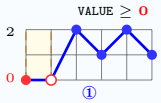
Constraint MIN_HEIGHT_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the minimum of all minimum values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6\rangle)$

Figure 4.753 provides an example where the MIN_HEIGHT_STEADY $(1, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

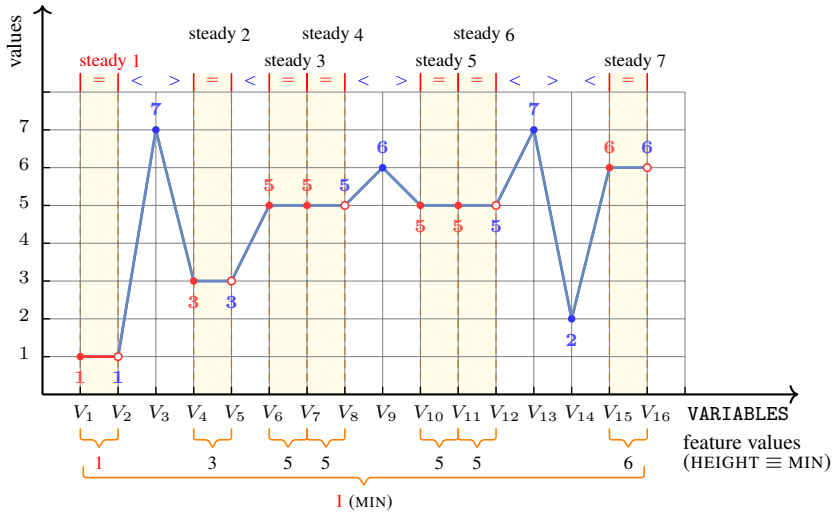


Figure 4.753: Illustrating the MIN_HEIGHT_STEADY constraint of the **Example** slot

Automaton

Figures 4.754 and 4.755 respectively depict the automaton associated with the constraint MIN_HEIGHT_STEADY and its simplified form.

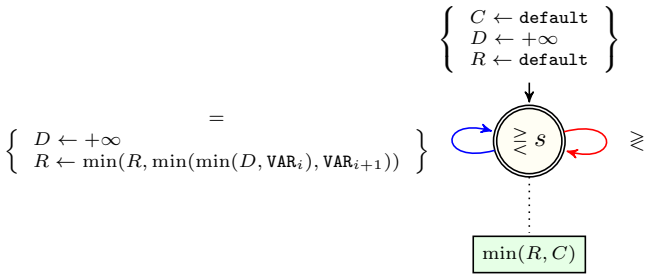


Figure 4.754: Automaton for the MIN_HEIGHT_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is $+\infty$

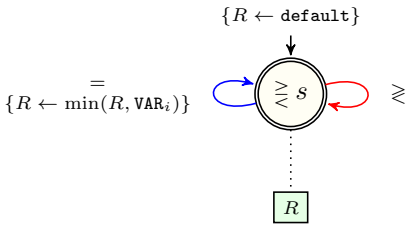


Figure 4.755: Simplified automaton for the MIN_HEIGHT_STEADY constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.129: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the MIN_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

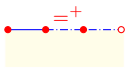
	s
s	$+\infty$

Table 4.130: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the MIN_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_HEIGHT_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

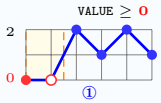
Constraint MIN_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

VALUE is the minimum of all minimum values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $(1, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle)$

Figure 4.756 provides an example where the MIN_HEIGHT_STEADY_SEQUENCE $(1, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

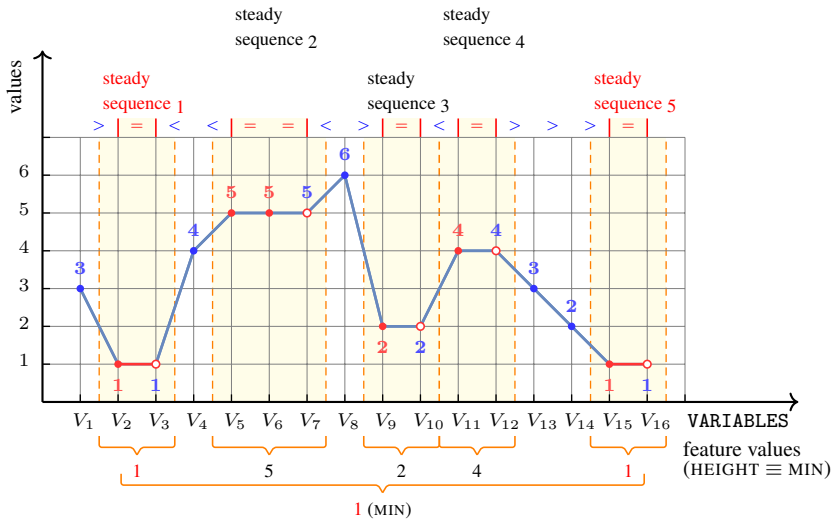


Figure 4.756: Illustrating the MIN_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.757 and 4.758 respectively depict the automaton associated with the constraint MIN_HEIGHT_STEADY_SEQUENCE and its simplified form.

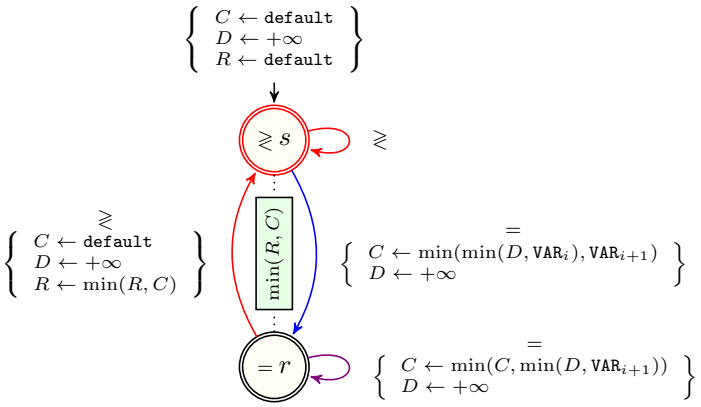


Figure 4.757: Automaton for the MIN_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $+\infty$

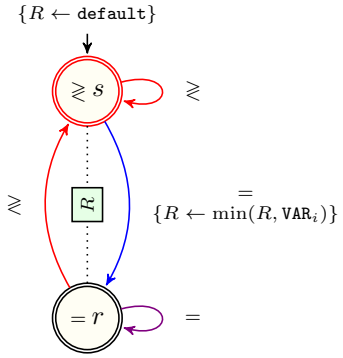


Figure 4.758: Simplified automaton for the MIN_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.131: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MIN_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$ M

Table 4.132: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MIN_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint MIN_INCREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.759 provides an example where the MIN_INCREASING $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 3, 0, 0, 0, 0, 1, 0, 3, 4, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

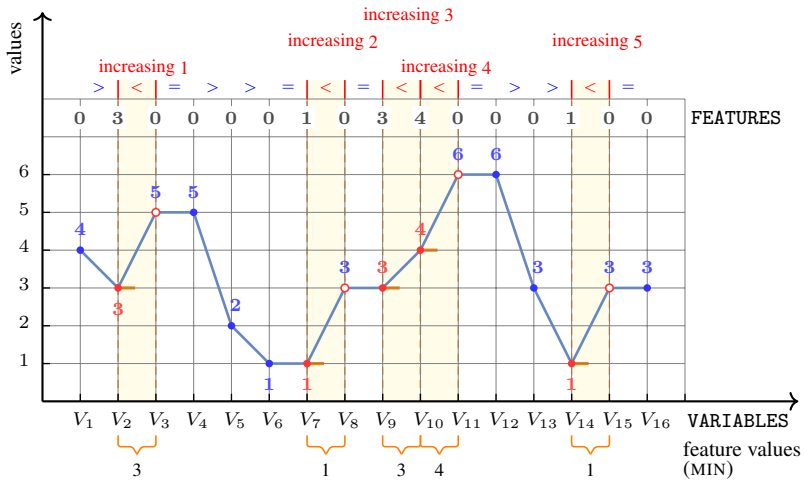
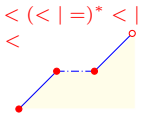


Figure 4.759: Illustrating the MIN_INCREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint MIN_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_SEQUENCE.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.760 provides an example where the MIN_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 3, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

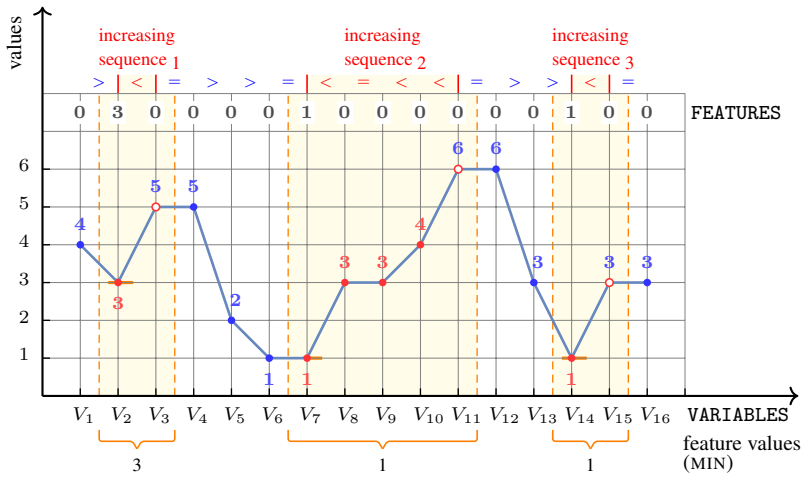
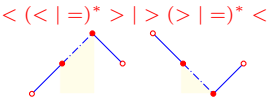


Figure 4.760: Illustrating the MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_INFLEXION



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INFLEXION pattern.

Constraint MIN_INFLEXION(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv
DEFAULT < minv ∨ DEFAULT > maxv
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INFLEXION is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INFLEXION.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.761 provides an example where the MIN_INFLEXION $([1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4],[0,0,0,2,0,0,3,5,2,5,1,5,0,3,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

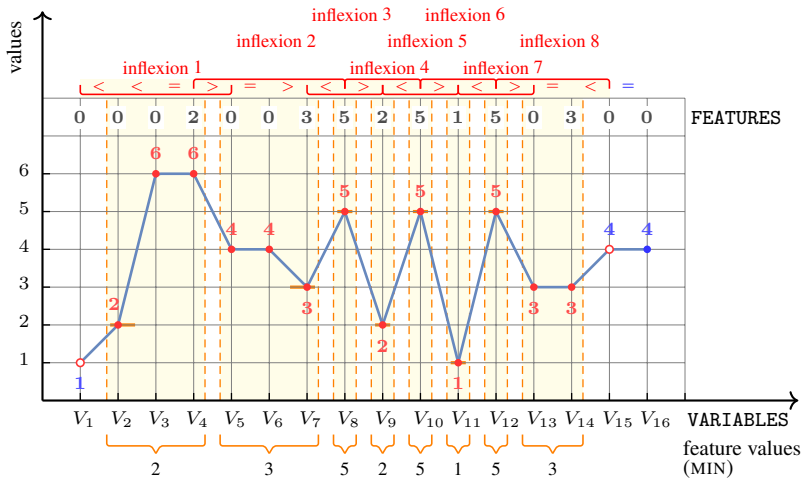
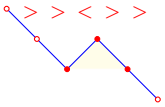


Figure 4.761: Illustrating the MIN_INFLEXION constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MIN_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

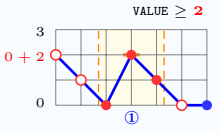
Constraint MIN_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 2$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(5, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.762 provides an example where the MIN_MAX_BUMP_ON_DECREASING_SEQUENCE $(5, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

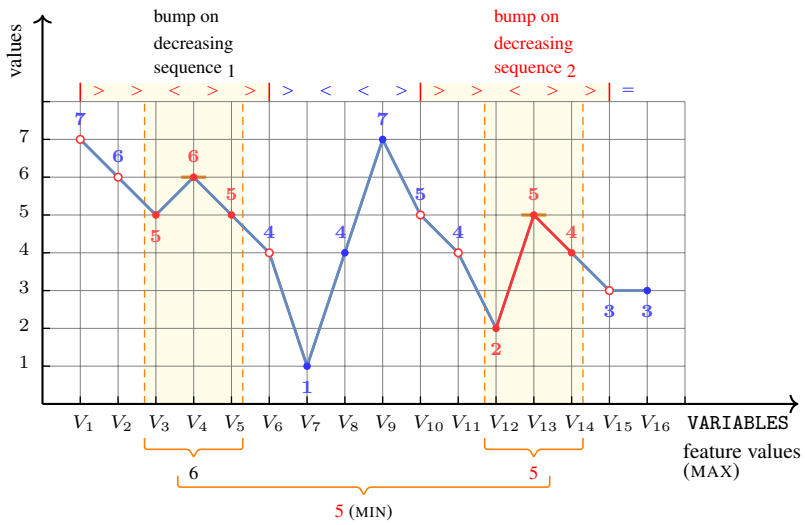


Figure 4.762: Illustrating the MIN_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.763 and 4.764 respectively depict the automaton associated with the constraint MIN_MAX_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

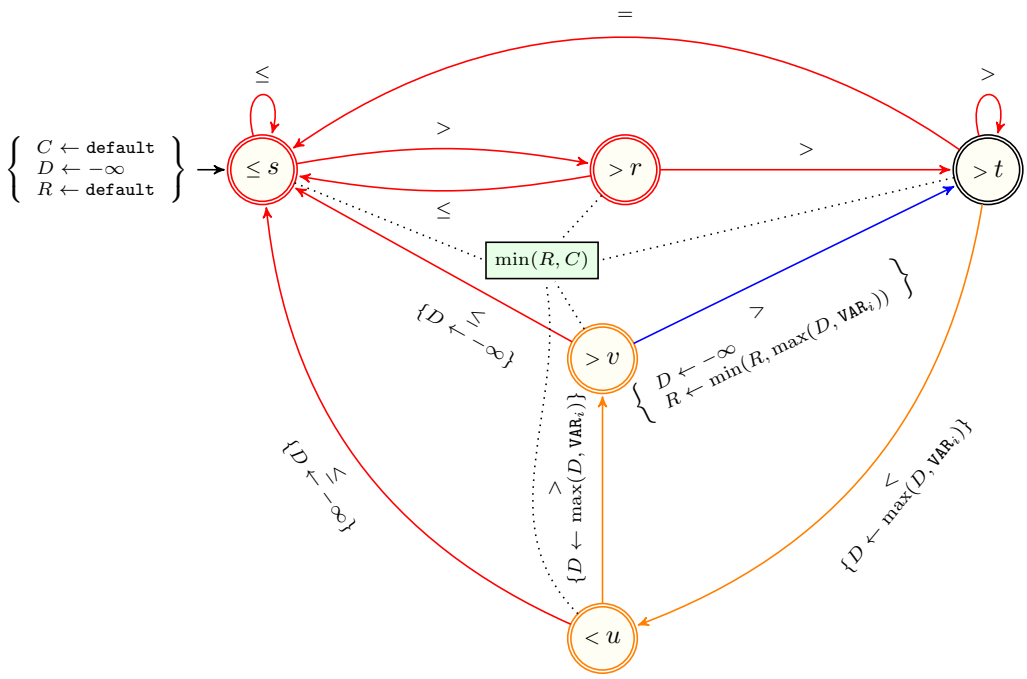


Figure 4.763: Automaton for the MIN_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$

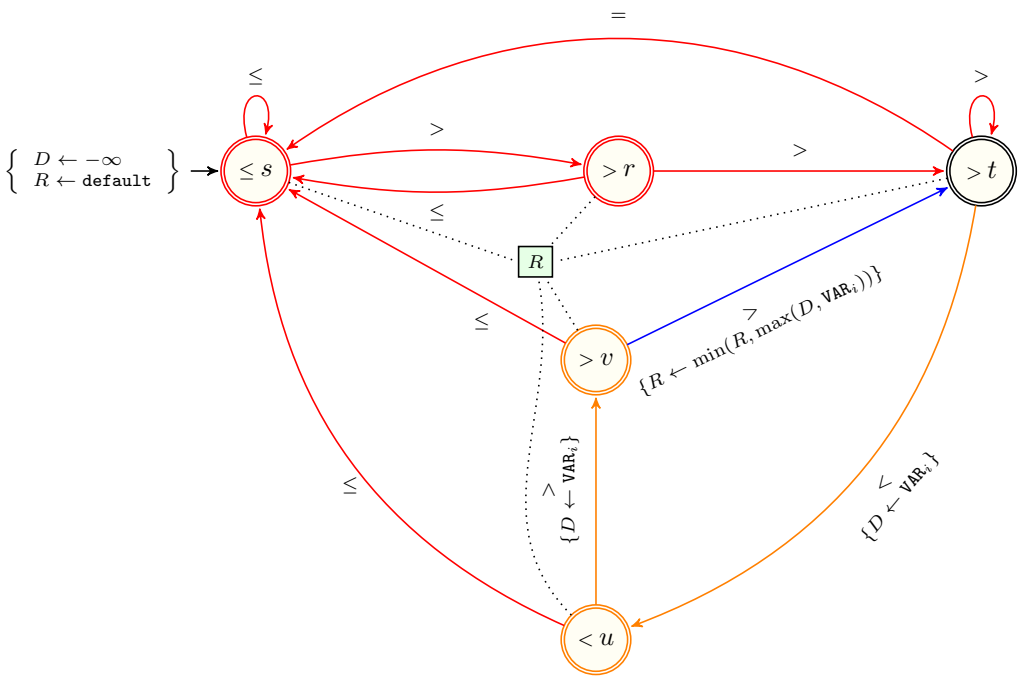


Figure 4.764: Simplified automaton for the MIN_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MAX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

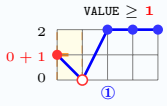
Constraint MIN_MAX_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(3, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.765 provides an example where the MIN_MAX_DECREASING $(3, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

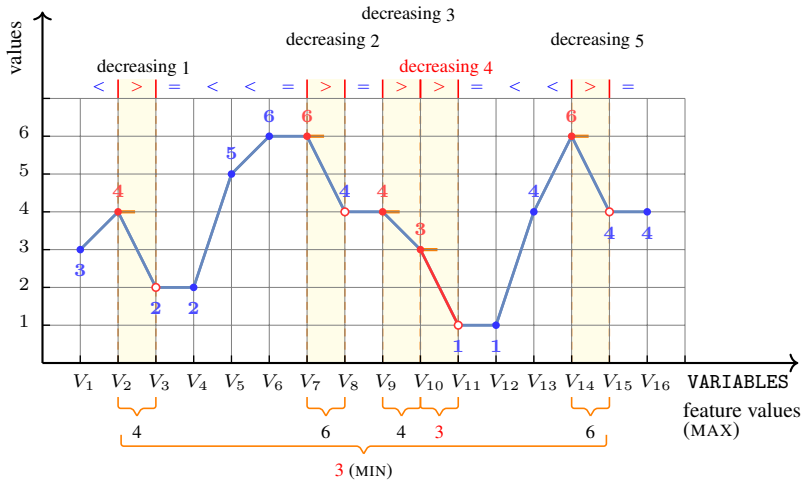


Figure 4.765: Illustrating the MIN_MAX_DECREASING constraint of the **Example** slot

Automaton

Figures 4.766 and 4.767 respectively depict the automaton associated with the constraint MIN_MAX_DECREASING and its simplified form.

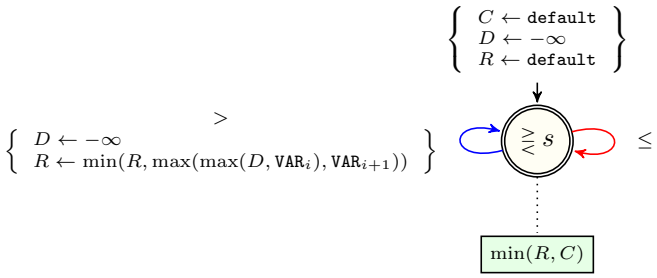


Figure 4.766: Automaton for the MIN_MAX_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $+\infty$

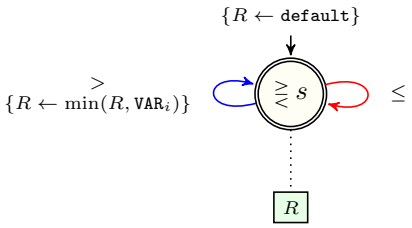


Figure 4.767: Simplified automaton for the MIN_MAX_DECREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

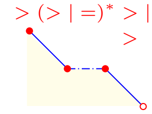
Table 4.133: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MIN_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.134: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MIN_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

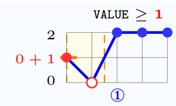
Origin Based on the DECREASING_SEQUENCE pattern.

Constraint MIN_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE : dvar
VARIABLES : collection(var−dvar)

Restrictions



$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 1$ ①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
where
minv = minval(VARIABLES.var)
maxv = maxval(VARIABLES.var)
sv = |VARIABLES|
rv = range(VARIABLES.var)

Purpose

VALUE is the minimum of all maximum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$(4, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.768 provides an example where the MIN_MAX_DECREASING_SEQUENCE (4, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4]) constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

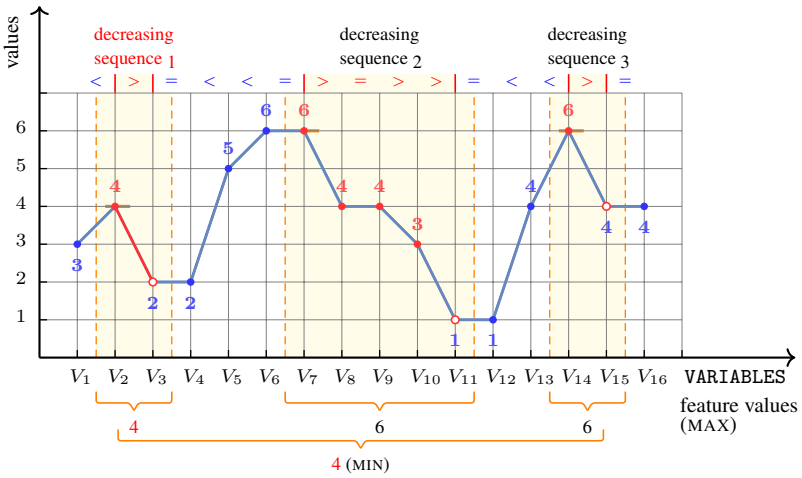


Figure 4.768: Illustrating the MIN_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.769 and 4.770 respectively depict the automaton associated with the constraint MIN_MAX_DECREASING_SEQUENCE and its simplified form.

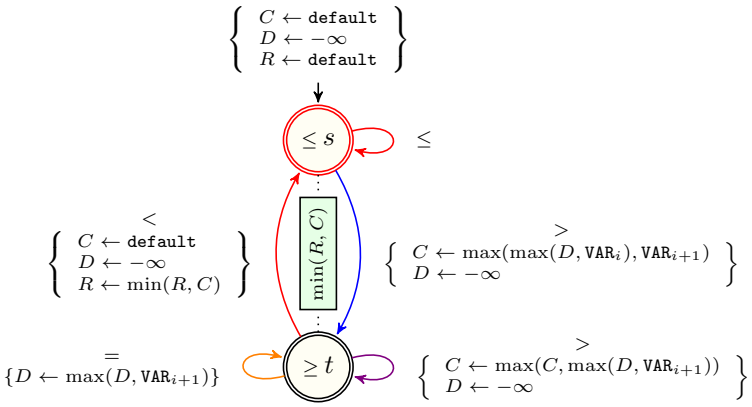


Figure 4.769: Automaton for the MIN_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$

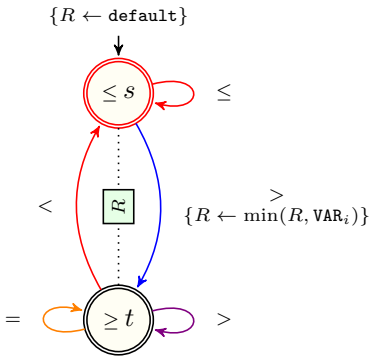


Figure 4.770: Simplified automaton for the MIN_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

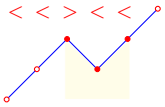
Table 4.135: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MIN_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	$+\infty$	$\overleftarrow{C}$
t	$+\infty$	$+\infty$ M

Table 4.136: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the MIN_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

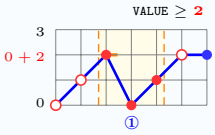
Constraint MIN_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 2$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(5, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.771 provides an example where the MIN_MAX_DIP_ON_INCREASING_SEQUENCE $(5, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

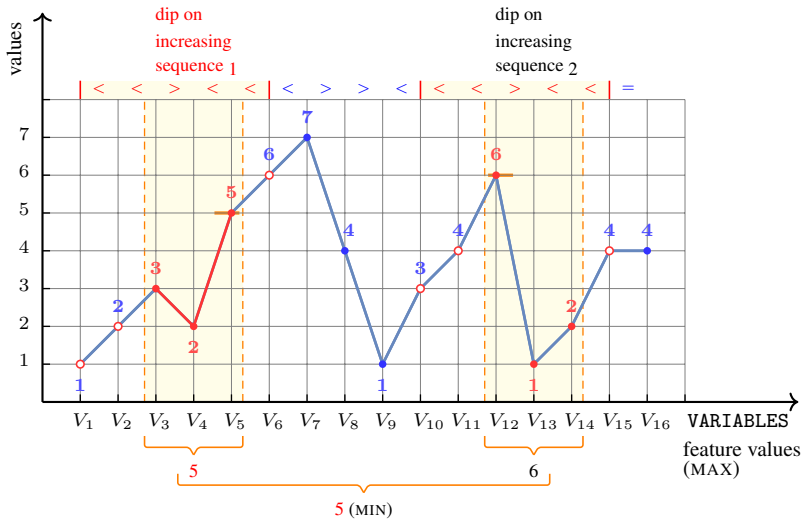


Figure 4.771: Illustrating the MIN_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.772 and 4.773 respectively depict the automaton associated with the constraint MIN_MAX_DIP_ON_INCREASING_SEQUENCE and its simplified form.

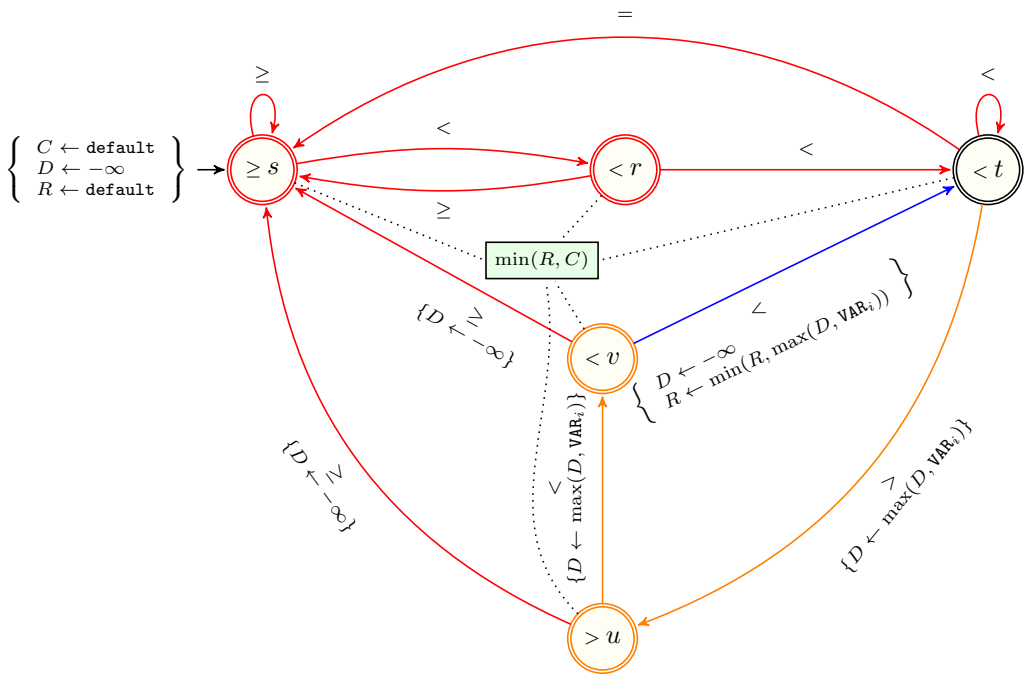


Figure 4.772: Automaton for the MIN_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$

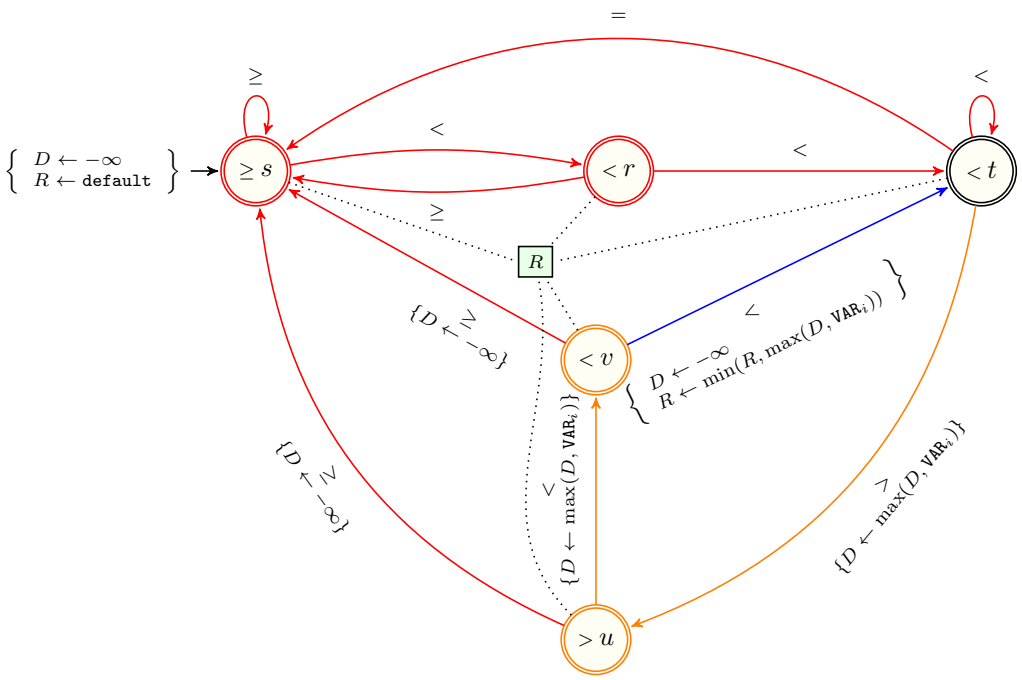


Figure 4.773: Simplified automaton for the MIN_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint MIN_MAX_INCREASING(VALUE, VARIABLES)

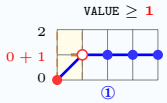
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)

where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all maximum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.774 provides an example where the MIN_MAX_INCREASING $(3, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

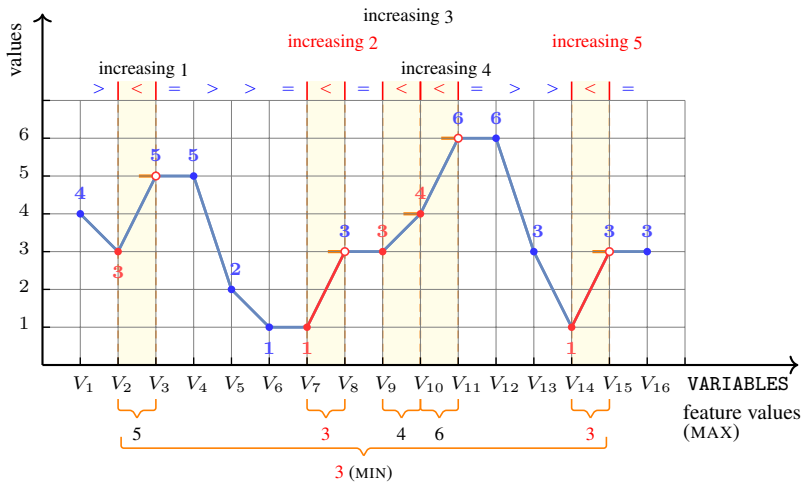


Figure 4.774: Illustrating the MIN_MAX_INCREASING constraint of the **Example** slot

Automaton

Figures 4.775 and 4.776 respectively depict the automaton associated with the constraint MIN_MAX_INCREASING and its simplified form.

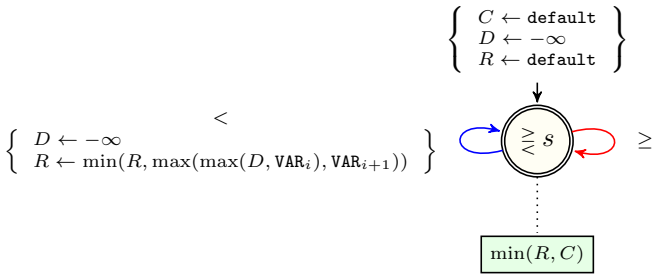


Figure 4.775: Automaton for the MIN_MAX_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $+\infty$

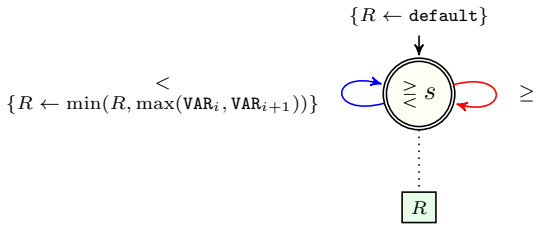


Figure 4.776: Simplified automaton for the MIN_MAX_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

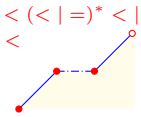
Table 4.137: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MIN_MAX_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.138: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MIN_MAX_INCREASING constraint defined as the composition of the INCREASING pattern , the feature MAX , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

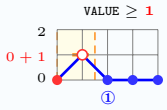
Constraint MIN_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.777 provides an example where the MIN_MAX_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

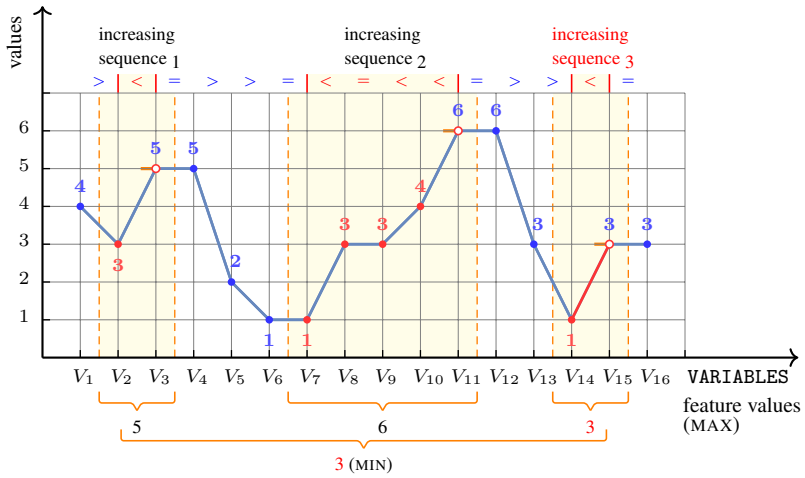


Figure 4.777: Illustrating the MIN_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.778 and 4.779 respectively depict the automaton associated with the constraint MIN_MAX_INCREASING_SEQUENCE and its simplified form.

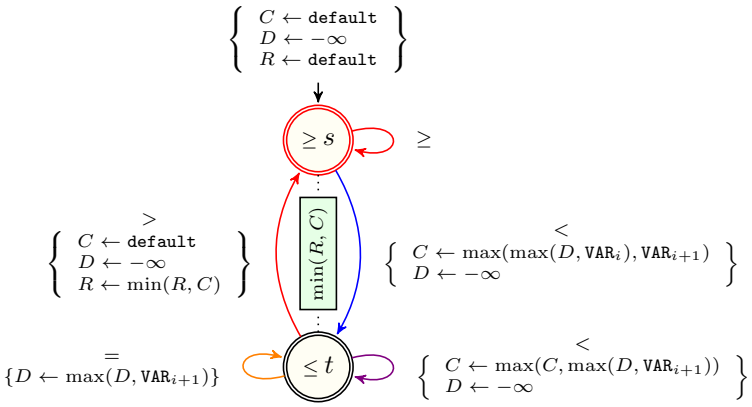


Figure 4.778: Automaton for the MIN_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$

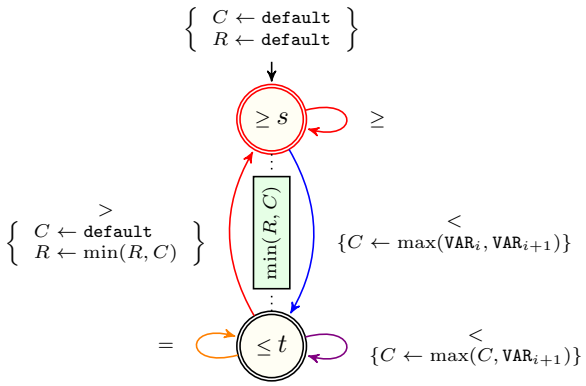


Figure 4.779: Simplified automaton for the MIN_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.139: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MIN_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

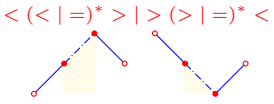
	s	t
s	$+\infty$	$+\infty$
t	$\overrightarrow{C}$	$+\infty$ M

Table 4.140: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the MIN_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

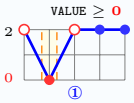
Constraint MIN_MAX_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimum of all maximum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(1, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.780 provides an example where the MIN_MAX_INFLEXION $(1, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

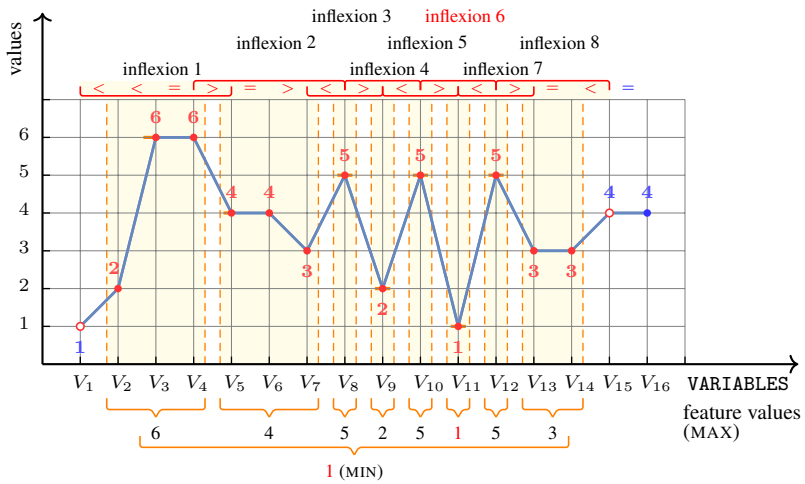


Figure 4.780: Illustrating the MIN_MAX_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.781 and 4.782 respectively depict the automaton associated with the constraint MIN_MAX_INFLEXION and its simplified form.

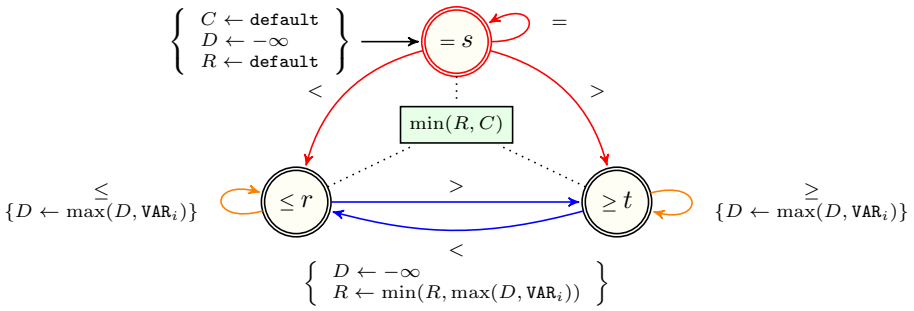


Figure 4.781: Automaton for the MIN_MAX_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

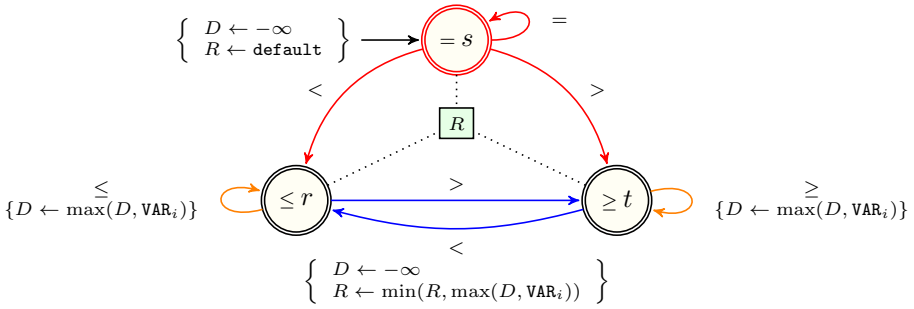
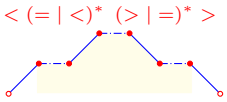


Figure 4.782: Simplified automaton for the MIN_MAX_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MAX_PEAK



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PEAK pattern.

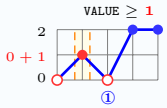
Constraint MIN_MAX_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimum of all maximum values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(3, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.783 provides an example where the MIN_MAX_PEAK $(3, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

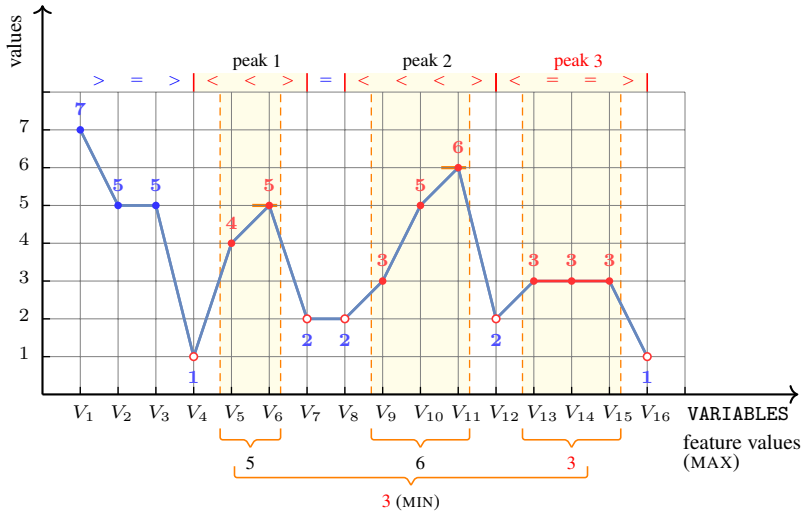


Figure 4.783: Illustrating the MIN_MAX_PEAK constraint of the **Example** slot

Automaton

Figures 4.784 and 4.785 respectively depict the automaton associated with the constraint MIN_MAX_PEAK and its simplified form.

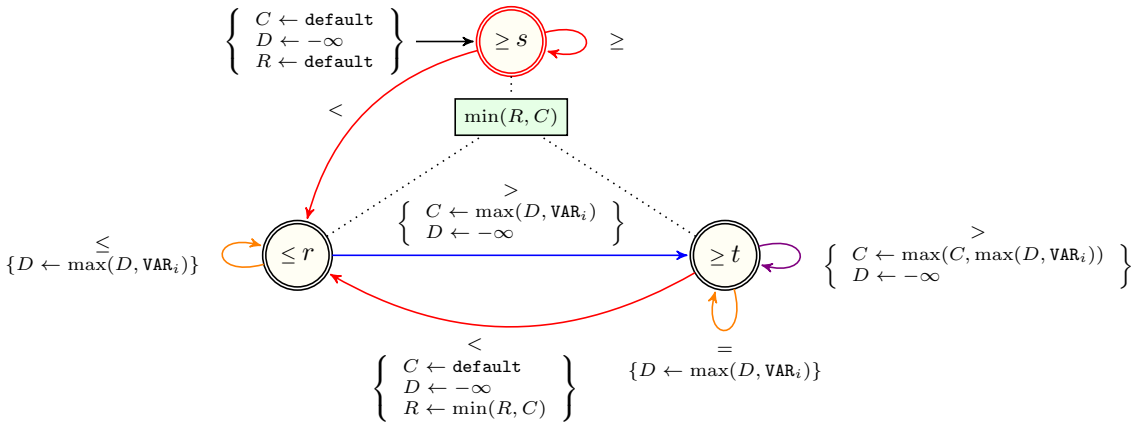


Figure 4.784: Automaton for the MIN_MAX_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is $+\infty$

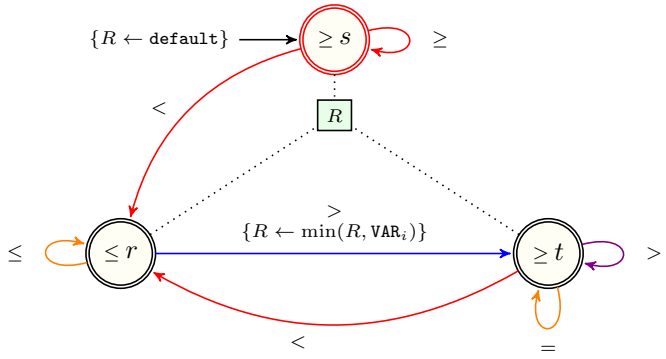


Figure 4.785: Simplified automaton for the MIN_MAX_PEAK constraint obtained by applying decoration Table 3.39 to the seed transducer of the PEAK pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.141: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MIN_MAX_PEAK constraint defined as the composition of the PEAK pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\texttt{VAR}_{i+1}$ C	$+\infty$ R
t	$+\infty$	$+\infty$ L	$+\infty$

Table 4.142: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the MIN_MAX_PEAK constraint defined as the composition of the PEAK pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

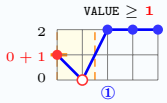
Constraint MIN_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(4, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3\rangle)$

Figure 4.786 provides an example where the MIN_MAX_STRICTLY_DECREASING_SEQUENCE $(4, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

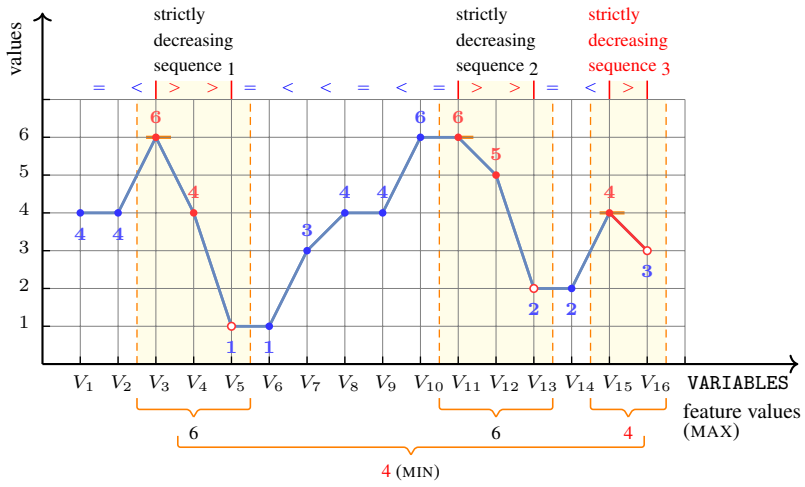


Figure 4.786: Illustrating the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.787 and 4.788 respectively depict the automaton associated with the constraint MIN_MAX_STRICTLY_DECREASING_SEQUENCE and its simplified form.

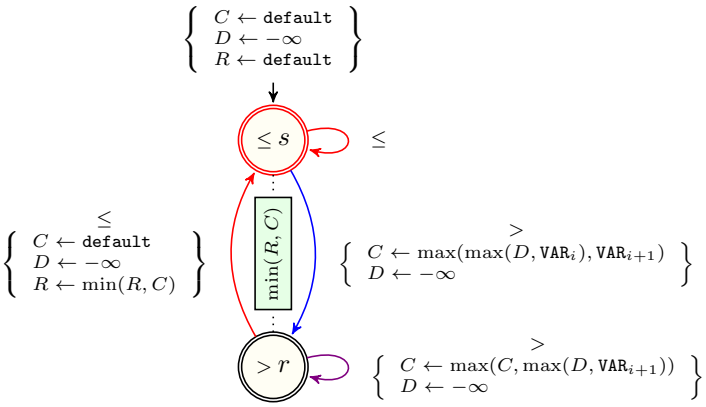


Figure 4.787: Automaton for the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$

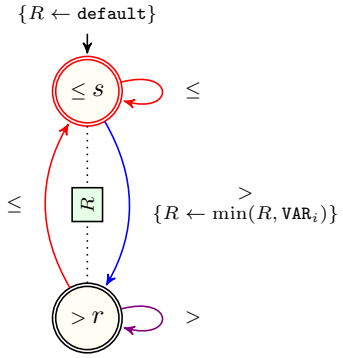


Figure 4.788: Simplified automaton for the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.143: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$+\infty$	$\overleftarrow{C}$
r	$+\infty$	$+\infty$ M

Table 4.144: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

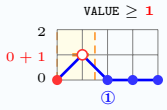
Constraint MIN_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all maximum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.789 provides an example where the MIN_MAX_STRICTLY_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

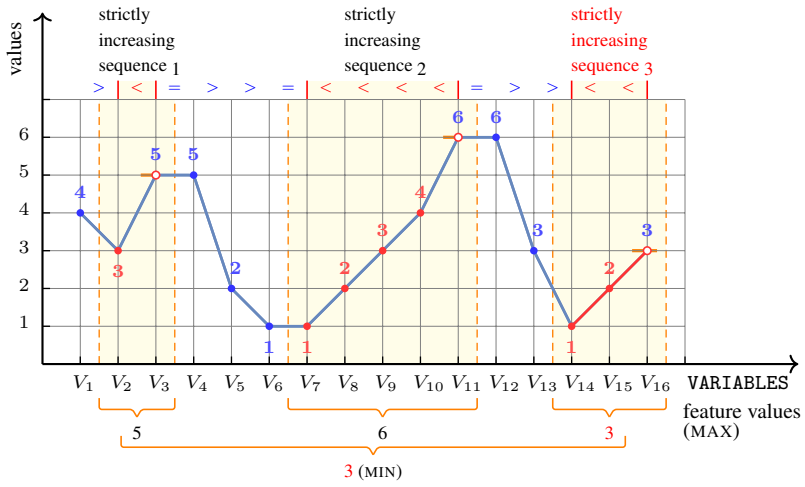


Figure 4.789: Illustrating the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.790 and 4.791 respectively depict the automaton associated with the constraint MIN_MAX_STRICTLY_INCREASING_SEQUENCE and its simplified form.

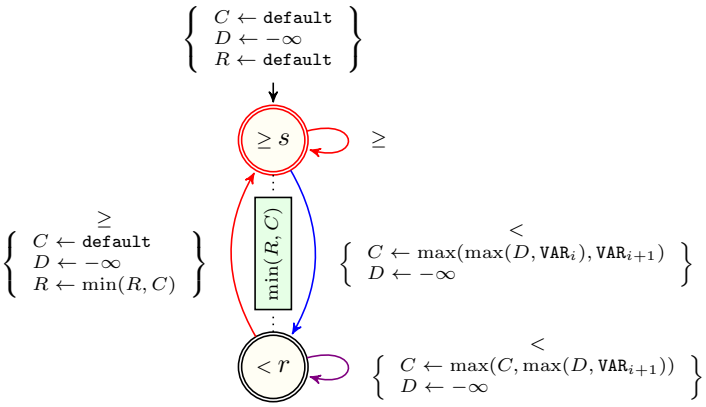


Figure 4.790: Automaton for the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$

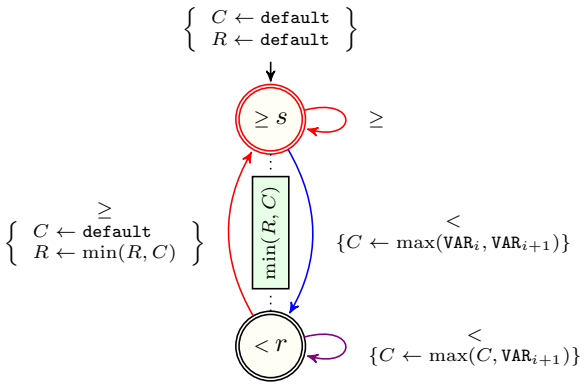


Figure 4.791: Simplified automaton for the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.145: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

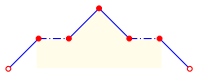
	s	r
s	$+\infty$	$+\infty$
r	$\overrightarrow{C}$	$+\infty$ M

Table 4.146: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint MIN_MAX_SUMMIT(VALUE, VARIABLES)

Arguments

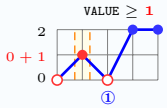
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

sv ≤ 2 ∨ rv ≤ 1 ⇒ VALUE = +∞
VALUE ≥ minv + 1①
VALUE = +∞ ∨ VALUE ≤ maxv
required(VARIABLES, var)

where

minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all maximum values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value +∞.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '(< | < (= | <)* <)(> | > (= | >)* >)'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 1$ to index j.

Example $(3, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1\rangle)$

Figure 4.792 provides an example where the MIN_MAX_SUMMIT $(3, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

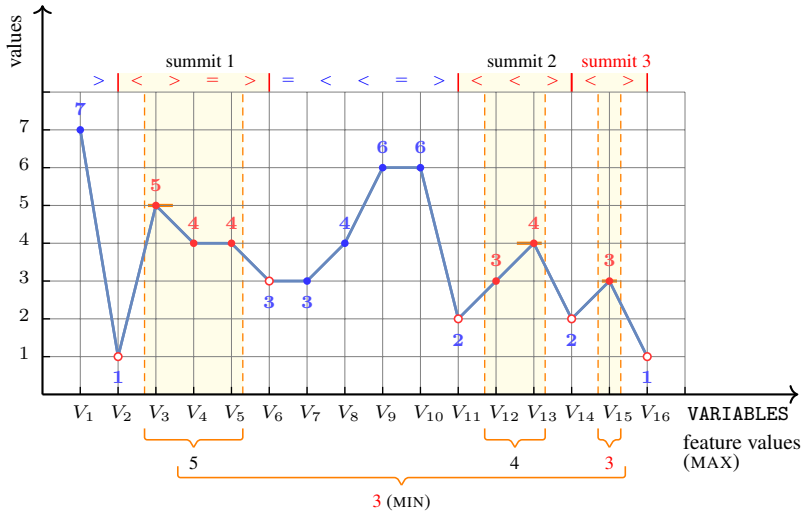


Figure 4.792: Illustrating the MIN_MAX_SUMMIT constraint of the **Example** slot

Automaton Figures 4.793 and 4.794 respectively depict the automaton associated with the constraint MIN_MAX_SUMMIT and its simplified form.

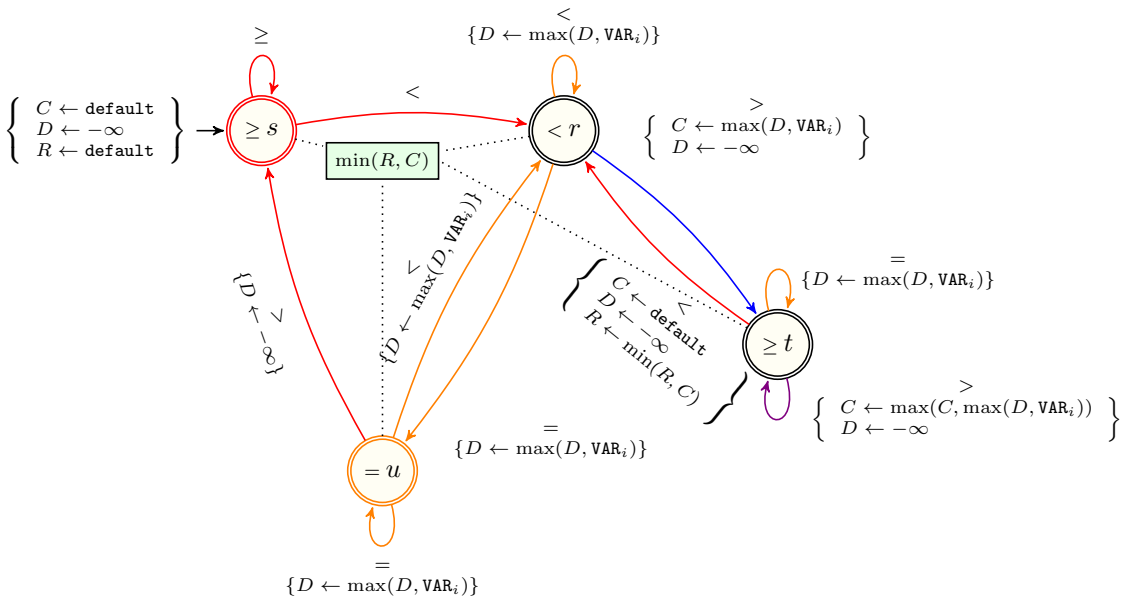


Figure 4.793: Automaton for the MIN_MAX_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is $+\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.147: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MIN_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

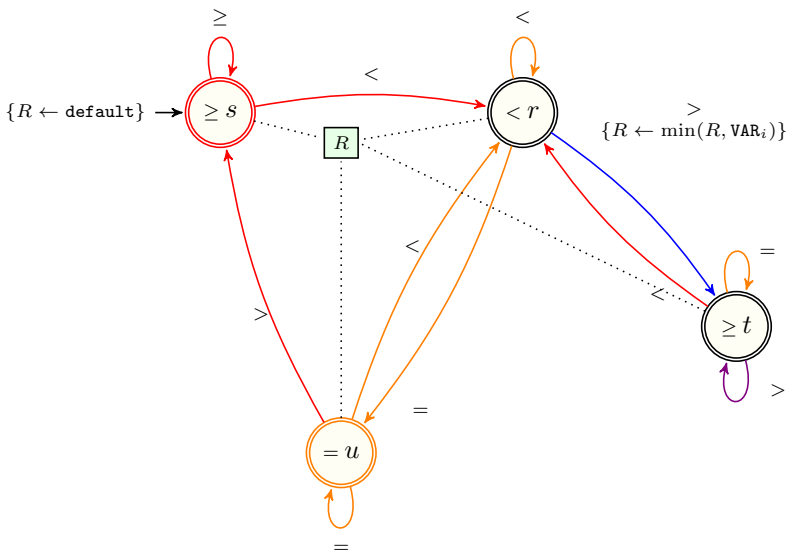


Figure 4.794: Simplified automaton for the MIN_MAX_SUMMIT constraint obtained by applying decoration Table 3.39 to the seed transducer of the SUMMIT pattern where `default` is $+\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

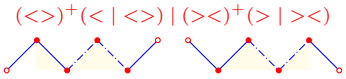
	s	r	t	u
s	$+\infty$	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\mathtt{VAR}_{i+1}$ C	$+\infty$ R	$+\infty$
t	$+\infty$	$+\infty$ L	$+\infty$	$+\infty$ L
u	$+\infty$	$+\infty$	$+\infty$ R	$+\infty$

Table 4.148: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the MIN_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MAX_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint MIN_MAX_ZIGZAG(VALUE, VARIABLES)

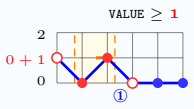
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)

where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimum of all maximum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(3, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.795 provides an example where the MIN_MAX_ZIGZAG $(3, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

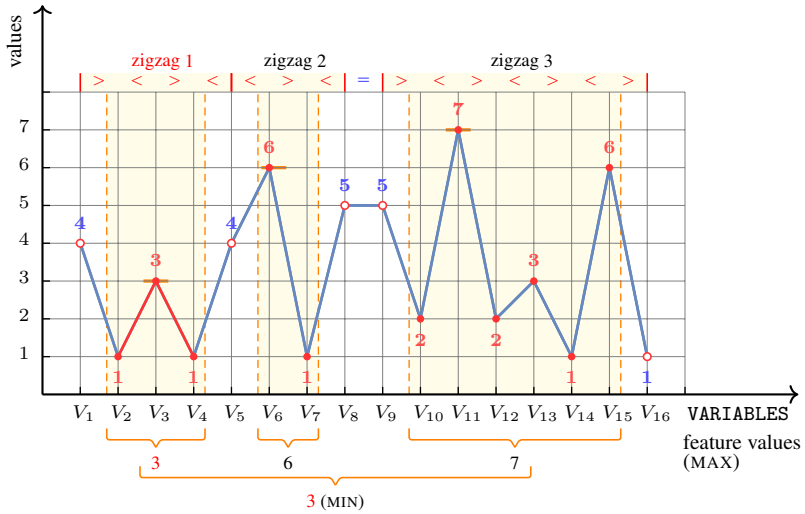


Figure 4.795: Illustrating the MIN_MAX_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.796 and 4.797 respectively depict the automaton associated with the constraint MIN_MAX_ZIGZAG and its simplified form.

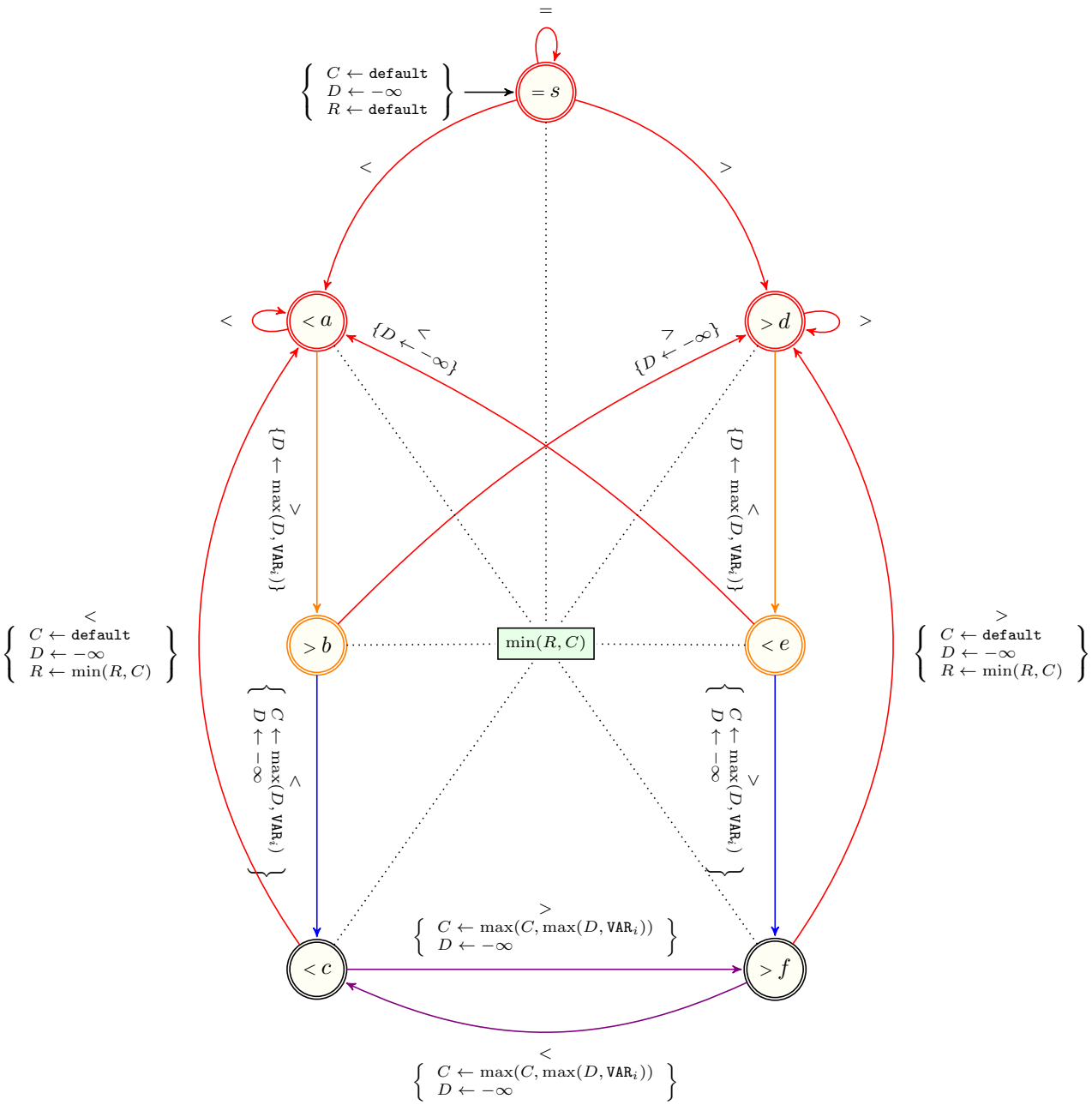


Figure 4.796: Automaton for the MIN_MAX_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $+\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

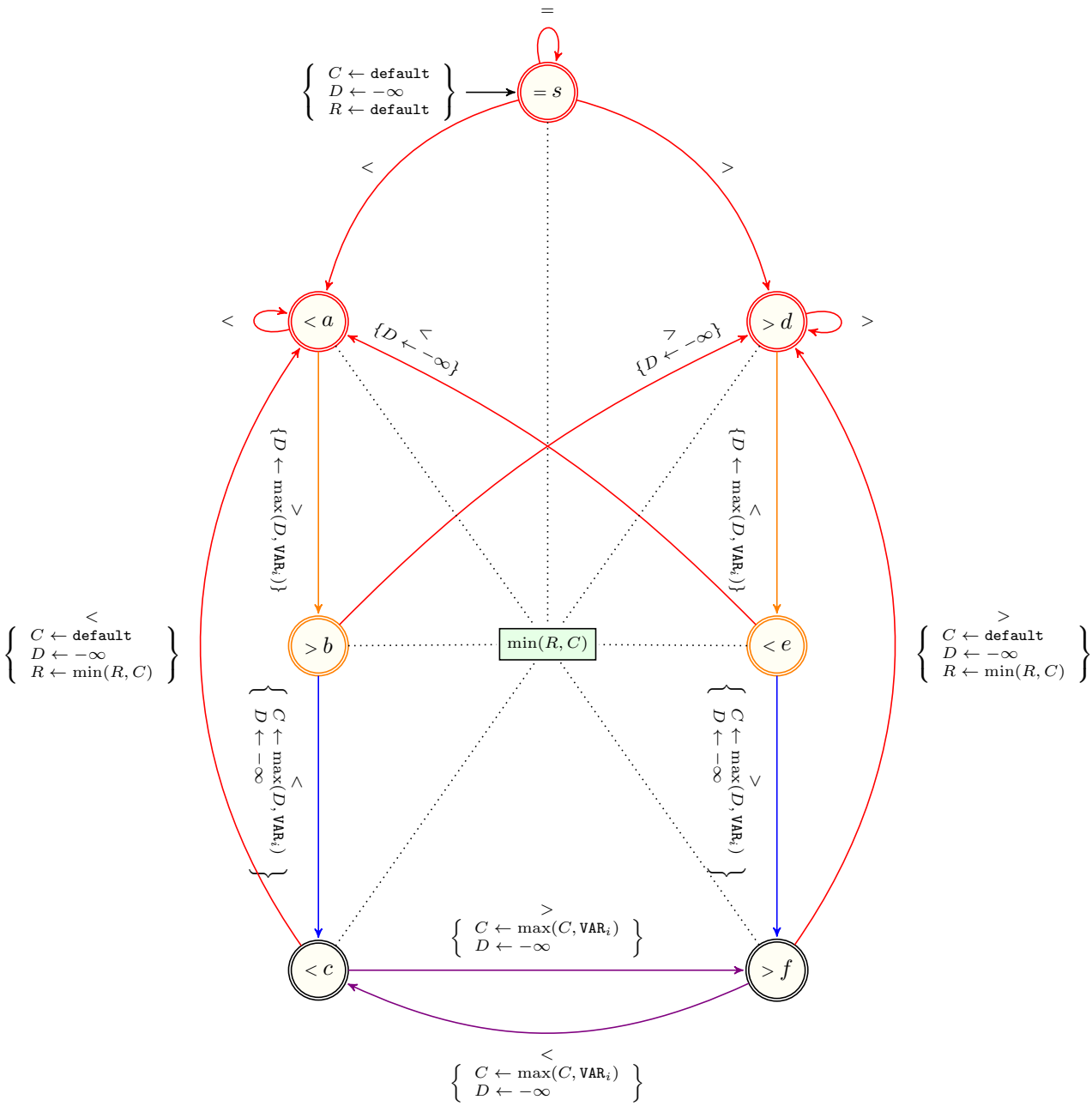


Figure 4.797: Simplified automaton for the MIN_MAX_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is $+\infty$; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

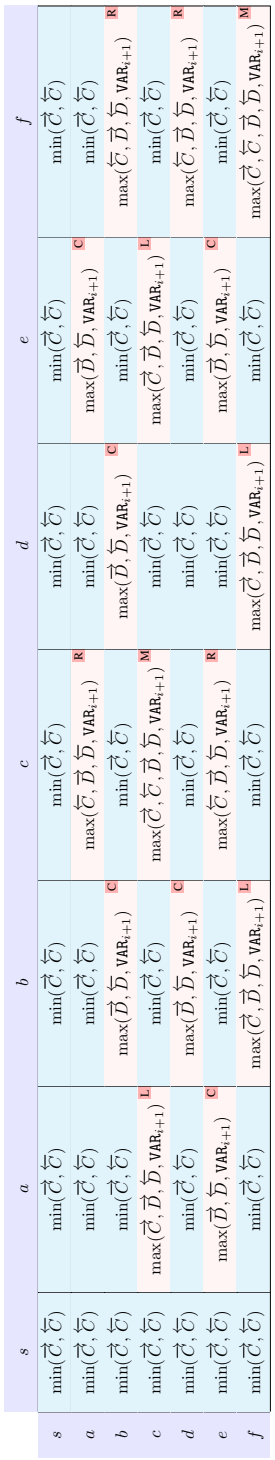
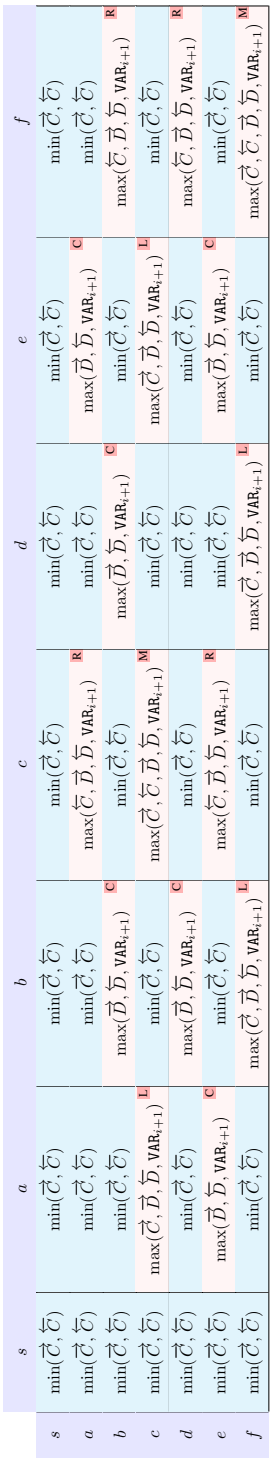


Table 4.149: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MIN_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

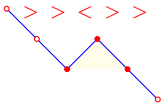


	s	a	b	c	d	e	f
s	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$
a	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\max(\overleftarrow{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\vec{C}, \overleftarrow{C})$
b	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\vec{C}, \overleftarrow{C})$	$\max(\overleftarrow{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
c	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \overleftarrow{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ M	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\vec{C}, \overleftarrow{C})$
d	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\max(\overleftarrow{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
e	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$
f	$\min(\vec{C}, \overleftarrow{C})$	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\vec{C}, \overleftarrow{C})$	$\max(\vec{C}, \overleftarrow{C}, \vec{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ M

Table 4.150: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MIN_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MAX, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

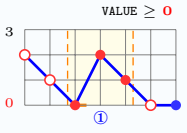
Constraint MIN_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 2$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all minimum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(2, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3 \rangle)$

Figure 4.798 provides an example where the MIN_MIN_BUMP_ON_DECREASING_SEQUENCE $(2, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
range(VARIABLES.var) > 2

Arg. properties Functional dependency: VALUE determined by VARIABLES.

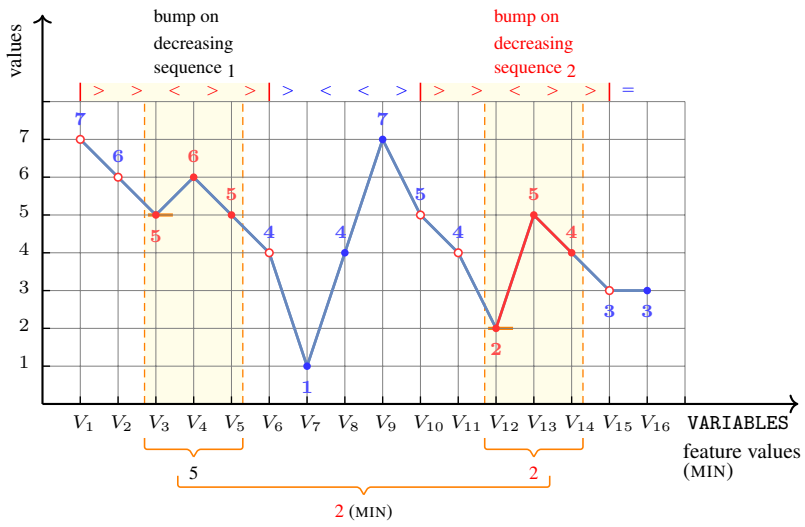


Figure 4.798: Illustrating the MIN_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.799 and 4.800 respectively depict the automaton associated with the constraint MIN_MIN_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

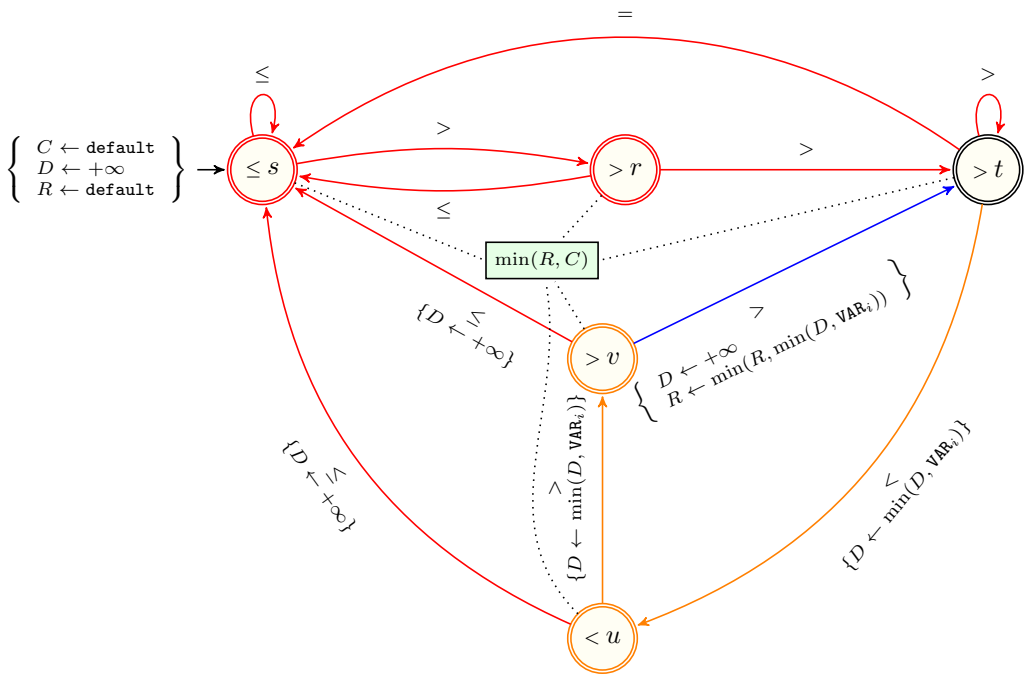


Figure 4.799: Automaton for the MIN_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$

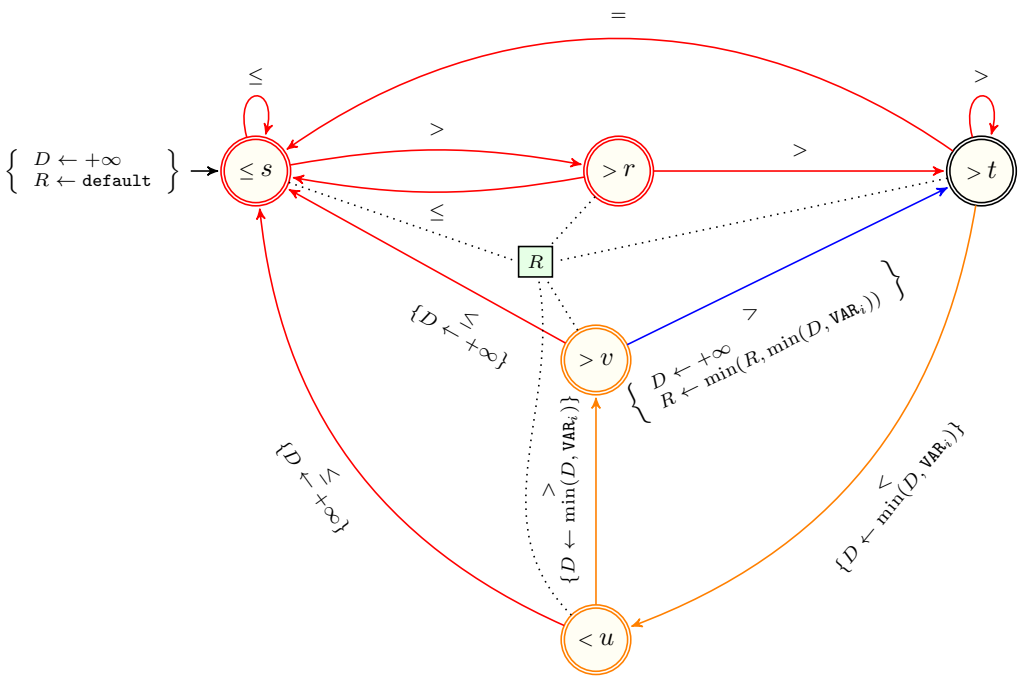


Figure 4.800: Simplified automaton for the MIN_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

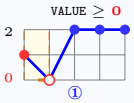
Constraint MIN_MIN_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.801 provides an example where the MIN_MIN_DECREASING $(1, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

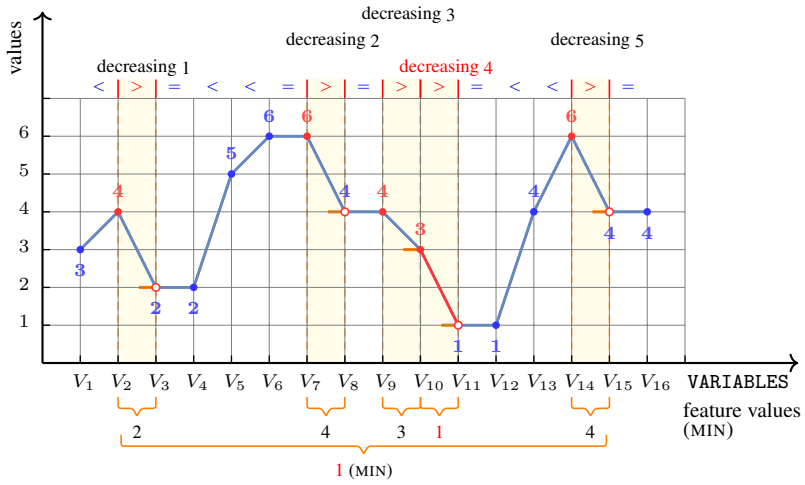


Figure 4.801: Illustrating the MIN_MIN_DECREASING constraint of the **Example** slot

Automaton

Figures 4.802 and 4.803 respectively depict the automaton associated with the constraint MIN_MIN_DECREASING and its simplified form.

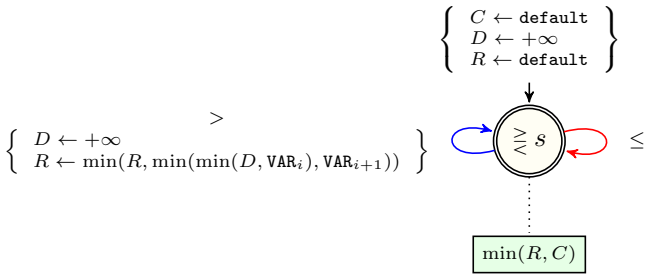


Figure 4.802: Automaton for the MIN_MIN_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $+\infty$

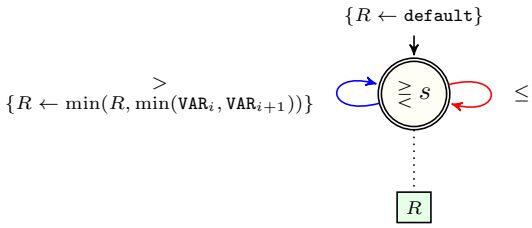


Figure 4.803: Simplified automaton for the MIN_MIN_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

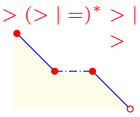
Table 4.151: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MIN_MIN_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.152: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MIN_MIN_DECREASING constraint defined as the composition of the DECREASING pattern , the feature MIN , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

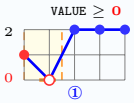
Constraint MIN_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.804 provides an example where the MIN_MIN_DECREASING_SEQUENCE $(1, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

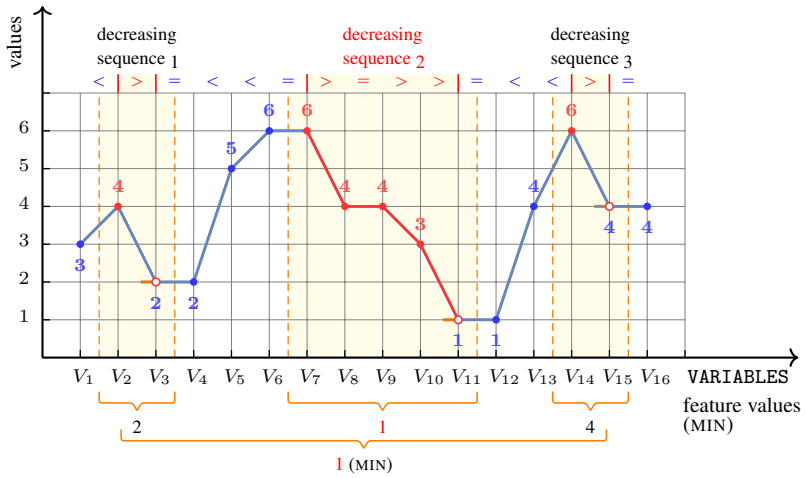


Figure 4.804: Illustrating the MIN_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.805 and 4.806 respectively depict the automaton associated with the constraint MIN_MIN_DECREASING_SEQUENCE and its simplified form.

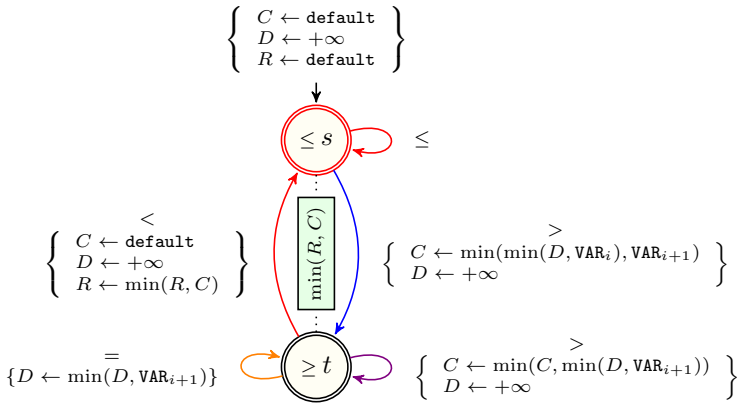


Figure 4.805: Automaton for the MIN_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$

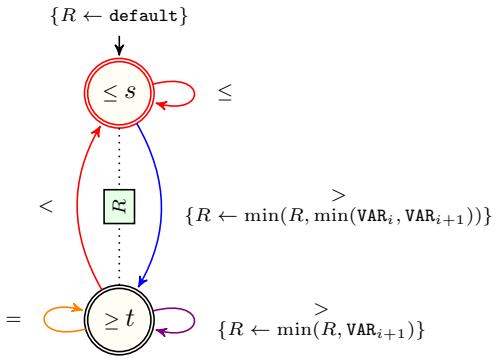


Figure 4.806: Simplified automaton for the MIN_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

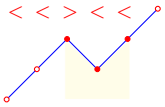
Table 4.153: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MIN_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	$+\infty$	$+\infty$
t	$+\infty$	$+\infty$ M

Table 4.154: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the MIN_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

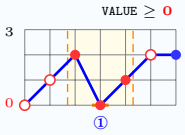
Constraint MIN_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 2$
required(VARIABLES, var)
where
minv = minval(VARIABLES.var)
maxv = maxval(VARIABLES.var)
sv = |VARIABLES|
rv = range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(1, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.807 provides an example where the MIN_MIN_DIP_ON_INCREASING_SEQUENCE $(1, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

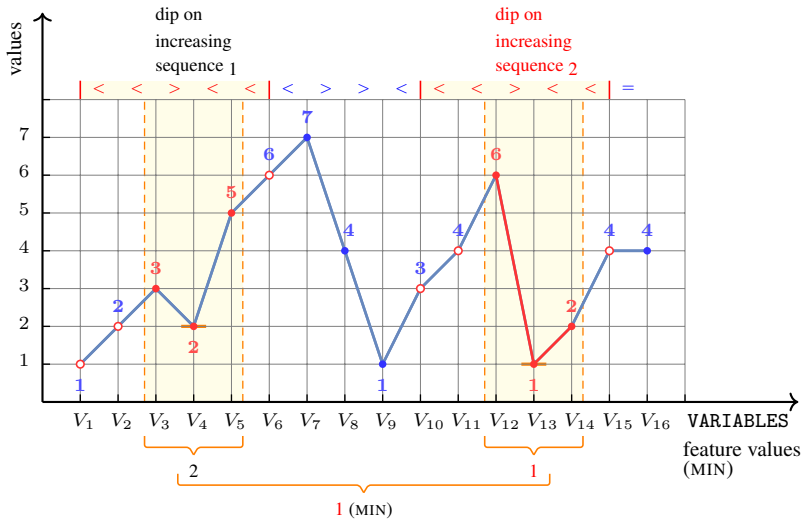


Figure 4.807: Illustrating the MIN_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.808 and 4.809 respectively depict the automaton associated with the constraint MIN_MIN_DIP_ON_INCREASING_SEQUENCE and its simplified form.

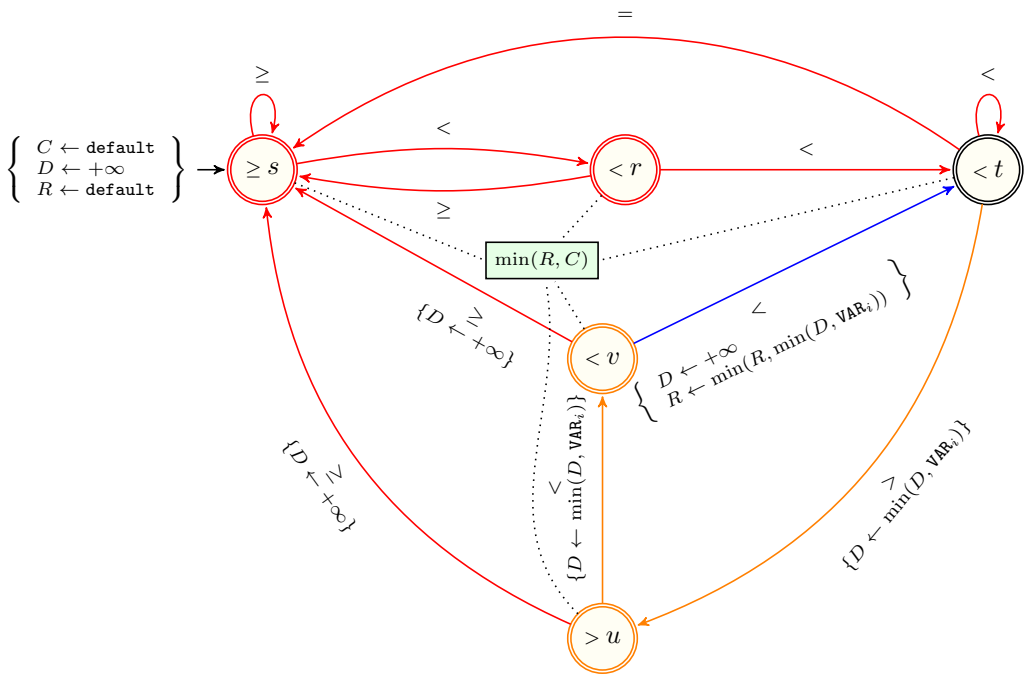


Figure 4.808: Automaton for the MIN_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$

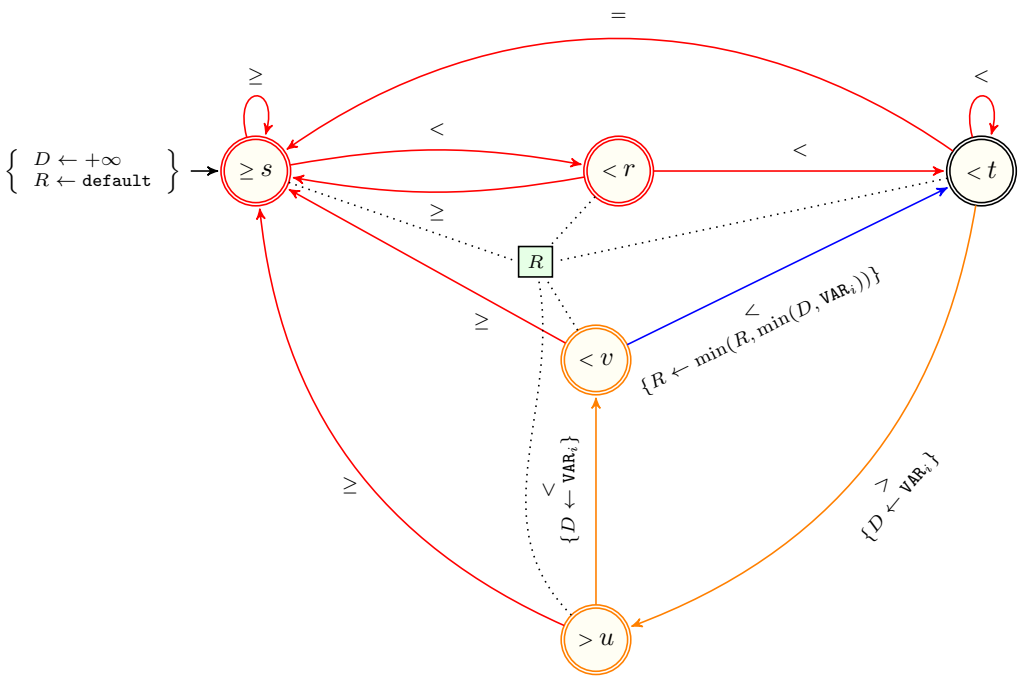
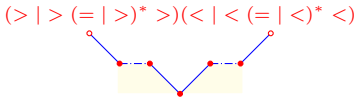


Figure 4.809: Simplified automaton for the MIN_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MIN_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the GORGE pattern.

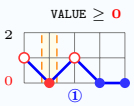
Constraint MIN_MIN_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.810 provides an example where the MIN_MIN_GORGE $(3, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

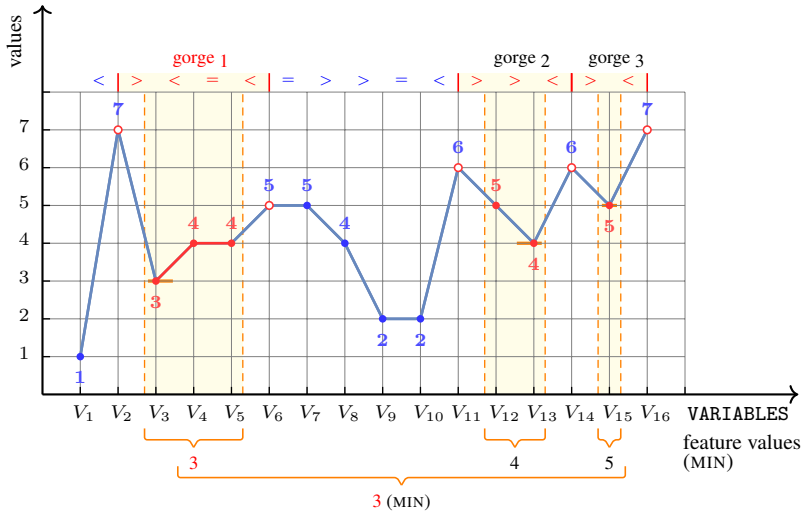


Figure 4.810: Illustrating the MIN_MIN_GORGE constraint of the **Example** slot

Automaton Figures 4.811 and 4.812 respectively depict the automaton associated with the constraint MIN_MIN_GORGE and its simplified form.

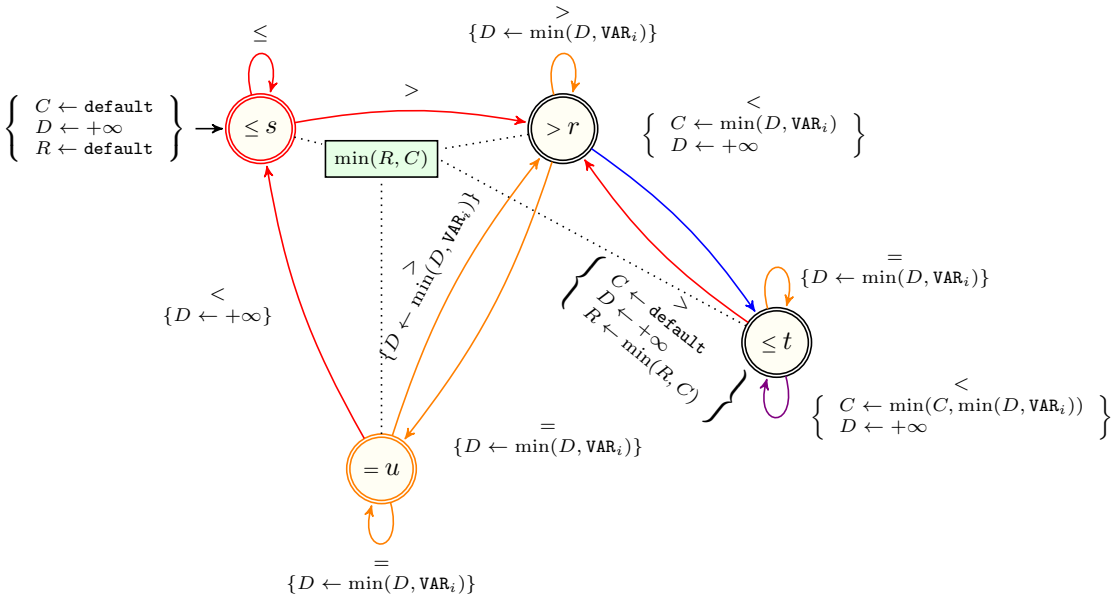


Figure 4.811: Automaton for the MIN_MIN_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is $+\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

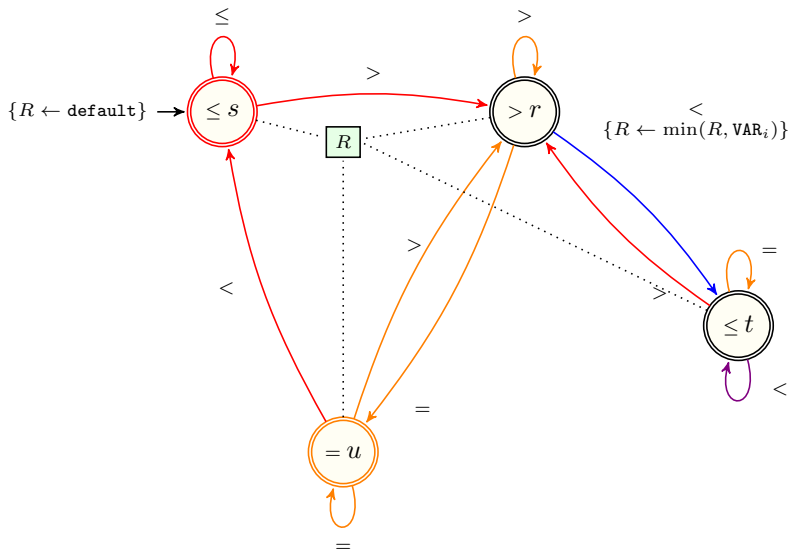


Figure 4.812: Simplified automaton for the MIN_MIN_GORGE constraint obtained by applying decoration Table 3.39 to the seed transducer of the GORGE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.155: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MIN_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	$+\infty$	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\mathtt{VAR}_{i+1}$ C	$+\infty$ R	$+\infty$
t	$+\infty$	$+\infty$ L	$+\infty$	$+\infty$ L
u	$+\infty$	$+\infty$	$+\infty$ R	$+\infty$

Table 4.156: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the MIN_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

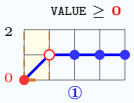
Constraint MIN_MIN_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all minimum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.813 provides an example where the MIN_MIN_INCREASING $(1, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

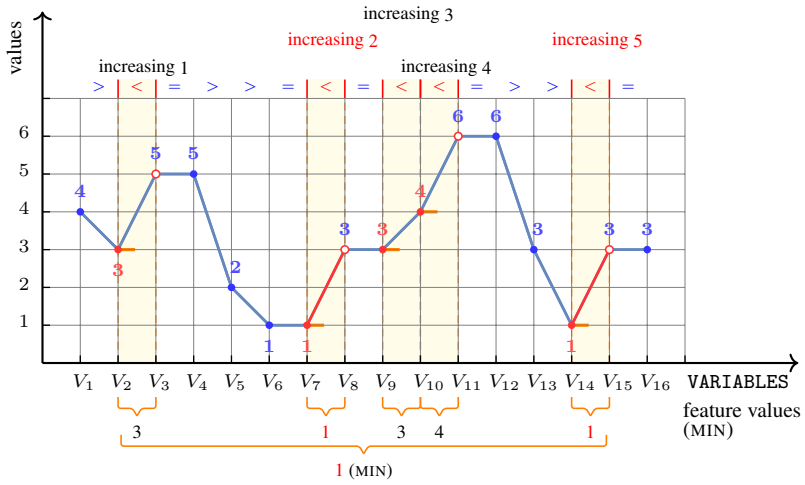


Figure 4.813: Illustrating the MIN_MIN_INCREASING constraint of the **Example** slot

Automaton

Figures 4.814 and 4.815 respectively depict the automaton associated with the constraint MIN_MIN_INCREASING and its simplified form.

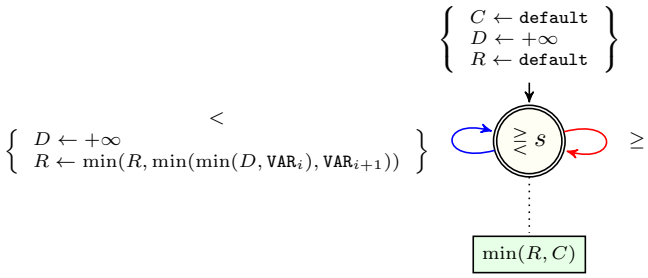


Figure 4.814: Automaton for the MIN_MIN_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $+\infty$

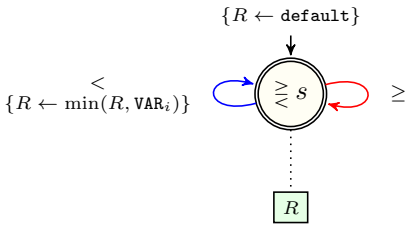


Figure 4.815: Simplified automaton for the MIN_MIN_INCREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

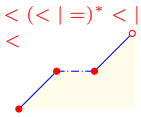
Table 4.157: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MIN_MIN_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.158: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MIN_MIN_INCREASING constraint defined as the composition of the INCREASING pattern , the feature MIN , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING_SEQUENCE pattern.

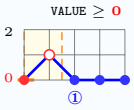
Constraint MIN_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all minimum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.816 provides an example where the MIN_MIN_INCREASING_SEQUENCE $(1, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

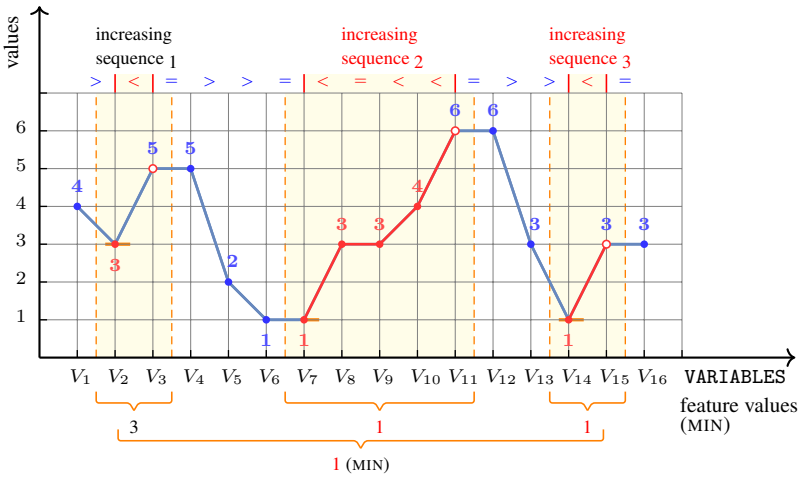


Figure 4.816: Illustrating the MIN_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.817 and 4.818 respectively depict the automaton associated with the constraint MIN_MIN_INCREASING_SEQUENCE and its simplified form.

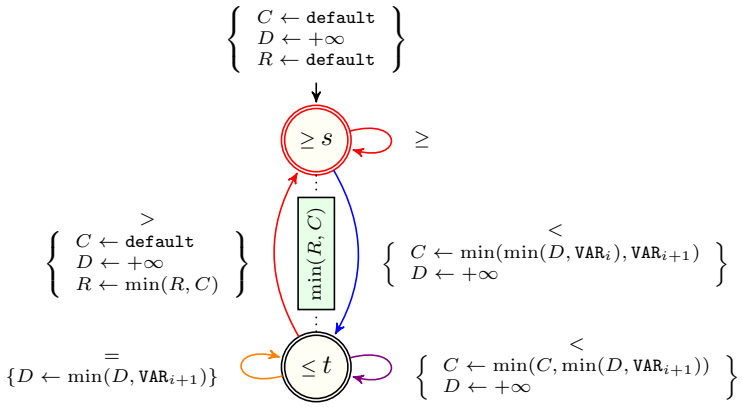


Figure 4.817: Automaton for the MIN_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$

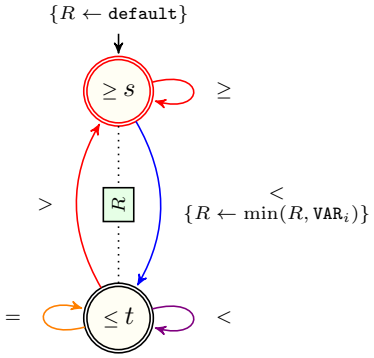


Figure 4.818: Simplified automaton for the MIN_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.159: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MIN_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

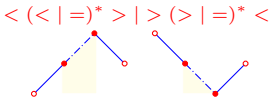
	s	t
s	$+\infty$	$+\infty$
t	$+\infty$	$+\infty$ M

Table 4.160: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the MIN_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

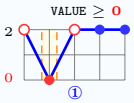
Constraint MIN_MIN_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the minimum of all minimum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(1, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.819 provides an example where the MIN_MIN_INFLEXION $(1, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

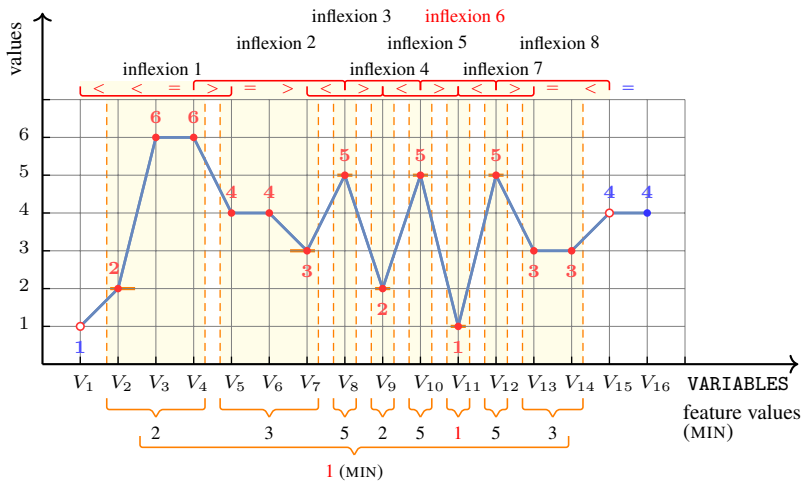


Figure 4.819: Illustrating the MIN_MIN_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.820 and 4.821 respectively depict the automaton associated with the constraint MIN_MIN_INFLEXION and its simplified form.

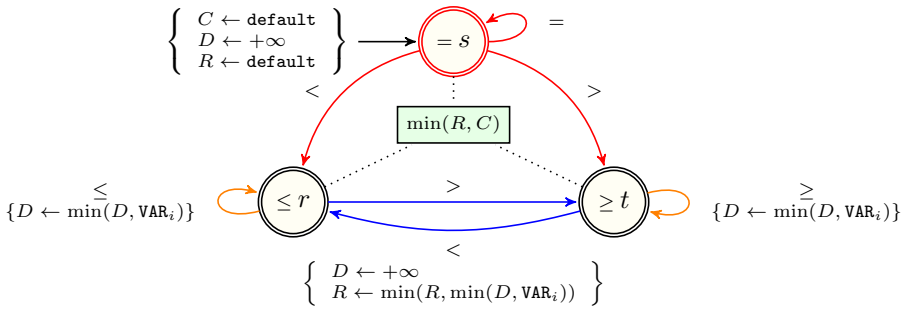


Figure 4.820: Automaton for the MIN_MIN_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

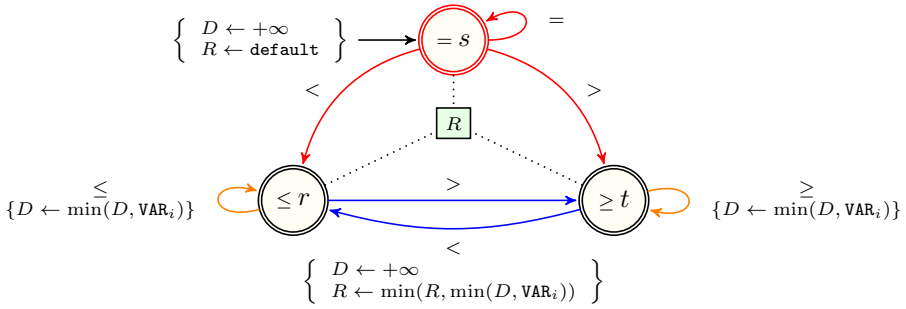


Figure 4.821: Simplified automaton for the MIN_MIN_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

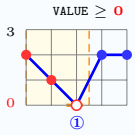
Constraint MIN_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all minimum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^{+}$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example $(1, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle)$

Figure 4.822 provides an example where the MIN_MIN_STRICTLY_DECREASING_SEQUENCE $(1, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

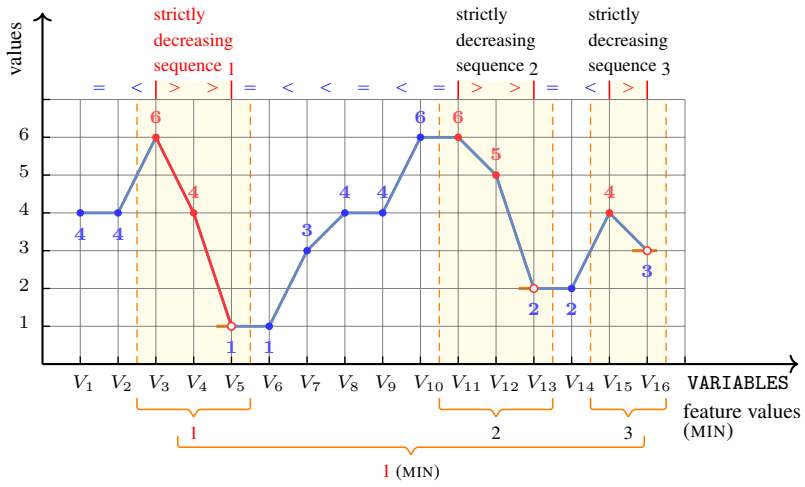


Figure 4.822: Illustrating the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.823 and 4.824 respectively depict the automaton associated with the constraint MIN_MIN_STRICTLY_DECREASING_SEQUENCE and its simplified form.

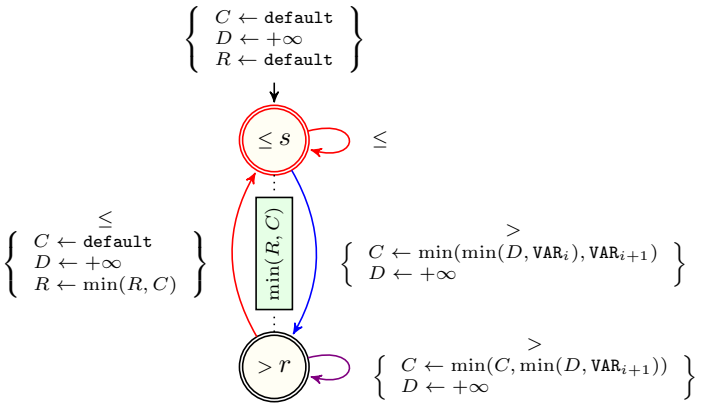


Figure 4.823: Automaton for the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$

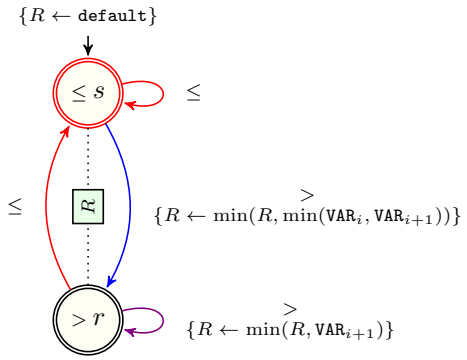


Figure 4.824: Simplified automaton for the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.161: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$ M

Table 4.162: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

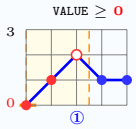
Constraint MIN_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum of all minimum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(1, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.825 provides an example where the MIN_MIN_STRICTLY_INCREASING_SEQUENCE $(1, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

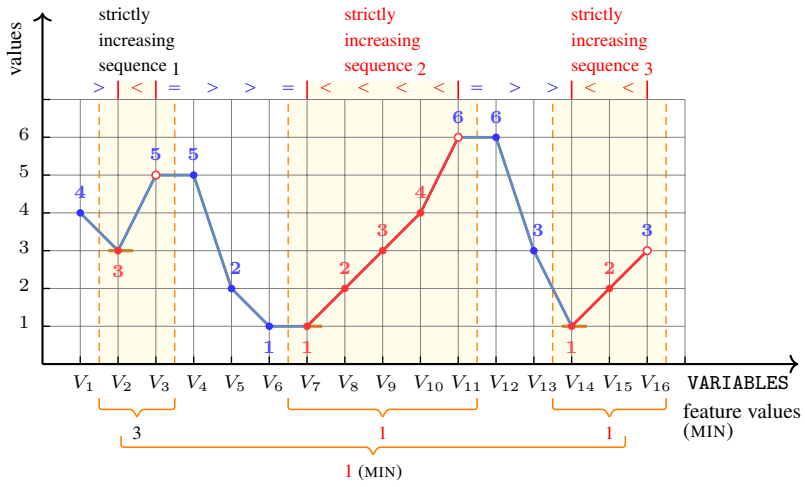


Figure 4.825: Illustrating the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.826 and 4.827 respectively depict the automaton associated with the constraint MIN_MIN_STRICTLY_INCREASING_SEQUENCE and its simplified form.

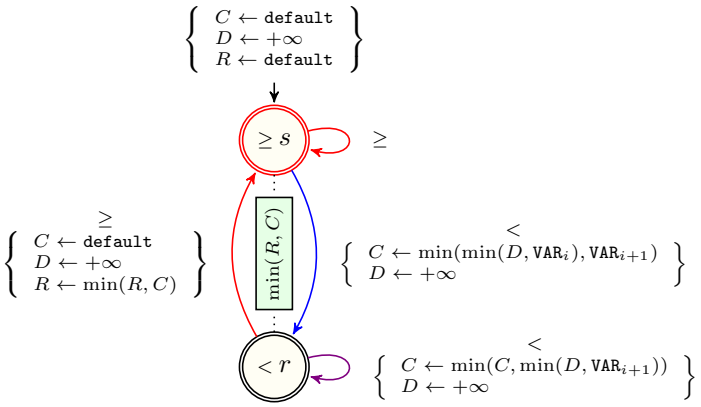


Figure 4.826: Automaton for the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$

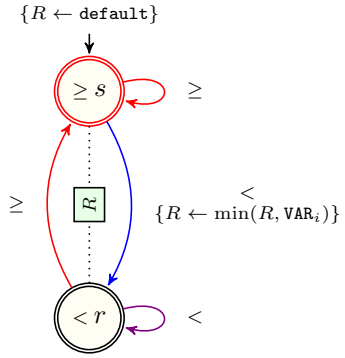


Figure 4.827: Simplified automaton for the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.163: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

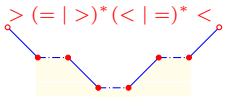
	s	r
s	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$ M

Table 4.164: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

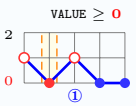
Constraint MIN_MIN_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(2, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.828 provides an example where the MIN_MIN_VALLEY $(2, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

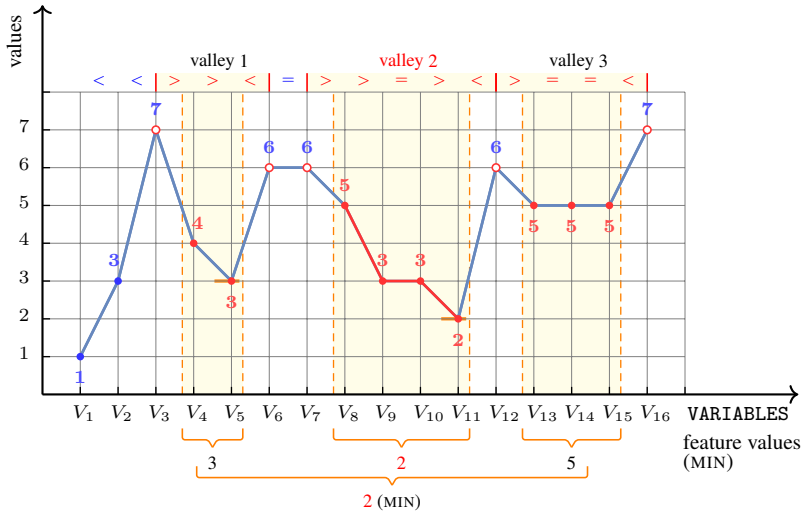


Figure 4.828: Illustrating the MIN_MIN_VALLEY constraint of the **Example** slot

Automaton Figures 4.829 and 4.830 respectively depict the automaton associated with the constraint MIN_MIN_VALLEY and its simplified form.

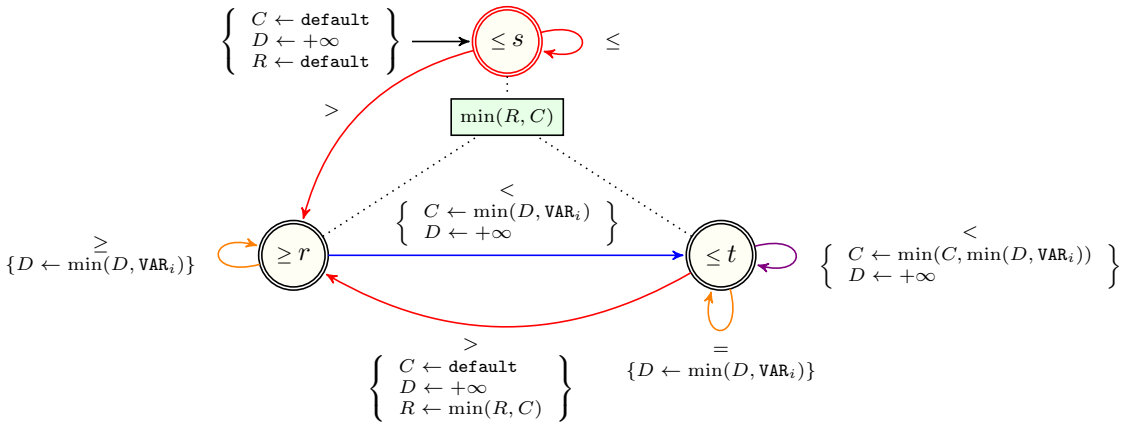


Figure 4.829: Automaton for the MIN_MIN_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is $+\infty$

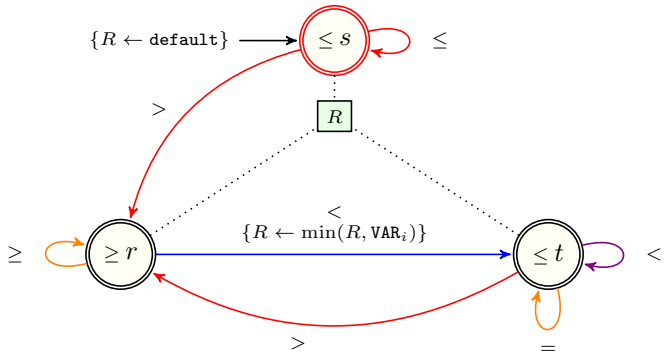


Figure 4.830: Simplified automaton for the MIN_MIN_VALLEY constraint obtained by applying decoration Table 3.39 to the seed transducer of the VALLEY pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.165: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MIN_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

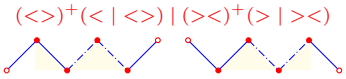
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\texttt{VAR}_{i+1}$ C	$+\infty$ R
t	$+\infty$	$+\infty$ L	$+\infty$

Table 4.166: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the MIN_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_MIN_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint MIN_MIN_ZIGZAG(VALUE, VARIABLES)

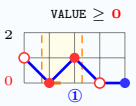
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)

where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum of all minimum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(1, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1 \rangle)$

Figure 4.831 provides an example where the MIN_MIN_ZIGZAG $(1, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

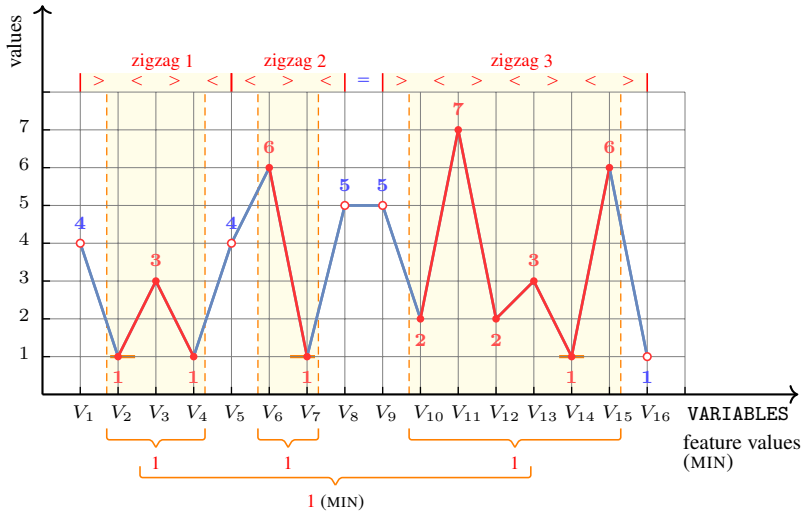


Figure 4.831: Illustrating the MIN_MIN_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.832 and 4.833 respectively depict the automaton associated with the constraint MIN_MIN_ZIGZAG and its simplified form.

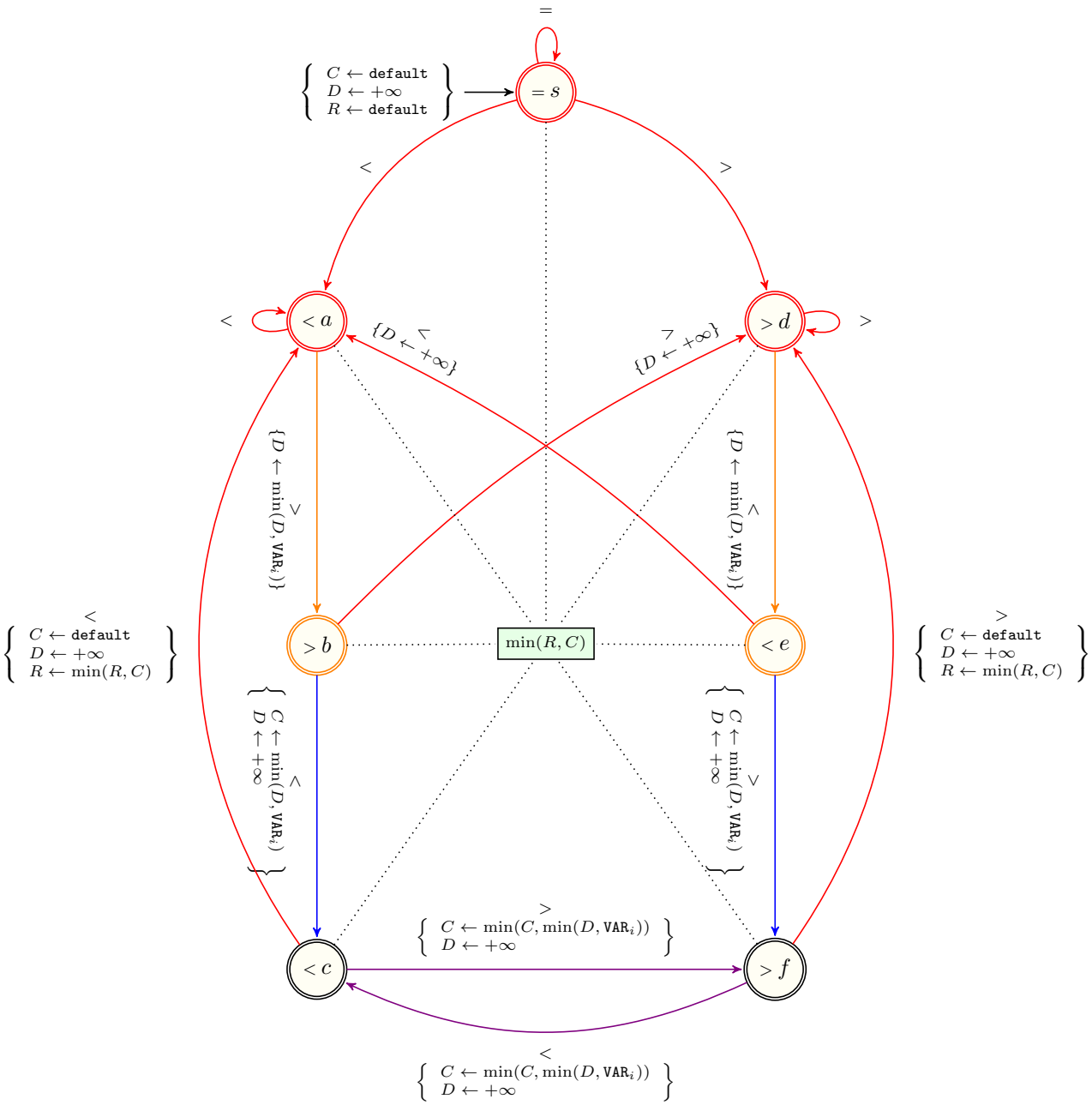


Figure 4.832: Automaton for the MIN_MIN_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $+\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

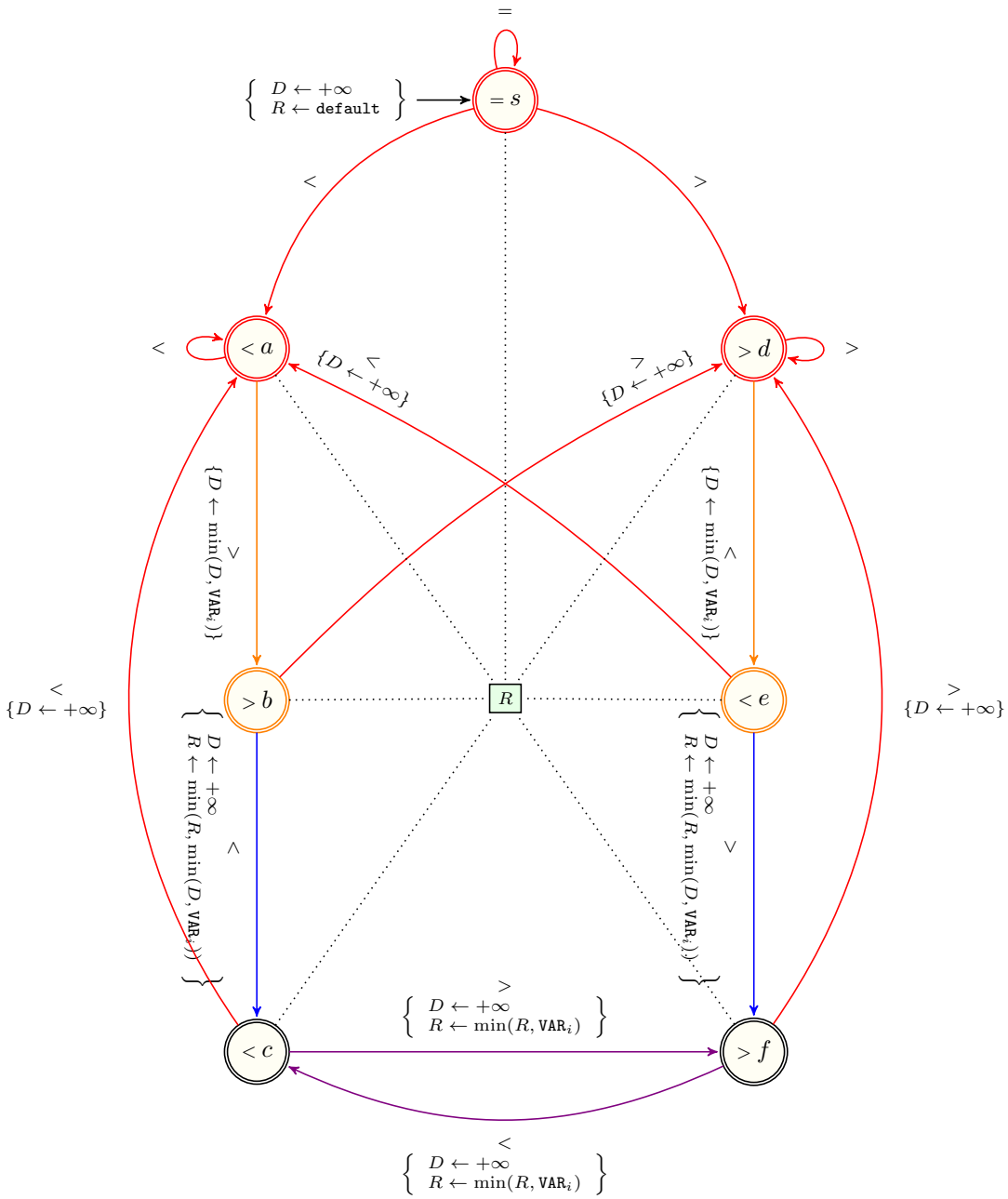


Figure 4.833: Simplified automaton for the MIN_MIN_ZIGZAG constraint obtained by applying decoration Table 3.29 to the seed transducer of the ZIGZAG pattern where default is $+\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

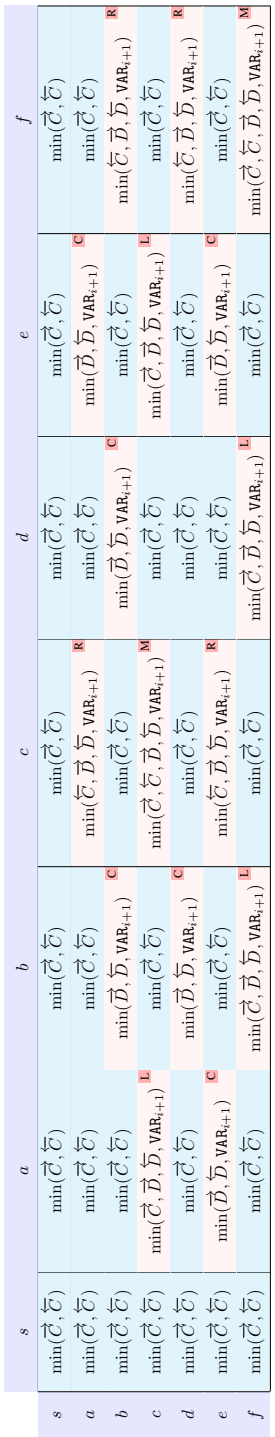


Table 4.167: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MIN_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

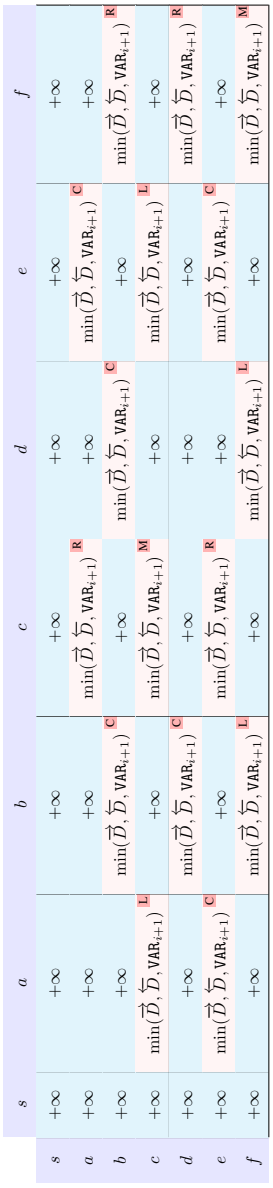


Table 4.168: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MIN_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MIN, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint MIN_RANGE_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$

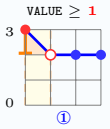
$\texttt{VALUE} \geq 1$①

required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$

rv =range(VARIABLES.var)



Purpose

VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(1, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.834 provides an example where the MIN_RANGE_DECREASING $(1, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$

range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

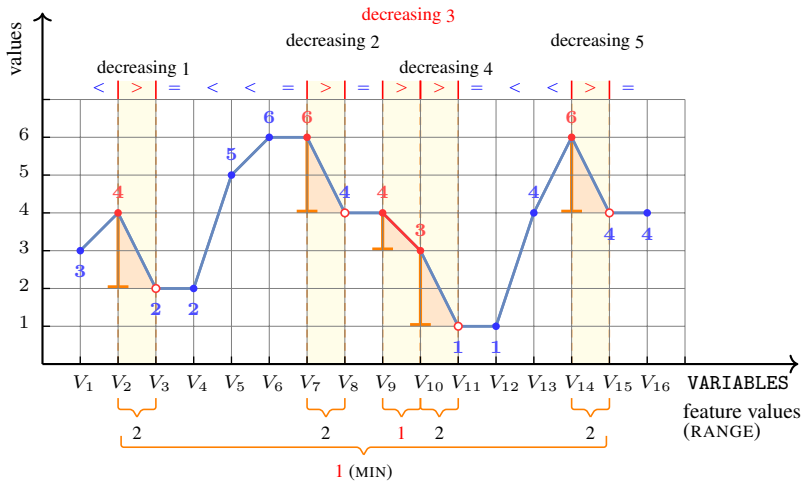


Figure 4.834: Illustrating the MIN_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figures 4.835 and 4.836 respectively depict the automaton associated with the constraint MIN_RANGE_DECREASING and its simplified form.

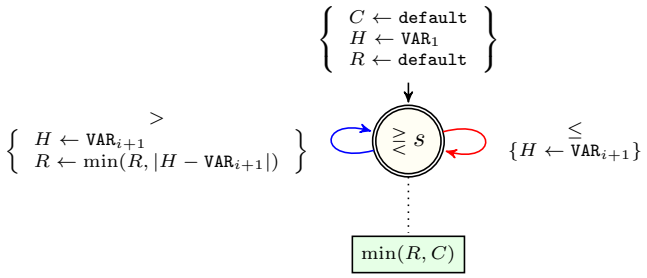


Figure 4.835: Automaton for the MIN_RANGE_DECREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING pattern where `default` is $+\infty$

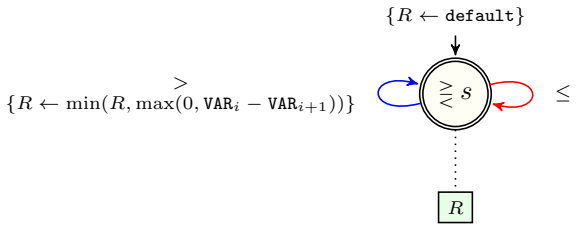
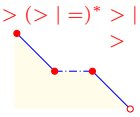


Figure 4.836: Simplified automaton for the MIN_RANGE_DECREASING constraint obtained by applying decoration Table 3.46 to the seed transducer of the DECREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

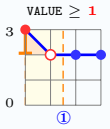
Constraint MIN_RANGE_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 1$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(2, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.837 provides an example where the MIN_RANGE_DECREASING_SEQUENCE $(2, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

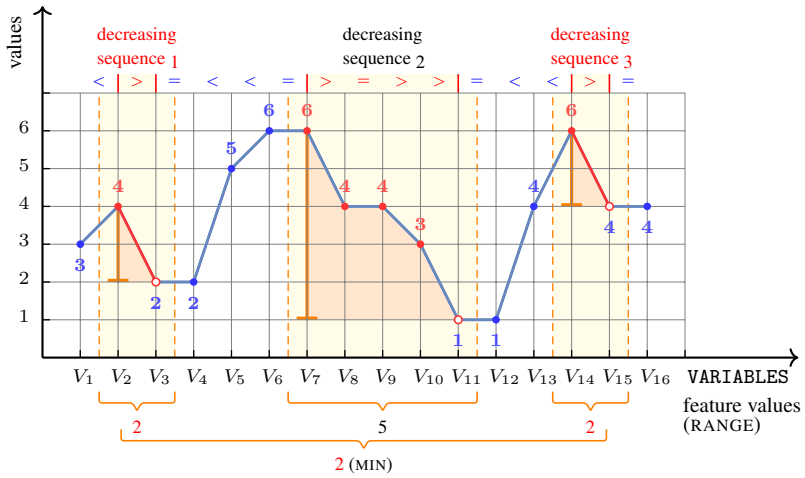


Figure 4.837: Illustrating the MIN_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.838 and 4.839 respectively depict the automaton associated with the constraint MIN_RANGE_DECREASING_SEQUENCE and its simplified form.

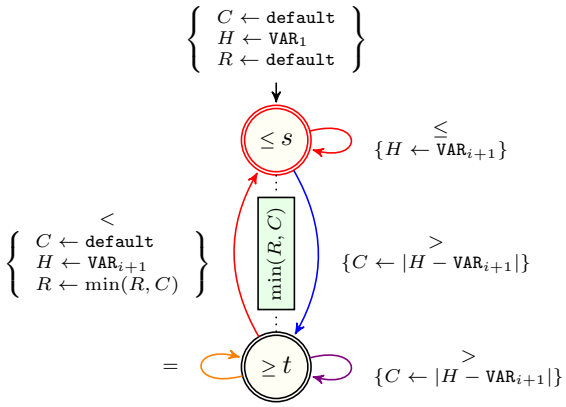


Figure 4.838: Automaton for the MIN_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$

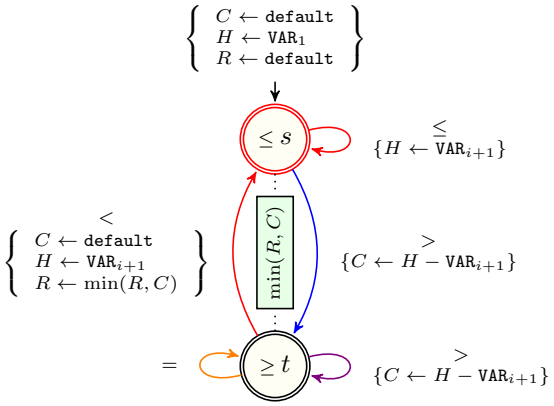


Figure 4.839: Simplified automaton for the MIN_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.42 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

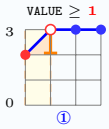
Constraint MIN_RANGE_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

sv ≤ 1 ∨ rv ≤ 1 ⇒ VALUE = +∞
VALUE ≥ 1①
required(VARIABLES, var)
where
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(1, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3\rangle)$

Figure 4.840 provides an example where the MIN_RANGE_INCREASING $(1, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3])$ constraint holds.

Typical |VARIABLES| > 1
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

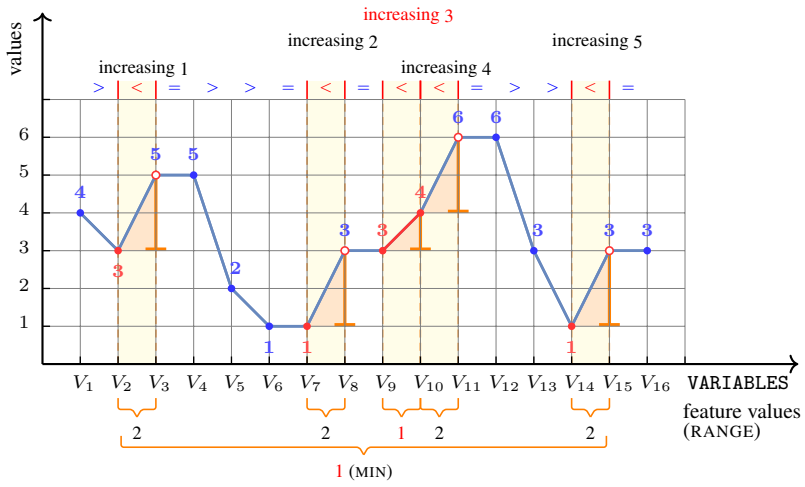


Figure 4.840: Illustrating the MIN_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figures 4.841 and 4.842 respectively depict the automaton associated with the constraint MIN_RANGE_INCREASING and its simplified form.

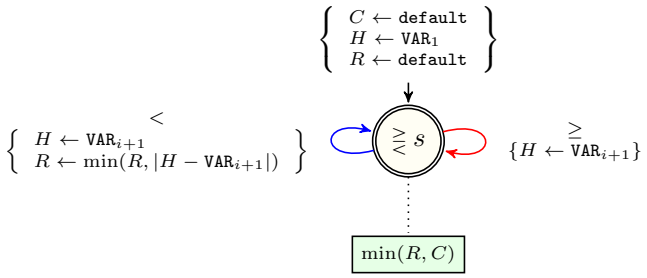


Figure 4.841: Automaton for the MIN_RANGE_INCREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING pattern where `default` is $+\infty$

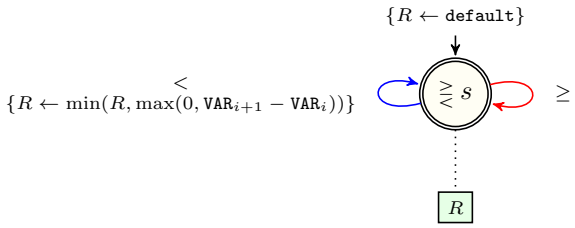
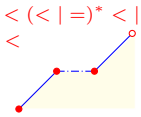


Figure 4.842: Simplified automaton for the MIN_RANGE_INCREASING constraint obtained by applying decoration Table 3.47 to the seed transducer of the INCREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

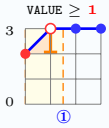
Constraint MIN_RANGE_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 1$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$. An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(2, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.843 provides an example where the MIN_RANGE_INCREASING_SEQUENCE $(2, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

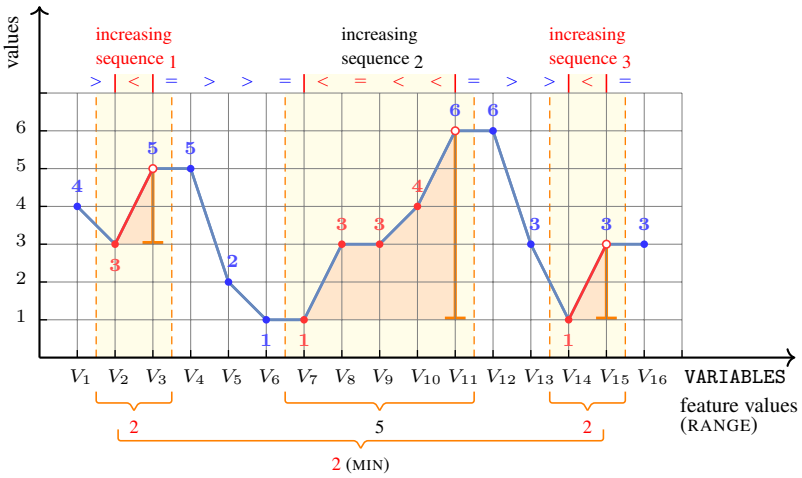


Figure 4.843: Illustrating the MIN_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.844 and 4.845 respectively depict the automaton associated with the constraint MIN_RANGE_INCREASING_SEQUENCE and its simplified form.

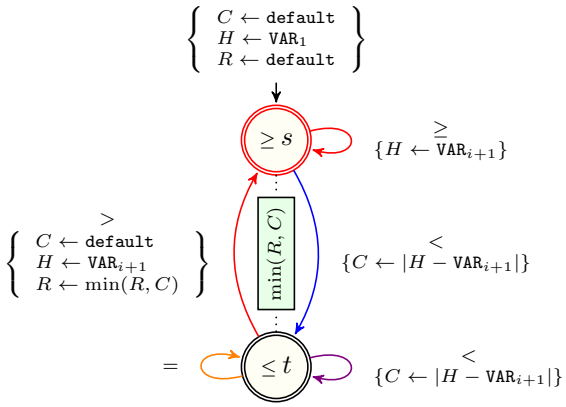


Figure 4.844: Automaton for the MIN_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$

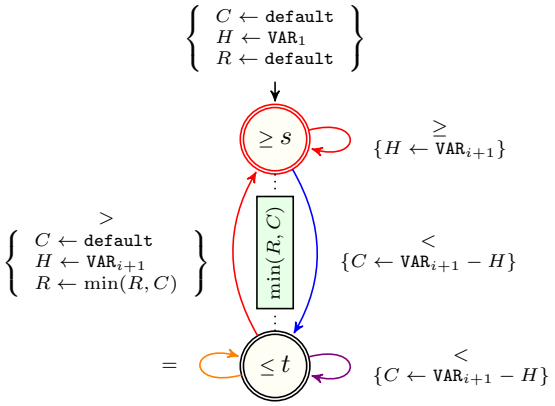


Figure 4.845: Simplified automaton for the MIN_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.43 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint MIN_RANGE_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$

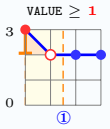
$\texttt{VALUE} \geq 1$①

required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$

rv =range(VARIABLES.var)



Purpose

VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.

An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.

Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(1, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3\rangle)$

Figure 4.846 provides an example where the MIN_RANGE_STRICTLY_DECREASING_SEQUENCE $(1, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$

range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

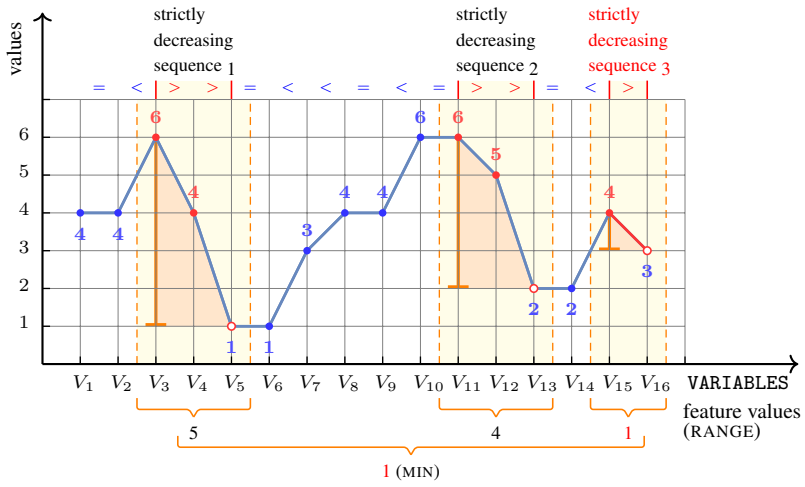


Figure 4.846: Illustrating the MIN_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.847 and 4.848 respectively depict the automaton associated with the constraint MIN_RANGE_STRICTLY_DECREASING_SEQUENCE and its simplified form.

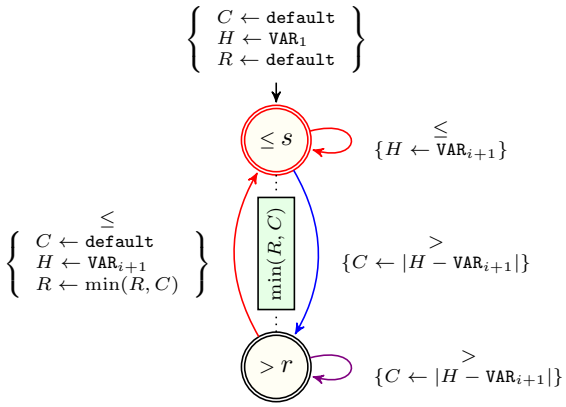


Figure 4.847: Automaton for the MIN_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$

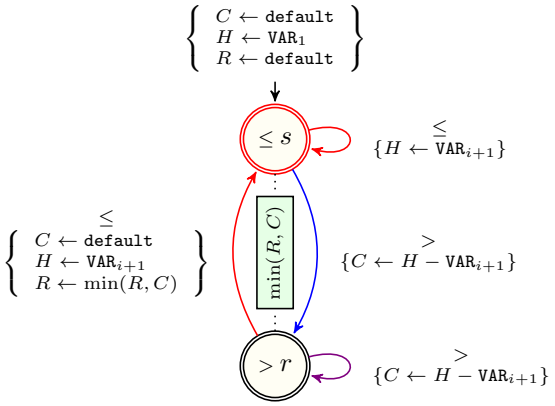


Figure 4.848: Simplified automaton for the MIN_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.42 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

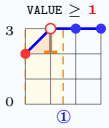
Constraint MIN_RANGE_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 1$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimum value of the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j+1$.

Example $(2, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.849 provides an example where the MIN_RANGE_STRICTLY_INCREASING_SEQUENCE $(2, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

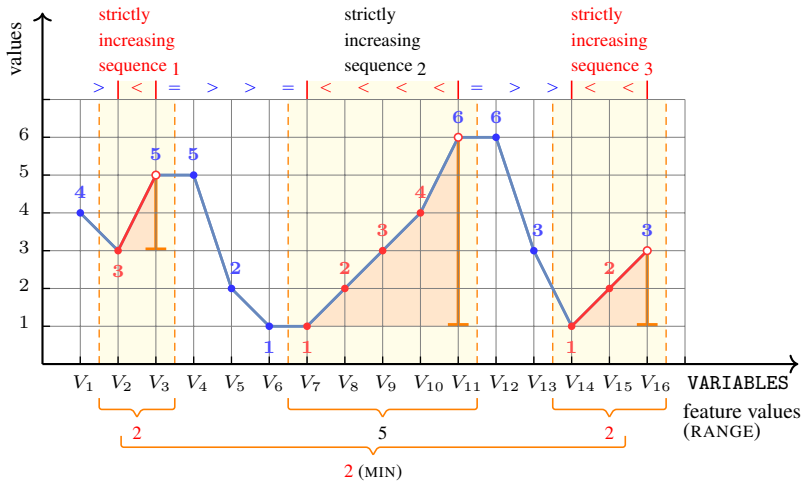


Figure 4.849: Illustrating the MIN_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.850 and 4.851 respectively depict the automaton associated with the constraint MIN_RANGE_STRICTLY_INCREASING_SEQUENCE and its simplified form.

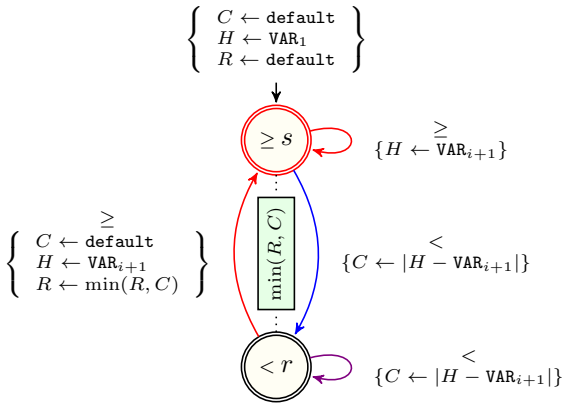


Figure 4.850: Automaton for the MIN_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$

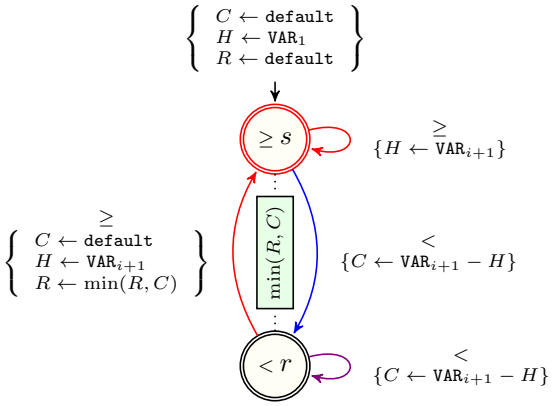


Figure 4.851: Simplified automaton for the MIN_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.43 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

FEATURE PATTERN

MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint MIN_STRICTLY_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_DECREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.852 provides an example where the MIN_STRICTLY_DECREASING_SEQUENCE $([4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3], [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

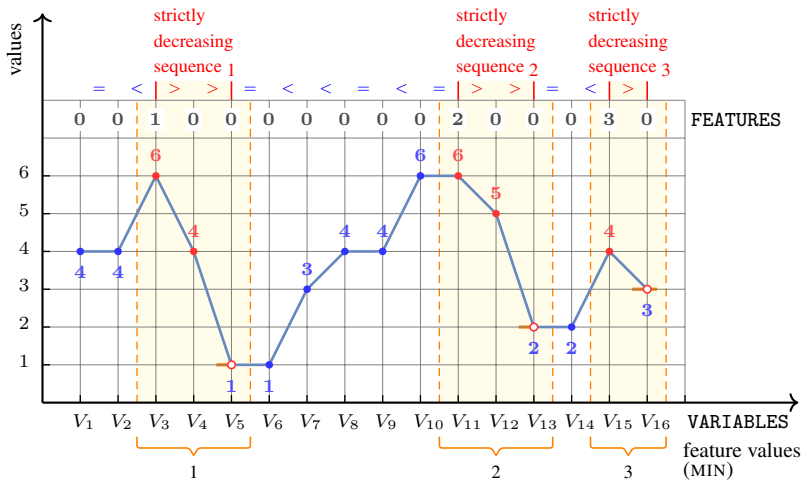


Figure 4.852: Illustrating the MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint MIN_STRICTLY_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1

where

maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES[i] is the default value DEFAULT; otherwise FEATURES[i] gives the feature value of the corresponding occurrence of STRICTLY_INCREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example Figure 4.853 provides an example where the MIN_STRICTLY_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3], [0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

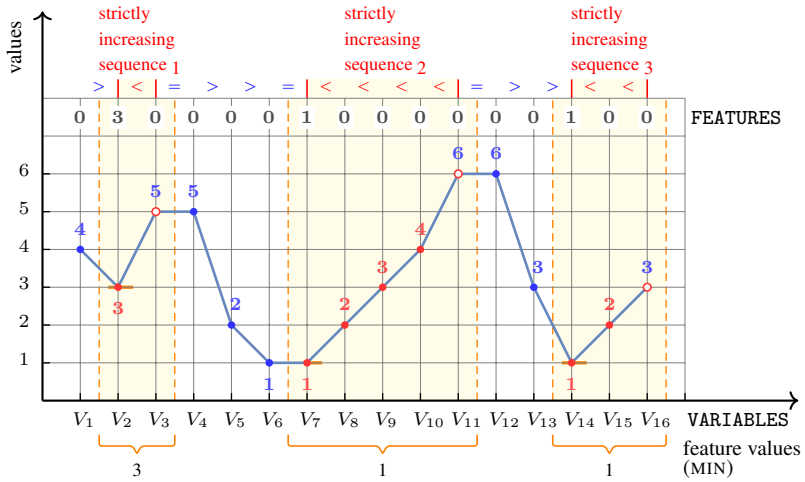


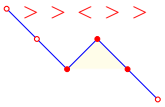
Figure 4.853: Illustrating the MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MIN_SURF_BUMP_ON_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint MIN_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

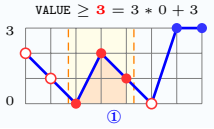
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 3 * \texttt{minv} + 3$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 3 * \texttt{maxv} - 3$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 3 - \texttt{max}(\texttt{maxv} - 3, 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(11, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3 \rangle)$

Figure 4.854 provides an example where the MIN_SURF_BUMP_ON_DECREASING_SEQUENCE $(11, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

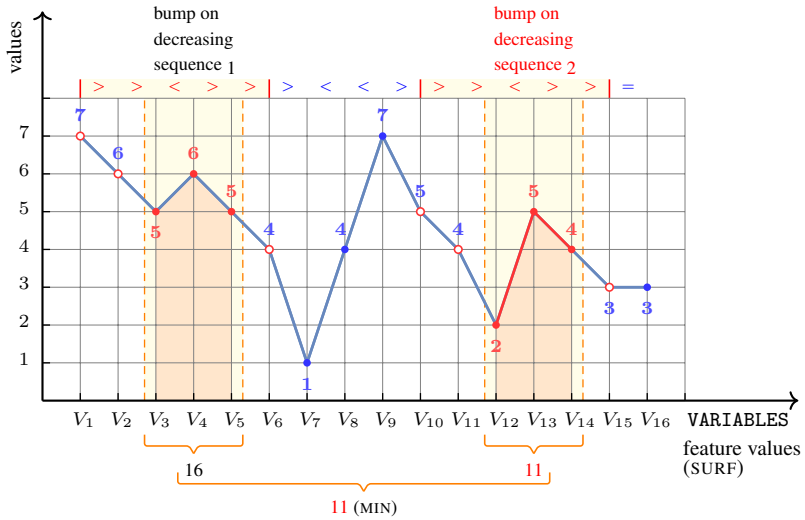


Figure 4.854: Illustrating the MIN_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.855 and 4.856 respectively depict the automaton associated with the constraint MIN_SURF_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

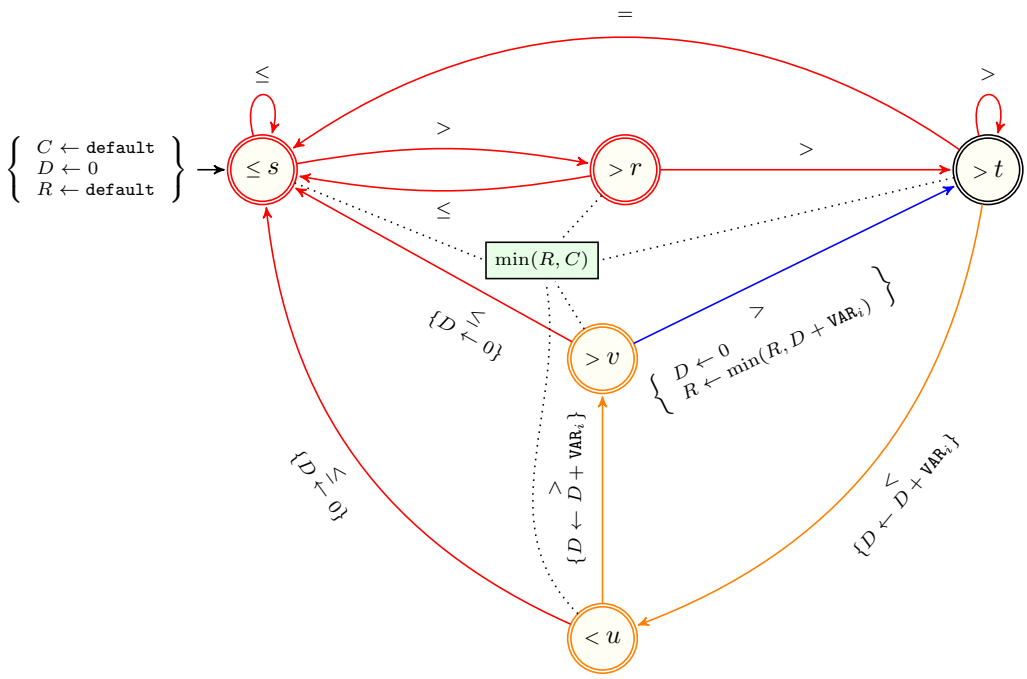


Figure 4.855: Automaton for the MIN_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$

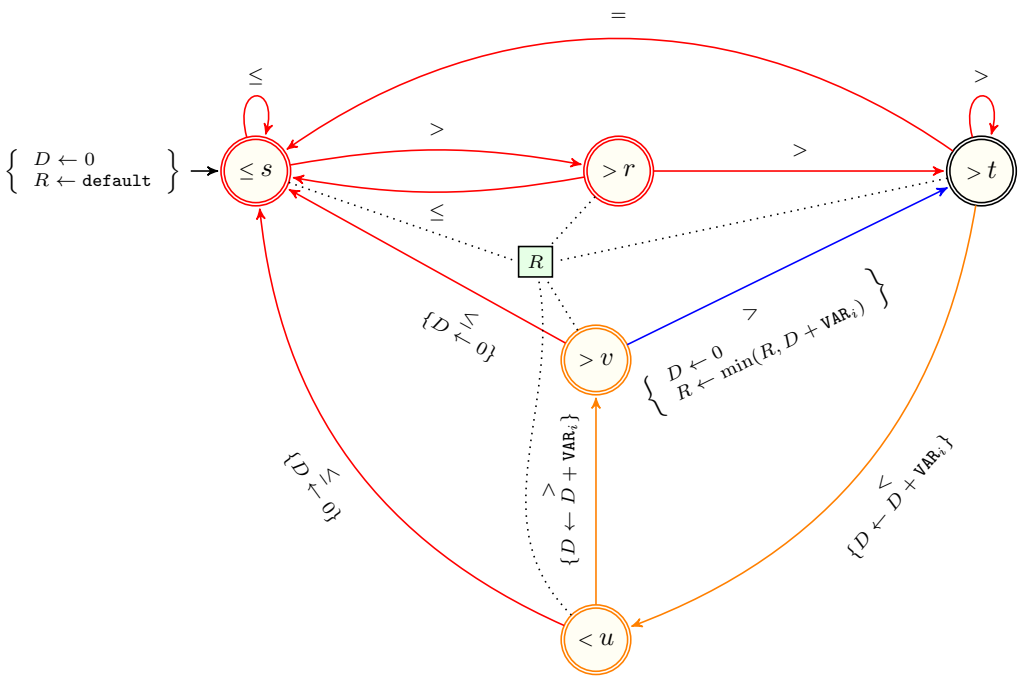


Figure 4.856: Simplified automaton for the MIN_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_SURF_DECREASING

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

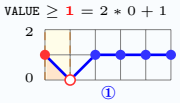
Constraint MIN_SURF_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 2 * \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal surface of occurrences of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(4, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.857 provides an example where the MIN_SURF_DECREASING $(4, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

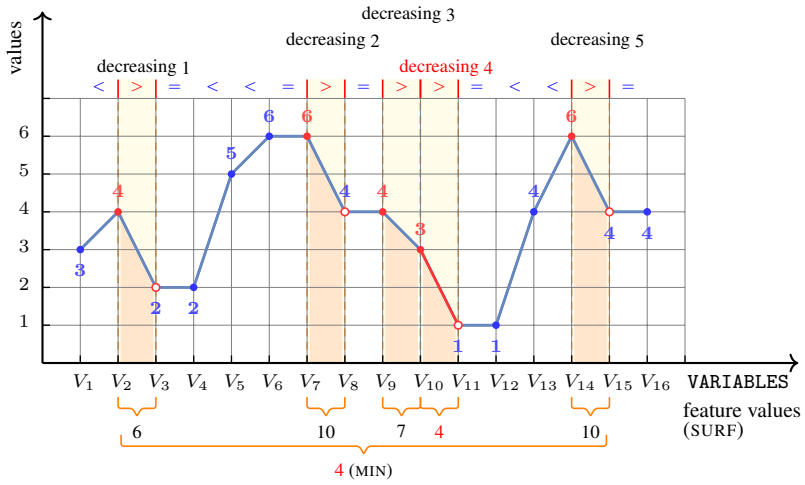


Figure 4.857: Illustrating the MIN_SURF_DECREASING constraint of the **Example** slot

Automaton

Figures 4.858 and 4.859 respectively depict the automaton associated with the constraint MIN_SURF_DECREASING and its simplified form.

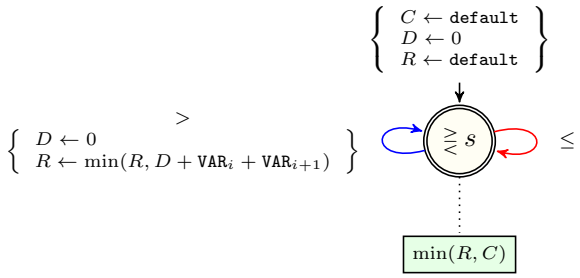


Figure 4.858: Automaton for the MIN_SURF_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is $+\infty$

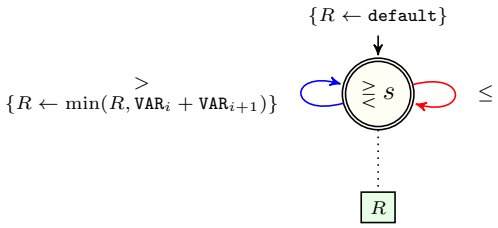


Figure 4.859: Simplified automaton for the MIN_SURF_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

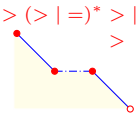
Table 4.169: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the MIN_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.170: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the MIN_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint MIN_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

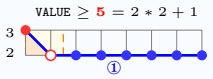
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \geq 2 * \mathtt{minv} + 1$①

$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \geq \min(2 * \mathtt{minv} + 1$②$, \mathtt{sv} * (\mathtt{minv} + 1)$③$)$



$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq 2 * \mathtt{maxv} - 1$

$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \max(2 * \mathtt{maxv} - 1, \mathtt{sv} * (\mathtt{maxv} - 1))$

$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$

$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{maxv} = 1) \Rightarrow$
$\mathtt{n1} \geq \mathtt{VALUE} - \max(0, 2 * \mathtt{maxv} - 3)$

$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - 1 - \mathtt{sv} * (\mathtt{maxv} - 2)$

$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{minv}, \mathtt{minv} + 1 \rangle)$

$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{minv} = -1) \Rightarrow$
$\mathtt{n2} \geq \min(0, 2 * \mathtt{minv} + 3) - \mathtt{VALUE}$

$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq \mathtt{sv} * (\mathtt{minv} + 2) - 1 - \mathtt{VALUE}$

$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$

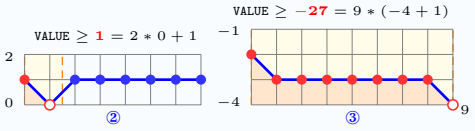
where

$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$

$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$

$\mathtt{sv} = |\mathtt{VARIABLES}|$

$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(6, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.860 provides an example where the MIN_SURF_DECREASING_SEQUENCE $(6, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

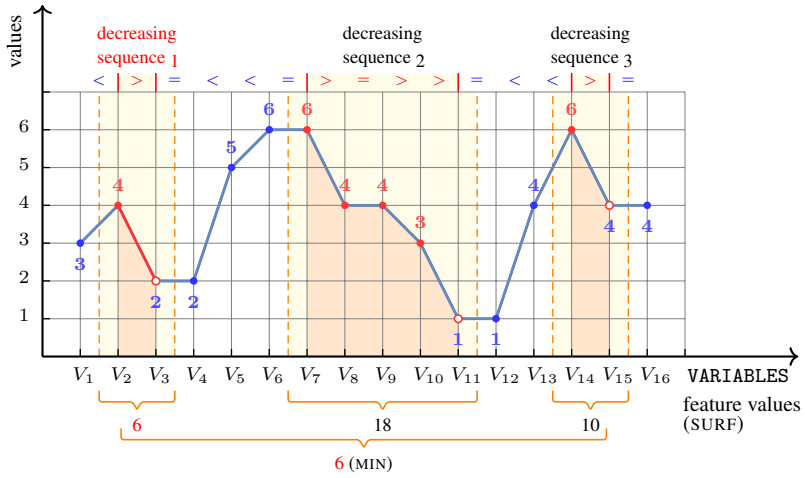


Figure 4.860: Illustrating the MIN_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figure 4.861 depicts the automaton associated with the constraint MIN_SURF_DECREASING_SEQUENCE.

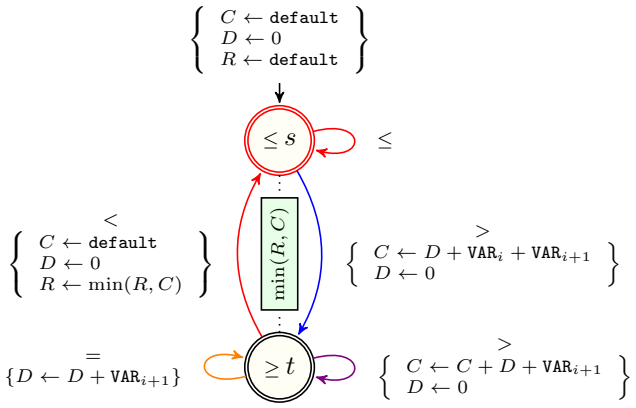


Figure 4.861: Automaton for the MIN_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

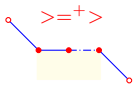
	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.171: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MIN_SURF_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_DECREASING_TERRACE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint MIN_SURF_DECREASING_TERRACE(VALUE, VARIABLES)

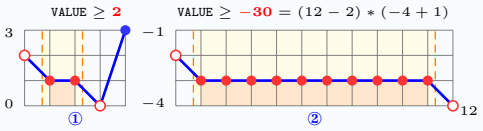
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1), (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(4, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3 \rangle)$

Figure 4.862 provides an example where the MIN_SURF_DECREASING_TERRACE $(4, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

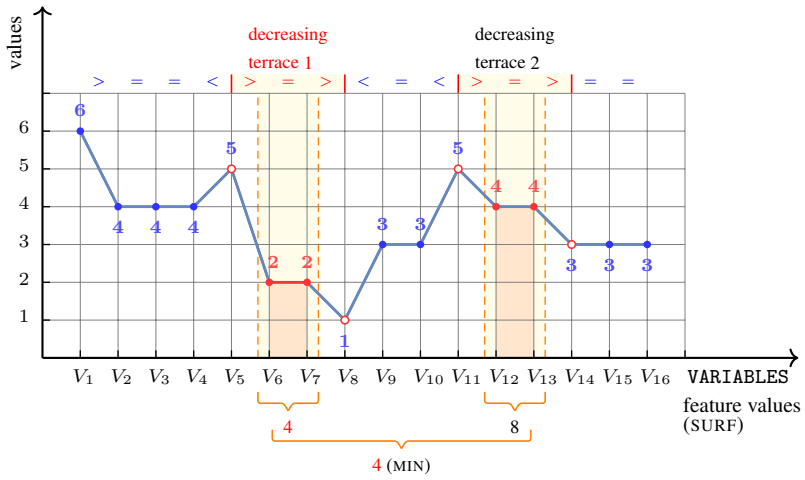


Figure 4.862: Illustrating the MIN_SURF_DECREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.863 and 4.864 respectively depict the automaton associated with the constraint MIN_SURF_DECREASING_TERRACE and its simplified form.

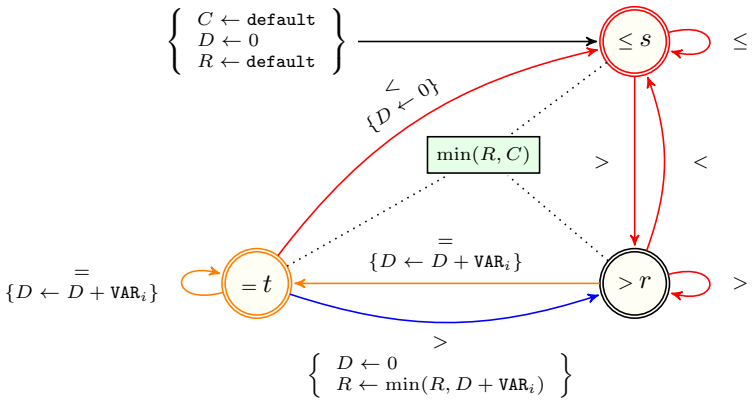


Figure 4.863: Automaton for the MIN_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $+\infty$

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.172: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MIN_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

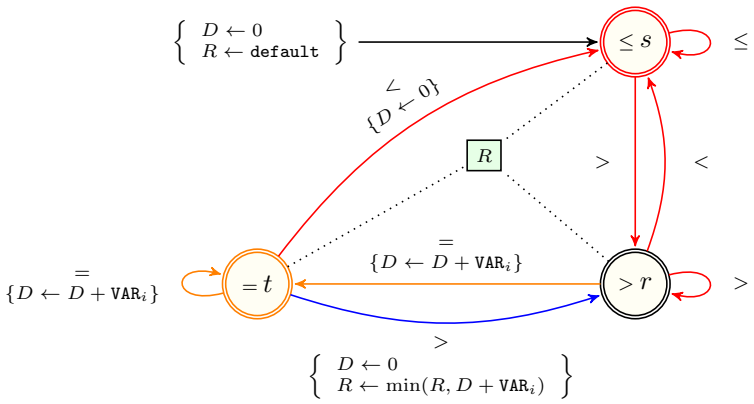


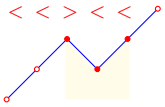
Figure 4.864: Simplified automaton for the MIN_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.173: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MIN_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

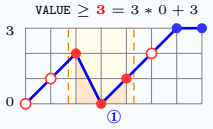
Constraint MIN_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 3 * \texttt{minv} + 3$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 3 * \texttt{maxv} - 3$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 3 - \texttt{max}(\texttt{maxv} - 3, 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(9, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle)$

Figure 4.865 provides an example where the MIN_SURF_DIP_ON_INCREASING_SEQUENCE $(9, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

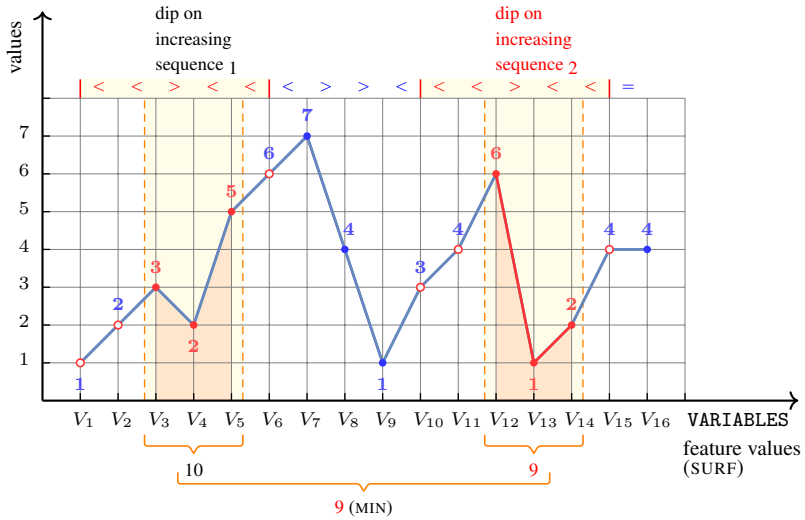


Figure 4.865: Illustrating the MIN_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.866 and 4.867 respectively depict the automaton associated with the constraint MIN_SURF_DIP_ON_INCREASING_SEQUENCE and its simplified form.

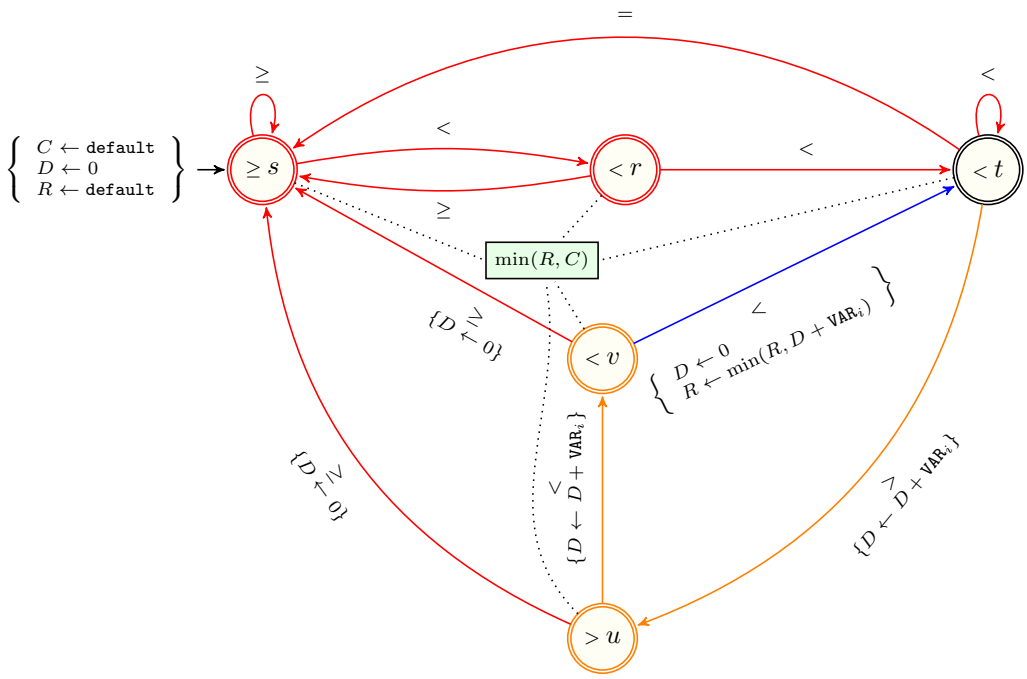


Figure 4.866: Automaton for the MIN_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$

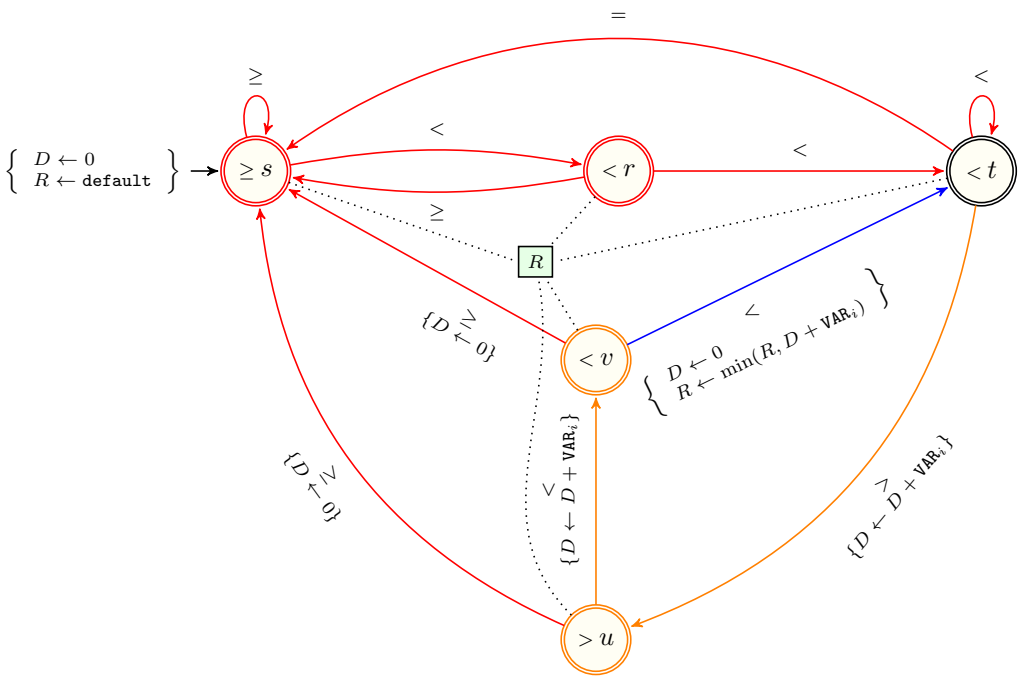
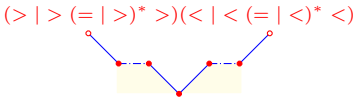


Figure 4.867: Simplified automaton for the MIN_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_SURF_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

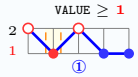
Constraint MIN_SURF_GORGE(VALUE, VARIABLES)

Arguments

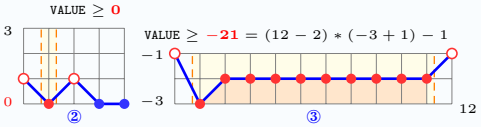
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \geq \texttt{minv}$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \geq \min(\texttt{minv}$②$, (\texttt{sv}-2) * (\texttt{minv}+1) - 1$③$)$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
$\texttt{rv} \geq 3 \Rightarrow$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}-1, (\texttt{sv}-2) * (\texttt{maxv}-1) - 1)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{maxv} = 1) \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{maxv}-2)$
$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - (\texttt{sv}-2) * (\texttt{maxv}-2)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}, \texttt{minv}+1 \rangle)$
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{minv} = -1) \Rightarrow \texttt{n2} \geq \min(0, \texttt{minv}+1) - \texttt{VALUE}$
$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow$
$\texttt{n2} \geq (\texttt{sv}-2) * (\texttt{minv}+2) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$



where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose

VALUE is the minimal surface of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(5, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.868 provides an example where the MIN_SURF_GORGE $(5, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

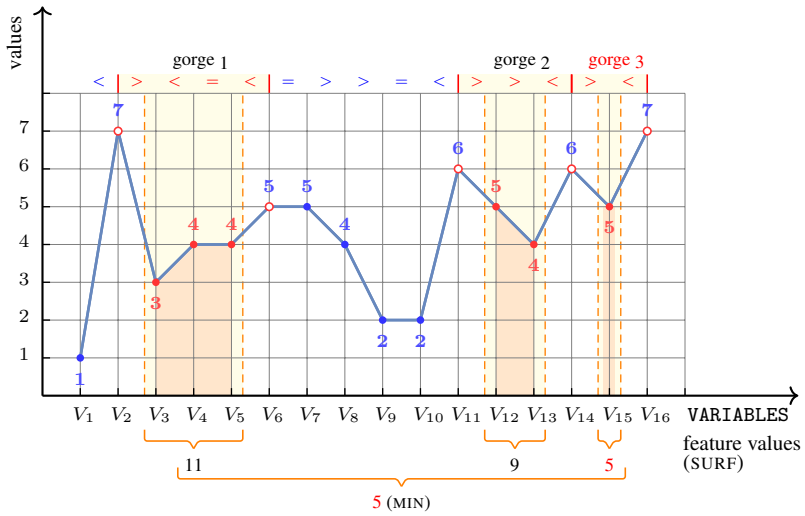


Figure 4.868: Illustrating the MIN_SURF_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.869 depicts the automaton associated with the constraint MIN_SURF_GORGE.

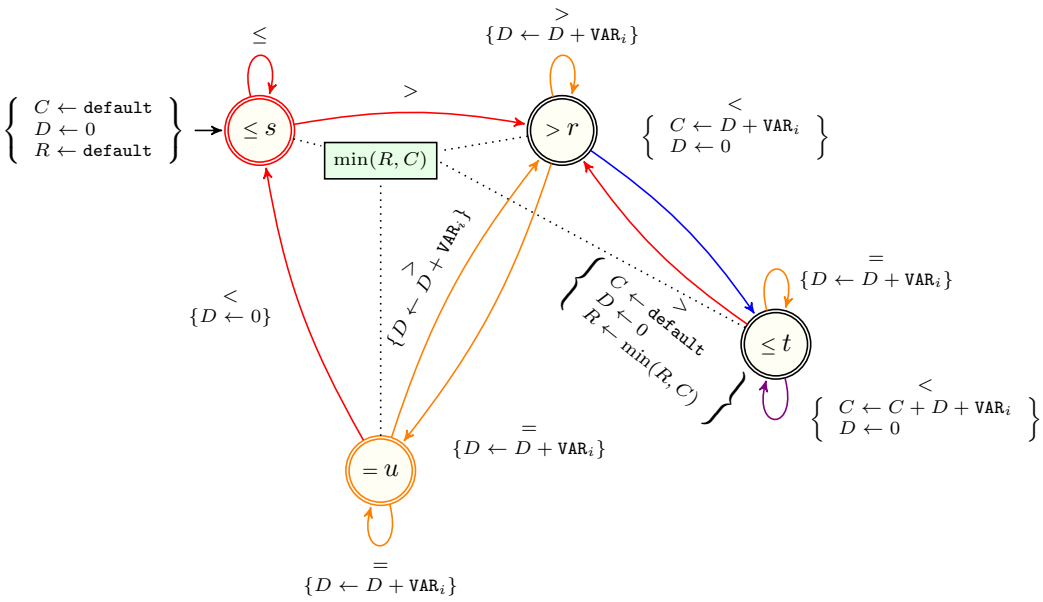


Figure 4.869: Automaton for the MIN_SURF_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is $+\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.174: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MIN_SURF_GORGE constraint defined as the composition of the GORGE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint MIN_SURF_INCREASING(VALUE, VARIABLES)

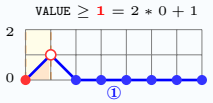
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 2 * \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
required(VARIABLES, var)

where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal surface of occurrences of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example $(4, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.870 provides an example where the MIN_SURF_INCREASING $(4, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Arg. properties Functional dependency: VALUE determined by VARIABLES.

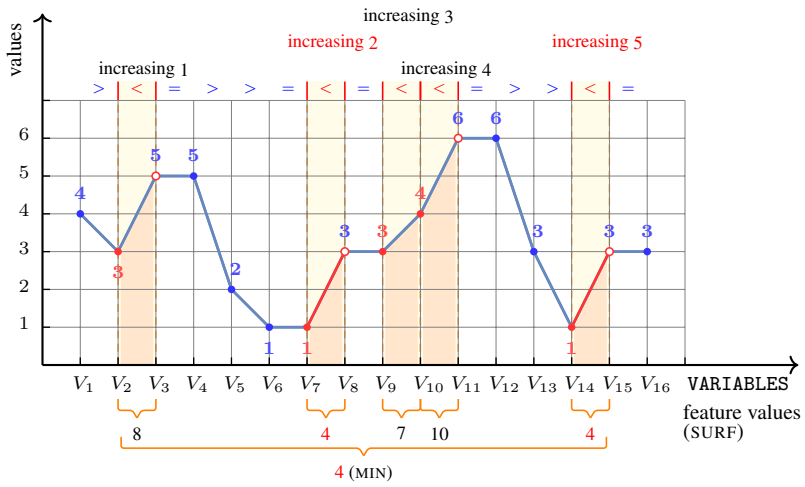


Figure 4.870: Illustrating the MIN_SURF_INCREASING constraint of the **Example** slot

Automaton

Figures 4.871 and 4.872 respectively depict the automaton associated with the constraint MIN_SURF_INCREASING and its simplified form.

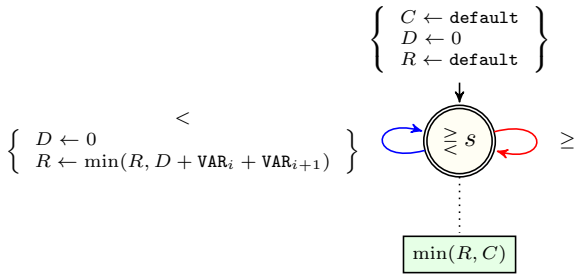


Figure 4.871: Automaton for the MIN_SURF_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is $+\infty$

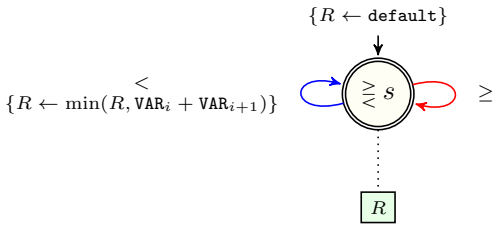


Figure 4.872: Simplified automaton for the MIN_SURF_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

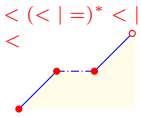
Table 4.175: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the MIN_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	$+\infty$

Table 4.176: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the MIN_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint MIN_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

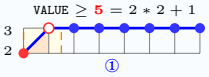
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$

$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$①

$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \geq \min(2 * \texttt{minv} + 1$②$, \texttt{sv} * (\texttt{minv} + 1)$③$)$



$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$

$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv} - 1, \texttt{sv} * (\texttt{maxv} - 1))$

$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$

$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{maxv} = 1) \Rightarrow$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, 2 * \texttt{maxv} - 3)$

$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 1 - \texttt{sv} * (\texttt{maxv} - 2)$

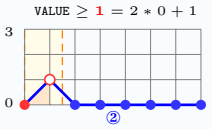
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$

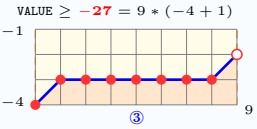
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{minv} = -1) \Rightarrow$
$\texttt{n2} \geq \min(0, 2 * \texttt{minv} + 3) - \texttt{VALUE}$

$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - 1 - \texttt{VALUE}$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$





Purpose VALUE is the minimal surface of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(4, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.873 provides an example where the MIN_SURF_INCREASING_SEQUENCE $(4, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

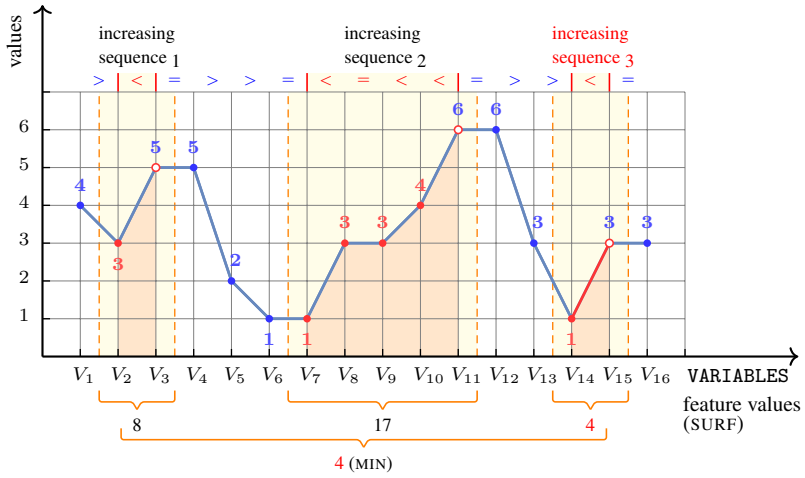


Figure 4.873: Illustrating the MIN_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton Figure 4.874 depicts the automaton associated with the constraint MIN_SURF_INCREASING_SEQUENCE.

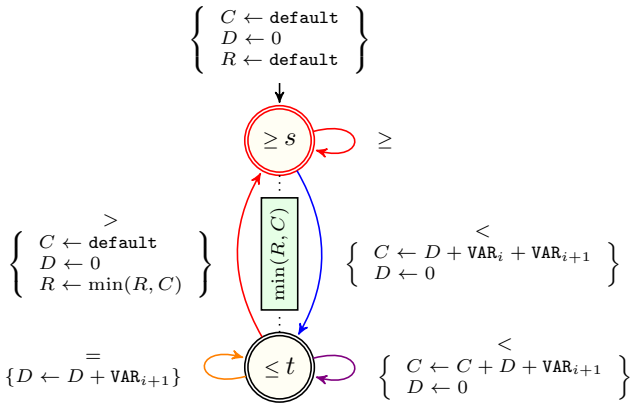


Figure 4.874: Automaton for the MIN_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

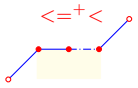
	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.177: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MIN_SURF_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint MIN_SURF_INCREASING_TERRACE(VALUE, VARIABLES)

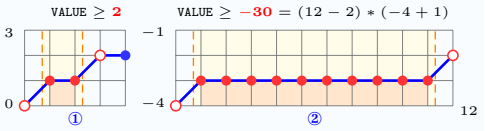
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1), (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(9, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.875 provides an example where the MIN_SURF_INCREASING_TERRACE $(9, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

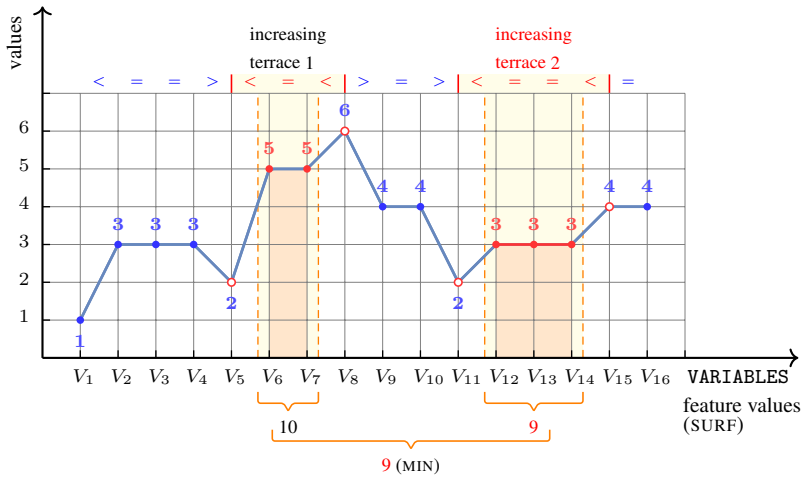


Figure 4.875: Illustrating the MIN_SURF_INCREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.876 and 4.877 respectively depict the automaton associated with the constraint MIN_SURF_INCREASING_TERRACE and its simplified form.

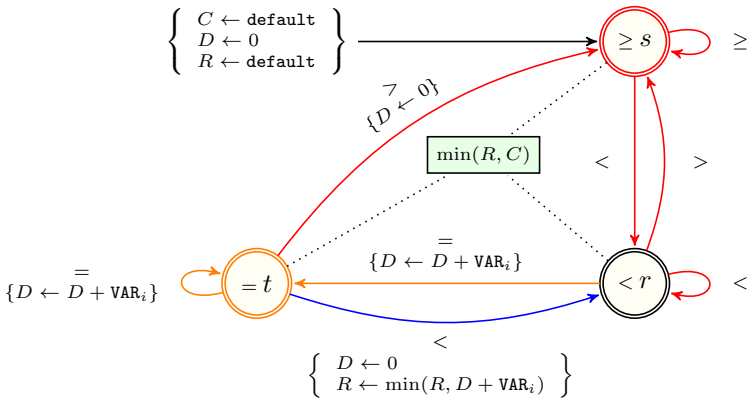


Figure 4.876: Automaton for the MIN_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where default is $+\infty$

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C

Table 4.178: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MIN_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

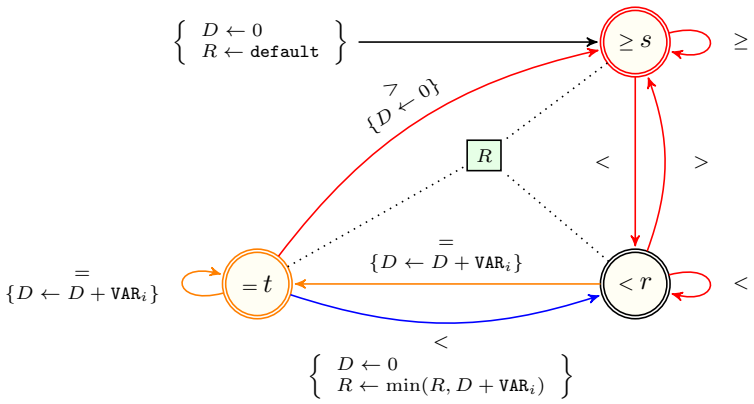


Figure 4.877: Simplified automaton for the MIN_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

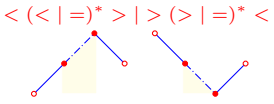
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.179: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MIN_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

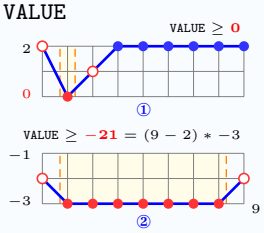
Constraint MIN_SURF_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(\texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}, (\texttt{sv}-2) * \texttt{maxv})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(1, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.878 provides an example where the MIN_SURF_INFLEXION $(1, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

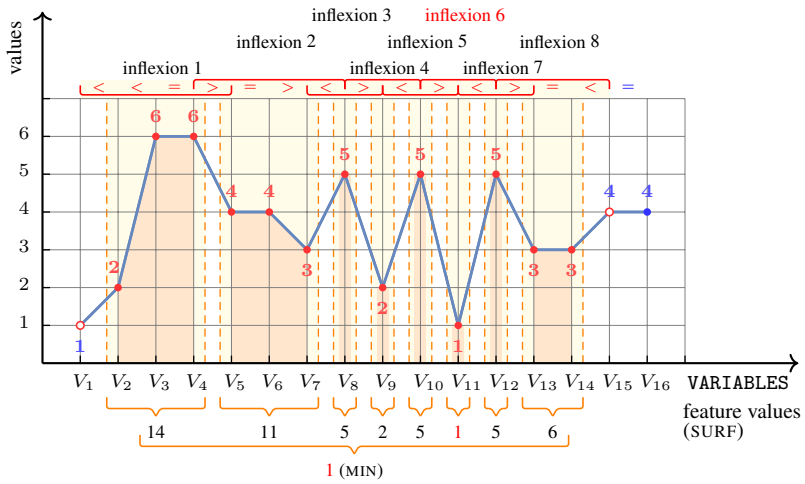


Figure 4.878: Illustrating the MIN_SURF_INFLEXION constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.879 and 4.880 respectively depict the automaton associated with the constraint MIN_SURF_INFLEXION and its simplified form.

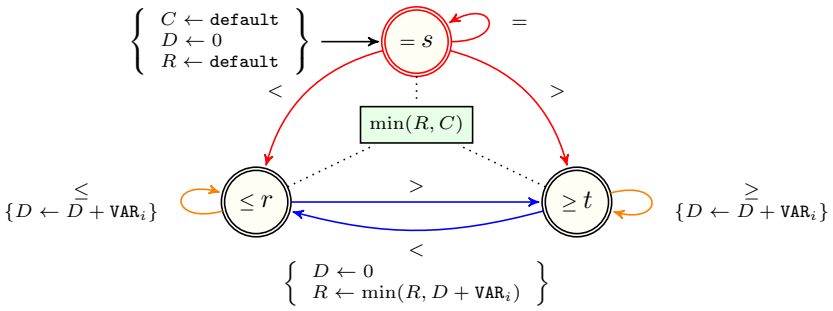


Figure 4.879: Automaton for the MIN_SURF_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

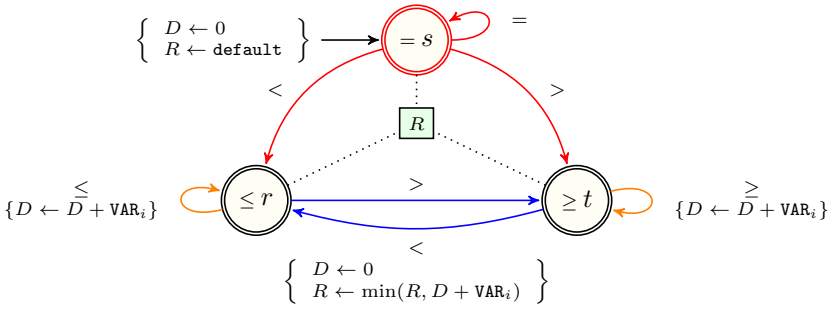
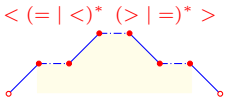


Figure 4.880: Simplified automaton for the MIN_SURF_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where default is $+\infty$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_SURF_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

Constraint MIN_SURF_PEAK(VALUE, VARIABLES)

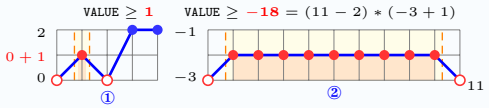
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, (\texttt{sv} - 2) * \texttt{maxv})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(9, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.881 provides an example where the MIN_SURF_PEAK $(9, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

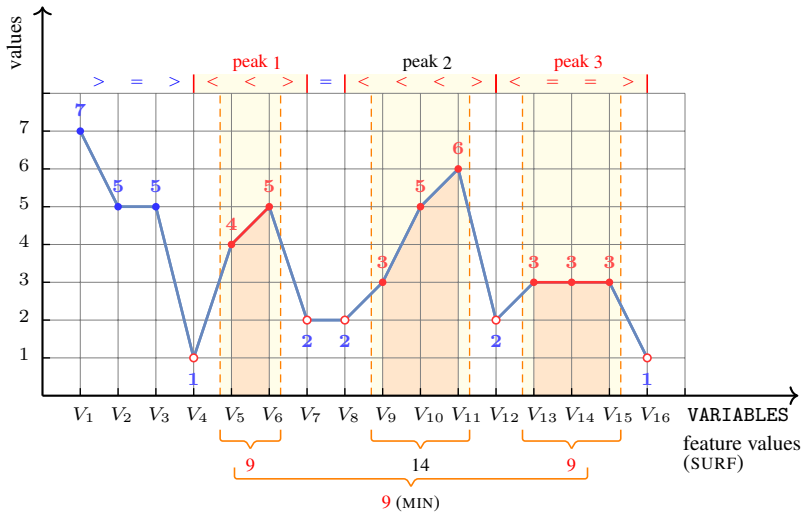


Figure 4.881: Illustrating the MIN_SURF_PEAK constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.882 depicts the automaton associated with the constraint MIN_SURF_PEAK.

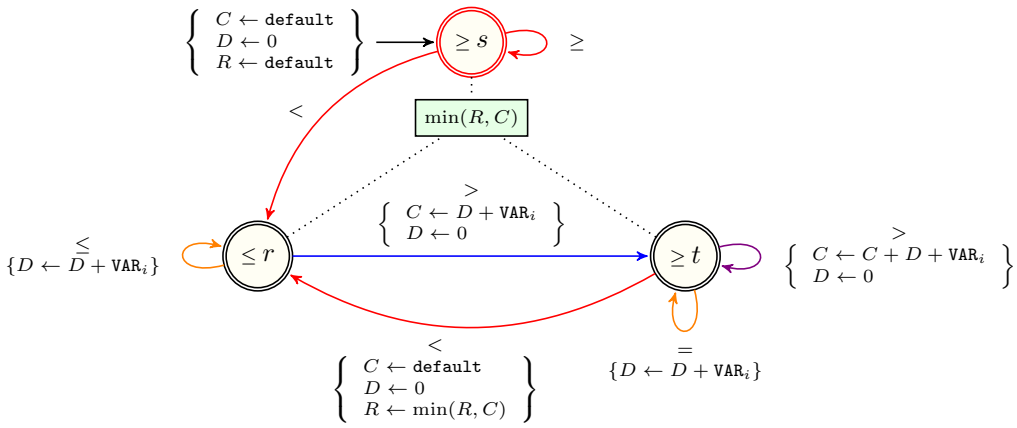


Figure 4.882: Automaton for the MIN_SURF_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.180: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MIN_SURF_PEAK constraint defined as the composition of the PEAK pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_PLAIN

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint MIN_SURF_PLAIN(VALUE, VARIABLES)

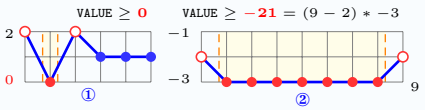
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}-1, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(4, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3 \rangle)$

Figure 4.883 provides an example where the MIN_SURF_PLAIN $(4, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

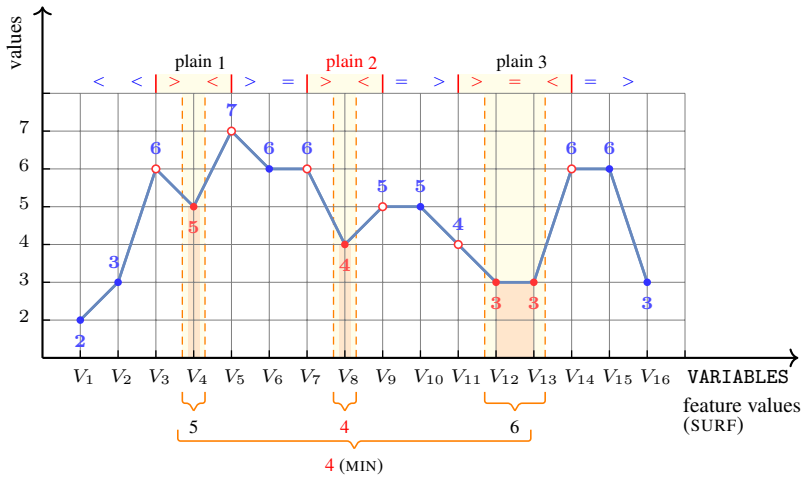


Figure 4.883: Illustrating the MIN_SURF_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.884 and 4.885 respectively depict the automaton associated with the constraint MIN_SURF_PLAIN and its simplified form.

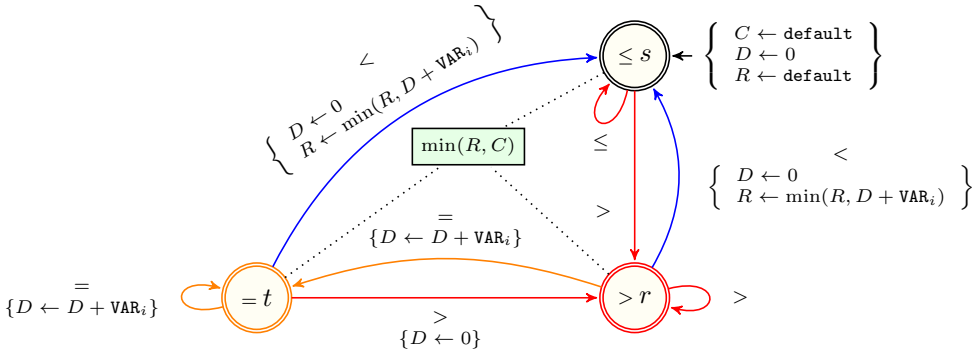


Figure 4.884: Automaton for the MIN_SURF_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is $+\infty$

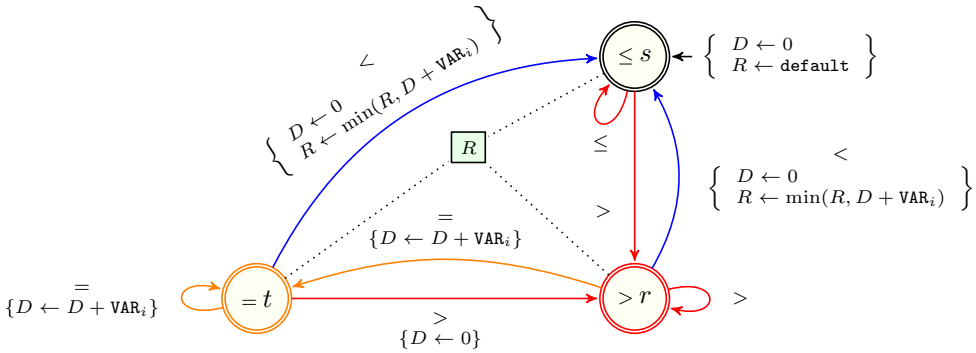


Figure 4.885: Simplified automaton for the MIN_SURF_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLAIN pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.181: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MIN_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.182: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MIN_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_PLATEAU

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint MIN_SURF_PLATEAU(VALUE, VARIABLES)

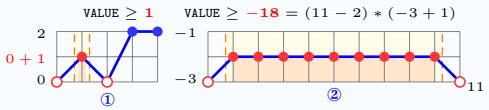
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, (\texttt{sv} - 2) * \texttt{maxv})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(3, \langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5 \rangle)$

Figure 4.886 provides an example where the MIN_SURF_PLATEAU $(3, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

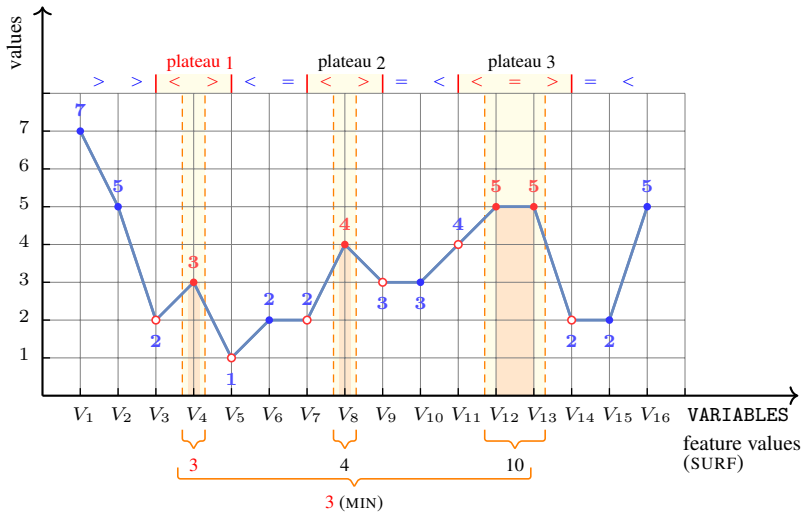


Figure 4.886: Illustrating the MIN_SURF_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.887 and 4.888 respectively depict the automaton associated with the constraint MIN_SURF_PLATEAU and its simplified form.

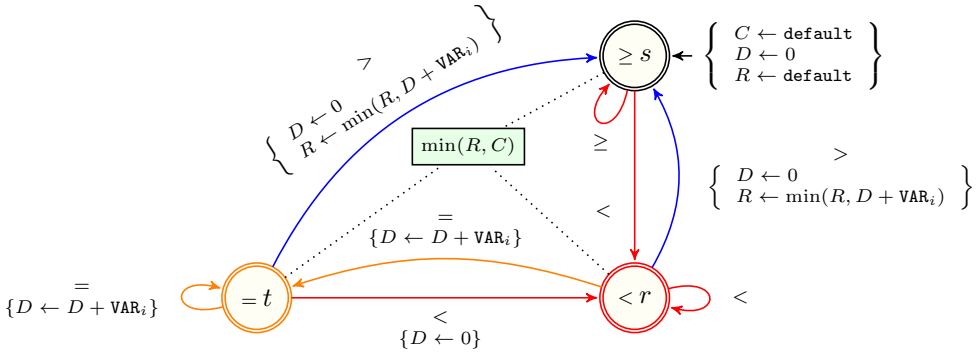


Figure 4.887: Automaton for the MIN_SURF_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is $+\infty$

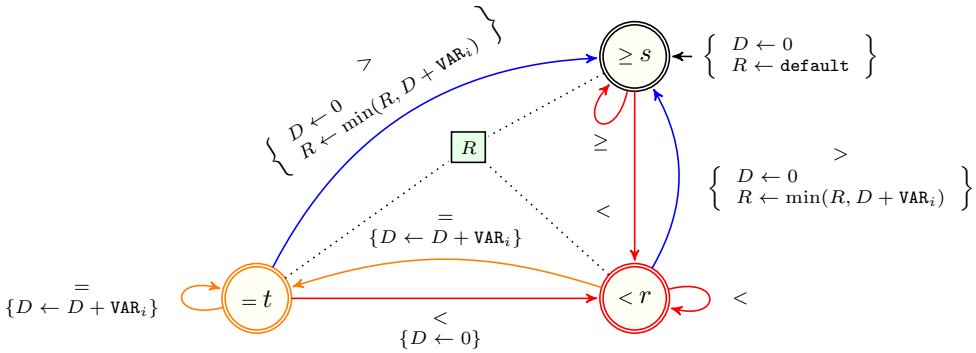


Figure 4.888: Simplified automaton for the MIN_SURF_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLATEAU pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.183: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MIN_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

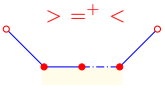
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.184: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MIN_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint MIN_SURF_PROPER_PLAIN(VALUE, VARIABLES)

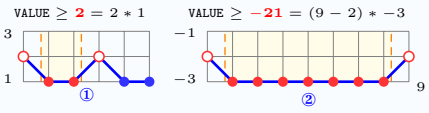
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(2 * \texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv}-1), (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(8, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5 \rangle)$

Figure 4.889 provides an example where the MIN_SURF_PROPER_PLAIN $(8, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

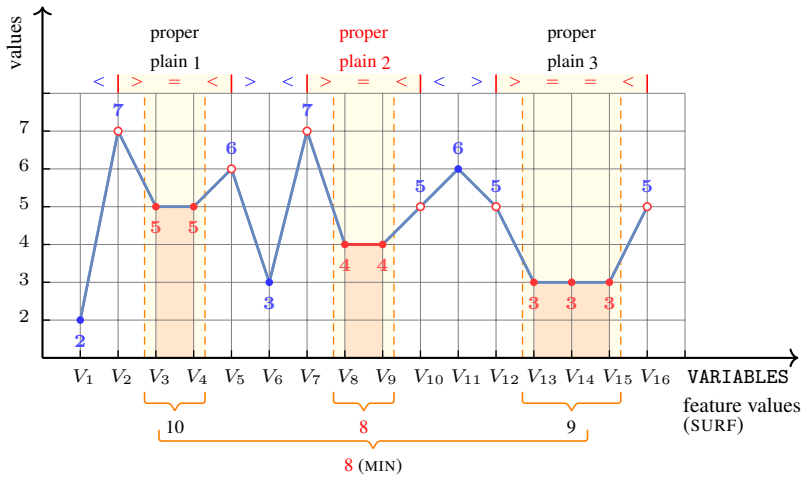


Figure 4.889: Illustrating the MIN_SURF_PROPER_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.890 and 4.891 respectively depict the automaton associated with the constraint MIN_SURF_PROPER_PLAIN and its simplified form.

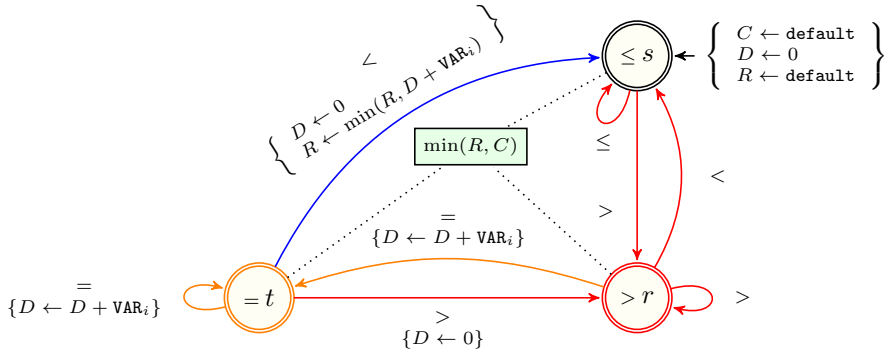


Figure 4.890: Automaton for the MIN_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is $+\infty$

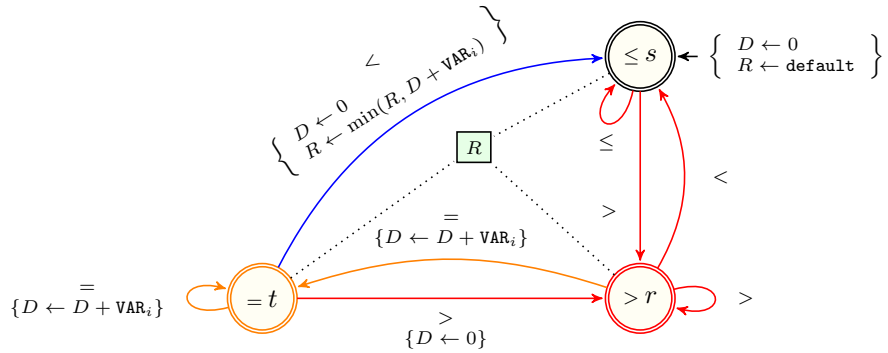


Figure 4.891: Simplified automaton for the MIN_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLAIN pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.185: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MIN_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

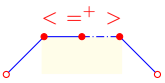
	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.186: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MIN_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_PROPER_PLATEAU

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

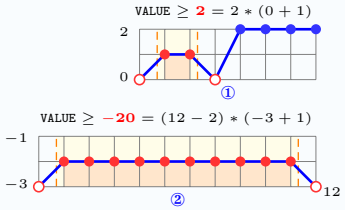
Constraint MIN_SURF_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv}, (\texttt{sv} - 2) * \texttt{maxv})$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
among(n1, VARIABLES[2, sv − 1], ⟨maxv⟩)
among(n2, VARIABLES[2, sv − 1], ⟨minv + 1⟩)
required(VARIABLES, var)
where
minv =minval(VARIABLES.var)
sv = |VARIABLES|
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the minimal surface of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(6, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3\rangle)$

Figure 4.892 provides an example where the MIN_SURF_PROPER_PLATEAU $(6, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
|VARIABLES| > 3
range(VARIABLES.var) > 1

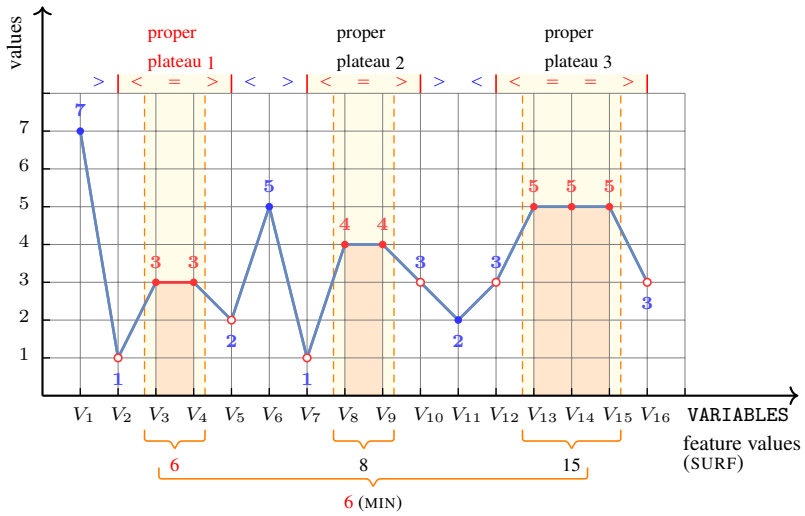


Figure 4.892: Illustrating the MIN_SURF_PROPER_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.893 and 4.894 respectively depict the automaton associated with the constraint MIN_SURF_PROPER_PLATEAU and its simplified form.

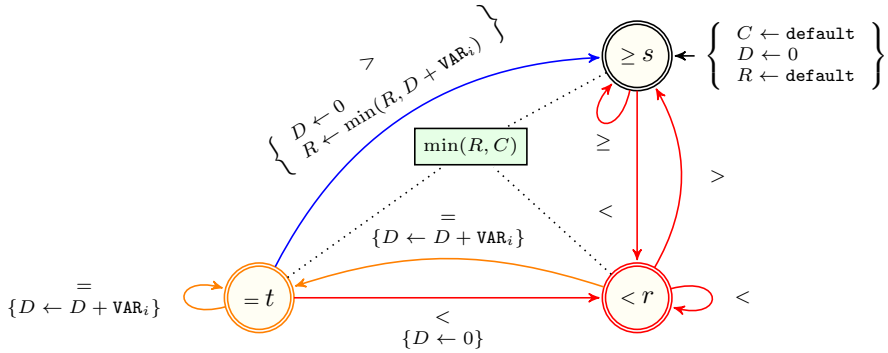


Figure 4.893: Automaton for the MIN_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $+\infty$

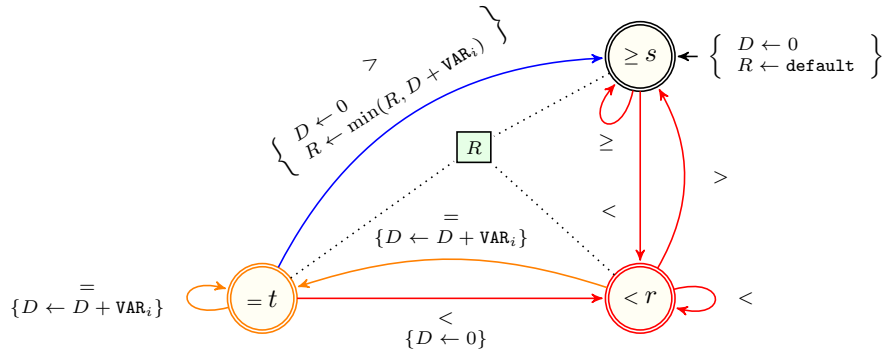


Figure 4.894: Simplified automaton for the MIN_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.187: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MIN_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$+\infty$	$+\infty$	$+\infty$
r	$+\infty$	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$+\infty$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.188: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MIN_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

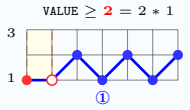
Constraint MIN_SURF_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 2 * \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv}$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 2 * (\texttt{maxv} - 1)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq 2 * (\texttt{minv} + 1) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the minimal surface of occurrences of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(2, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6 \rangle)$

Figure 4.895 provides an example where the MIN_SURF_STEADY $(2, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

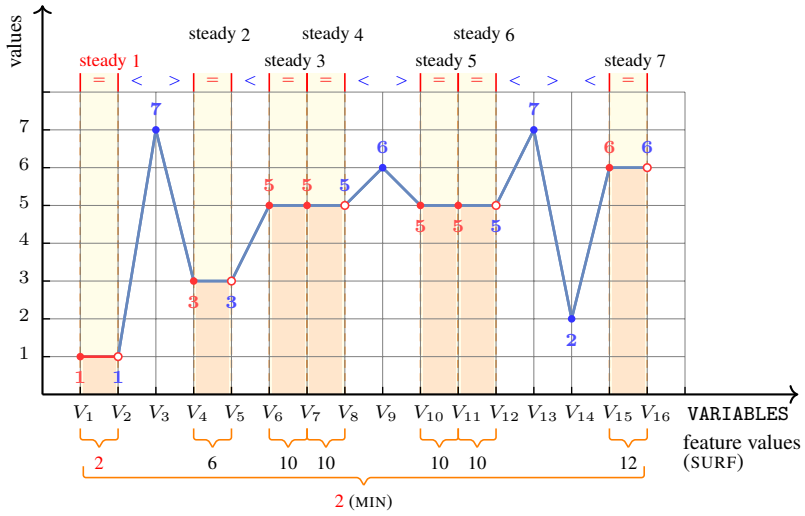


Figure 4.895: Illustrating the MIN_SURF_STEADY constraint of the **Example** slot

Automaton

Figures 4.896 and 4.897 respectively depict the automaton associated with the constraint MIN_SURF_STEADY and its simplified form.

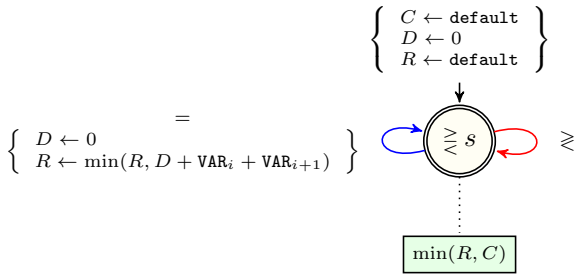


Figure 4.896: Automaton for the MIN_SURF_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is $+\infty$

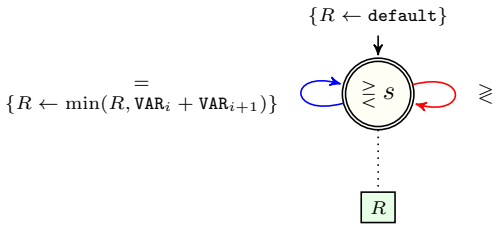


Figure 4.897: Simplified automaton for the MIN_SURF_STEADY constraint obtained by applying decoration Table 3.40 to the seed transducer of the STEADY pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s
s	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.189: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the MIN_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

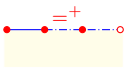
	s
s	$+\infty$

Table 4.190: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the MIN_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_STEADY_SEQUENCE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the STEADY_SEQUENCE pattern.

Constraint MIN_SURF_STEADY_SEQUENCE(VALUE, VARIABLES)

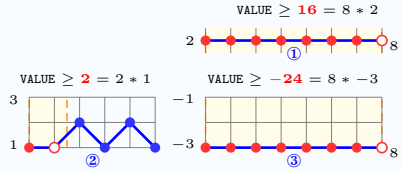
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv} * \texttt{minv}①$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}②, \texttt{sv} * \texttt{minv}③)$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{sv} * \texttt{maxv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}, \texttt{sv} * \texttt{maxv})$
among(n1, VARIABLES[1, sv], ⟨maxv⟩)
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{sv} * (\texttt{maxv} - 1))$
among(n2, VARIABLES[2, sv], ⟨minv⟩)
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{sv} * (\texttt{minv} + 1)) - \texttt{VALUE}$
required(VARIABLES, var)

where
minv =minval(VARIABLES.var)
rv =range(VARIABLES.var)
sv = |VARIABLES|
maxv =maxval(VARIABLES.var)



Purpose VALUE is the minimal surface of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(2, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.898 provides an example where the MIN_SURF_STEADY_SEQUENCE $(2, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical |VARIABLES| > 1

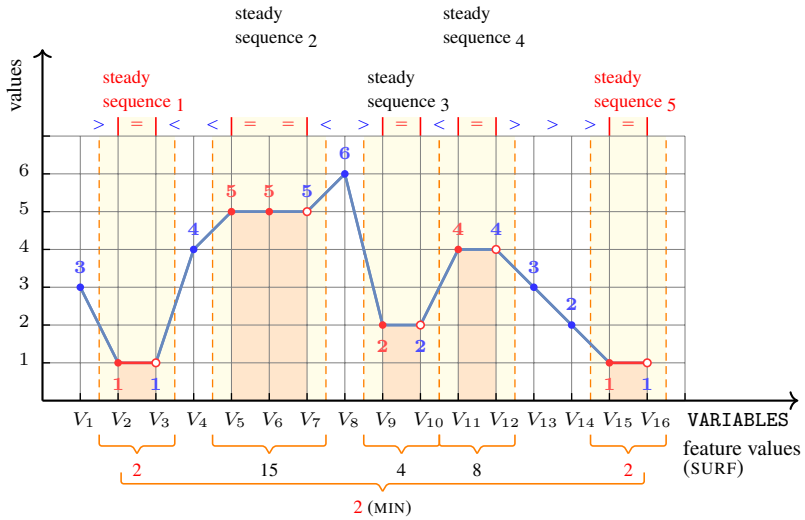


Figure 4.898: Illustrating the MIN_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.899 and 4.900 respectively depict the automaton associated with the constraint MIN_SURF_STEADY_SEQUENCE and its simplified form.

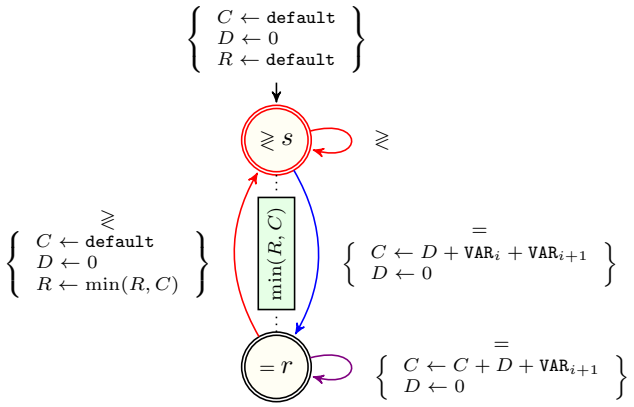


Figure 4.899: Automaton for the MIN_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $+\infty$

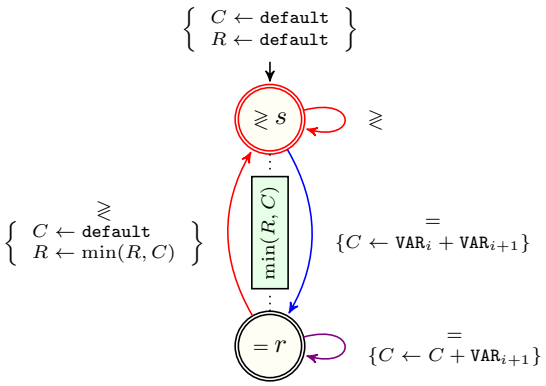


Figure 4.900: Simplified automaton for the MIN_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.191: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MIN_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \texttt{VAR}_{i+1}$ M

Table 4.192: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MIN_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint MIN_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

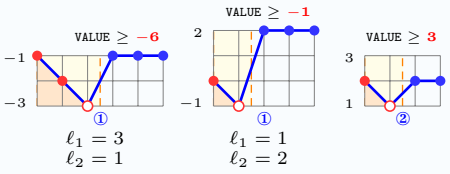
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$①
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$②
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose

VALUE is the minimal surface of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.

An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.

Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(7, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.901 provides an example where the MIN_SURF_STRICTLY_DECREASING_SEQUENCE $(7, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

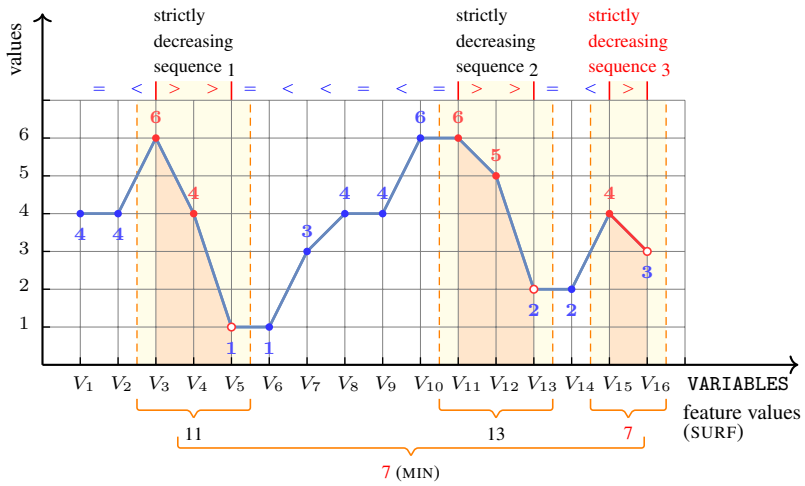


Figure 4.901: Illustrating the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.902 and 4.903 respectively depict the automaton associated with the constraint MIN_SURF_STRICTLY_DECREASING_SEQUENCE and its simplified form.

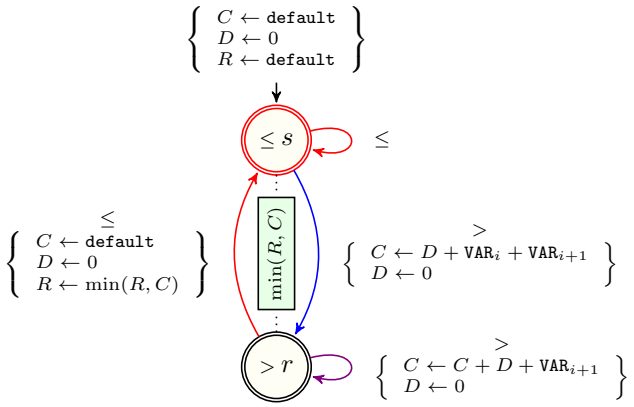


Figure 4.902: Automaton for the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$

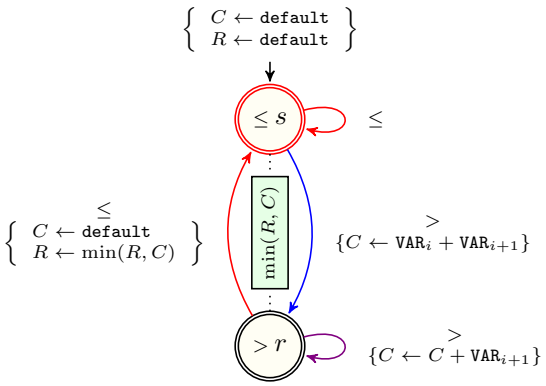


Figure 4.903: Simplified automaton for the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \mathtt{VAR}_{i+1}$ M

Table 4.193: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \mathtt{VAR}_{i+1}$ M

Table 4.194: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_STRICTLY_INCREASING_SEQUENCE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint MIN_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

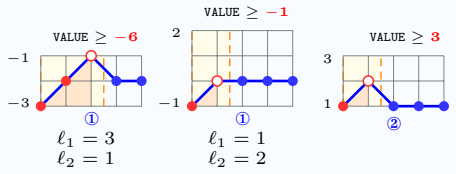
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$①
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$②
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.904 provides an example where the MIN_SURF_STRICTLY_INCREASING_SEQUENCE $(6, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

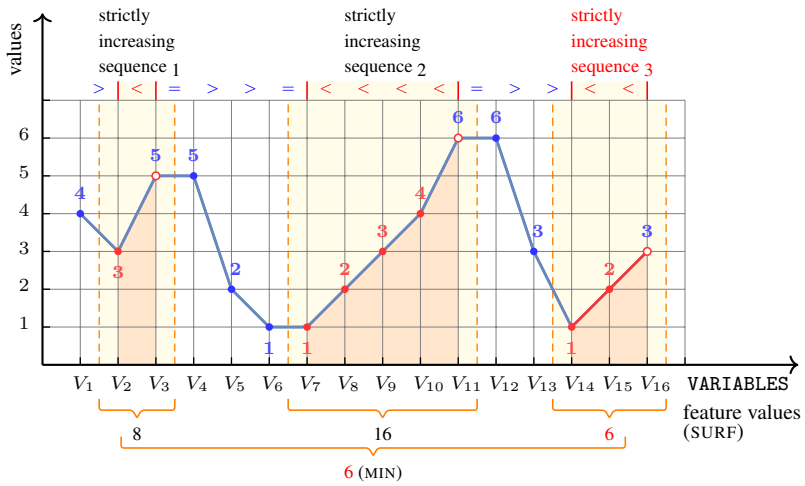


Figure 4.904: Illustrating the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.905 and 4.906 respectively depict the automaton associated with the constraint MIN_SURF_STRICTLY_INCREASING_SEQUENCE and its simplified form.

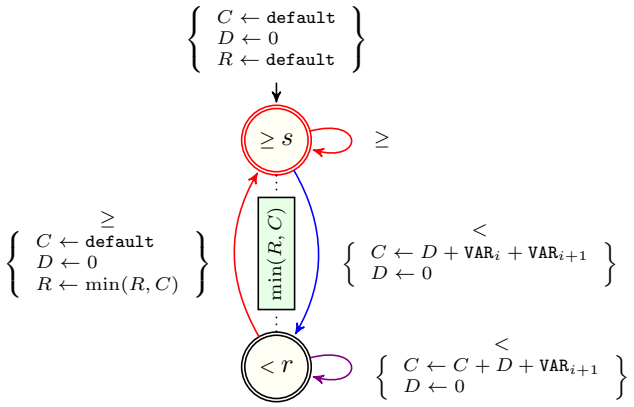


Figure 4.905: Automaton for the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$

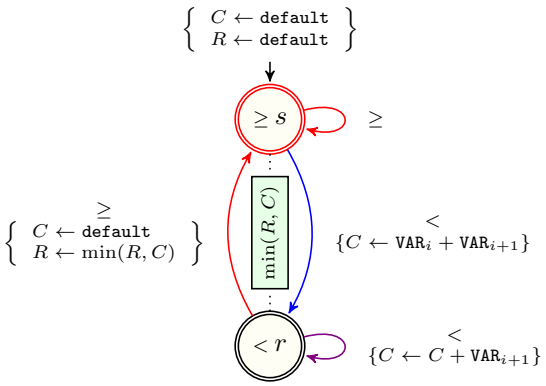


Figure 4.906: Simplified automaton for the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.195: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

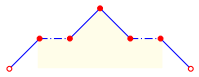
	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - \texttt{VAR}_{i+1}$ M

Table 4.196: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint MIN_SURF_SUMMIT(VALUE, VARIABLES)

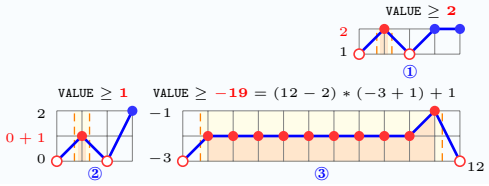
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \geq \min(\texttt{minv} + 1$①$, (\texttt{sv} - 2) * (\texttt{minv} + 1) + 1$②$)$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}, (\texttt{sv} - 2) * (\texttt{maxv} - 1) + 1)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{maxv} = 1) \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{maxv} - 1)$
$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow$
$\texttt{n1} \geq \texttt{VALUE} - (\texttt{sv} - 2) * (\texttt{maxv} - 2) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{minv} = -1) \Rightarrow \texttt{n2} \geq \min(0, \texttt{minv} + 2) - \texttt{VALUE}$
$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq (\texttt{sv} - 2) * (\texttt{minv} + 2) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose VALUE is the minimal surface of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(3, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.907 provides an example where the MIN_SURF_SUMMIT $(3, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

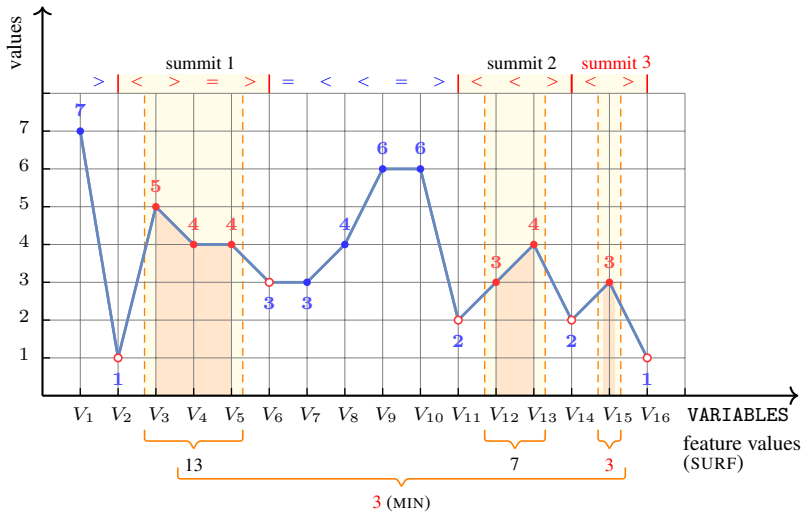


Figure 4.907: Illustrating the MIN_SURF_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.908 depicts the automaton associated with the constraint MIN_SURF_SUMMIT.

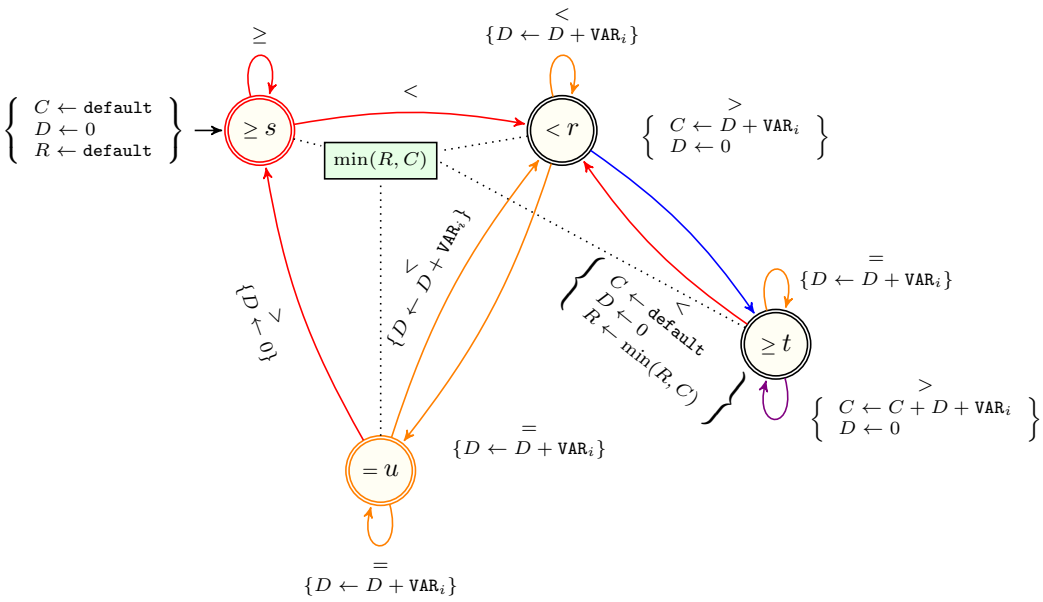


Figure 4.908: Automaton for the MIN_SURF_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is $+\infty$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

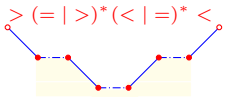
	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.197: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MIN_SURF_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint MIN_SURF_VALLEY(VALUE, VARIABLES)

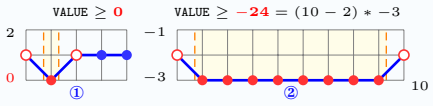
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}-1, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the minimal surface of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(7, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.909 provides an example where the MIN_SURF_VALLEY $(7, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

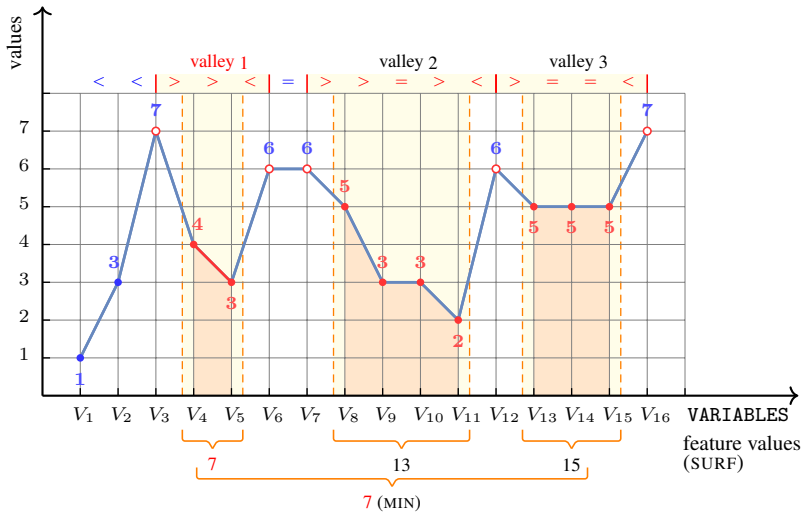


Figure 4.909: Illustrating the MIN_SURF_VALLEY constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figure 4.910 depicts the automaton associated with the constraint MIN_SURF_VALLEY.

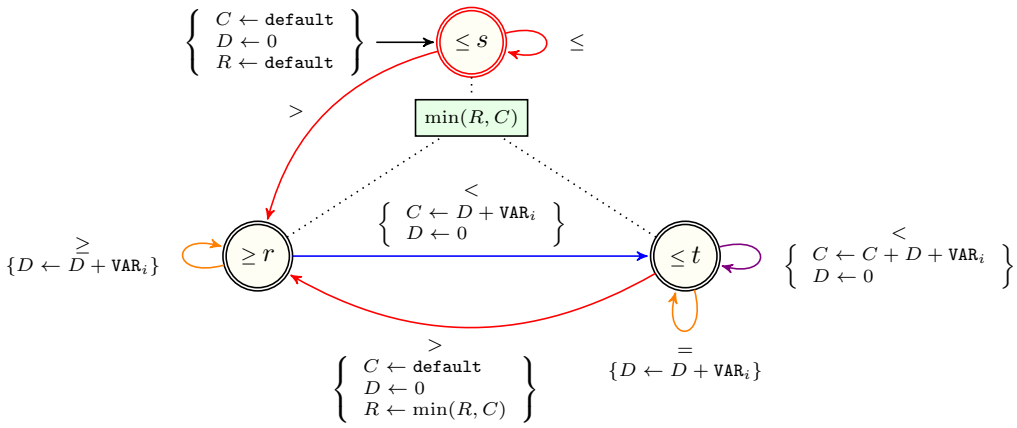


Figure 4.910: Automaton for the MIN_SURF_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is $+\infty$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

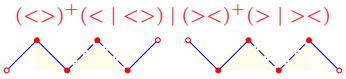
	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.198: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MIN_SURF_VALLEY constraint defined as the composition of the VALLEY pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_SURF_ZIGZAG

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint MIN_SURF_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$$

$$\texttt{VALUE} \geq \min \left(\begin{array}{l} 2 * \texttt{minv} + 1 ①, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{minv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{minv}+1) ② \end{array} \right)$$

$$\bigvee \left(\begin{array}{l} \texttt{VALUE} = +\infty, \\ \texttt{VALUE} \leq \max \left(\begin{array}{l} 2 * \texttt{maxv} - 1, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{maxv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{maxv}-1) \end{array} \right) \end{array} \right)$$

among(n1, VARIABLES[2, sv − 1], ⟨maxv − 1, maxv⟩)

$$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \lfloor (\texttt{sv}-1)/2 \rfloor - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$$

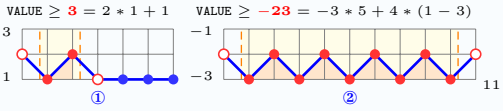
among(n2, VARIABLES[2, sv − 1], ⟨minv, minv + 1⟩)

$$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \lfloor (\texttt{sv}-1)/2 \rfloor - \texttt{VALUE}$$

required(VARIABLES, var)

where

minv = minval(VARIABLES.var)
sv = |VARIABLES|
maxv = maxval(VARIABLES.var)
rv = range(VARIABLES.var)



Purpose VALUE is the minimal surface of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $+\infty$.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(5, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.911 provides an example where the MIN_SURF_ZIGZAG $(5, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

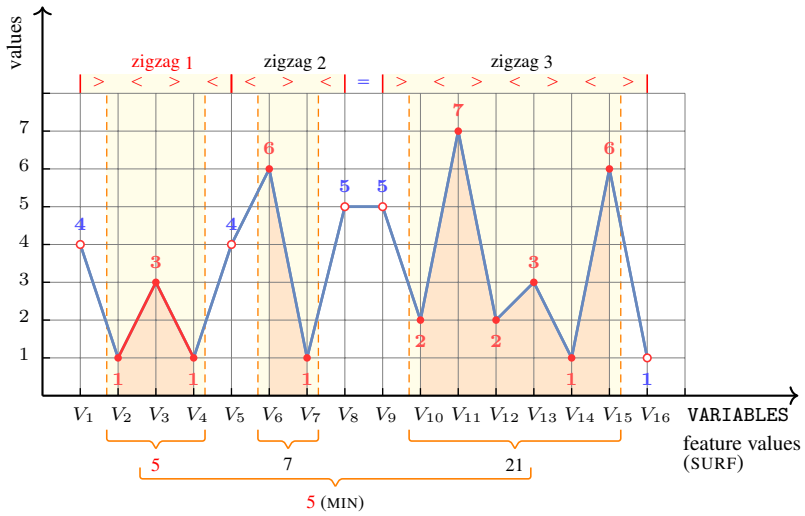


Figure 4.911: Illustrating the MIN_SURF_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.912 and 4.913 respectively depict the automaton associated with the constraint MIN_SURF_ZIGZAG and its simplified form.

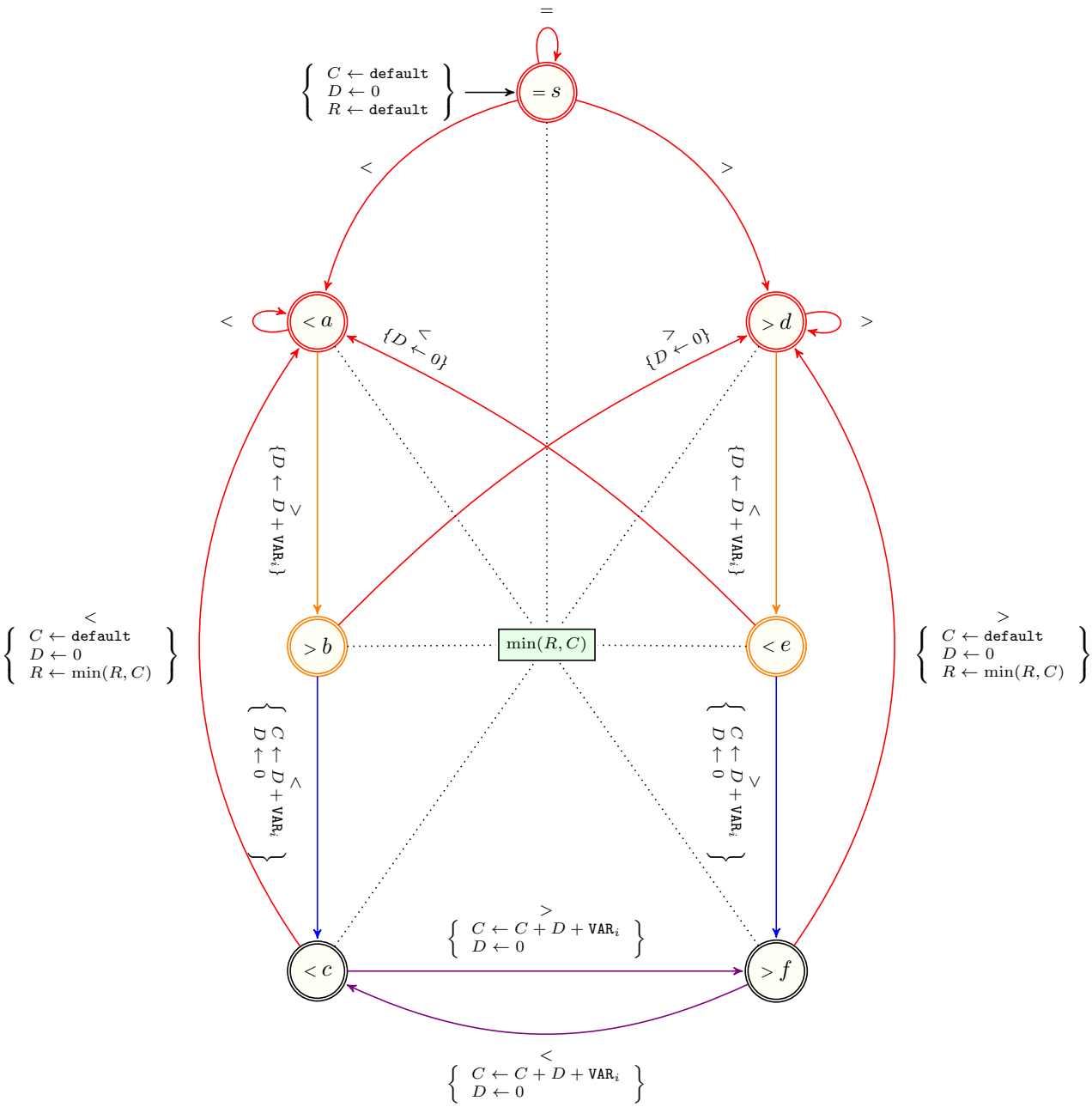


Figure 4.912: Automaton for the MIN_SURF_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $+\infty$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

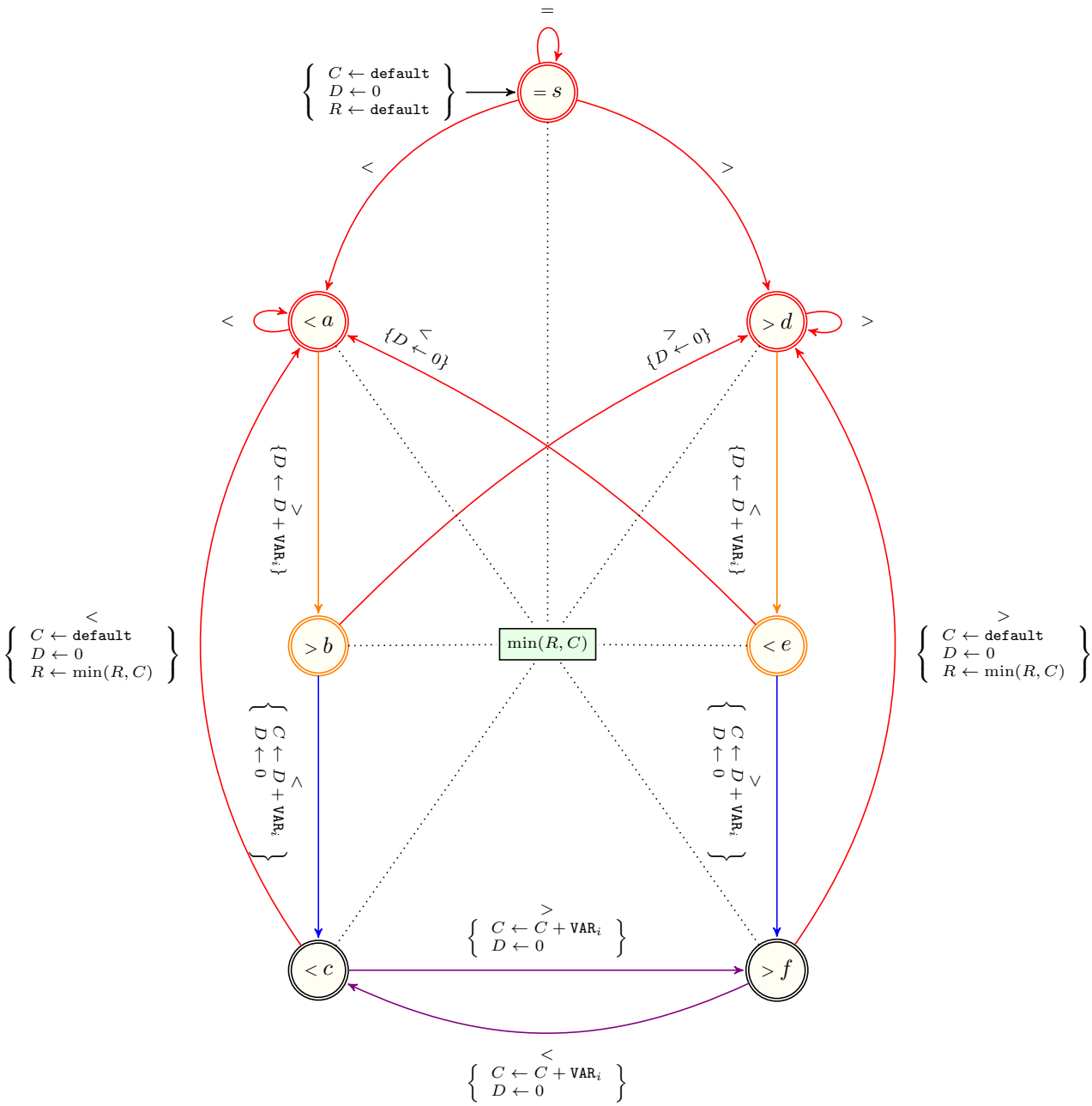


Figure 4.913: Simplified automaton for the MIN_SURF_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is $+\infty$; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

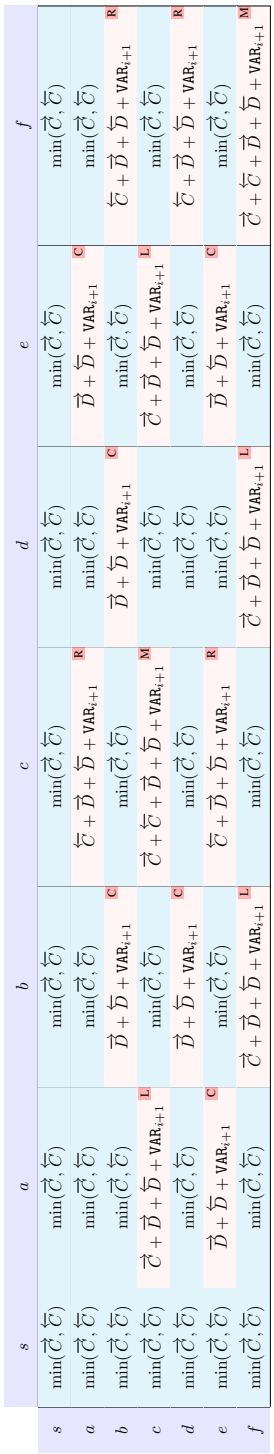


Table 4.199: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MIN_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

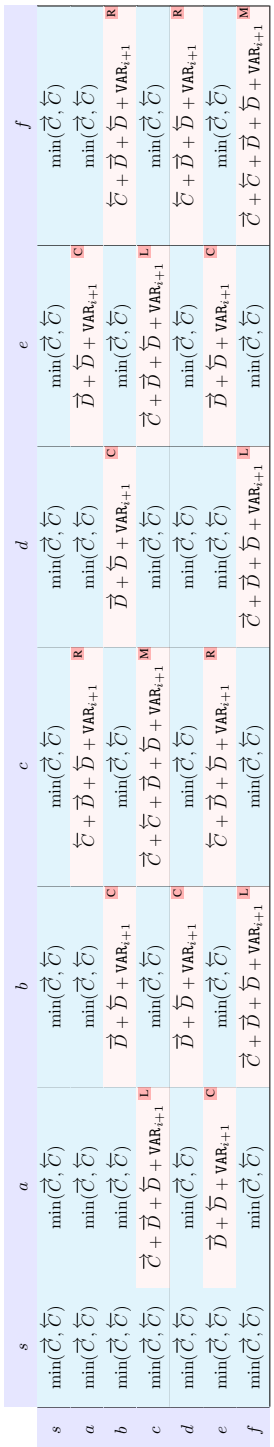
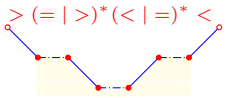


Table 4.200: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MIN_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature SURF, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MIN_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint MIN_VALLEY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of VALLEY is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of VALLEY.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> \ (= \mid >)^* (< \mid =)^* \ <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.914 provides an example where the MIN_VALLEY $([1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7],[0,0,0,0,3,0,0,0,0,0,2,0,0,0,5,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

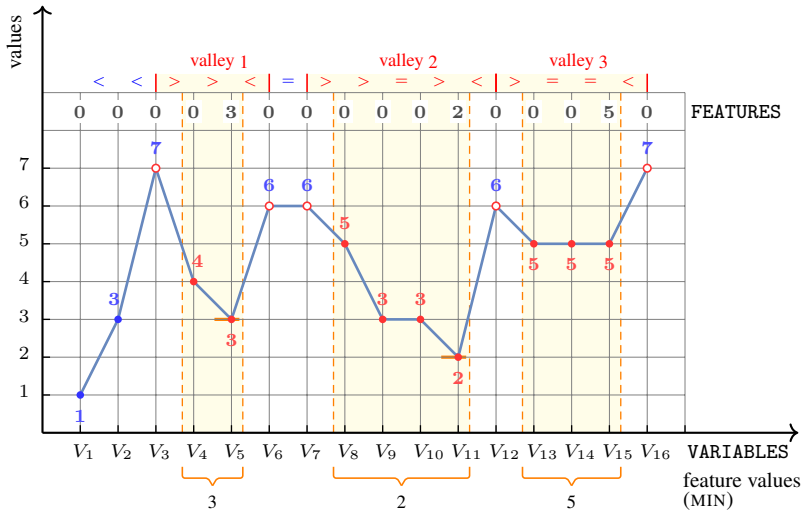
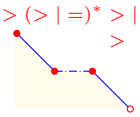


Figure 4.914: Illustrating the MIN_VALLEY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

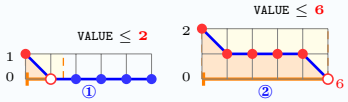
Constraint MIN_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the minimal width of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(2, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.915 provides an example where the MIN_WIDTH_DECREASING_SEQUENCE $(2, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

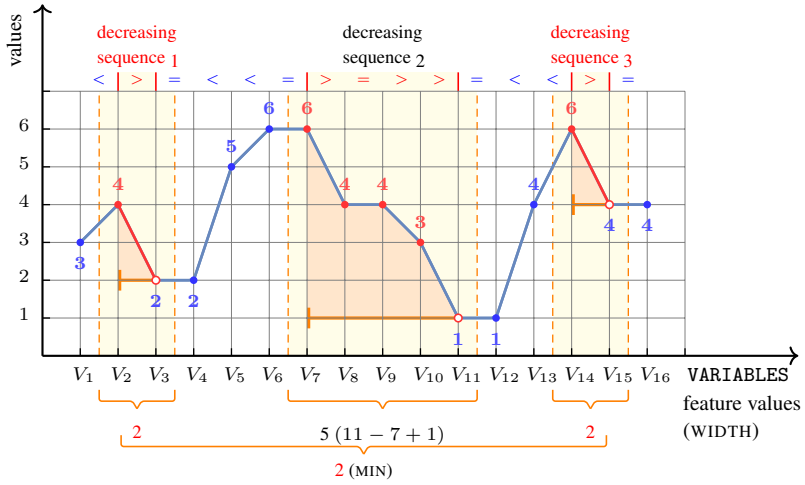


Figure 4.915: Illustrating the MIN_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.916 depicts the automaton associated with the constraint MIN_WIDTH_DECREASING_SEQUENCE.

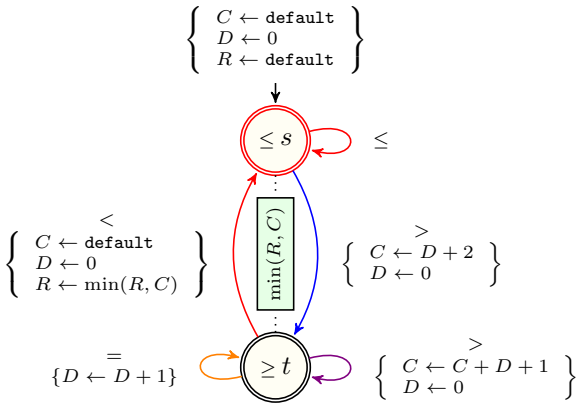


Figure 4.916: Automaton for the MIN_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

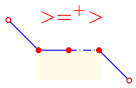
	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.201: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the MIN_WIDTH_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

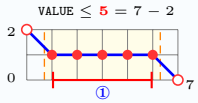
Constraint MIN_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3 \rangle)$

Figure 4.917 provides an example where the MIN_WIDTH_DECREASING_TERRACE $(2, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

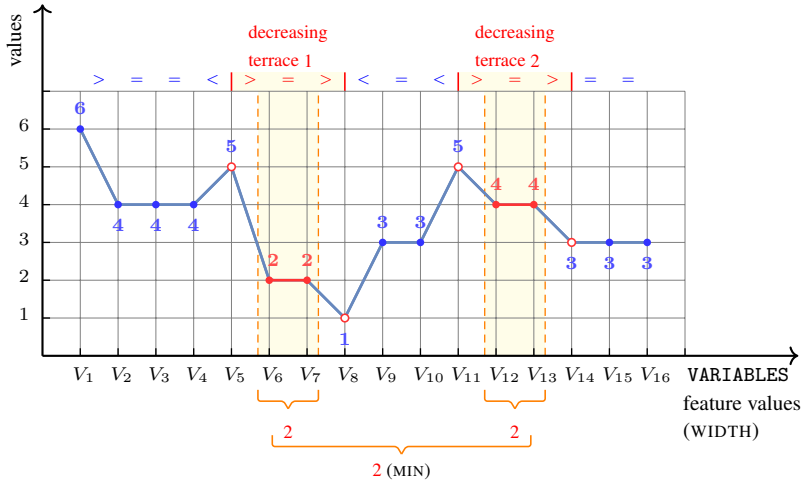


Figure 4.917: Illustrating the MIN_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.918 and 4.919 respectively depict the automaton associated with the constraint MIN_WIDTH_DECREASING_TERRACE and its simplified form.

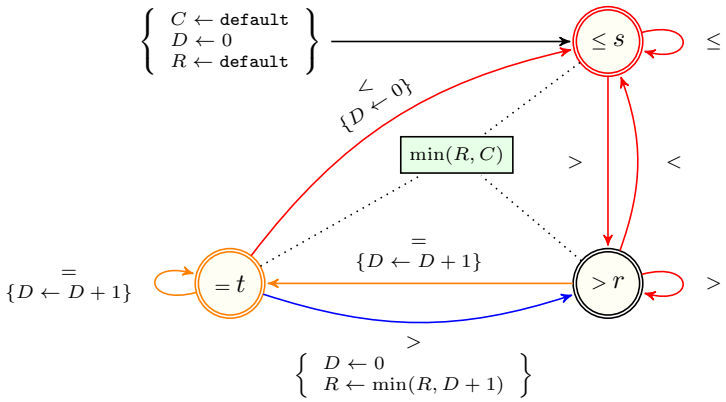


Figure 4.918: Automaton for the MIN_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $|\texttt{VARIABLES}| + 1$

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.202: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the MIN_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

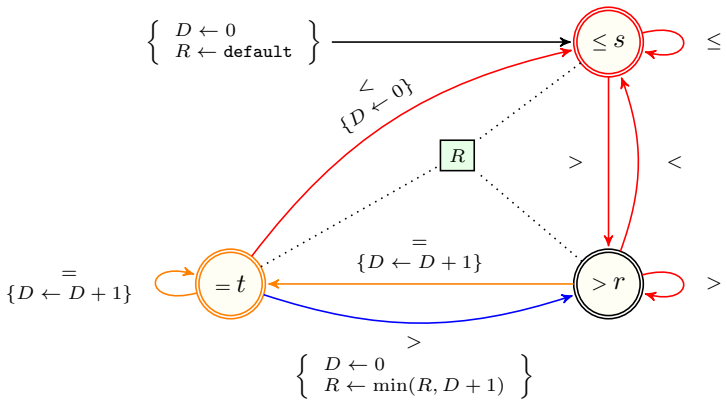


Figure 4.919: Simplified automaton for the MIN_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DECREASING_TERRACE pattern where `default` is $|\texttt{VARIABLES}|+1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

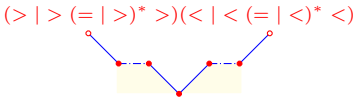
	s	r	t
s	$\|\texttt{VARIABLES}\|+1$	$\|\texttt{VARIABLES}\|+1$	$\|\texttt{VARIABLES}\|+1$
r	$\|\texttt{VARIABLES}\|+1$	$\|\texttt{VARIABLES}\|+1$	$\overrightarrow{D}+\overleftarrow{D}+1$ C
t	$\|\texttt{VARIABLES}\|+1$	$\overrightarrow{D}+\overleftarrow{D}+1$ C	$\overrightarrow{D}+\overleftarrow{D}+1$ C

Table 4.203: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the MIN_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

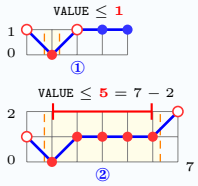
Constraint MIN_WIDTH_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(1, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.920 provides an example where the MIN_WIDTH_GORGE $(1, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

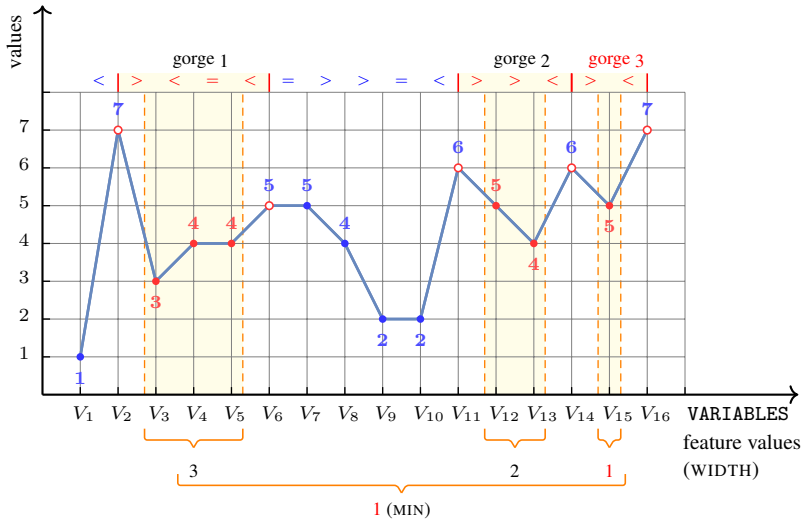


Figure 4.920: Illustrating the MIN_WIDTH_GORGE constraint of the **Example** slot

Automaton Figure 4.921 depicts the automaton associated with the constraint MIN_WIDTH_GORGE.

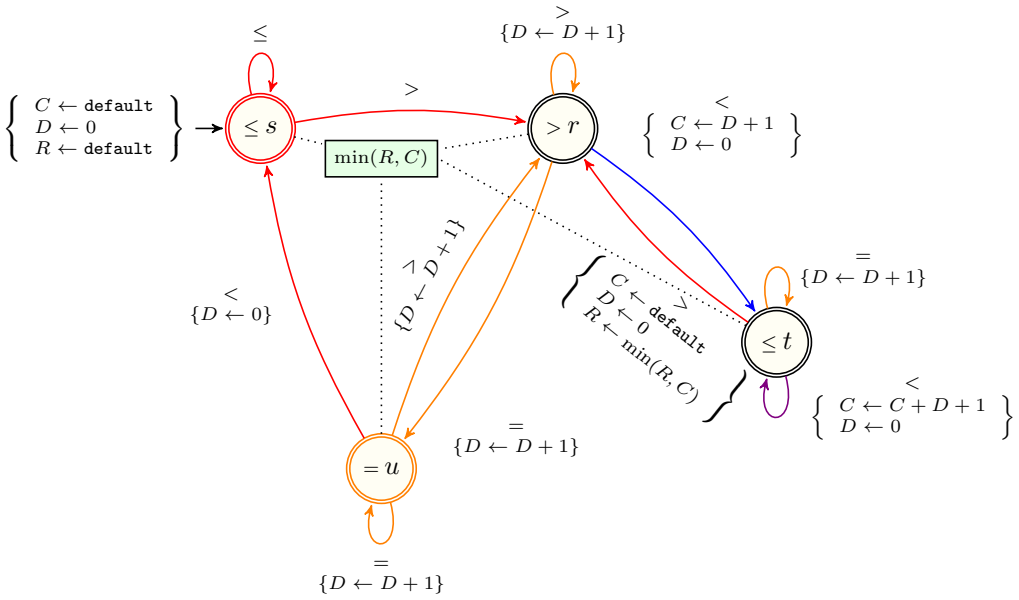


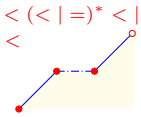
Figure 4.921: Automaton for the MIN_WIDTH_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is $|\texttt{VARIABLES}| + 1$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.204: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the MIN_WIDTH_GORGE constraint defined as the composition of the GORGE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

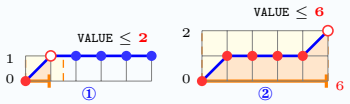
Constraint MIN_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the minimal width of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(2, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.922 provides an example where the MIN_WIDTH_INCREASING_SEQUENCE $(2, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

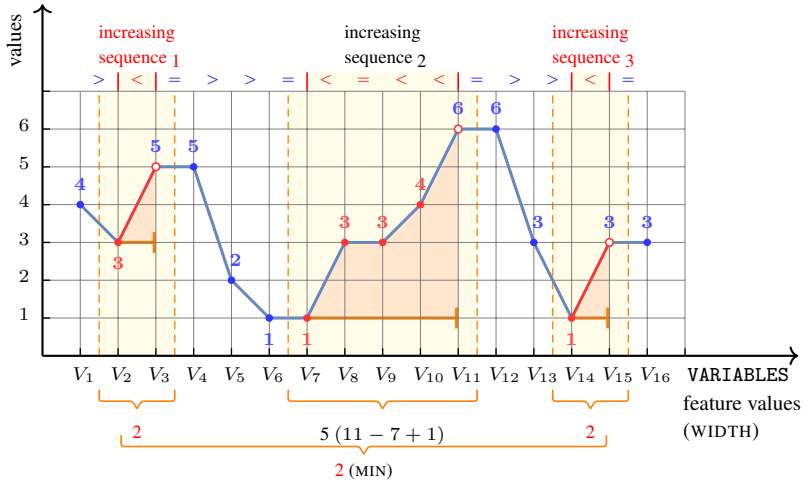


Figure 4.922: Illustrating the MIN_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Figure 4.923 depicts the automaton associated with the constraint MIN_WIDTH_INCREASING_SEQUENCE.

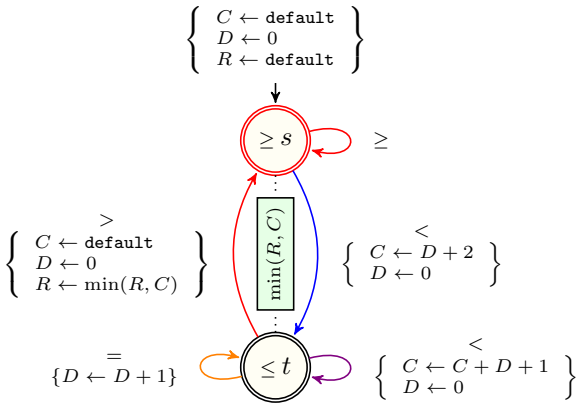


Figure 4.923: Automaton for the MIN_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

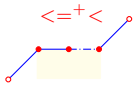
	s	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.205: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the MIN_WIDTH_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

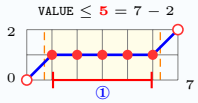
Constraint MIN_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.924 provides an example where the MIN_WIDTH_INCREASING_TERRACE $(2, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

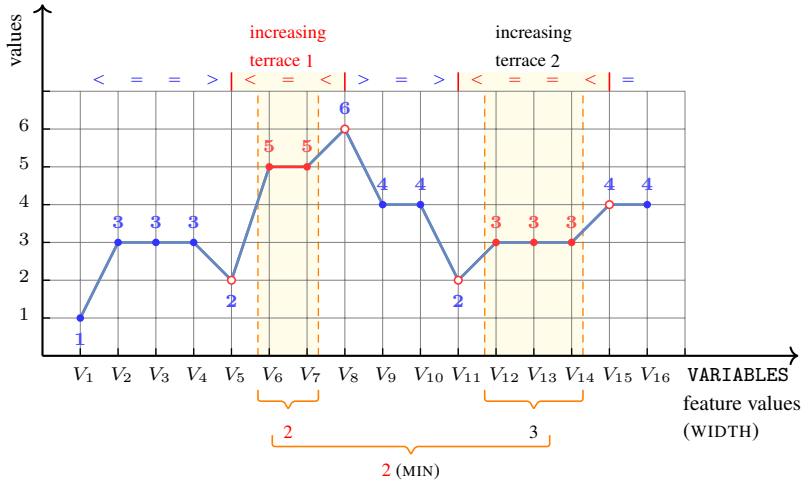


Figure 4.924: Illustrating the MIN_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.925 and 4.926 respectively depict the automaton associated with the constraint MIN_WIDTH_INCREASING_TERRACE and its simplified form.

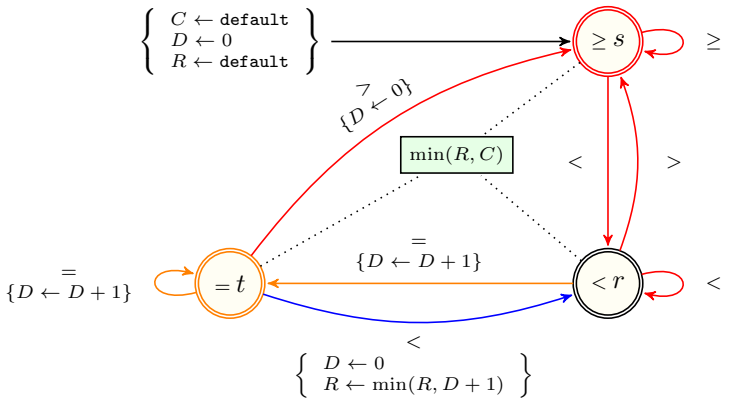


Figure 4.925: Automaton for the MIN_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $|\texttt{VARIABLES}| + 1$

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.206: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the MIN_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

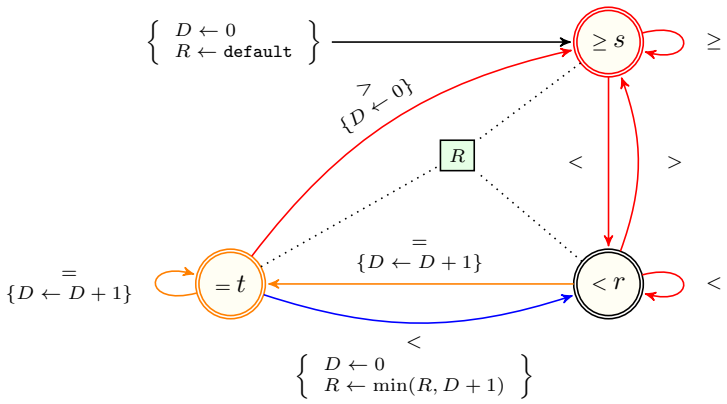


Figure 4.926: Simplified automaton for the MIN_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.26 to the seed transducer of the INCREASING_TERRACE pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

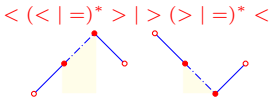
	s	r	t
s	$\|\texttt{VARIABLES}\| + 1$	$\|\texttt{VARIABLES}\| + 1$	$\|\texttt{VARIABLES}\| + 1$
r	$\|\texttt{VARIABLES}\| + 1$	$\|\texttt{VARIABLES}\| + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\|\texttt{VARIABLES}\| + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.207: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the MIN_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

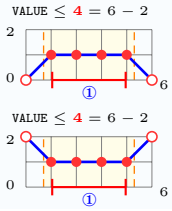
Constraint MIN_WIDTH_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(1, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.927 provides an example where the MIN_WIDTH_INFLEXION $(1, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

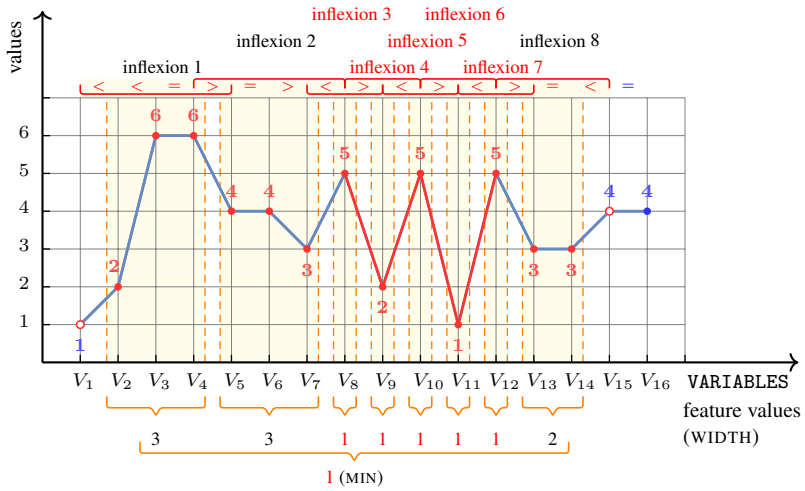


Figure 4.927: Illustrating the MIN_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.928 and 4.929 respectively depict the automaton associated with the constraint MIN_WIDTH_INFLEXION and its simplified form.

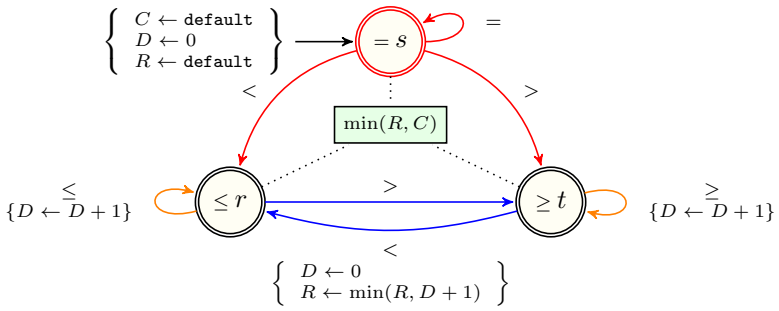


Figure 4.928: Automaton for the MIN_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where default is $|\texttt{VARIABLES}| + 1$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

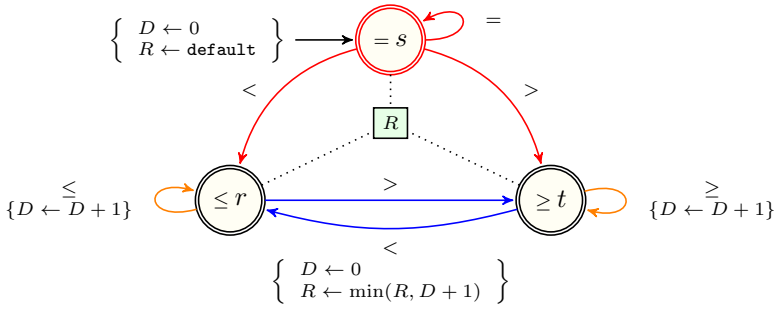
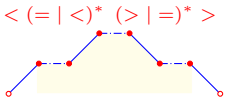


Figure 4.929: Simplified automaton for the MIN_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where default is $|\texttt{VARIABLES}| + 1$ (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

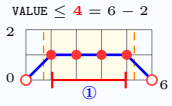
Constraint MIN_WIDTH_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.930 provides an example where the MIN_WIDTH_PEAK $(2, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

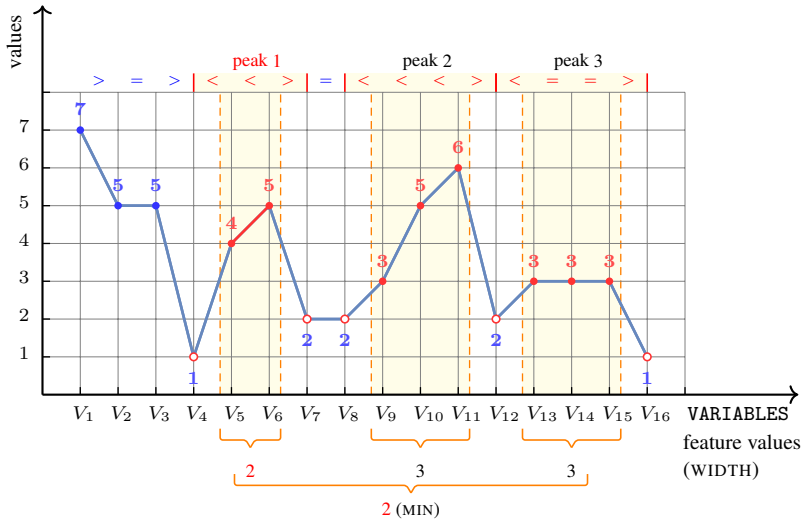


Figure 4.930: Illustrating the MIN_WIDTH_PEAK constraint of the **Example** slot

Automaton Figure 4.931 depicts the automaton associated with the constraint MIN_WIDTH_PEAK.

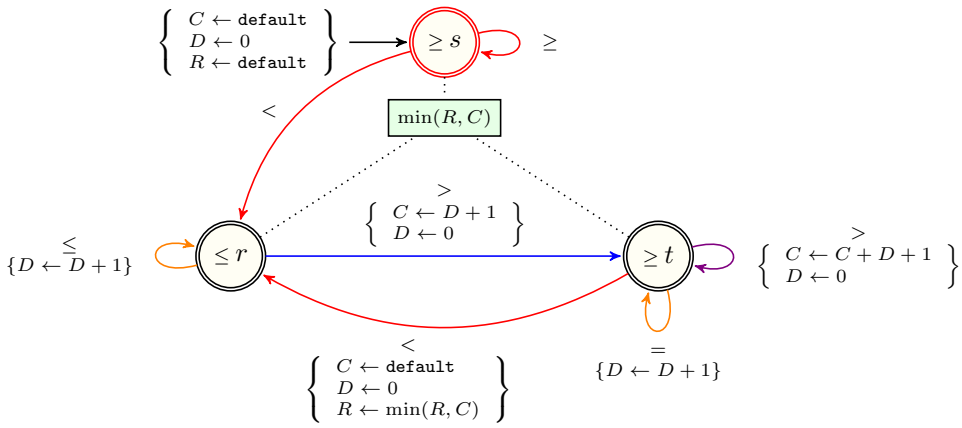


Figure 4.931: Automaton for the MIN_WIDTH_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where default is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.208: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the MIN_WIDTH_PEAK constraint defined as the composition of the PEAK pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

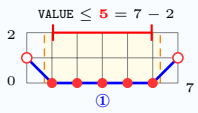
Constraint MIN_WIDTH_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(1, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3 \rangle)$

Figure 4.932 provides an example where the MIN_WIDTH_PLAIN $(1, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

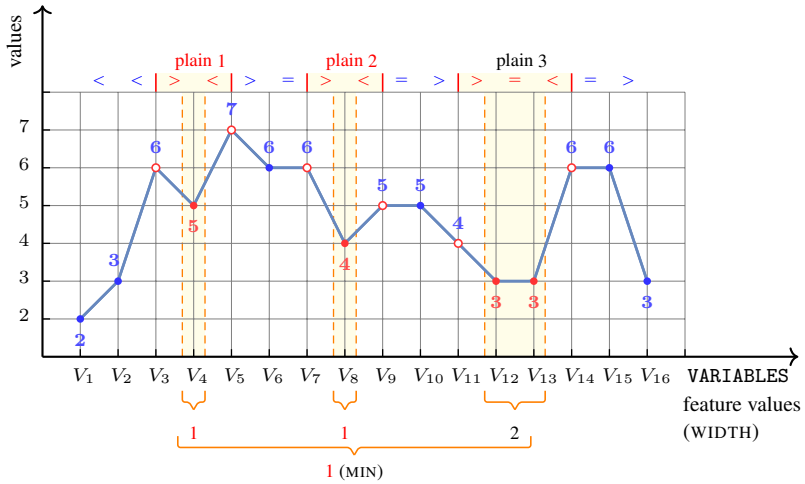


Figure 4.932: Illustrating the MIN_WIDTH_PLAIN constraint of the **Example** slot

Automaton

Figures 4.933 and 4.934 respectively depict the automaton associated with the constraint MIN_WIDTH_PLAIN and its simplified form.

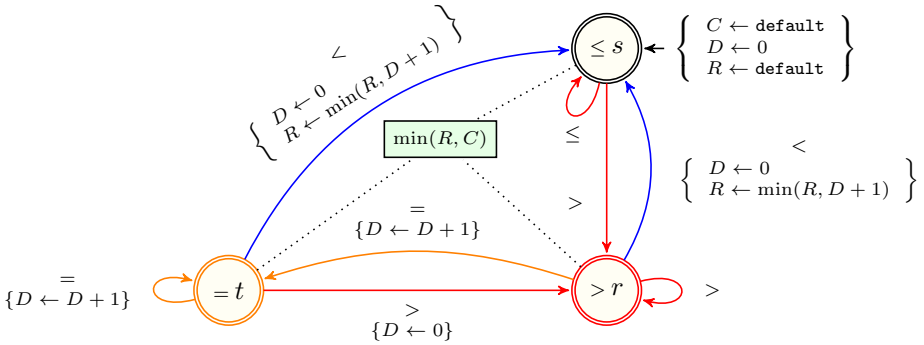


Figure 4.933: Automaton for the MIN_WIDTH_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is $|\texttt{VARIABLES}| + 1$

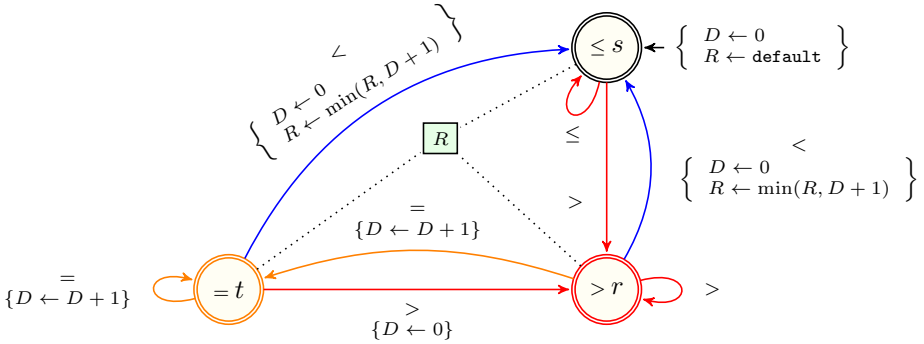


Figure 4.934: Simplified automaton for the MIN_WIDTH_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLAIN pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.209: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the MIN_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$
r	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.210: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the MIN_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_PLATEAU

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

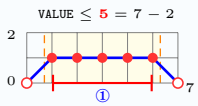
Constraint MIN_WIDTH_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(3, \langle 1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5 \rangle)$

Figure 4.935 provides an example where the MIN_WIDTH_PLATEAU $(3, [1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

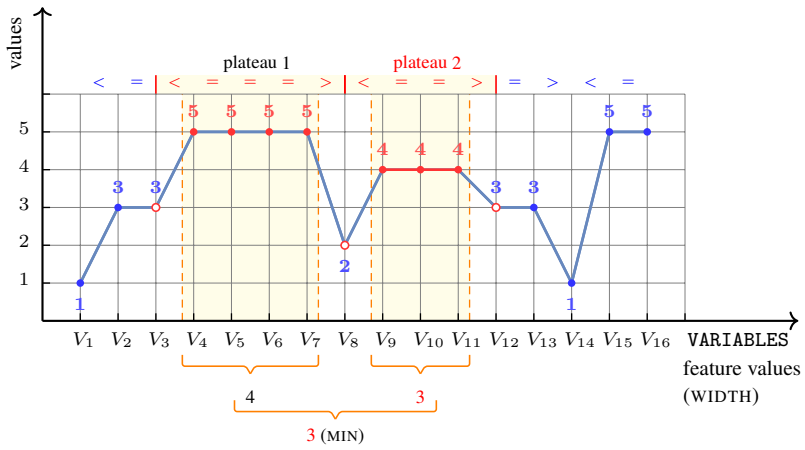


Figure 4.935: Illustrating the MIN_WIDTH_PLATEAU constraint of the **Example** slot

Automaton Figures 4.936 and 4.937 respectively depict the automaton associated with the constraint MIN_WIDTH_PLATEAU and its simplified form.

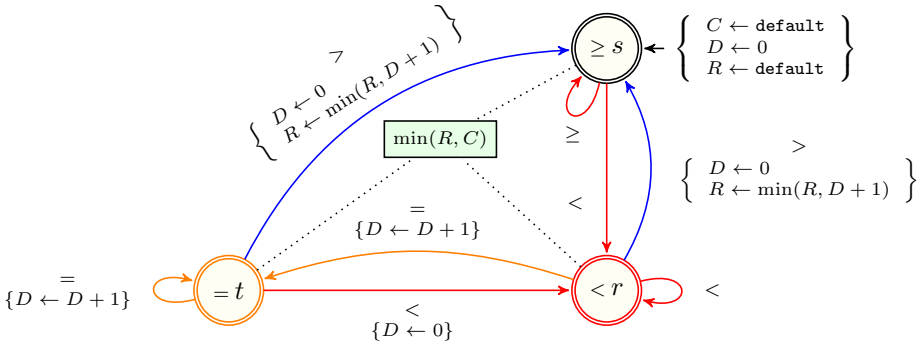


Figure 4.936: Automaton for the MIN_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is $|\texttt{VARIABLES}| + 1$

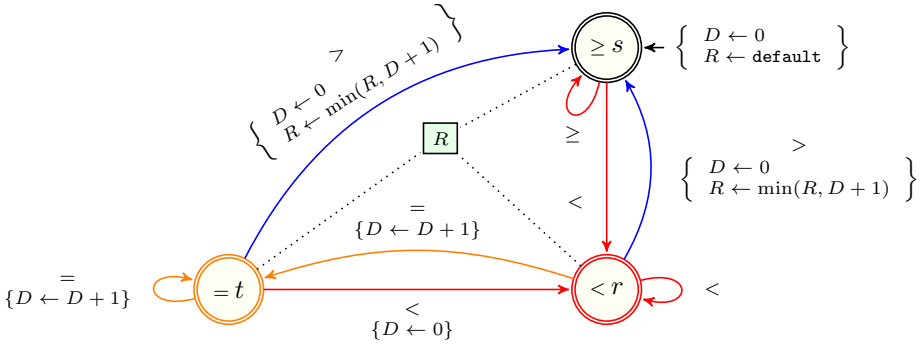


Figure 4.937: Simplified automaton for the MIN_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PLATEAU pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.211: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the MIN_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

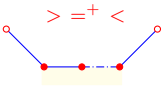
	s	r	t
s	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$
r	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.212: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the MIN_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

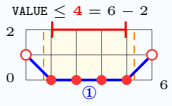
Constraint MIN_WIDTH_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5 \rangle)$

Figure 4.938 provides an example where the MIN_WIDTH_PROPER_PLAIN $(2, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

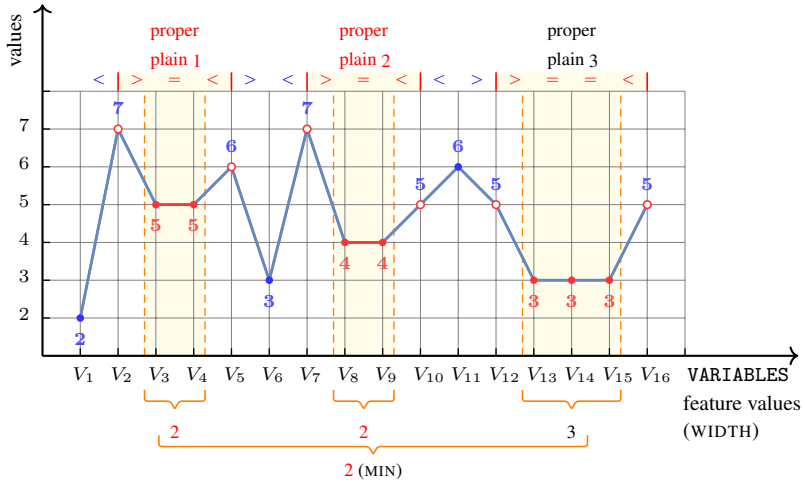


Figure 4.938: Illustrating the MIN_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.939 and 4.940 respectively depict the automaton associated with the constraint MIN_WIDTH_PROPER_PLAIN and its simplified form.

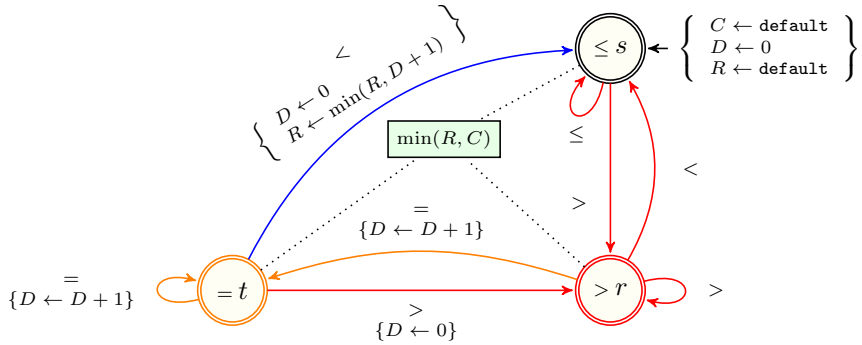


Figure 4.939: Automaton for the MIN_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is $|\texttt{VARIABLES}| + 1$

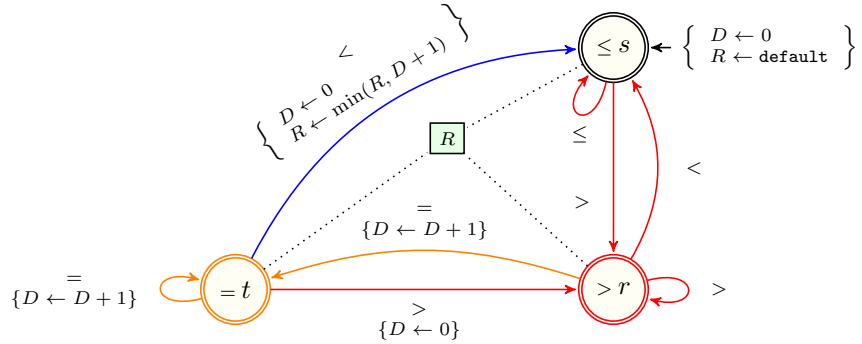


Figure 4.940: Simplified automaton for the MIN_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLAIN pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.213: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the MIN_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

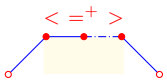
	s	r	t
s	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$
r	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.214: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the MIN_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

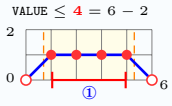
Constraint MIN_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3 \rangle)$

Figure 4.941 provides an example where the MIN_WIDTH_PROPER_PLATEAU $(2, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

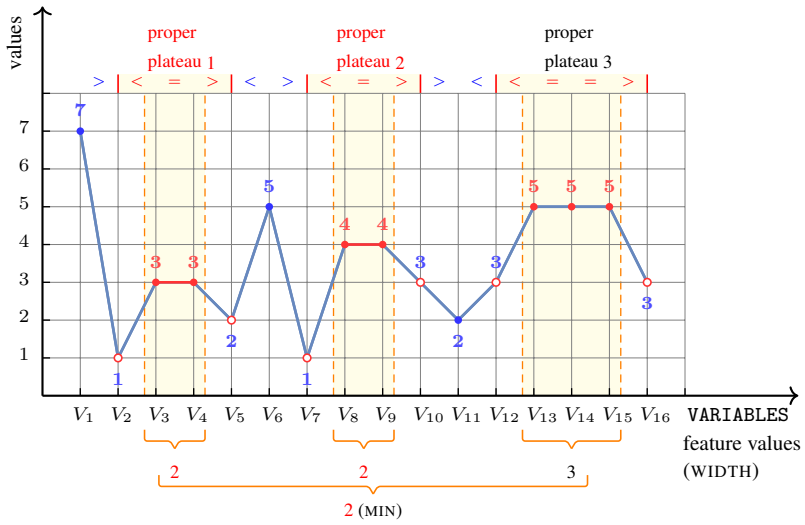


Figure 4.941: Illustrating the MIN_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.942 and 4.943 respectively depict the automaton associated with the constraint MIN_WIDTH_PROPER_PLATEAU and its simplified form.

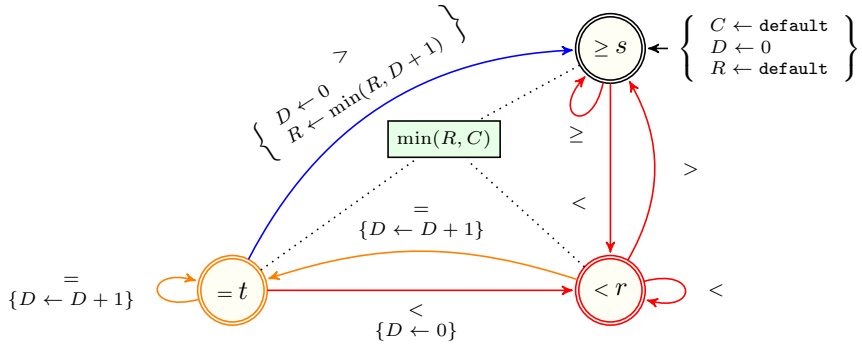


Figure 4.942: Automaton for the MIN_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $|\texttt{VARIABLES}| + 1$

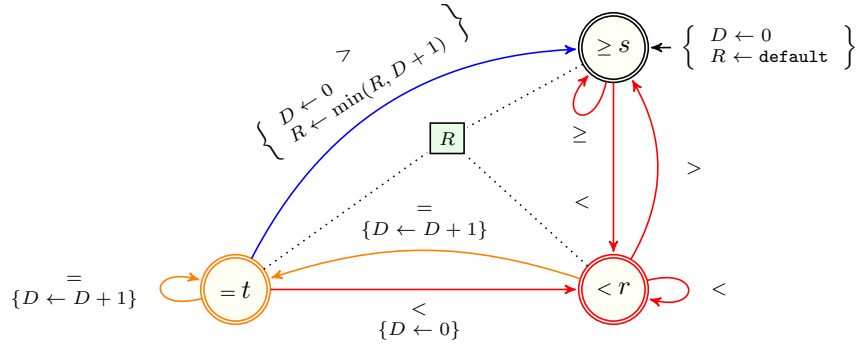


Figure 4.943: Simplified automaton for the MIN_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.26 to the seed transducer of the PROPER_PLATEAU pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.215: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the MIN_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

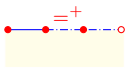
	s	r	t
s	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$
r	$\lvert\texttt{VARIABLES}\rvert + 1$	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\lvert\texttt{VARIABLES}\rvert + 1$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.216: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the MIN_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

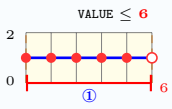
Constraint MIN_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$①
required(VARIABLES, var)
where
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the minimal width of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(2, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.944 provides an example where the MIN_WIDTH_STEADY_SEQUENCE $(2, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

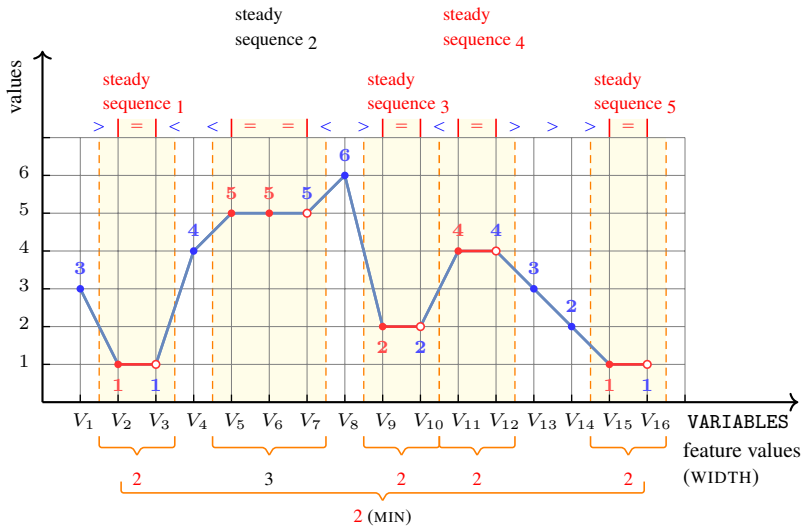


Figure 4.944: Illustrating the MIN_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.945 and 4.946 respectively depict the automaton associated with the constraint MIN_WIDTH_STEADY_SEQUENCE and its simplified form.

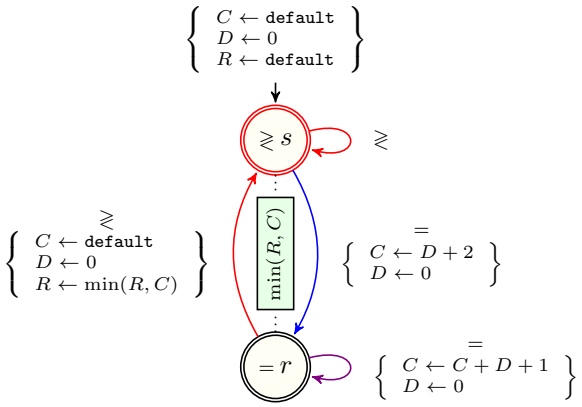


Figure 4.945: Automaton for the MIN_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where default is $|\texttt{VARIABLES}| + 1$

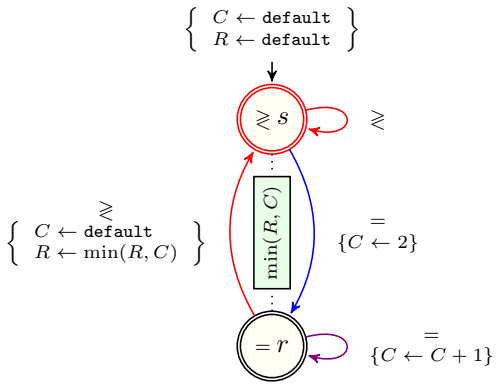


Figure 4.946: Simplified automaton for the MIN_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STEADY_SEQUENCE pattern where default is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.217: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the MIN_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - 1$ M

Table 4.218: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the MIN_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

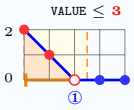
Constraint MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}[1])$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the minimal width of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(2, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3\rangle)$

Figure 4.947 provides an example where the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE $(2, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

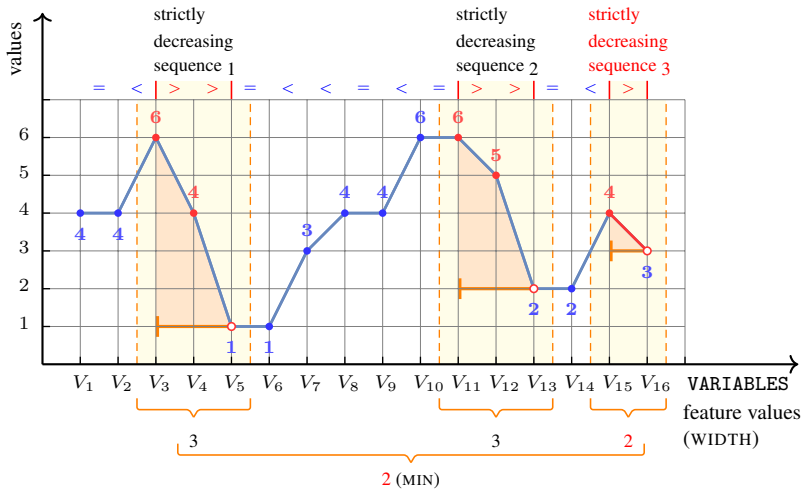


Figure 4.947: Illustrating the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.948 and 4.949 respectively depict the automaton associated with the constraint MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE and its simplified form.

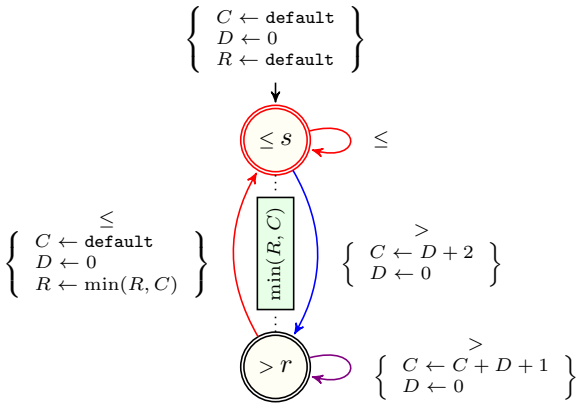


Figure 4.948: Automaton for the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$

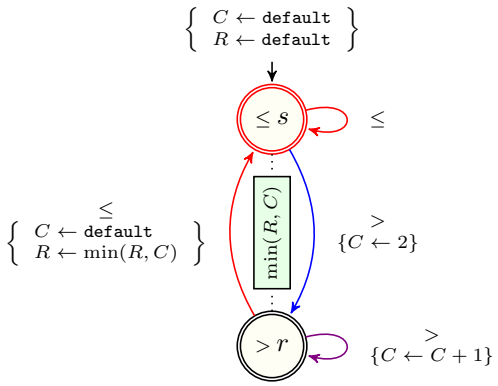


Figure 4.949: Simplified automaton for the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.219: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - 1$ M

Table 4.220: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

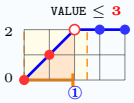
Constraint MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(2, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.950 provides an example where the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE $(2, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

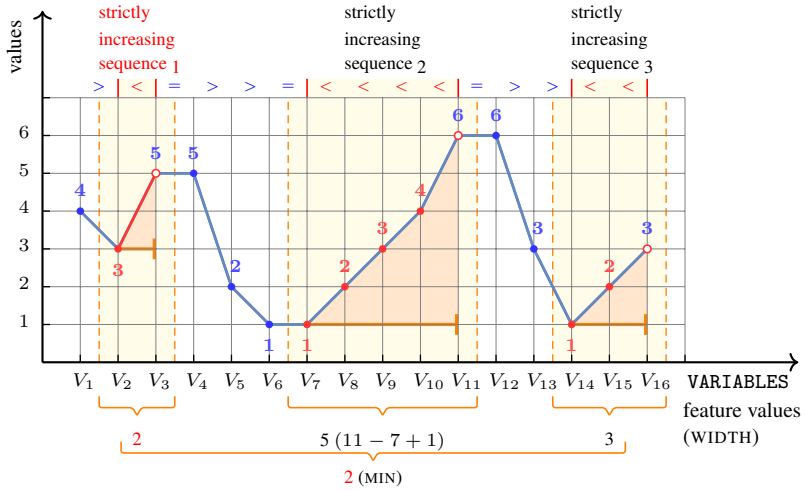


Figure 4.950: Illustrating the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.951 and 4.952 respectively depict the automaton associated with the constraint MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE and its simplified form.

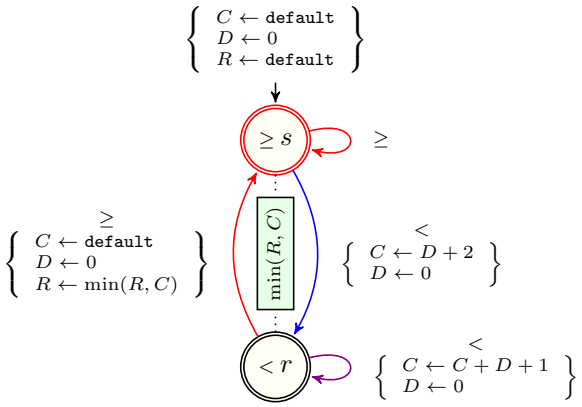


Figure 4.951: Automaton for the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$

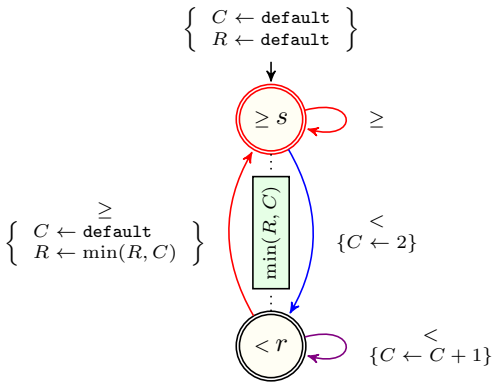


Figure 4.952: Simplified automaton for the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.221: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overleftarrow{C} - 1$ M

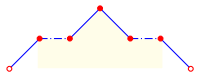
Table 4.222: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_SUMMIT

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



Origin Based on the SUMMIT pattern.

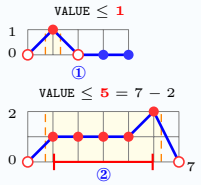
Constraint MIN_WIDTH_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(1, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.953 provides an example where the MIN_WIDTH_SUMMIT $(1, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

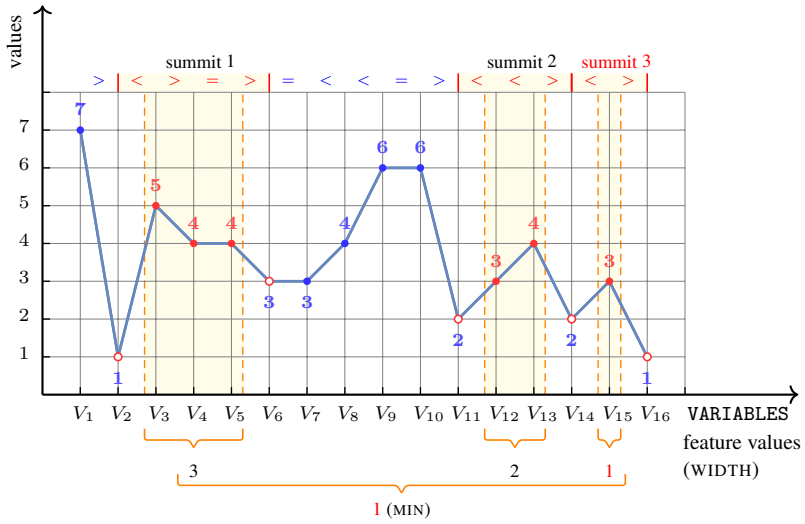


Figure 4.953: Illustrating the MIN_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figure 4.954 depicts the automaton associated with the constraint MIN_WIDTH_SUMMIT.

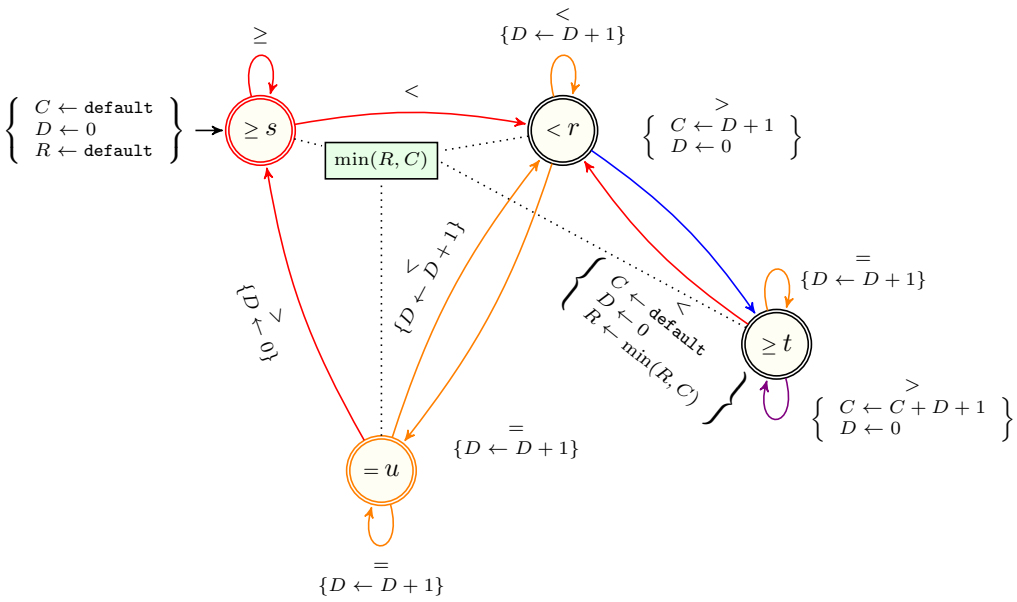


Figure 4.954: Automaton for the MIN_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is $|\texttt{VARIABLES}|+1$ (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $-R_i + R_{i-1} \geq 0$ is a linear invariant.

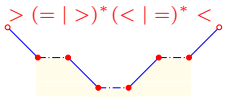
	s	r	t	u
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.223: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the MIN_WIDTH_SUMMIT constraint defined as the composition of the SUMMIT pattern , the feature WIDTH , and the aggregator min ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

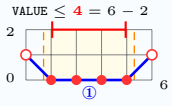
Constraint MIN_WIDTH_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.955 provides an example where the MIN_WIDTH_VALLEY $(2, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

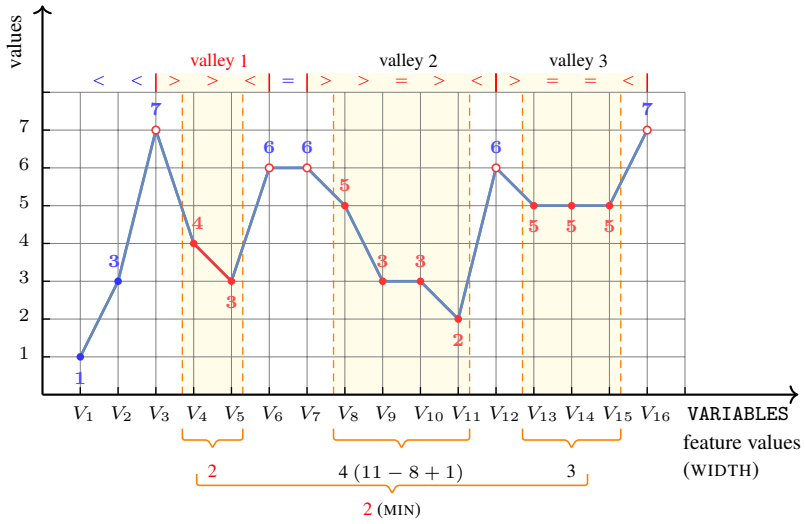


Figure 4.955: Illustrating the MIN_WIDTH_VALLEY constraint of the **Example** slot

Automaton Figure 4.956 depicts the automaton associated with the constraint MIN_WIDTH_VALLEY.

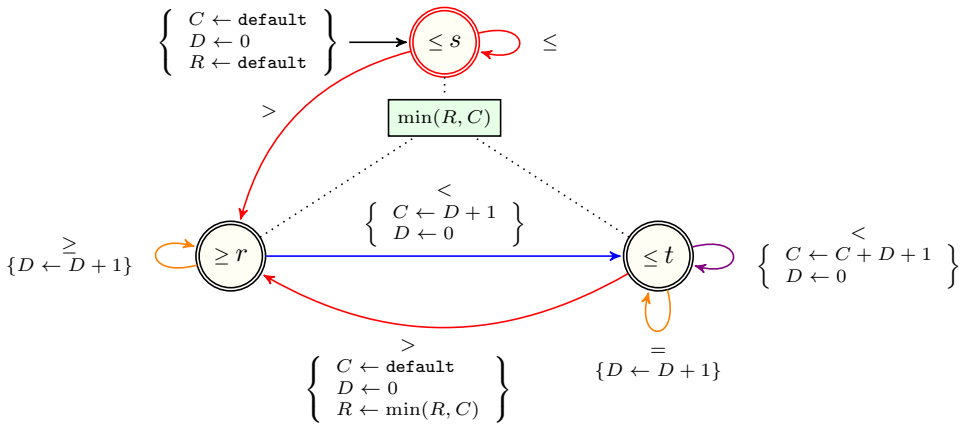


Figure 4.956: Automaton for the MIN_WIDTH_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is $|\texttt{VARIABLES}| + 1$; $-R_i + R_{i-1} \geq 0$ is a linear invariant.

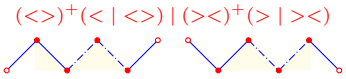
	s	r	t
s	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\min(\overrightarrow{C}, \overleftarrow{C})$
r	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\min(\overrightarrow{C}, \overleftarrow{C})$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\min(\overrightarrow{C}, \overleftarrow{C})$

Table 4.224: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the MIN_WIDTH_VALLEY constraint defined as the composition of the VALLEY pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

MIN_WIDTH_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

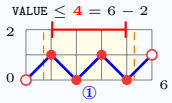
Constraint MIN_WIDTH_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

VALUE is the minimal width of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value $|\texttt{VARIABLES}| + 1$.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(2, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.957 provides an example where the MIN_WIDTH_ZIGZAG $(2, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

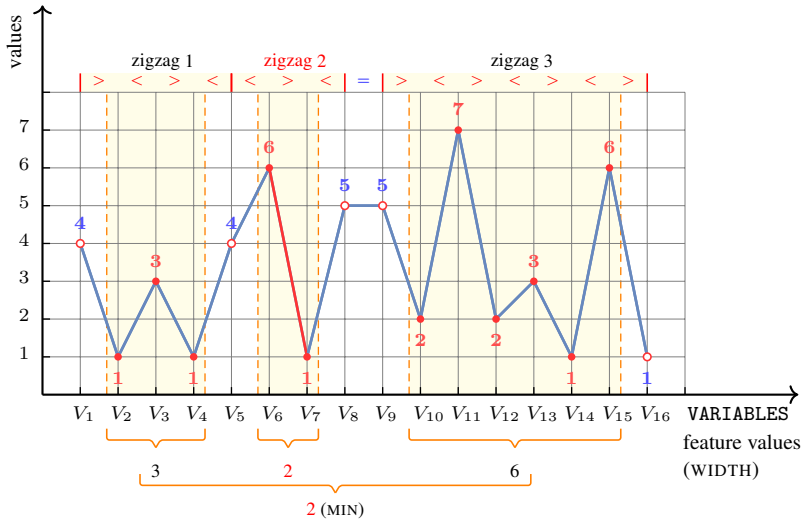


Figure 4.957: Illustrating the MIN_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.958 and 4.959 respectively depict the automaton associated with the constraint MIN_WIDTH_ZIGZAG and its simplified form.

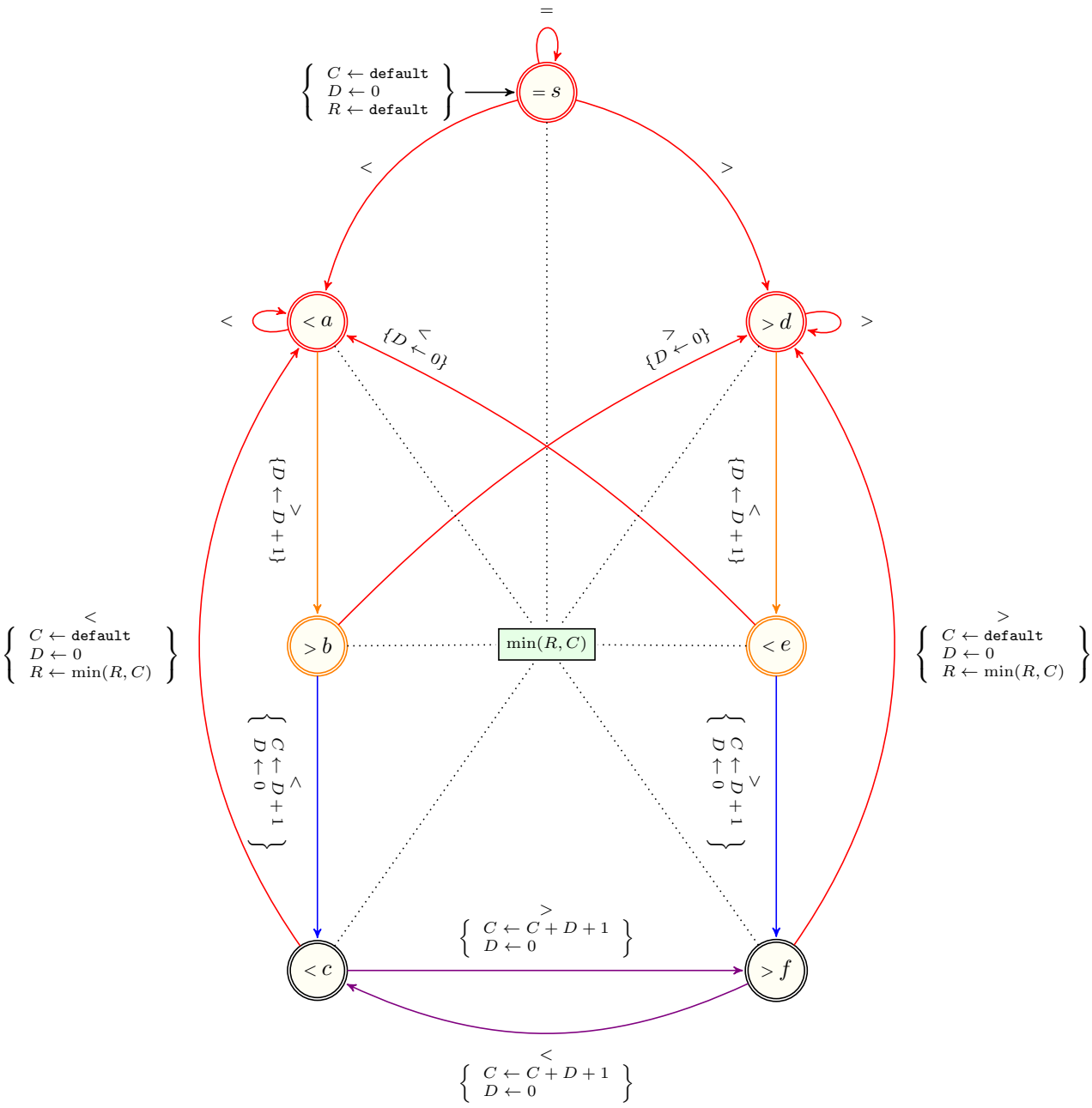


Figure 4.958: Automaton for the MIN_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is $|\texttt{VARIABLES}| + 1$; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

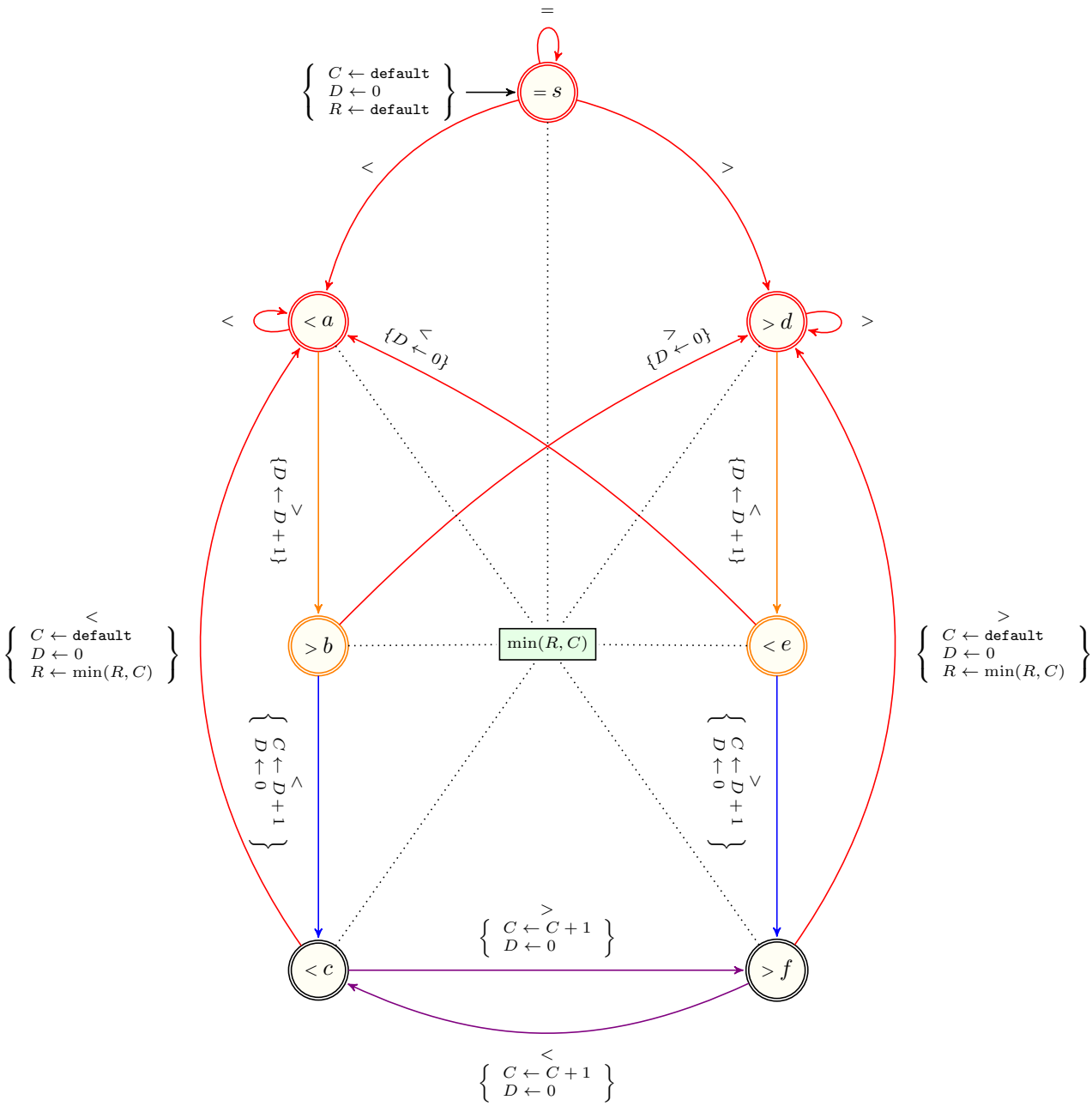
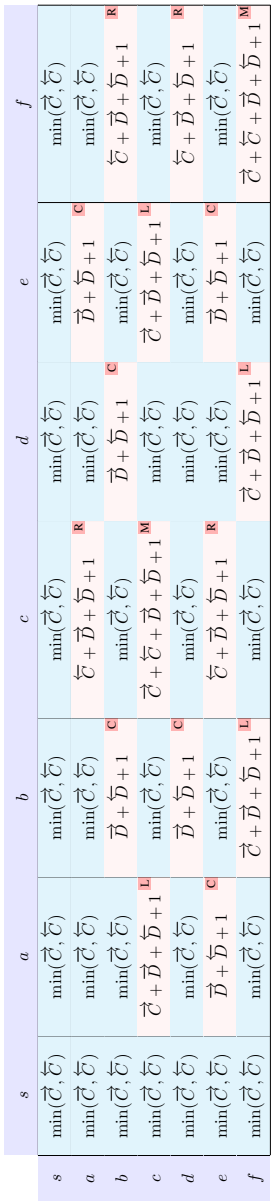


Figure 4.959: Simplified automaton for the MIN_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is $|\texttt{VARIABLES}| + 1$; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.; $-R_i + R_{i-1} \geq 0$ is a linear invariant.



	s	a	b	c	d	e	f
s	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$
a	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}}+\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ R	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$
b	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}}+\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ R
c	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}}+\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ L	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}}+\overleftarrow{\mathcal{C}}+\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ M	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}}+\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ L	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$
d	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$
e	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}}+\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ R	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ C	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$
f	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}}+\vec{\mathcal{D}}+\overleftarrow{\mathcal{D}}+1$ R	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}}+\overleftarrow{\mathcal{C}}+\vec{\mathcal{B}}+\overleftarrow{\mathcal{D}}+1$ M	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}},\overleftarrow{\mathcal{C}})$

Table 4.225: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the MIN_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

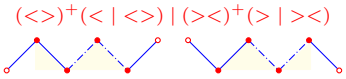
	s	a	b	c	d	e	f
s	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$
a	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ R	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$
b	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ R
c	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\overleftarrow{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ R	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \overleftarrow{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ M	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ L	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$
d	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ R
e	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ L	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ C	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$
f	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ L	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ L	$\min(\vec{\mathcal{C}}, \overleftarrow{\mathcal{C}})$	$\vec{\mathcal{C}} + \vec{\mathcal{C}} + \vec{\mathcal{D}} + \overleftarrow{\mathcal{D}} + 1$ M

Table 4.226: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the MIN_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator min; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

FEATURE PATTERN

MIN_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint MIN_ZIGZAG(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = DEFAULT
FEATURES.var = DEFAULT ∨ FEATURES.var ≥ minv
FEATURES.var = DEFAULT ∨ FEATURES.var ≤ maxv − 1
DEFAULT < minv ∨ DEFAULT > maxv − 1
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of ZIGZAG is identified (even if this occurrence of pattern is not complete) then FEATURES[i] is the default value DEFAULT; otherwise FEATURES[i] gives the feature value of the corresponding occurrence of ZIGZAG.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example Figure 4.960 provides an example where the MIN_ZIGZAG $([4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1],[0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

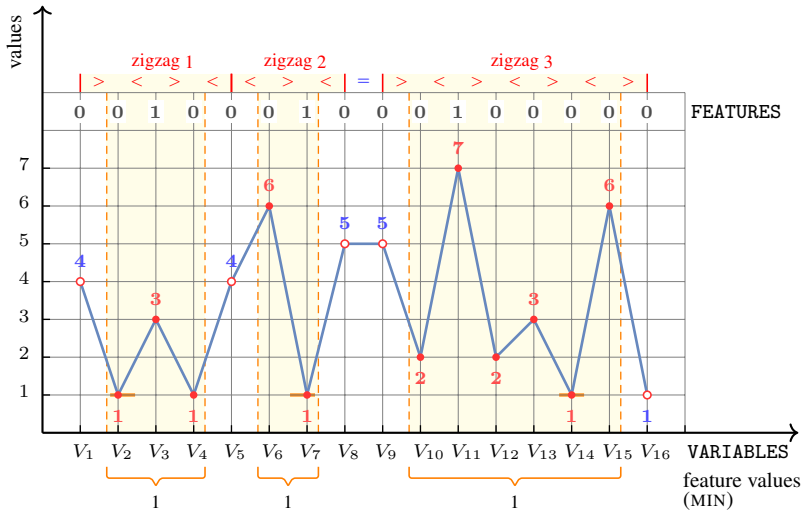
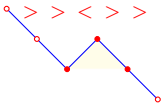


Figure 4.960: Illustrating the MIN_ZIGZAG constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

AGGREGATOR FEATURE PATTERN

NB_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

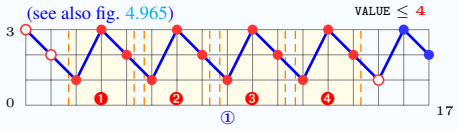
Constraint NB_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \lfloor(\texttt{sv} - 3)/3\rfloor$①)
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.

Example $(2, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.961 provides an example where the NB_BUMP_ON_DECREASING_SEQUENCE $(2, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

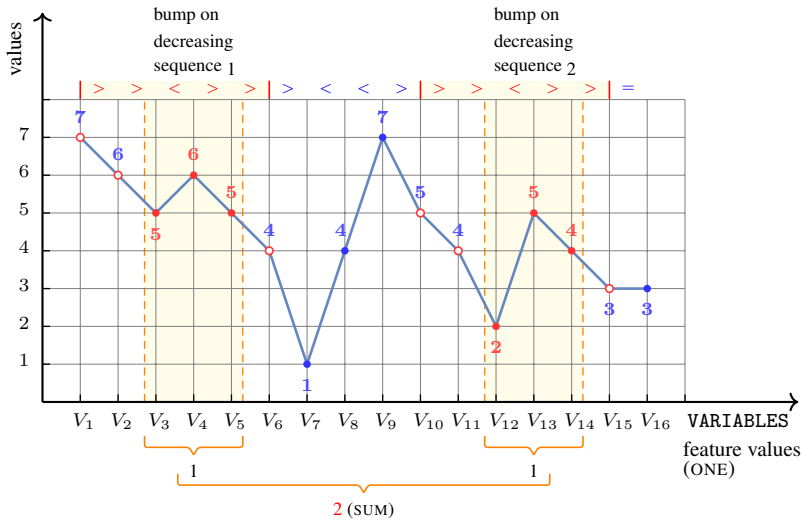


Figure 4.961: Illustrating the NB_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.962 and 4.963 respectively depict the automaton associated with the constraint NB_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

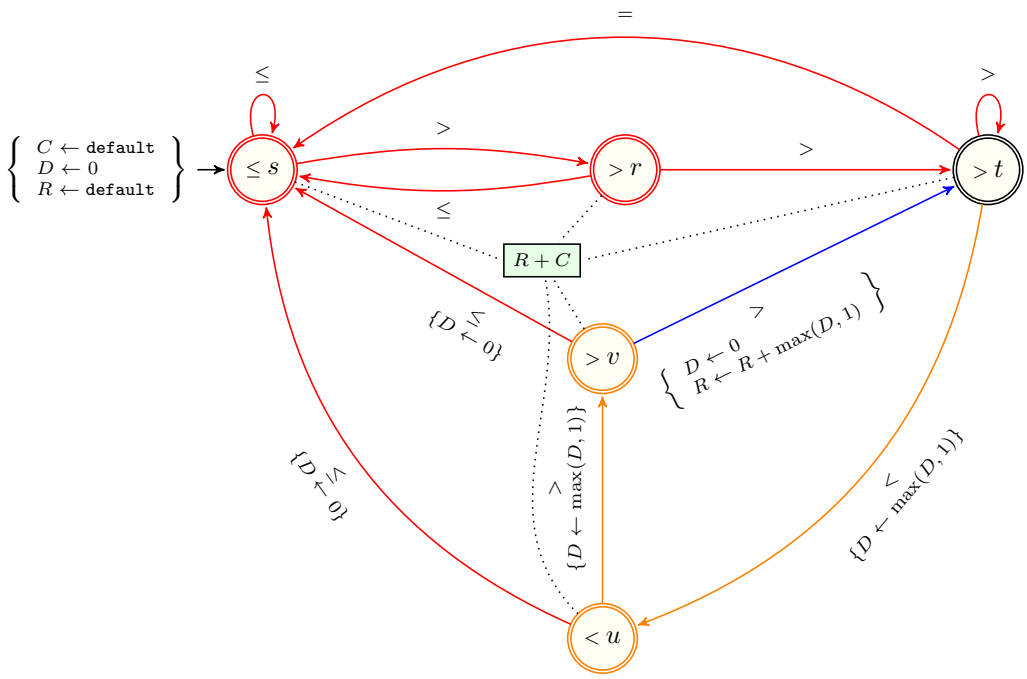


Figure 4.962: Automaton for the NB_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

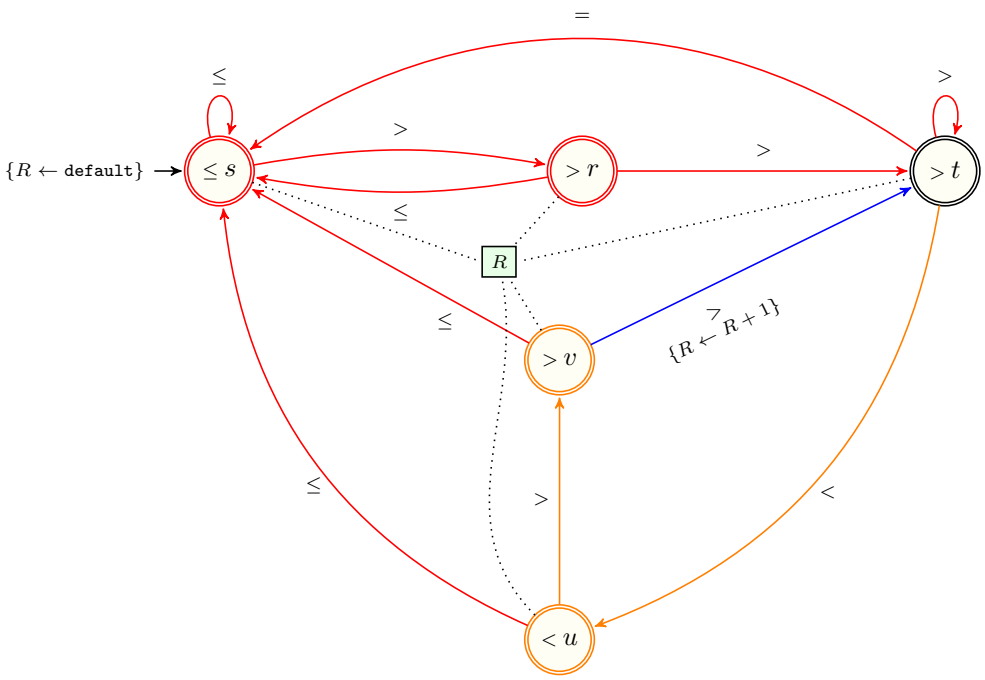


Figure 4.963: Simplified automaton for the NB_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-3} + 1 \geq 0$ are linear invariants.

Specialisation

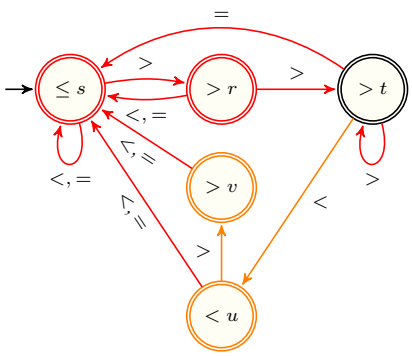
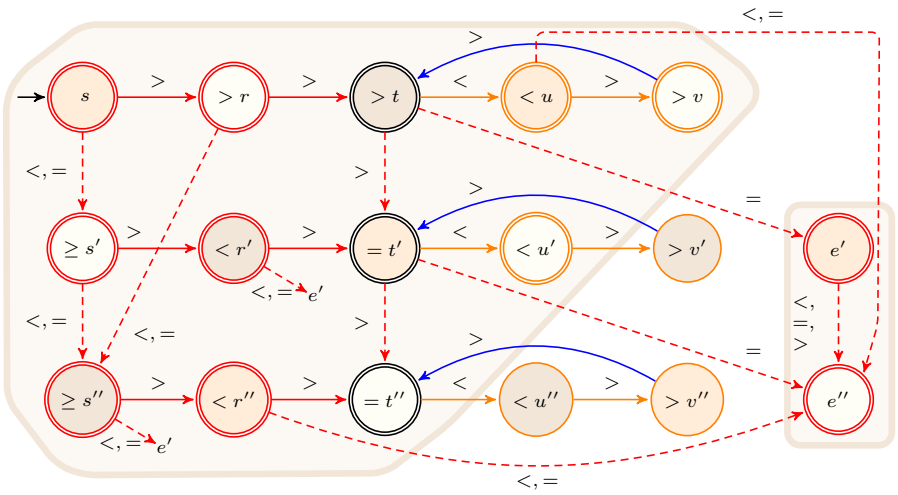


Figure 4.964: Automaton without registers for the NB_BUMP_ON_DECREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern by removing the register R and the **found** transition, i.e. the transition from state v to state t that increments R.



| > > < > > |	< > | > > < > > |	| > > < > > | = <
< | > > < > > |	= < | > > < > > |	| > > < > > | = =
= | > > < > > |	= = | > > < > > |	| > > < > > | = >
| > > < > > | <	= | > > < > > | <	| > > < > > | > <
| > > < > > | =	= | > > < > > | =	| > > < > > | > =
| > > < > > | >	= | > > < > > | >	| > > < > > | > >
> | > > < > > |	= > | > > < > > |	> | > > < > > | <
< < | > > < > > |	> < | > > < > > |	> | > > < > > | =
< = | > > < > > |	> = | > > < > > |	> | > > < > > | >
< | > > < > > | <	| > > < > > | < <	> > | > > < > > |
< | > > < > > | =	| > > < > > | < =	
< | > > < > > | >	| > > < > > | < >	

Figure 4.965: (**top**) Automaton without registers for the NB_BUMP_ON_DECREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-3}{3} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern, dashed transitions correspond to slack, and accepting states have a light-brown background; states s, u, t', e', r'' are accepting when $\mathtt{sv} \bmod 3 = 1$, states r, v, s', u', t'', e'' are accepting when $\mathtt{sv} \bmod 3 = 2$, states t, r', s'' are accepting when $\mathtt{sv} \bmod 3 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6, 7\}$.

AGGREGATOR FEATURE PATTERN

NB_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint NB_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$

$\texttt{VALUE} \geq 0$

$\texttt{VALUE} \leq \texttt{max}(0, (\ell - 1) * \lfloor \texttt{sv}/\ell \rfloor + \texttt{max}(0, \texttt{sv} \bmod \ell - 1))$①

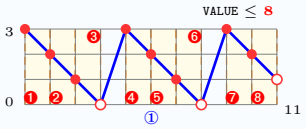
required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

$\ell = \texttt{min}(\texttt{sv}, \texttt{rv})$



Purpose VALUE is the number of occurrences of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.

Example $(5, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.966 provides an example where the NB_DECREASING $(5, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$

$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

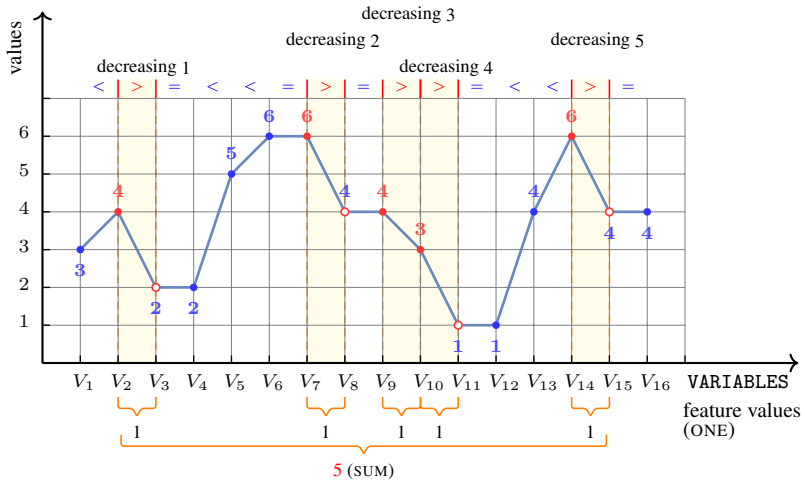


Figure 4.966: Illustrating the NB_DECREASING constraint of the **Example** slot

Automaton

Figures 4.967 and 4.968 respectively depict the automaton associated with the constraint NB_DECREASING and its simplified form.

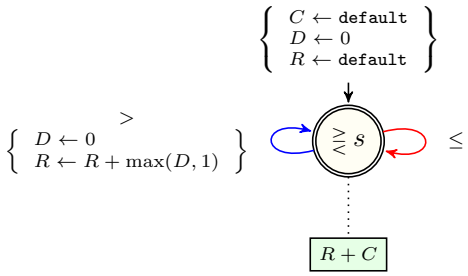


Figure 4.967: Automaton for the NB_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is 0

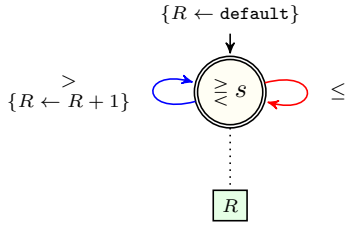


Figure 4.968: Simplified automaton for the NB_DECREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 1 \geq 0$ are linear invariants.

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.227: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the NB_DECREASING constraint defined as the composition of the DECREASING pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.228: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the NB_DECREASING constraint defined as the composition of the DECREASING pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

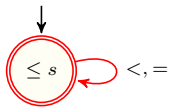


Figure 4.969: Automaton without registers for the NB_DECREASING_EQ_0 constraint; it describes all sequences containing no occurrence of the DECREASING pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the DECREASING pattern by removing the register R and the **found** transition, i.e. the transition from state s to state s that increments R.

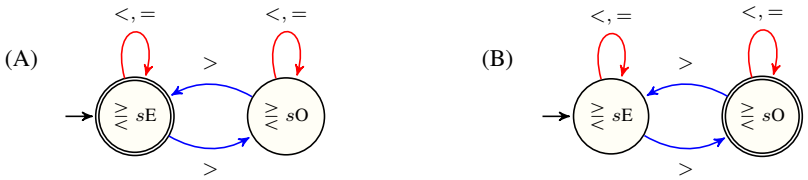
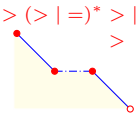


Figure 4.970: Automata without registers for the (A) NB_DECREASING_IS_EVEN and the (B) NB_DECREASING_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the DECREASING pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the DECREASING pattern.

AGGREGATOR FEATURE PATTERN

NB_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

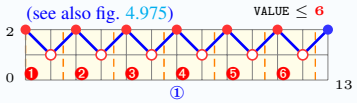
Constraint NB_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.

Example $(3, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.971 provides an example where the NB_DECREASING_SEQUENCE $(3, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

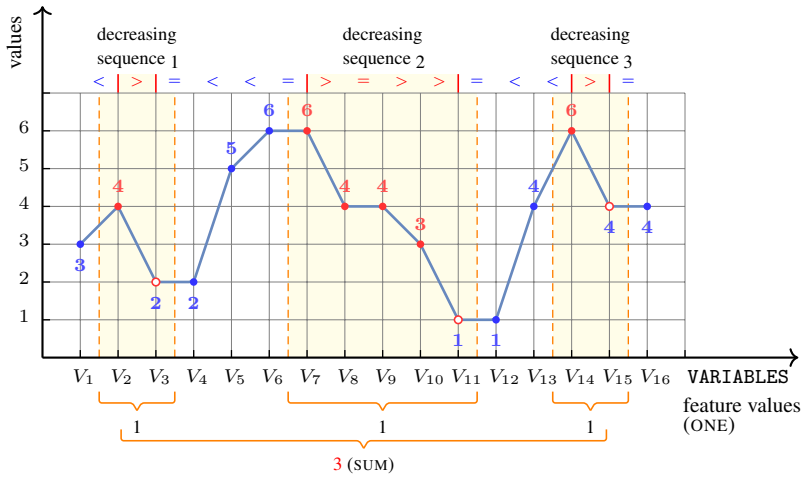


Figure 4.971: Illustrating the NB_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.972 and 4.973 respectively depict the automaton associated with the constraint NB_DECREASING_SEQUENCE and its simplified form.

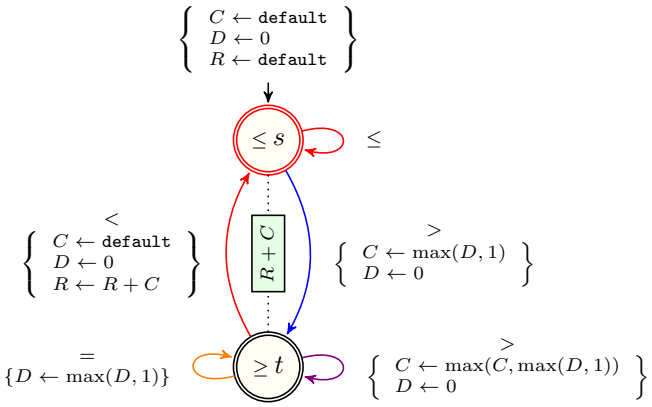


Figure 4.972: Automaton for the NB_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

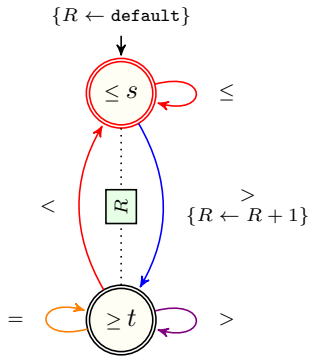


Figure 4.973: Simplified automaton for the NB_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	1 M

Table 4.229: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the NB_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	0	0
t	0	-1 M

Table 4.230: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the NB_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

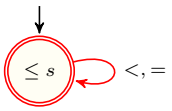


Figure 4.974: Automaton without registers for the NB_DECREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the DECREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the DECREASING_SEQUENCE pattern by removing the register R, the **found** transition from state s to t that increments R, and the state t that becomes unreachable after removing transition $s \rightarrow t$.

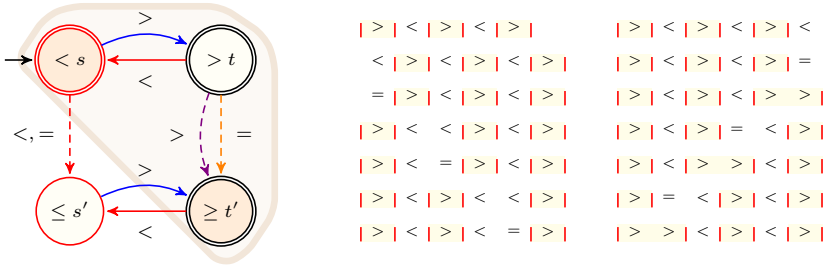


Figure 4.975: (**left**) Automaton without registers for the NB_DECREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the DECREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the DECREASING_SEQUENCE pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 0$, while states s and t' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6\}$.

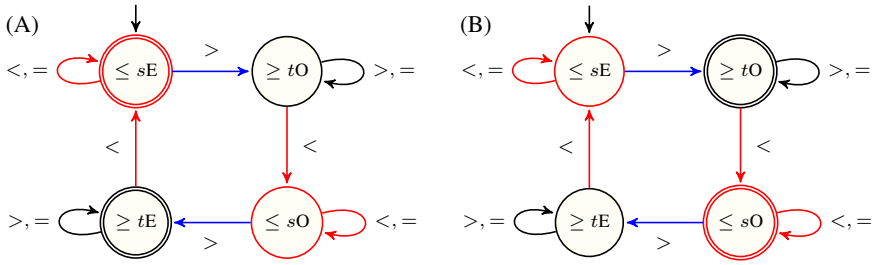
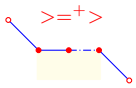


Figure 4.976: Automata without registers for the (A) NB_DECREASING_SEQUENCE_IS_EVEN and the (B) NB_DECREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the DECREASING_SEQUENCE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the DECREASING_SEQUENCE pattern.

AGGREGATOR FEATURE PATTERN

NB_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint NB_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$

$\texttt{VALUE} \geq 0$

$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\ell - 2)/2\rfloor * \lfloor \texttt{sv}/\ell \rfloor + \texttt{max}(0, \lfloor(\texttt{sv} \bmod \ell - 2)/2\rfloor))$①

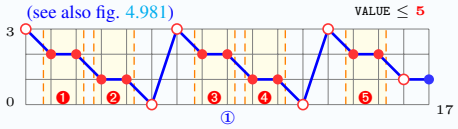
required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

$\ell = \texttt{max}(1, \texttt{min}(\texttt{sv}, 2 * \texttt{rv} - 2))$



Purpose VALUE is the number of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.

Example $(2, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3\rangle)$

Figure 4.977 provides an example where the NB_DECREASING_TERRACE $(2, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$

$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

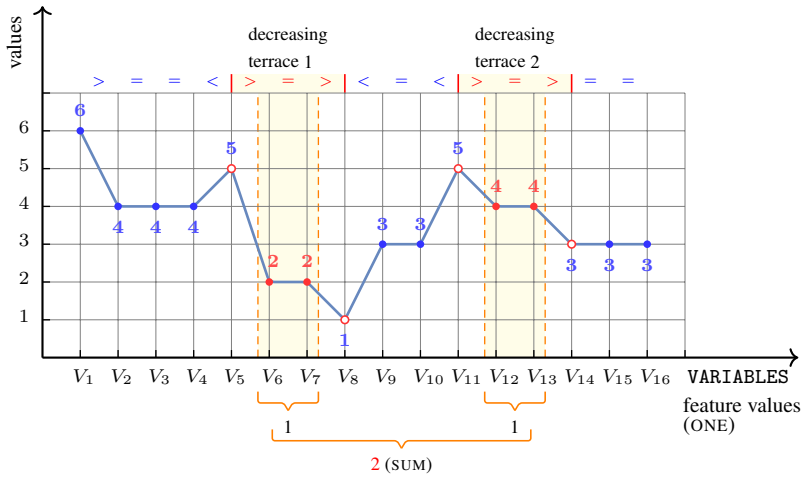


Figure 4.977: Illustrating the NB_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.978 and 4.979 respectively depict the automaton associated with the constraint NB_DECREASING_TERRACE and its simplified form.

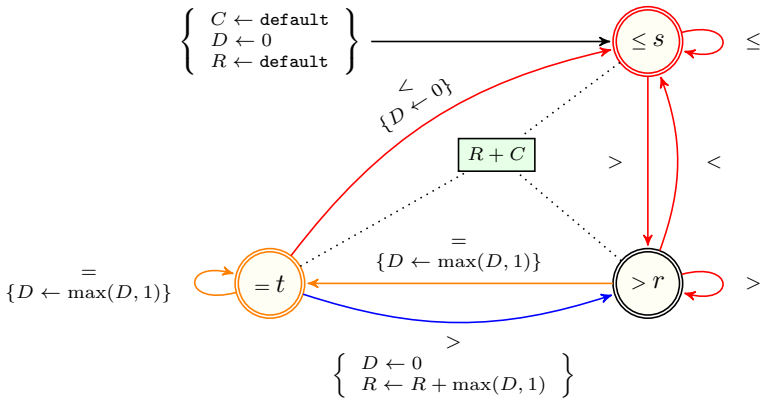


Figure 4.978: Automaton for the NB_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0

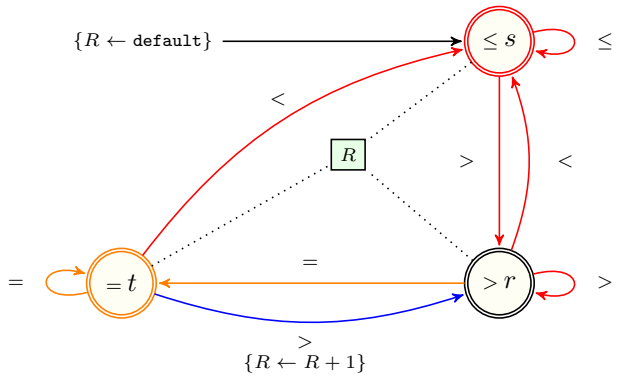


Figure 4.979: Simplified automaton for the NB_DECREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.231: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the NB_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	1 C
t	0	1 C	1 C

Table 4.232: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the NB_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

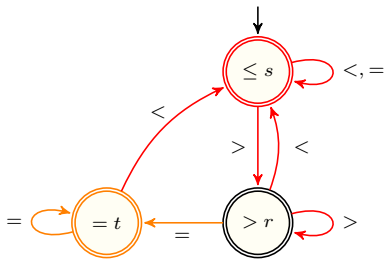


Figure 4.980: Automaton without registers for the NB_DECREASING_TERRACE_EQ_0 constraint; it describes all sequences containing no occurrence of the DECREASING_TERRACE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the DECREASING_TERRACE pattern by removing the register R and the **found** transition, i.e. the transition from state t to state r that increments R.

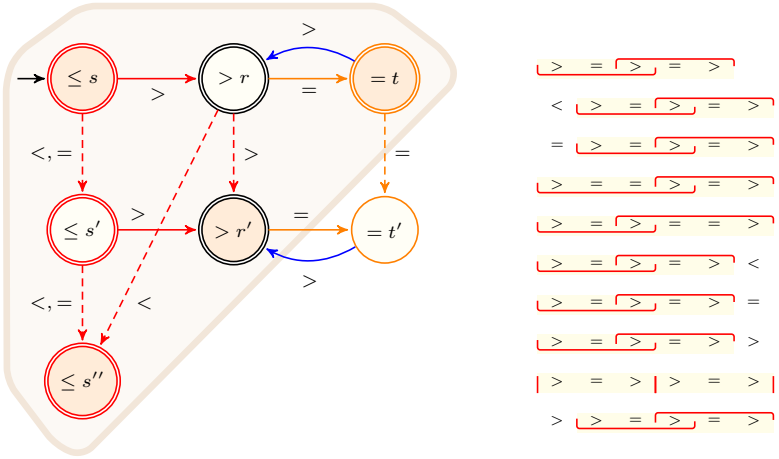


Figure 4.981: (**left**) Automaton without registers for the NB_DECREASING_TERRACE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the DECREASING_TERRACE pattern on a sequence of sv variables, assuming no restriction on the domain of the variables, i.e. the simplified bound $\max(0, \lfloor \frac{\mathtt{sv}-2}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the DECREASING_TERRACE pattern, dashed transitions to slack, and accepting states have a light-brown background; states r and s' are accepting when $\mathtt{sv} \bmod 2 = 0$, states s, t, r' and s'' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6\}$.

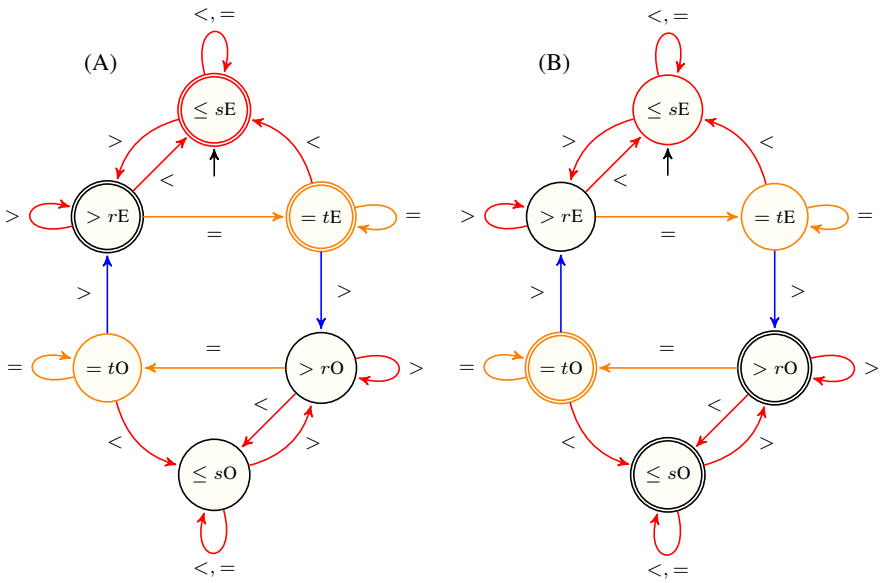
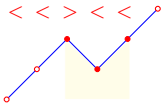


Figure 4.982: Automata without registers for the (A) NB_DECREASING_TERRACE_IS_EVEN and the (B) NB_DECREASING_TERRACE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the DECREASING_TERRACE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the DECREASING_TERRACE pattern.

AGGREGATOR FEATURE PATTERN

NB_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint NB_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

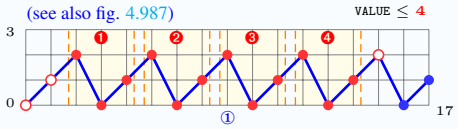
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \lfloor(\texttt{sv} - 3)/3\rfloor$①)
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)

(see also fig. 4.987)



Purpose VALUE is the number of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.

Example $(2, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.983 provides an example where the NB_DIP_ON_INCREASING_SEQUENCE $(2, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

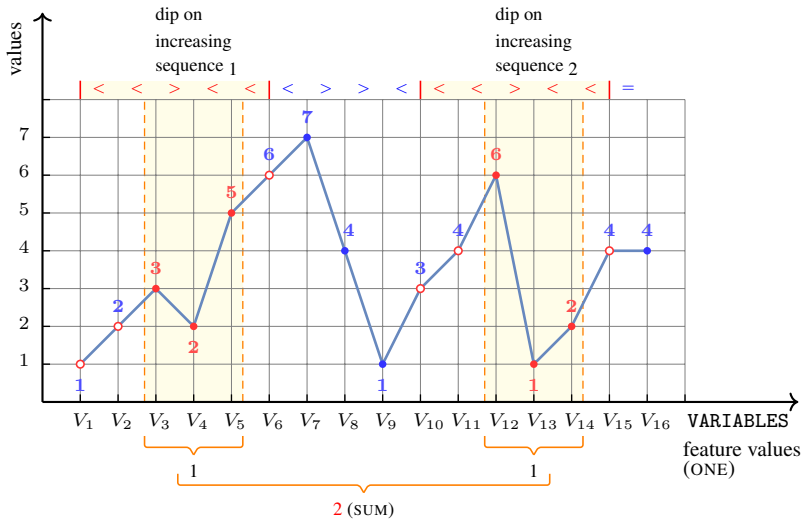


Figure 4.983: Illustrating the NB_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.984 and 4.985 respectively depict the automaton associated with the constraint NB_DIP_ON_INCREASING_SEQUENCE and its simplified form.

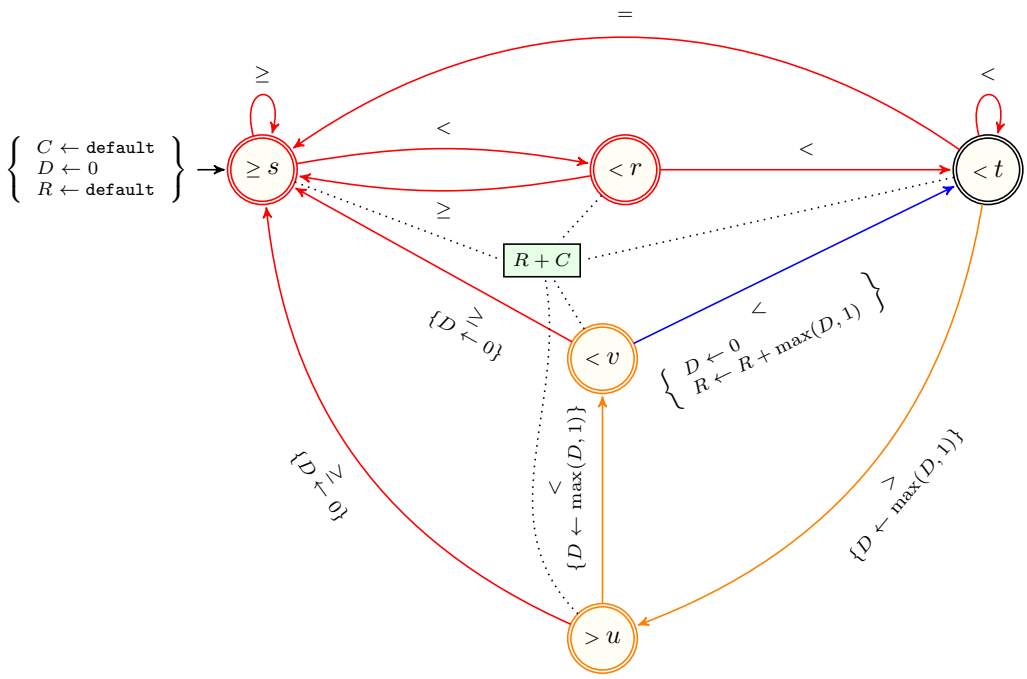


Figure 4.984: Automaton for the NB_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

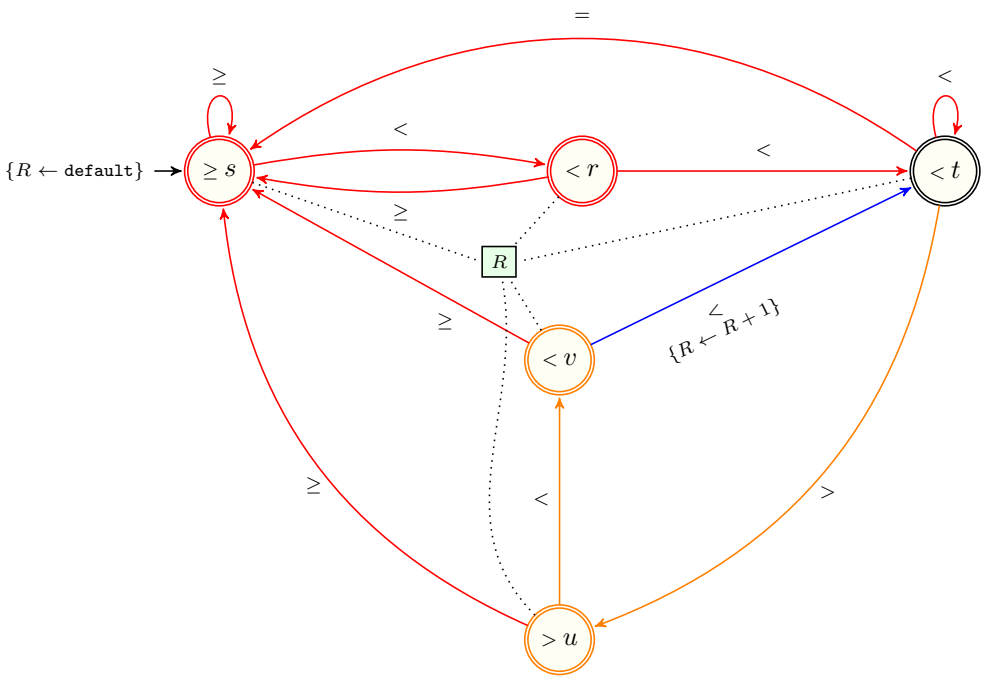


Figure 4.985: Simplified automaton for the NB_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-3} + 1 \geq 0$ are linear invariants.

Specialisation

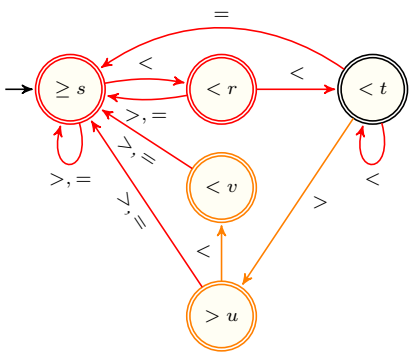
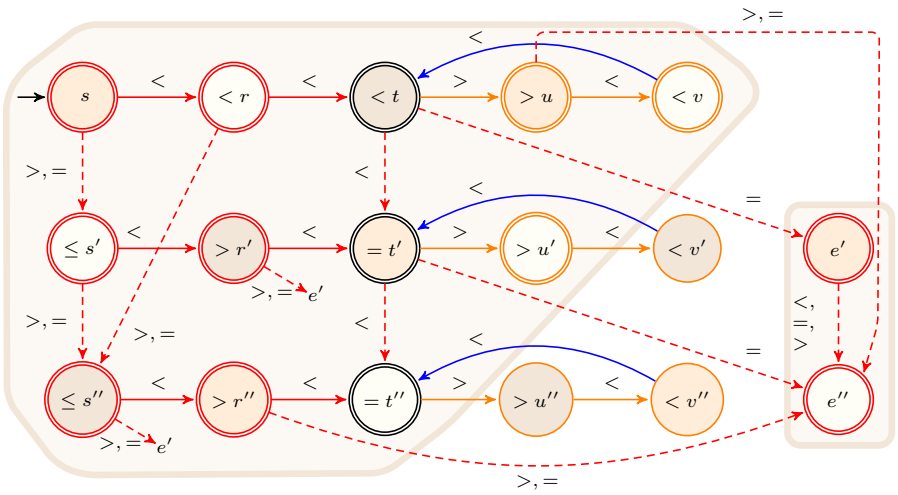


Figure 4.986: Automaton without registers for the NB_DIP_ON_INCREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the DIP_ON_INCREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern by removing the register R and the **found** transition, i.e. the transition from state v to state t that increments R.



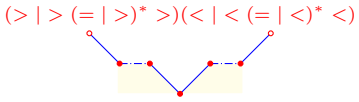
| < < > < < |	| < < > < < | < =	= | < < > < < | =
< | < < > < < |	| < < > < < | < >	= | < < > < < | >
| < < > < < | <	| < < > < < | = <	= = | < < > < < |
| < < > < < | =	| < < > < < | = =	= > | < < > < < |
| < < > < < | >	| < < > < < | = >	> < | < < > < < |
= | < < > < < |	| < < > < < | > <	> | < < > < < | <
> | < < > < < |	| < < > < < | > =	> | < < > < < | =
< < | < < > < < |	| < < > < < | > >	> | < < > < < | >
< | < < > < < | <	< = | < < > < < |	> = | < < > < < |
< | < < > < < | =	< > | < < > < < |	> > | < < > < < |
< | < < > < < | >	= < | < < > < < |	
| < < > < < | < <	= | < < > < < | <	

Figure 4.987: (**top**) Automaton without registers for the NB_DIP_ON_INCREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-3}{3} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the DIP_ON_INCREASING_SEQUENCE pattern, dashed transitions correspond to slack, and accepting states have a light-brown background; states s, u, t', e', r'' are accepting when $\mathtt{sv} \bmod 3 = 1$, states r, v, s', u', t'', e'' are accepting when $\mathtt{sv} \bmod 3 = 2$, states t, r', s'' are accepting when $\mathtt{sv} \bmod 3 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6, 7\}$.

AGGREGATOR FEATURE PATTERN

NB_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the GORGE pattern.

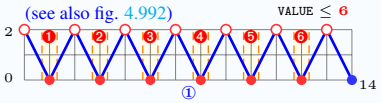
Constraint NB_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor$①)
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0. An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.

Example $(3, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7\rangle)$

Figure 4.988 provides an example where the NB_GORGE $(3, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

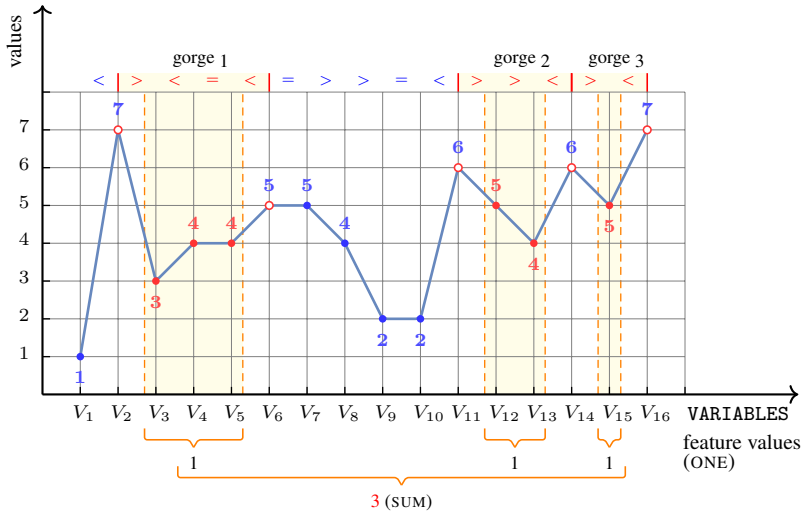


Figure 4.988: Illustrating the NB_GORGE constraint of the **Example** slot

Automaton Figures 4.989 and 4.990 respectively depict the automaton associated with the constraint NB_GORGE and its simplified form.

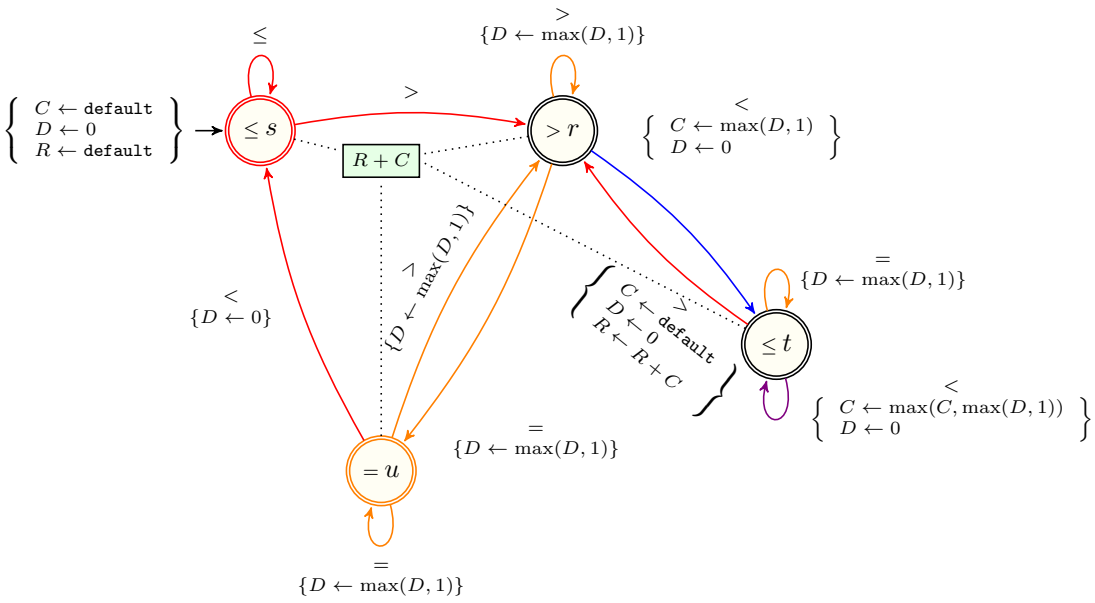


Figure 4.989: Automaton for the NB_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

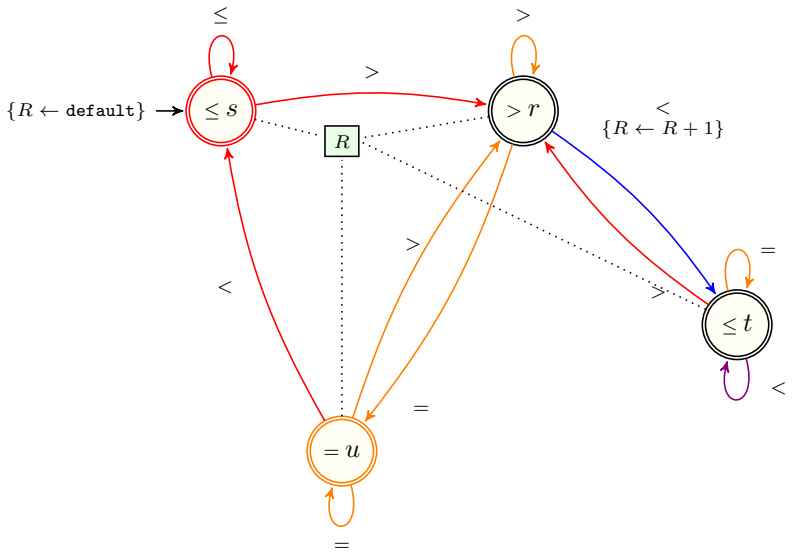


Figure 4.990: Simplified automaton for the NB_GORGE constraint obtained by applying decoration Table 3.39 to the seed transducer of the GORGE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	1 L	$\overrightarrow{C} + \overleftarrow{C}$	1 L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.233: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the NB_GORGE constraint defined as the composition of the GORGE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	1 C	0 R	0
t	0	0 L	0	0 L
u	0	0	0 R	0

Table 4.234: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the NB_GORGE constraint defined as the composition of the GORGE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

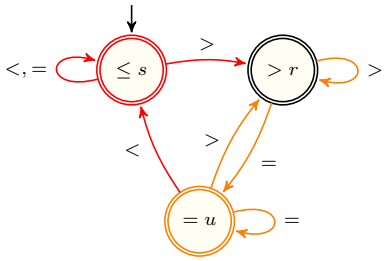


Figure 4.991: Automaton without registers for the NB_GORGE_EQ_0 constraint; it describes all sequences containing no occurrence of the GORGE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the GORGE pattern by removing the register R, the **found** transition from state r to t that increments R, and the state t that becomes unreachable after removing transition $r \rightarrow t$.

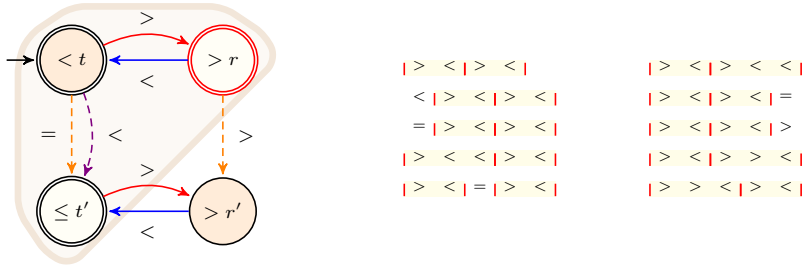


Figure 4.992: (**left**) Automaton without registers for the NB_GORGE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the GORGE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}-1}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the GORGE pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

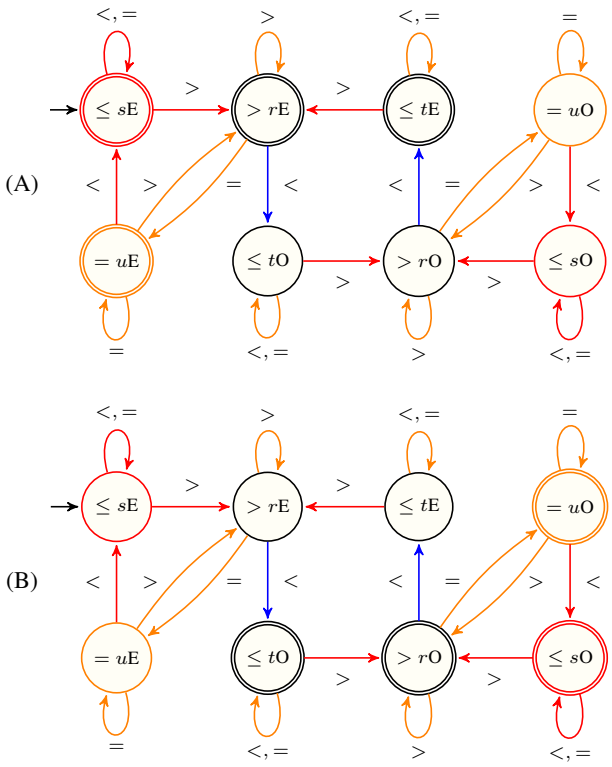


Figure 4.993: Automata without registers for the (A) NB_GORGE_IS_EVEN and the (B) NB_GORGE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the GORGE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the GORGE pattern.

AGGREGATOR FEATURE PATTERN

NB_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

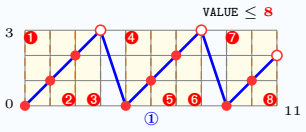
Constraint NB_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, (\ell - 1) * \lfloor \texttt{sv}/\ell \rfloor + \texttt{max}(0, \texttt{sv} \bmod \ell - 1))$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\ell = \texttt{min}(\texttt{sv}, \texttt{rv})$



Purpose VALUE is the number of occurrences of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.994 provides an example where the NB_INCREASING $(5, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

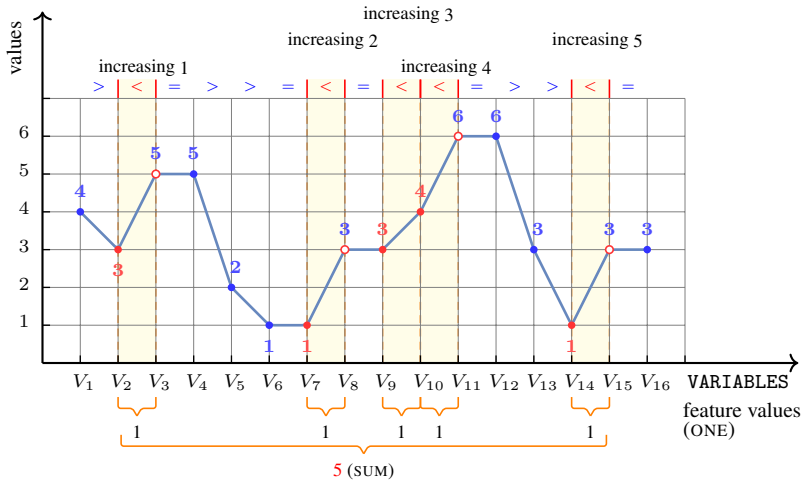


Figure 4.994: Illustrating the NB_INCREASING constraint of the **Example** slot

Automaton

Figures 4.995 and 4.996 respectively depict the automaton associated with the constraint NB_INCREASING and its simplified form.

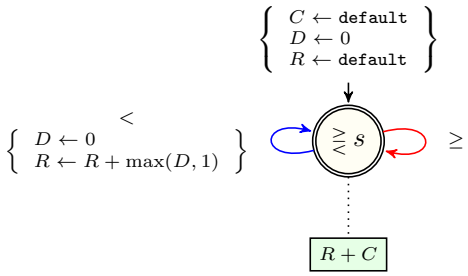


Figure 4.995: Automaton for the NB_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is 0

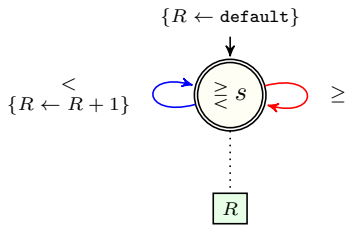


Figure 4.996: Simplified automaton for the NB_INCREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 1 \geq 0$ are linear invariants.

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.235: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the NB_INCREASING constraint defined as the composition of the INCREASING pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.236: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the NB_INCREASING constraint defined as the composition of the INCREASING pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

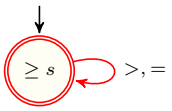


Figure 4.997: Automaton without registers for the NB_INCREASING_EQ_0 constraint; it describes all sequences containing no occurrence of the INCREASING pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the INCREASING pattern by removing the register R and the **found** transition, i.e. the transition from state s to state s that increments R.

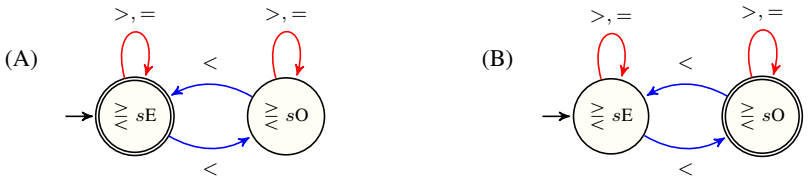
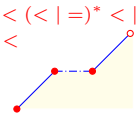


Figure 4.998: Automata without registers for the (A) NB_INCREASING_IS_EVEN and the (B) NB_INCREASING_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the INCREASING pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the INCREASING pattern.

AGGREGATOR FEATURE PATTERN

NB_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

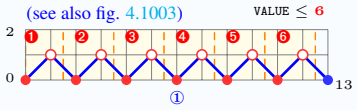
Constraint NB_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.999 provides an example where the NB_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

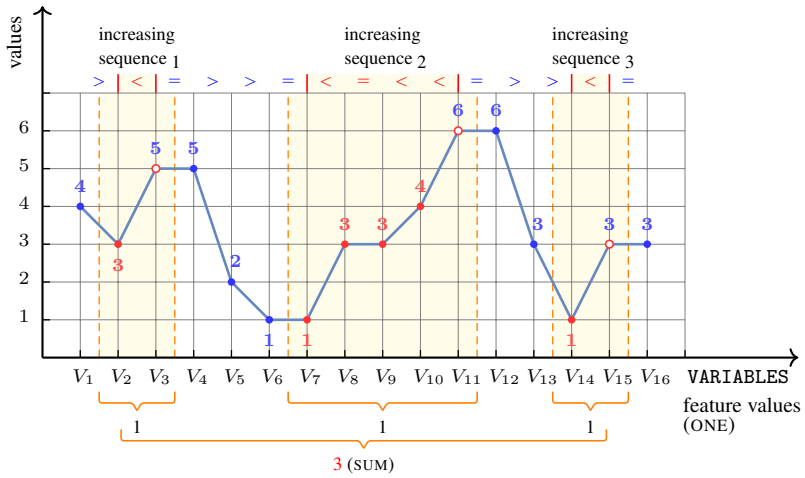


Figure 4.999: Illustrating the NB_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1000 and 4.1001 respectively depict the automaton associated with the constraint NB_INCREASING_SEQUENCE and its simplified form.

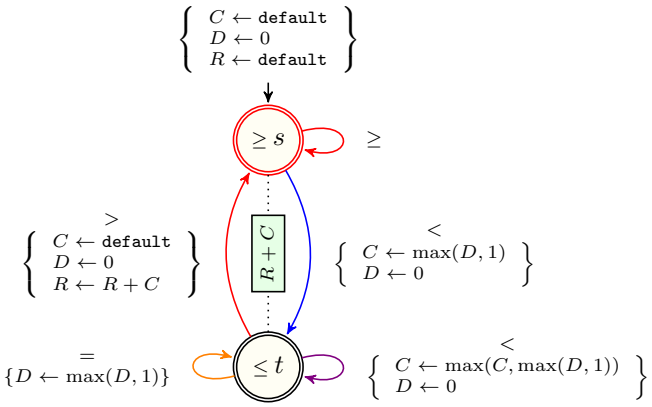


Figure 4.1000: Automaton for the NB_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

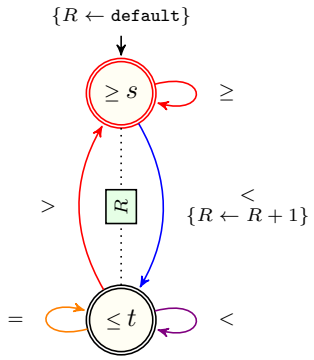


Figure 4.1001: Simplified automaton for the NB_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	1 M

Table 4.237: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the NB_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	0	0
t	0	-1 M

Table 4.238: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the NB_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

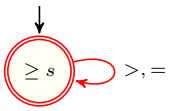


Figure 4.1002: Automaton without registers for the NB_INCREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the INCREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the INCREASING_SEQUENCE pattern by removing the register R, the **found** transition from state s to t that increments R, and the state t that becomes unreachable after removing transition $s \rightarrow t$.

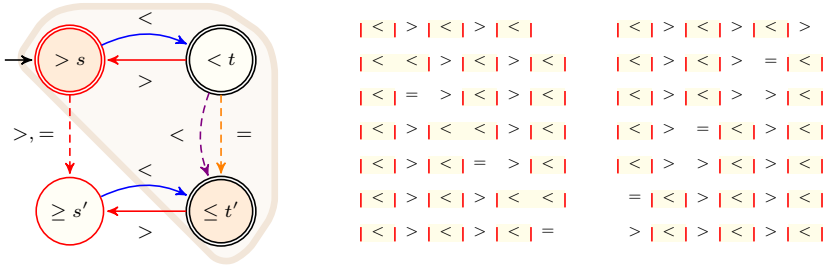


Figure 4.1003: (**left**) Automaton without registers for the NB_INCREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the INCREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the INCREASING_SEQUENCE pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 0$, while states s and t' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6\}$.

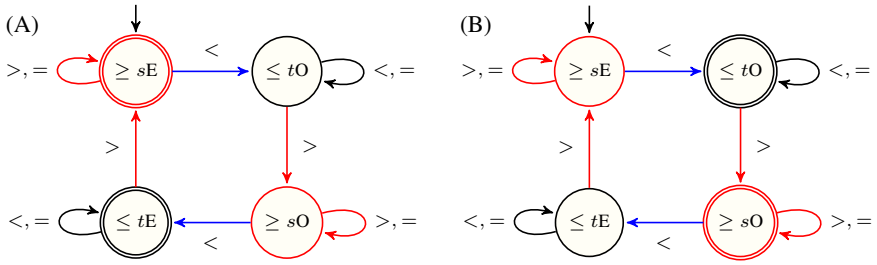
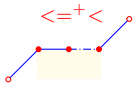


Figure 4.1004: Automata without registers for the (A) NB_INCREASING_SEQUENCE_IS_EVEN and the (B) NB_INCREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the INCREASING_SEQUENCE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the INCREASING_SEQUENCE pattern.

AGGREGATOR FEATURE PATTERN

NB_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint NB_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$

$\texttt{VALUE} \geq 0$

$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\ell - 2)/2\rfloor * \lfloor \texttt{sv}/\ell \rfloor + \texttt{max}(0, \lfloor(\texttt{sv} \bmod \ell - 2)/2\rfloor))$①

required(VARIABLES, var)

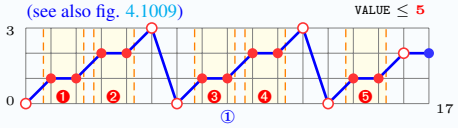
where

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

$\ell = \texttt{max}(1, \texttt{min}(\texttt{sv}, 2 * \texttt{rv} - 2))$

(see also fig. 4.1009)



Purpose VALUE is the number of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^+<$'.

Example $(2, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4\rangle)$

Figure 4.1005 provides an example where the NB_INCREASING_TERRACE $(2, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

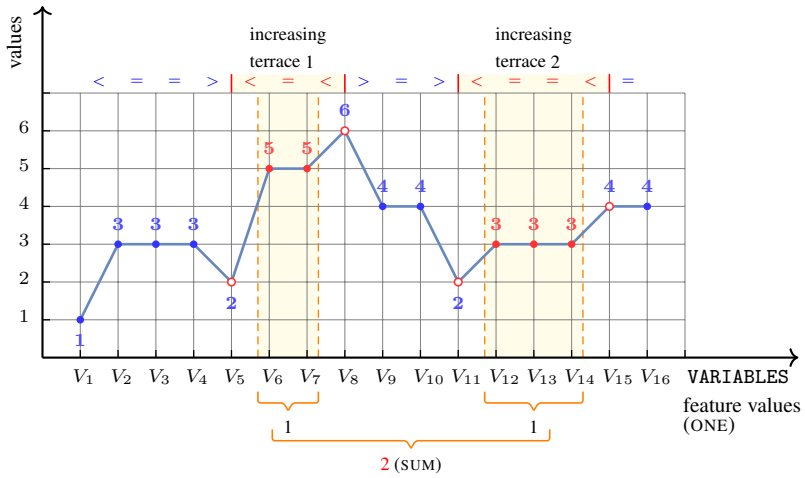


Figure 4.1005: Illustrating the NB_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.1006 and 4.1007 respectively depict the automaton associated with the constraint NB_INCREASING_TERRACE and its simplified form.

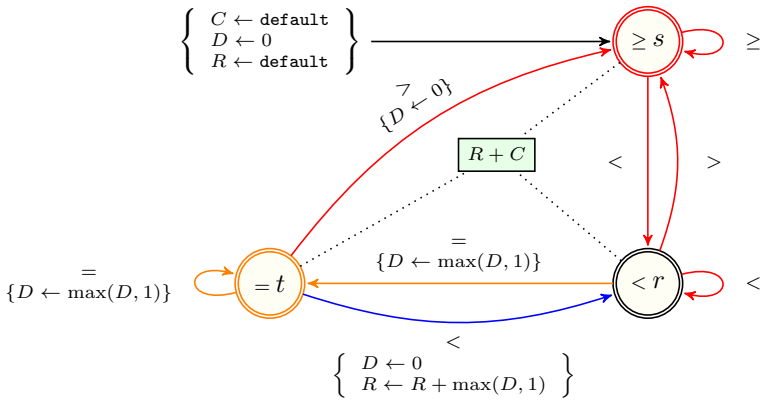


Figure 4.1006: Automaton for the NB_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0

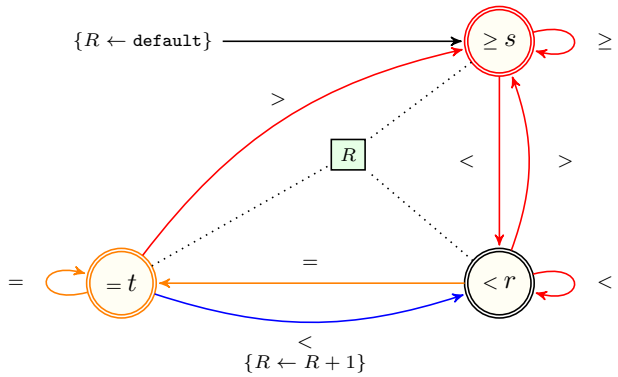


Figure 4.1007: Simplified automaton for the NB_INCREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.239: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the NB_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	1 C
t	0	1 C	1 C

Table 4.240: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the NB_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

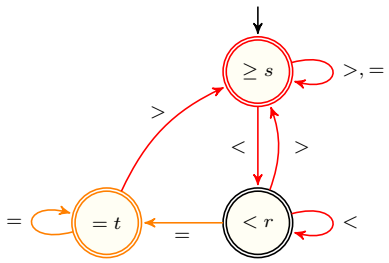


Figure 4.1008: Automaton without registers for the NB_INCREASING_TERRACE_EQ_0 constraint; it describes all sequences containing no occurrence of the INCREASING_TERRACE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the INCREASING_TERRACE pattern by removing the register R and the **found** transition, i.e. the transition from state t to state r that increments R.

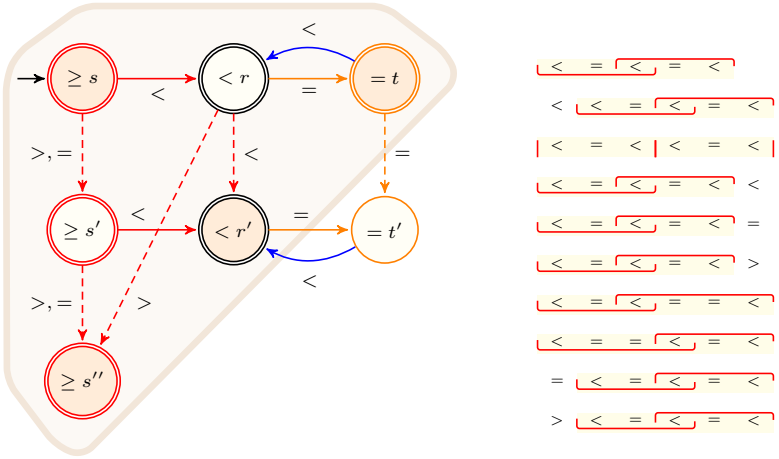


Figure 4.1009: (**left**) Automaton without registers for the NB_INCREASING_TERRACE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the INCREASING_TERRACE pattern on a sequence of sv variables, assuming no restriction on the domain of the variables, i.e. the simplified bound $\max(0, \lfloor \frac{\mathtt{sv}-2}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the INCREASING_TERRACE pattern, dashed transitions to slack, and accepting states have a light-brown background; states r and s' are accepting when $\mathtt{sv} \bmod 2 = 0$, states s, t, r' and s'' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{5, 6\}$.

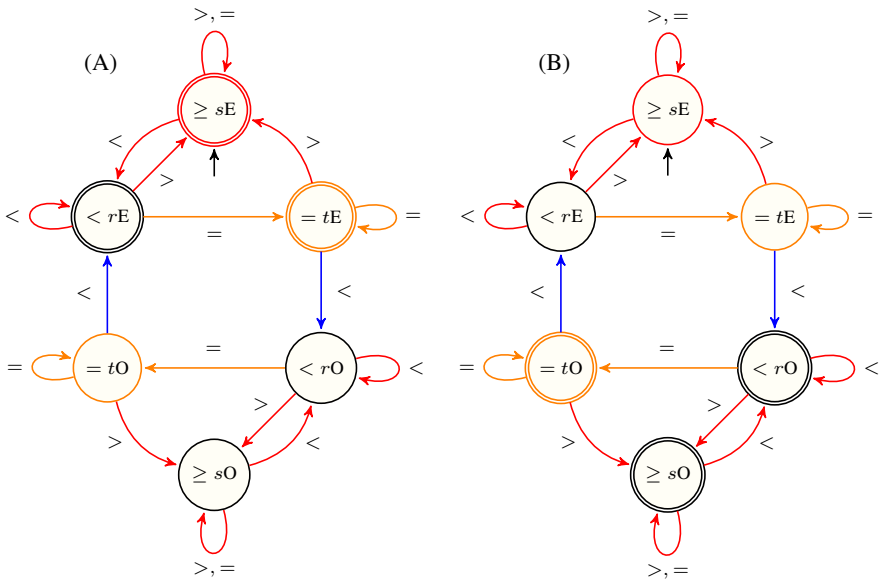
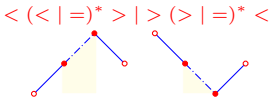


Figure 4.1010: Automata without registers for the (A) NB_INCREASING_TERRACE_IS_EVEN and the (B) NB_INCREASING_TERRACE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the INCREASING_TERRACE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the INCREASING_TERRACE pattern.

AGGREGATOR FEATURE PATTERN

NB_INFLEXION



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INFLEXION pattern.

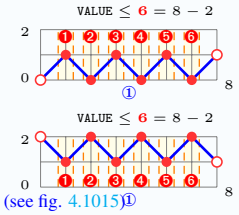
Constraint NB_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see fig. 4.1015)①

Purpose VALUE is the number of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.

Example $(8, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.1011 provides an example where the NB_INFLEXION $(8, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

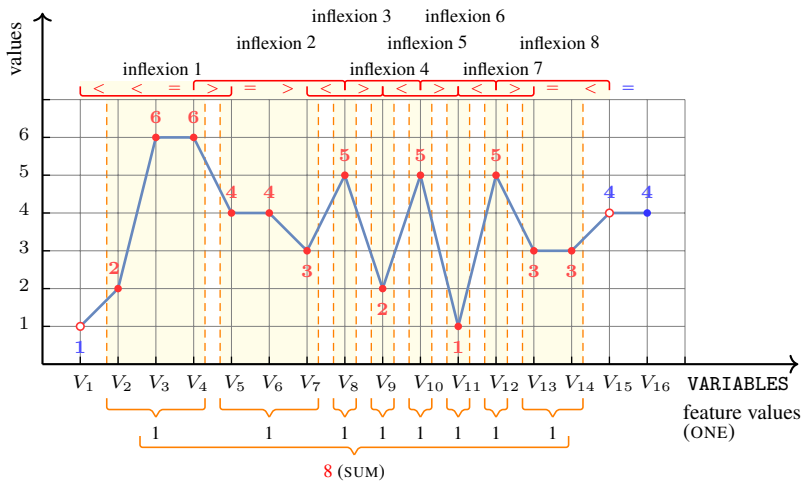


Figure 4.1011: Illustrating the NB_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.1012 and 4.1013 respectively depict the automaton associated with the constraint NB_INFLEXION and its simplified form.

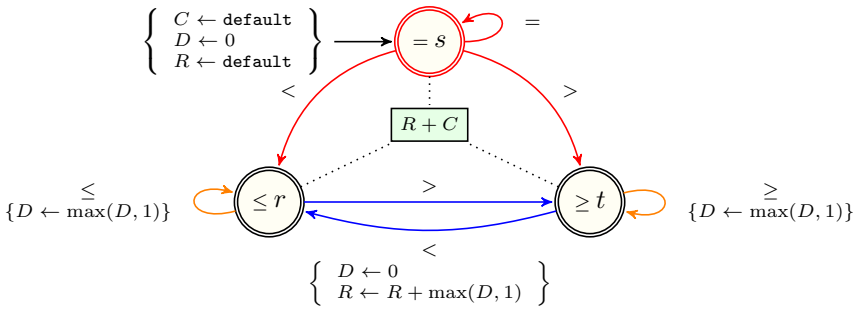


Figure 4.1012: Automaton for the NB_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

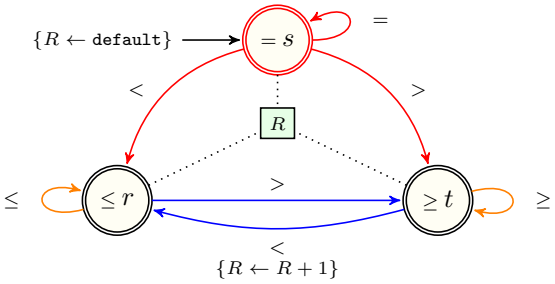


Figure 4.1013: Simplified automaton for the NB_INFLEXION constraint obtained by applying decoration Table 3.39 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same register update as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 1 \geq 0$ are linear invariants.

Specialisation

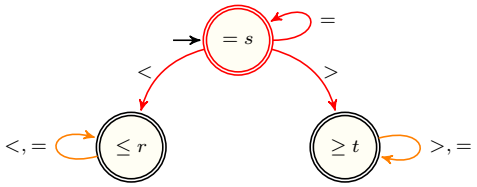


Figure 4.1014: Automaton without registers for the NB_INFLEXION_EQ_0 constraint; it describes all sequences containing no occurrence of the INFLEXION pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the INFLEXION pattern by removing the register R and the transitions labelled by **found**, i.e. transitions $r \to t$ and $t \to r$ that increment R.

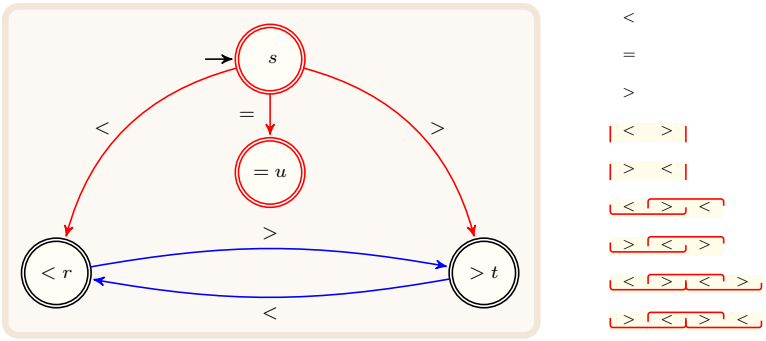


Figure 4.1015: (**left**) Automaton without registers for the NB_INFLEXION_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the INFLEXION pattern on a sequence of sv variables, i.e. $\max(0, \texttt{sv} - 2)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the INFLEXION pattern; the three accepting states have a light-brown background. (**right**) All corresponding solutions for $\texttt{sv} - 1 \in \{1, 2, 3, 4\}$.

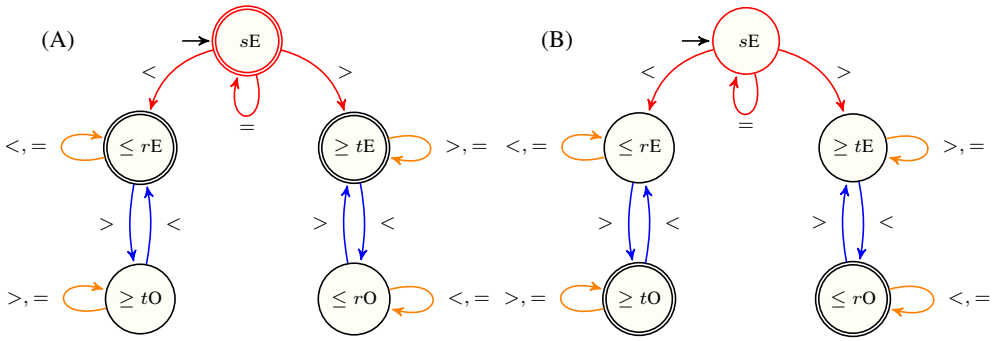
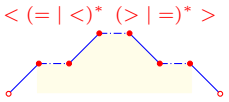


Figure 4.1016: Automata without registers for the (A) NB_INFLEXION_IS_EVEN and the (B) NB_INFLEXION_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the INFLEXION pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the INFLEXION pattern.

AGGREGATOR FEATURE PATTERN

NB_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

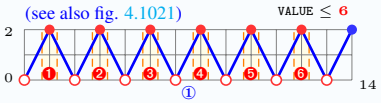
Constraint NB_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0. An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.

Example $(3, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1\rangle)$

Figure 4.1017 provides an example where the NB_PEAK $(3, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

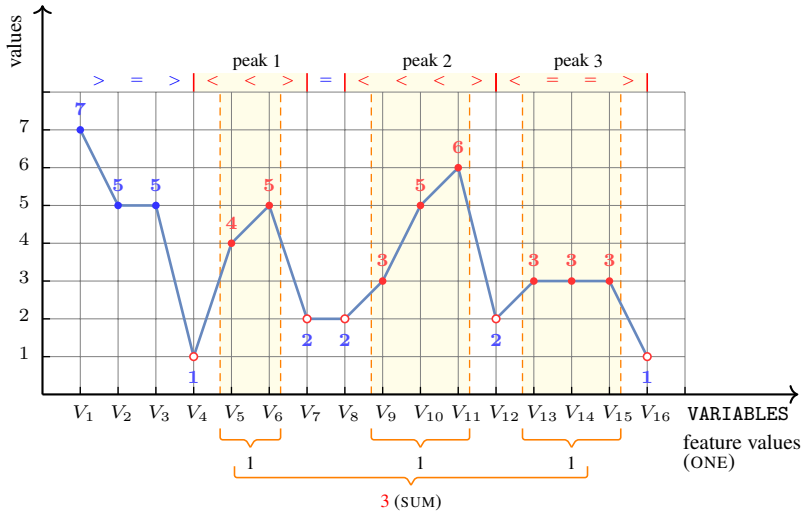


Figure 4.1017: Illustrating the NB_PEAK constraint of the **Example** slot

Automaton

Figures 4.1018 and 4.1019 respectively depict the automaton associated with the constraint NB_PEAK and its simplified form.

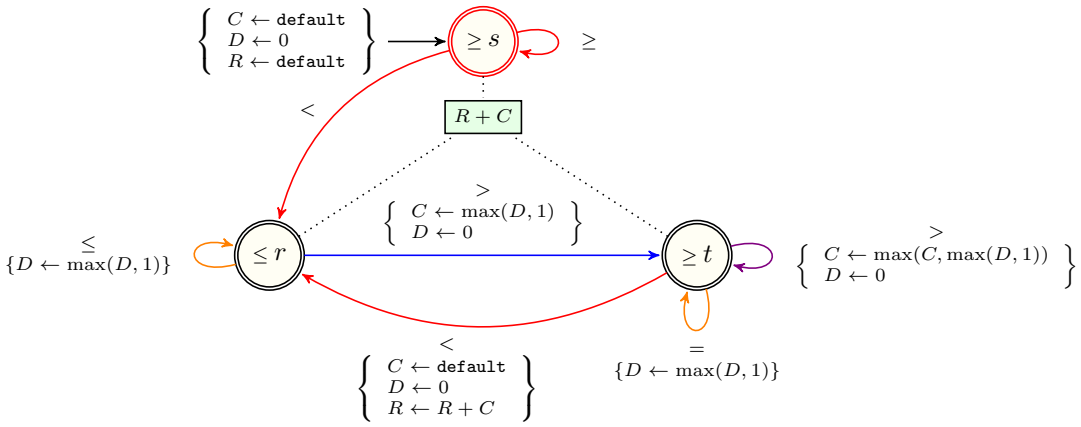


Figure 4.1018: Automaton for the NB_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is 0

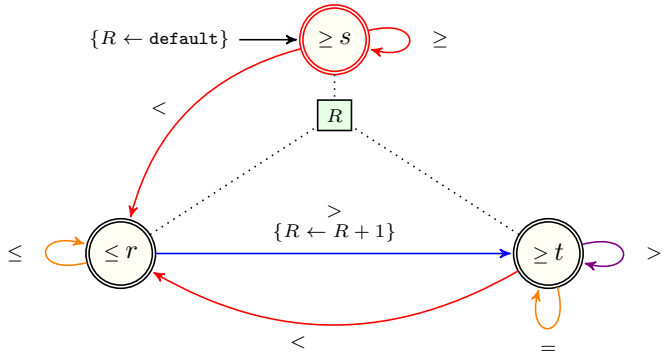


Figure 4.1019: Simplified automaton for the NB_PEAK constraint obtained by applying decoration Table 3.39 to the seed transducer of the PEAK pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 R
t	$\overrightarrow{C} + \overleftarrow{C}$	1 L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.241: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the NB_PEAK constraint defined as the composition of the PEAK pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	1 C	0 R
t	0	0 L	0

Table 4.242: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the NB_PEAK constraint defined as the composition of the PEAK pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

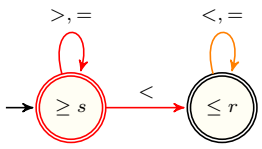


Figure 4.1020: Automaton without registers for the NB_PEAK_EQ_0 constraint; it describes all sequences containing no occurrence of the PEAK pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the PEAK pattern by removing the register R, the **found** transition from state r to t that increments R, and the state t that becomes unreachable after removing transition $r \to t$.

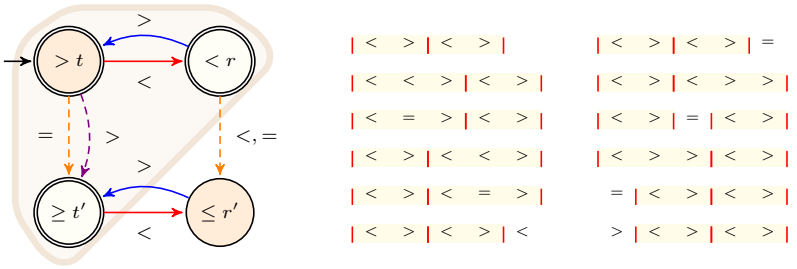


Figure 4.1021: (**left**) Automaton without registers for the NB_PEAK_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the PEAK pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the PEAK pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

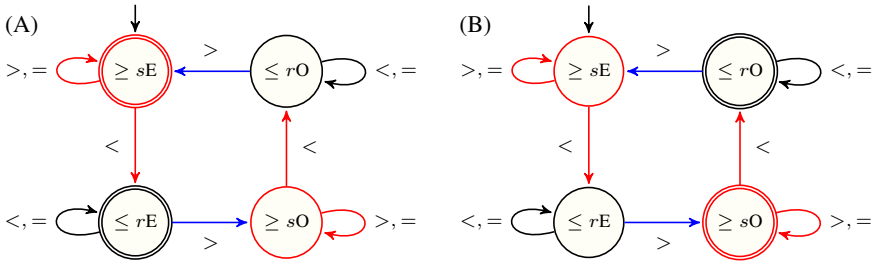


Figure 4.1022: Automata without registers for the (A) NB_PEAK_IS_EVEN and the (B) NB_PEAK_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the PEAK pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the PEAK pattern.

AGGREGATOR FEATURE PATTERN

NB_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

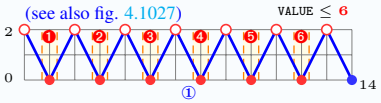
Constraint NB_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0. An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.

Example $(3, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3\rangle)$

Figure 4.1023 provides an example where the NB_PLAIN $(3, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

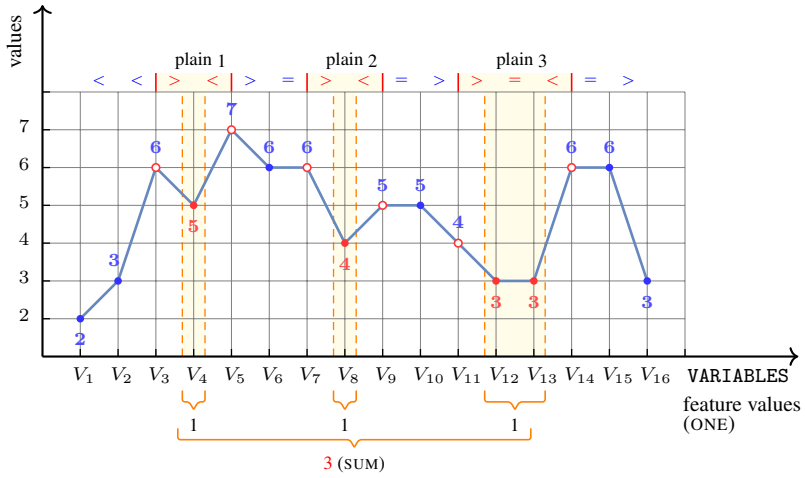


Figure 4.1023: Illustrating the NB_PLAIN constraint of the **Example** slot

Automaton

Figures 4.1024 and 4.1025 respectively depict the automaton associated with the constraint NB_PLAIN and its simplified form.

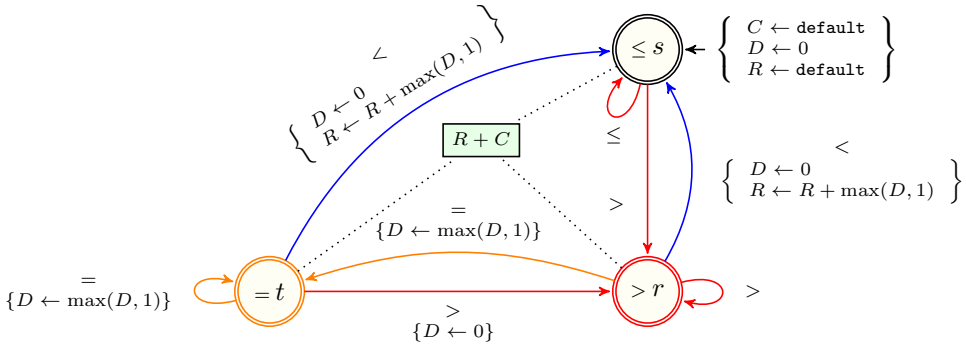


Figure 4.1024: Automaton for the NB_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is 0

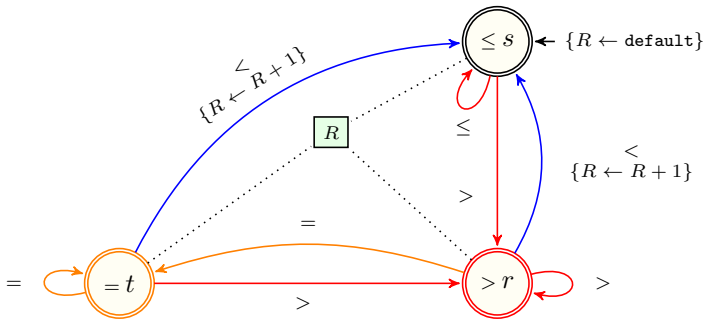


Figure 4.1025: Simplified automaton for the NB_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLAIN pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.243: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the NB_PLAIN constraint defined as the composition of the PLAIN pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	1 C	1 C
t	0	1 C	1 C

Table 4.244: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the NB_PLAIN constraint defined as the composition of the PLAIN pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

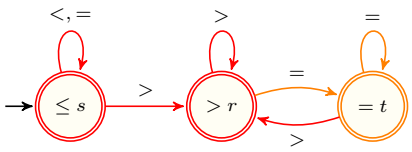


Figure 4.1026: Automaton without registers for the NB_PLAIN_EQ_0 constraint; it describes all sequences containing no occurrence of the PLAIN pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the PLAIN pattern by removing the register R and the transitions labelled by **found**, i.e. transitions $r \to s$ and $t \to s$ that increment R.

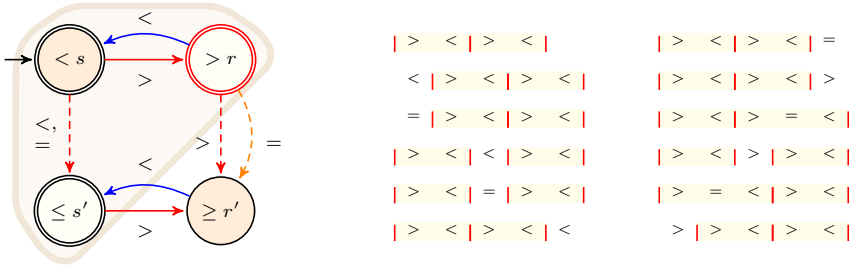


Figure 4.1027: (**left**) Automaton without registers for the NB_PLAIN_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the PLAIN pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the PLAIN pattern, dashed transitions to slack, and accepting states have a light-brown background; state s is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and s' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

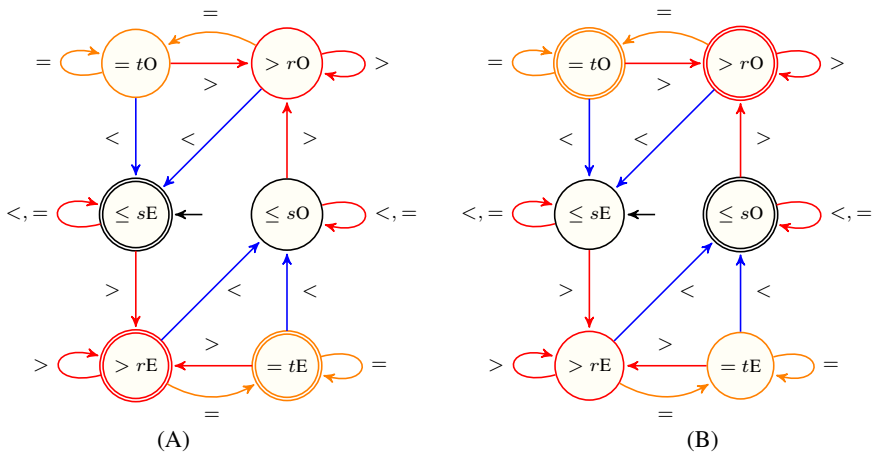


Figure 4.1028: Automata without registers for the (A) NB_PLAIN_IS_EVEN and the (B) NB_PLAIN_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the PLAIN pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the PLAIN pattern.

AGGREGATOR FEATURE PATTERN

NB_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

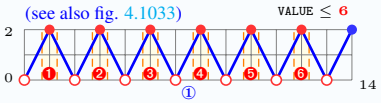
Constraint NB_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.

Example $(3, \langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5\rangle)$

Figure 4.1029 provides an example where the NB_PLATEAU $(3, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

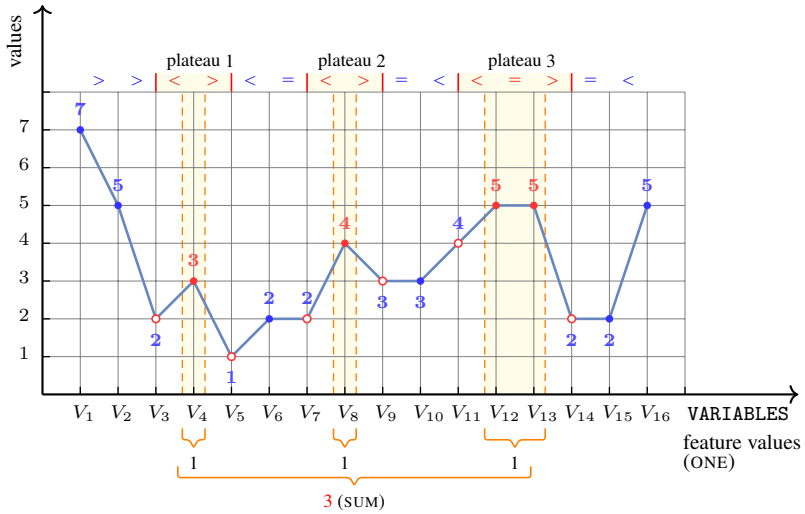


Figure 4.1029: Illustrating the NB_PLATEAU constraint of the **Example** slot

Automaton Figures 4.1030 and 4.1031 respectively depict the automaton associated with the constraint NB_PLATEAU and its simplified form.

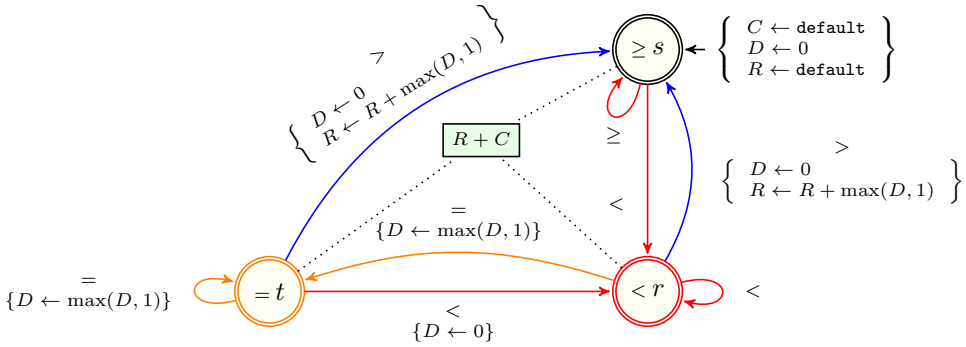


Figure 4.1030: Automaton for the NB_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is 0

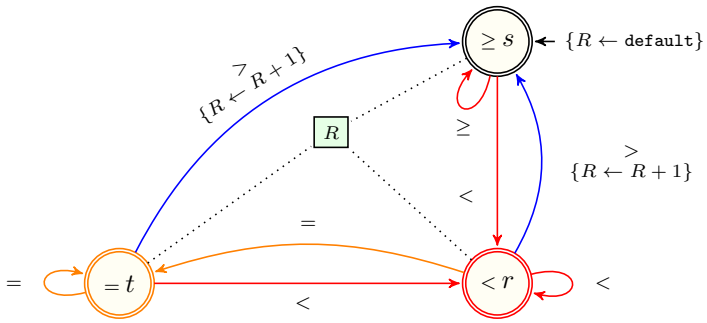


Figure 4.1031: Simplified automaton for the NB_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.245: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the NB_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	1 C	1 C
t	0	1 C	1 C

Table 4.246: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the NB_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

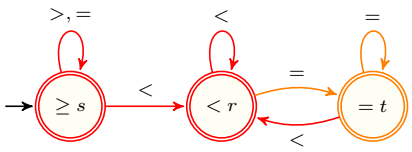


Figure 4.1032: Automaton without registers for the NB_PLATEAU_EQ_0 constraint; it describes all sequences containing no occurrence of the PLATEAU pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the PLATEAU pattern by removing the register R and the transitions labelled by **found**, i.e. transitions $r \to s$ and $t \to s$ that increment R.

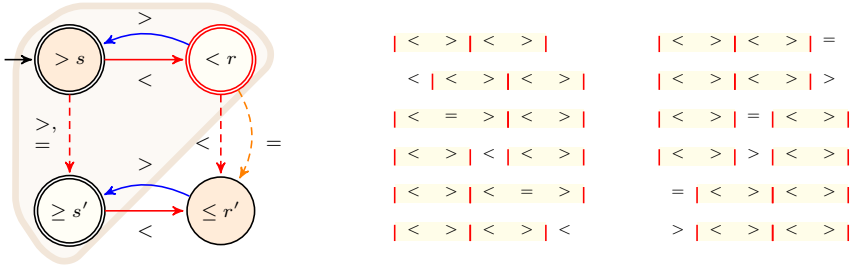


Figure 4.1033: (**left**) Automaton without registers for the NB_PLATEAU_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the PLATEAU pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the PLATEAU pattern, dashed transitions to slack, and accepting states have a light-brown background; state s is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and s' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

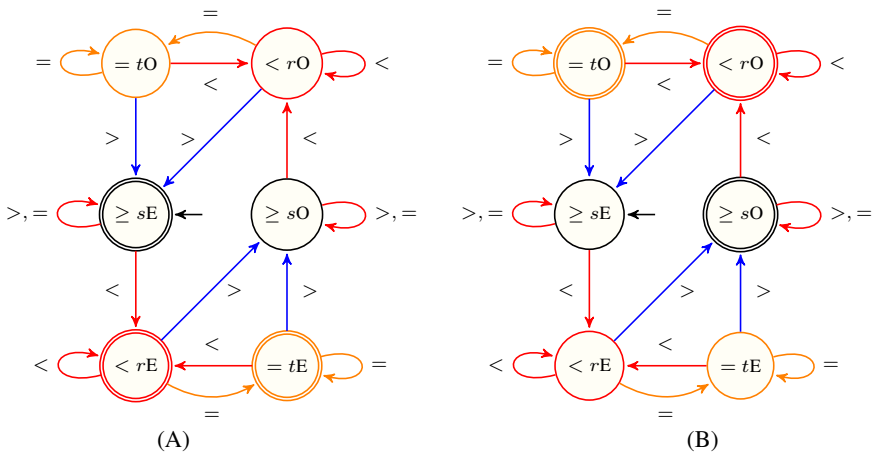
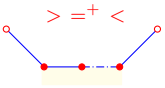


Figure 4.1034: Automata without registers for the (A) NB_PLATEAU_IS_EVEN and the (B) NB_PLATEAU_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the PLATEAU pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the PLATEAU pattern.

AGGREGATOR FEATURE PATTERN

NB_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

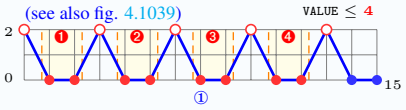
Constraint NB_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/3\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.

Example $(3, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5\rangle)$

Figure 4.1035 provides an example where the NB_PROPER_PLAIN $(3, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

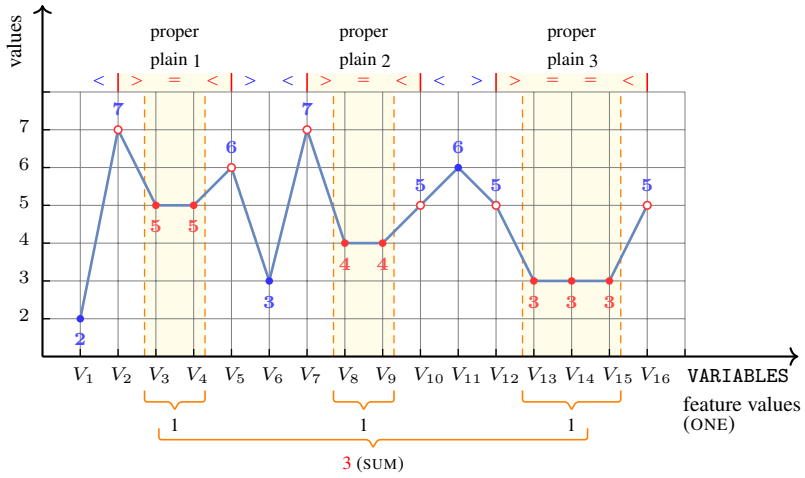


Figure 4.1035: Illustrating the NB_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.1036 and 4.1037 respectively depict the automaton associated with the constraint NB_PROPER_PLAIN and its simplified form.

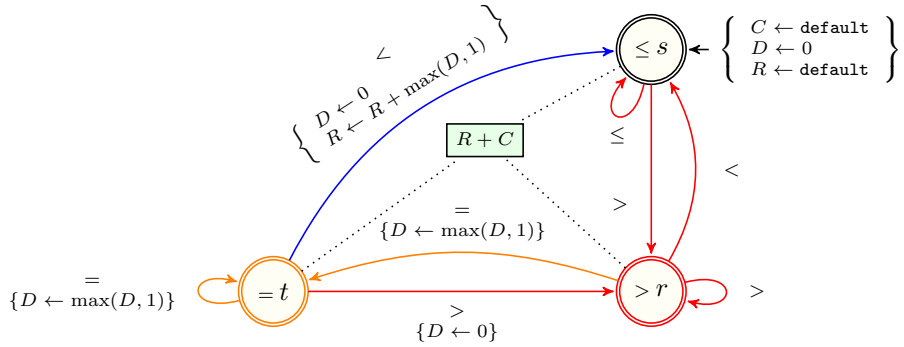


Figure 4.1036: Automaton for the NB_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where default is 0

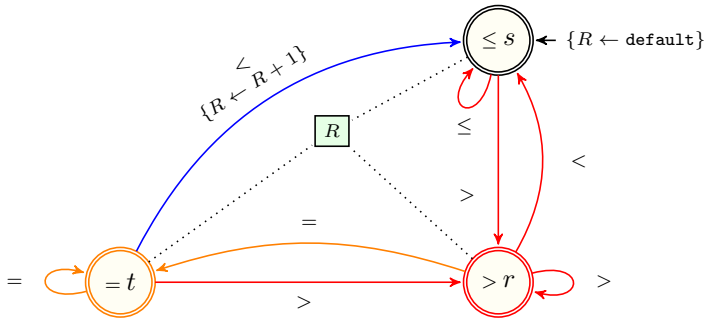


Figure 4.1037: Simplified automaton for the NB_PROPER_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLAIN pattern where default is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-3} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.247: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the NB_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	1 C
t	0	1 C	1 C

Table 4.248: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the NB_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

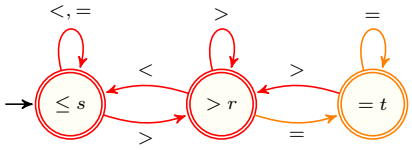


Figure 4.1038: Automaton without registers for the NB_PROPER_PLAIN_EQ_0 constraint; it describes all sequences containing no occurrence of the PROPER_PLAIN pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the PROPER_PLAIN pattern by removing the register R and the **found** transition, i.e. the transition from state t to state s that increments R.

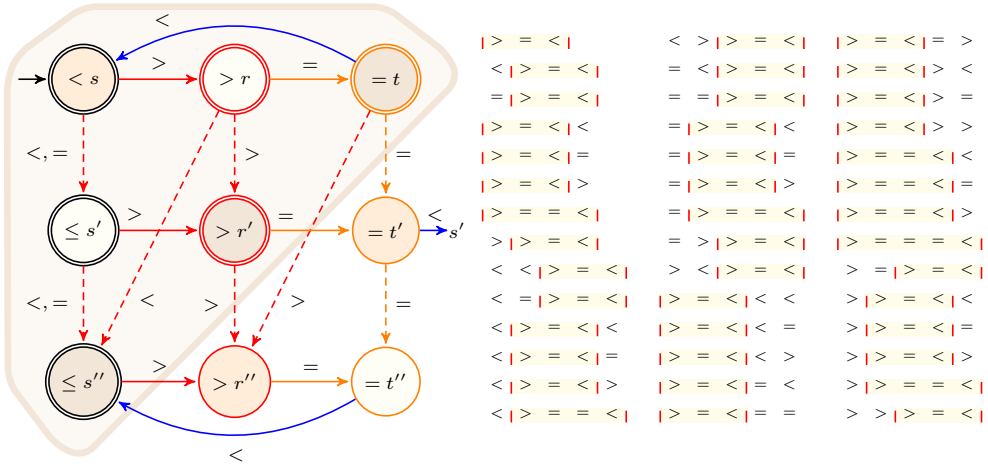


Figure 4.1039: (**left**) Automaton without registers for the NB_PROPER_PLAIN_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the PROPER_PLAIN pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{3} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the PROPER_PLAIN pattern, dashed transitions to slack, and accepting states have a light-brown background; state s is accepting when $\mathtt{sv} \bmod 3 = 1$, states r and s' are accepting when $\mathtt{sv} \bmod 3 = 2$, states t, r', s'' are accepting when $\mathtt{sv} \bmod 3 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4, 5\}$.

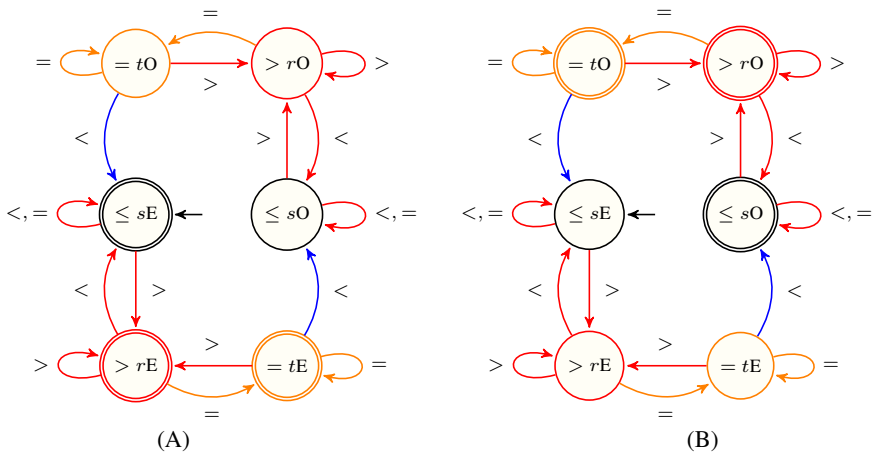
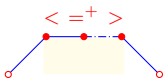


Figure 4.1040: Automata without registers for the (A) NB_PROPER_PLAIN_IS_EVEN and the (B) NB_PROPER_PLAIN_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the PROPER_PLAIN pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the PROPER_PLAIN pattern.

AGGREGATOR FEATURE PATTERN

NB_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

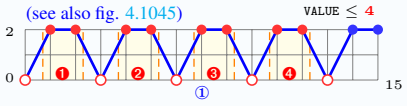
Constraint NB_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \lfloor(\texttt{sv} - 1)/3\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.

Example $(3, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3\rangle)$

Figure 4.1041 provides an example where the NB_PROPER_PLATEAU $(3, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

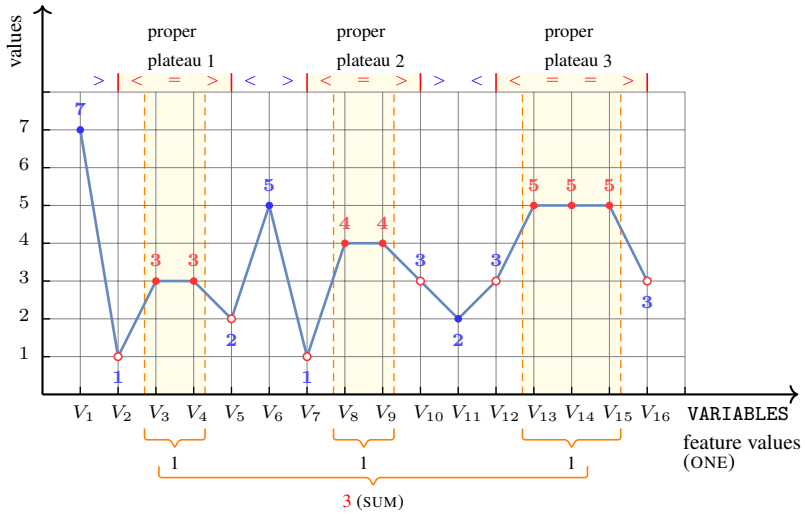


Figure 4.1041: Illustrating the NB_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.1042 and 4.1043 respectively depict the automaton associated with the constraint NB_PROPER_PLATEAU and its simplified form.

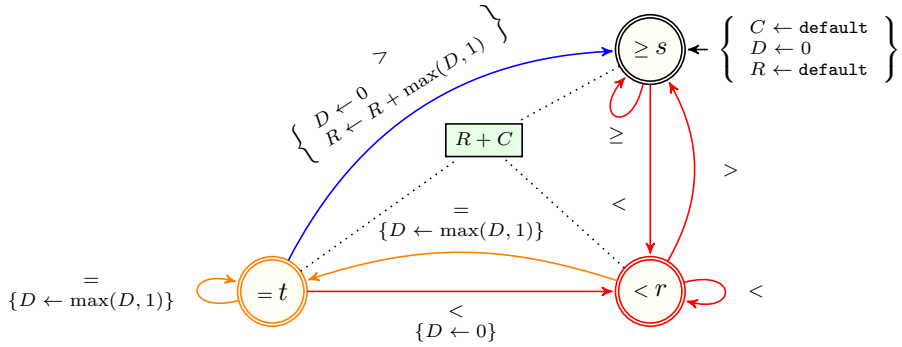


Figure 4.1042: Automaton for the NB_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

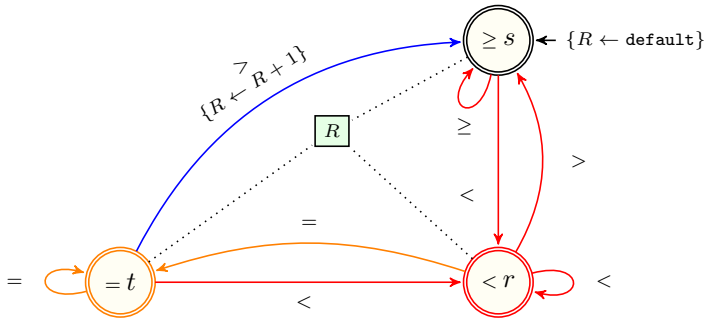


Figure 4.1043: Simplified automaton for the NB_PROPER_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-3} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 C
t	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 C

Table 4.249: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the NB_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	1 C
t	0	1 C	1 C

Table 4.250: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the NB_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

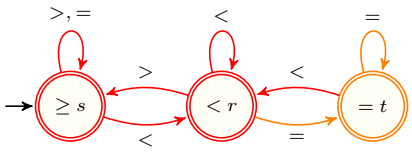


Figure 4.1044: Automaton without registers for the NB_PROPER_PLATEAU_EQ_0 constraint; it describes all sequences containing no occurrence of the PROPER_PLATEAU pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the PROPER_PLATEAU pattern by removing the register R and the **found** transition, i.e. the transition from state t to state s that increments R.

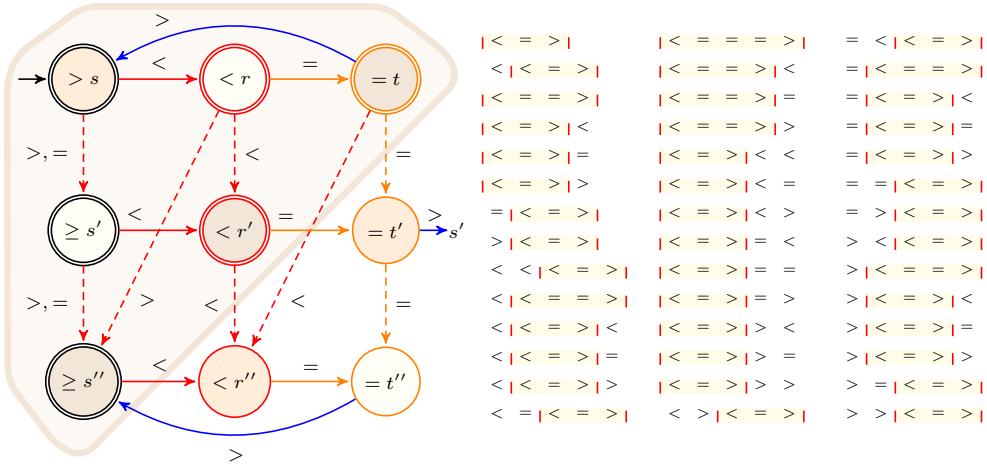


Figure 4.1045: (**left**) Automaton without registers for the NB_PROPER_PLATEAU_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the PROPER_PLATEAU pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{3} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the PROPER_PLATEAU pattern, dashed transitions to slack, and accepting states have a light-brown background; state s is accepting when $\mathtt{sv} \bmod 3 = 1$, states r and s' are accepting when $\mathtt{sv} \bmod 3 = 2$, states t, r', s'' are accepting when $\mathtt{sv} \bmod 3 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4, 5\}$.

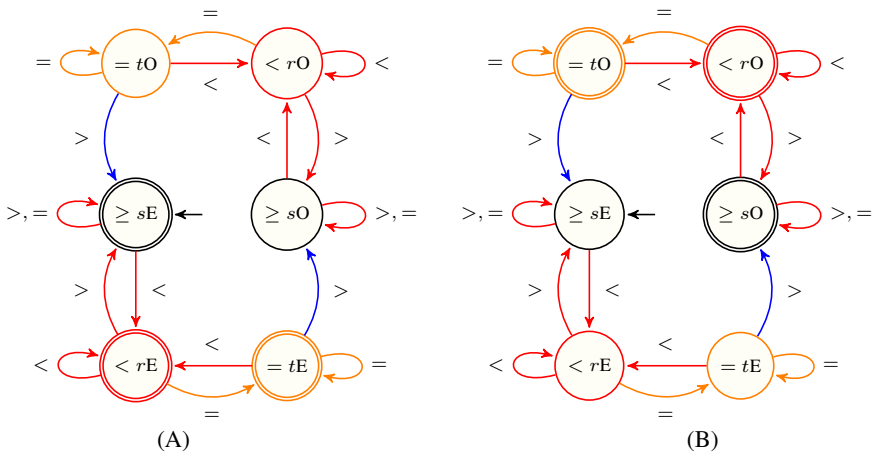


Figure 4.1046: Automata without registers for the (A) NB_PROPER_PLATEAU_IS_EVEN and the (B) NB_PROPER_PLATEAU_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the PROPER_PLATEAU pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the PROPER_PLATEAU pattern.

AGGREGATOR FEATURE PATTERN

NB_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

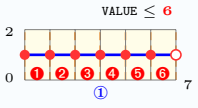
Constraint NB_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv} - 1$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 1)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.

Example $(7, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6\rangle)$

Figure 4.1047 provides an example where the NB_STEADY $(7, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

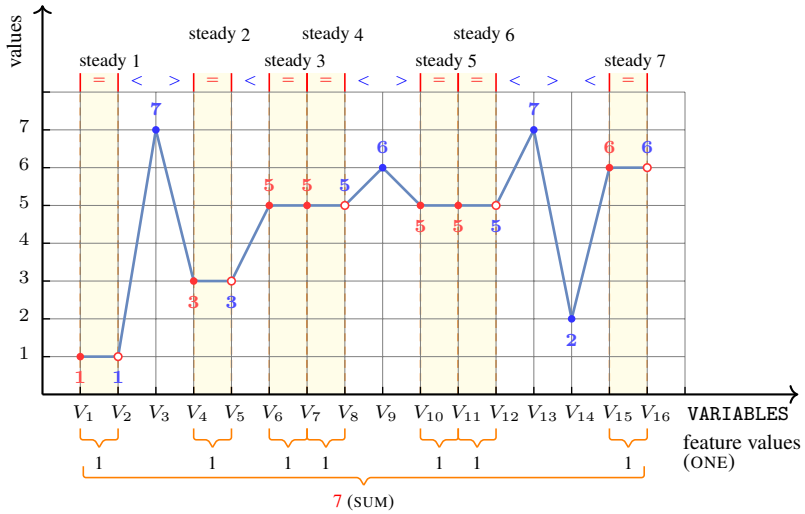


Figure 4.1047: Illustrating the NB_STEADY constraint of the **Example** slot

Automaton

Figures 4.1048 and 4.1049 respectively depict the automaton associated with the constraint NB_STEADY and its simplified form.

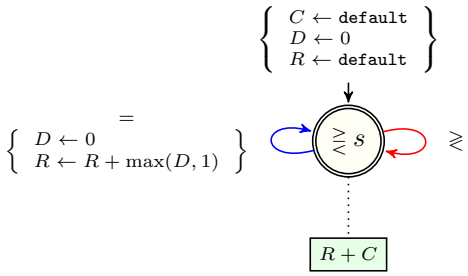


Figure 4.1048: Automaton for the NB_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is 0

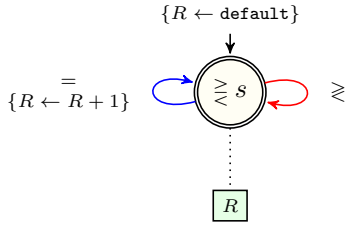


Figure 4.1049: Simplified automaton for the NB_STEADY constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 1 \geq 0$ are linear invariants.

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.251: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the NB_STEADY constraint defined as the composition of the STEADY pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.252: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the NB_STEADY constraint defined as the composition of the STEADY pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

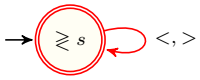


Figure 4.1050: Automaton without registers for the NB_STEADY_EQ_0 constraint; it describes all sequences containing no occurrence of the STEADY pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the STEADY pattern by removing the register R and the **found** transition, i.e. the transition from state s to state s that increments R.

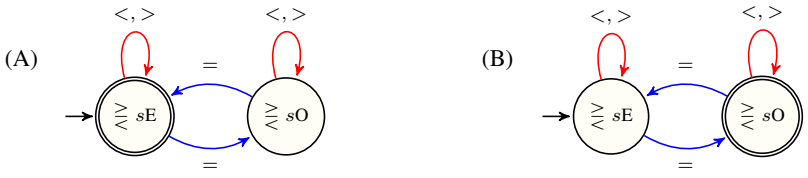
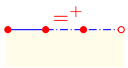


Figure 4.1051: Automata without registers for the (A) NB_STEADY_IS_EVEN and the (B) NB_STEADY_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the STEADY pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the STEADY pattern.

AGGREGATOR FEATURE PATTERN

NB_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint NB_STEADY_SEQUENCE(VALUE, VARIABLES)

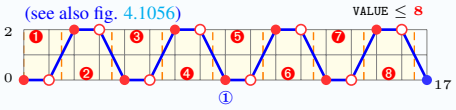
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq 1$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq 0$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \leq 1$
$\texttt{rv} \geq 2 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)

(see also fig. 4.1056)



Purpose VALUE is the number of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.

Example $(5, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.1052 provides an example where the NB_STEADY_SEQUENCE $(5, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

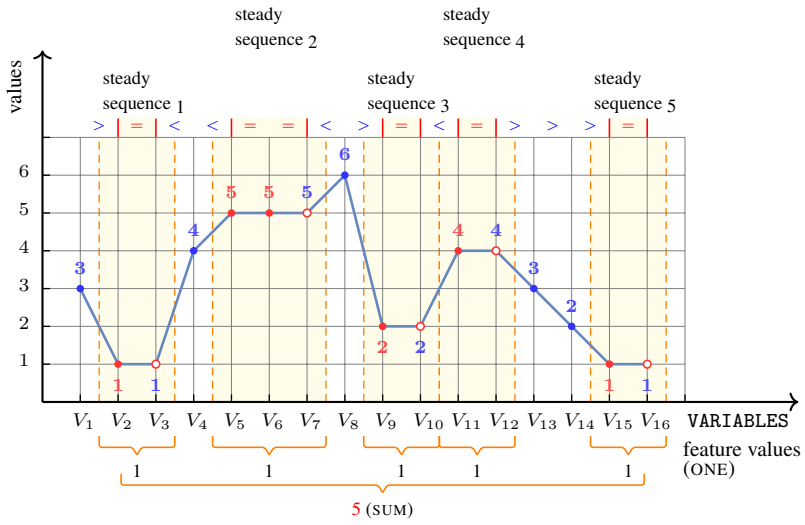


Figure 4.1052: Illustrating the NB_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1053 and 4.1054 respectively depict the automaton associated with the constraint NB_STEADY_SEQUENCE and its simplified form.

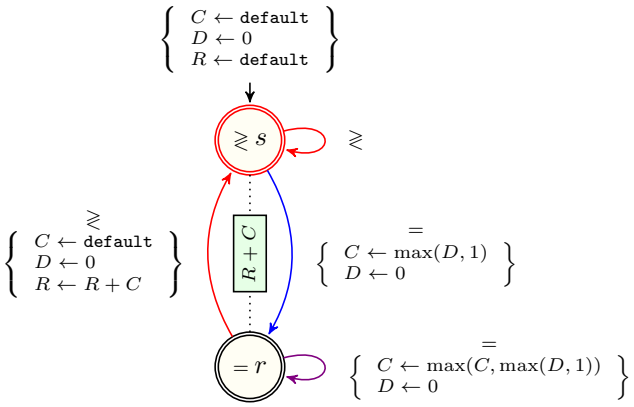


Figure 4.1053: Automaton for the NB_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

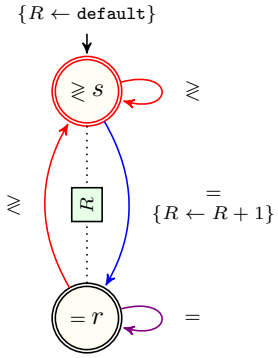


Figure 4.1054: Simplified automaton for the NB_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 M

Table 4.253: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the NB_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	-1 M

Table 4.254: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the NB_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

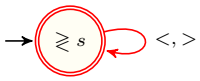


Figure 4.1055: Automaton without registers for the NB_STEADY_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the STEADY_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the STEADY_SEQUENCE pattern by removing the register R, the **found** transition from state s to r that increments R, and the state r that becomes unreachable after removing transition $s \rightarrow r$.

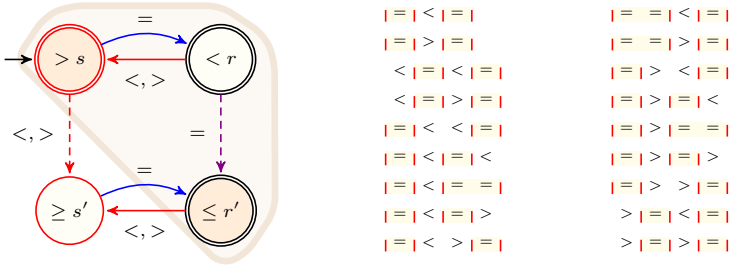


Figure 4.1056: (**left**) Automaton without registers for the NB_STEADY_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the STEADY_SEQUENCE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the STEADY_SEQUENCE pattern, dashed transitions to slack, and accepting states have a light-brown background; state r is accepting when $\mathtt{sv} \bmod 2 = 0$, while states s and r' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4\}$.

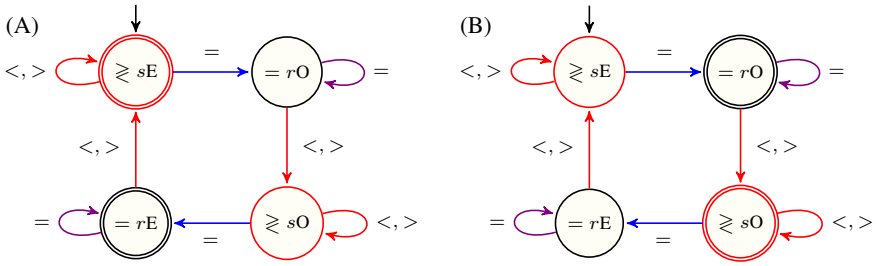


Figure 4.1057: Automata without registers for the (A) NB_STEADY_SEQUENCE_IS_EVEN and the (B) NB_STEADY_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the STEADY_SEQUENCE pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the STEADY_SEQUENCE pattern.

AGGREGATOR FEATURE PATTERN

NB_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

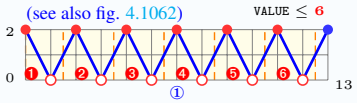
Constraint NB_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.

Example $(3, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1058 provides an example where the NB_STRICTLY_DECREASING_SEQUENCE $(3, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

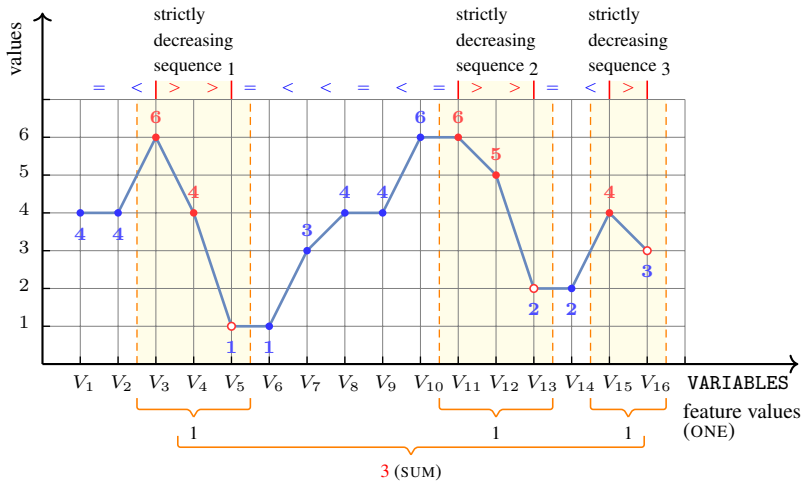


Figure 4.1058: Illustrating the NB_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1059 and 4.1060 respectively depict the automaton associated with the constraint NB_STRICTLY_DECREASING_SEQUENCE and its simplified form.

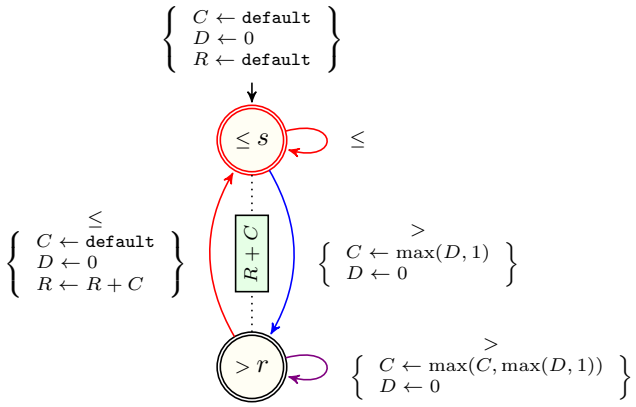


Figure 4.1059: Automaton for the NB_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 M

Table 4.255: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the NB_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

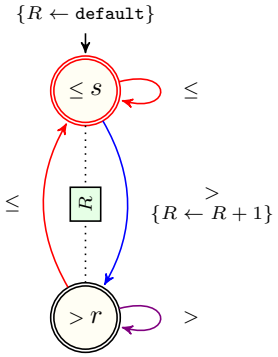


Figure 4.1060: Simplified automaton for the NB_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	-1 M

Table 4.256: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the NB_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

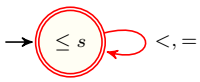


Figure 4.1061: Automaton without registers for the NB_STRICTLY_DECREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the STRICTLY_DECREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern by removing the register R, the **found** transition from state s to r that increments R, and the state r that becomes unreachable after removing transition $s \rightarrow r$.

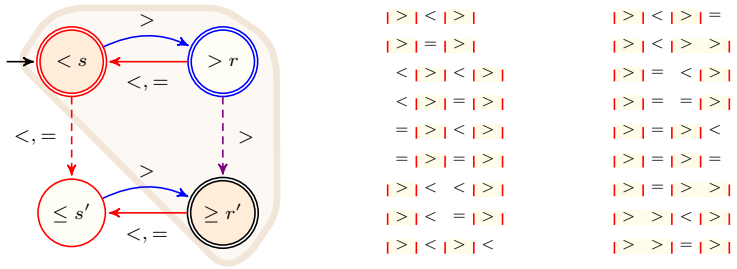


Figure 4.1062: (**left**) Automaton without registers for the NB_STRICTLY_DECREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the STRICTLY_DECREASING_SEQUENCE pattern, dashed transitions to slack, and accepting states have a light-brown background; state r is accepting when $\mathtt{sv} \bmod 2 = 0$, while states s and r' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4\}$.

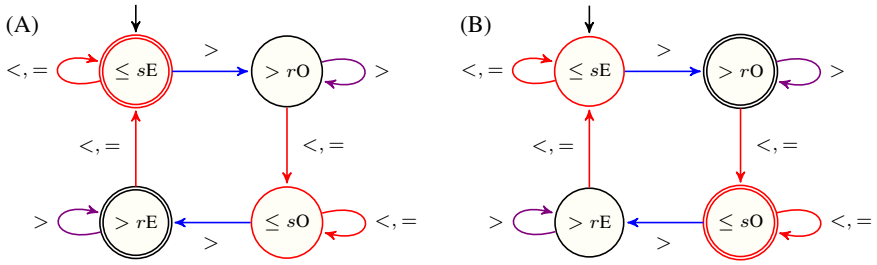


Figure 4.1063: Automata without registers for the (A) NB_STRICTLY_DECREASING_IS_EVEN and the (B) NB_STRICTLY_DECREASING_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the STRICTLY_DECREASING pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the STRICTLY_DECREASING pattern.

AGGREGATOR FEATURE PATTERN

NB_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

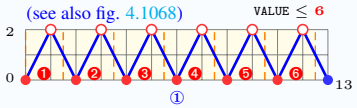
Constraint NB_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.

Example $(3, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1064 provides an example where the NB_STRICTLY_INCREASING_SEQUENCE $(3, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

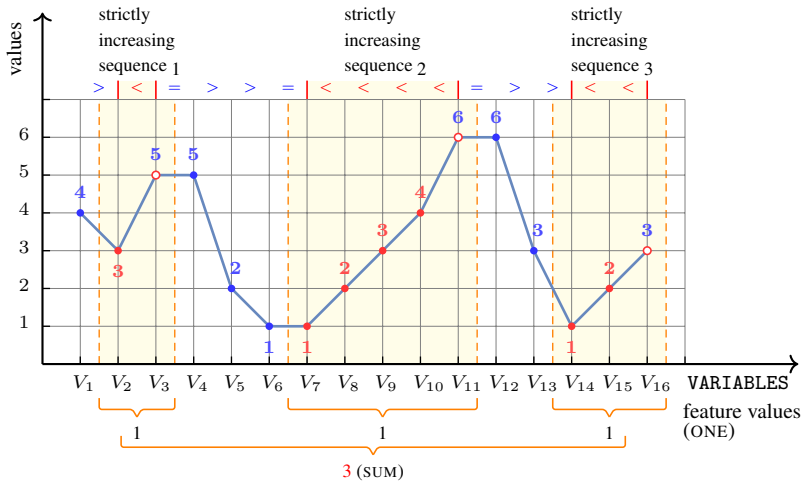


Figure 4.1064: Illustrating the NB_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1065 and 4.1066 respectively depict the automaton associated with the constraint NB_STRICTLY_INCREASING_SEQUENCE and its simplified form.

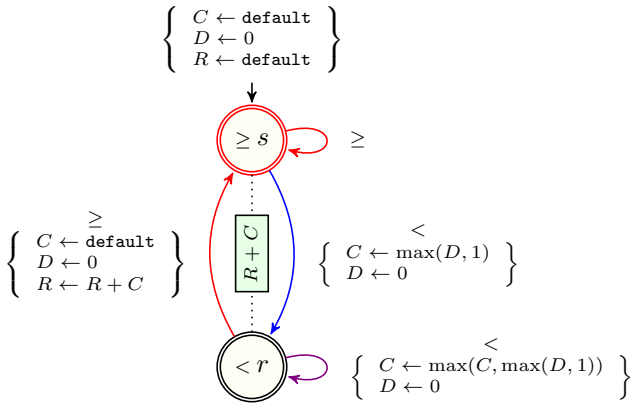


Figure 4.1065: Automaton for the NB_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 M

Table 4.257: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the NB_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

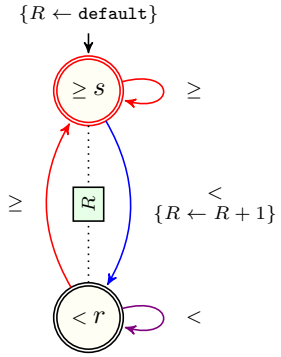


Figure 4.1066: Simplified automaton for the NB_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	-1 M

Table 4.258: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the NB_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

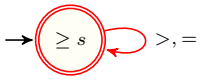


Figure 4.1067: Automaton without registers for the NB_STRICTLY_INCREASING_SEQUENCE_EQ_0 constraint; it describes all sequences containing no occurrence of the STRICTLY_INCREASING_SEQUENCE pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern by removing the register R, the **found** transition from state s to r that increments R, and the state r that becomes unreachable after removing transition $s \rightarrow r$.

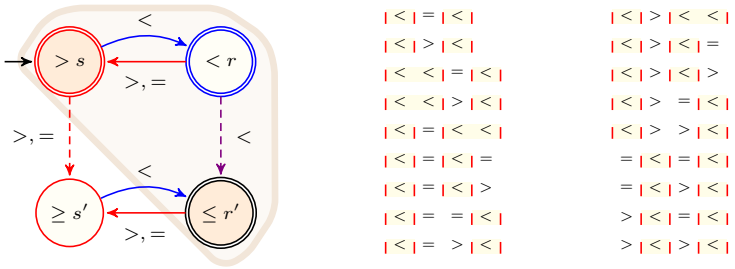


Figure 4.1068: (**left**) Automaton without registers for the NB_STRICTLY_INCREASING_SEQUENCE_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the STRICTLY_INCREASING_SEQUENCE pattern, dashed transitions to slack, and accepting states have a light-brown background; state r is accepting when $\mathtt{sv} \bmod 2 = 0$, while states s and r' are accepting when $\mathtt{sv} \bmod 2 = 1$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4\}$.

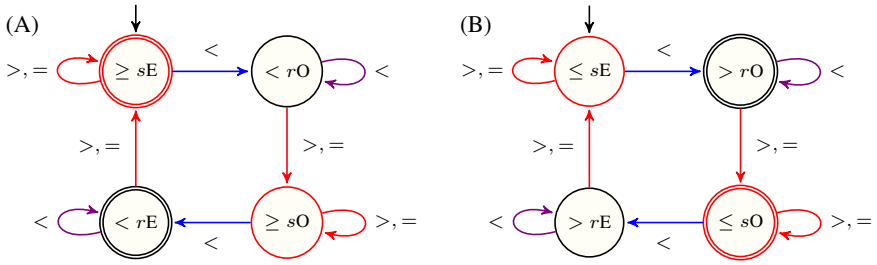
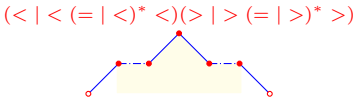


Figure 4.1069: Automata without registers for the (A) NB_STRICTLY_INCREASING_IS_EVEN and the (B) NB_STRICTLY_INCREASING_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the STRICTLY_INCREASING pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the STRICTLY_INCREASING pattern.

AGGREGATOR FEATURE PATTERN

NB_SUMMIT

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the SUMMIT pattern.

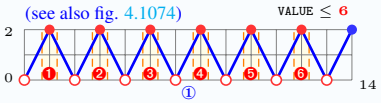
Constraint NB_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.

Example $(3, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1\rangle)$

Figure 4.1070 provides an example where the NB_SUMMIT $(3, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

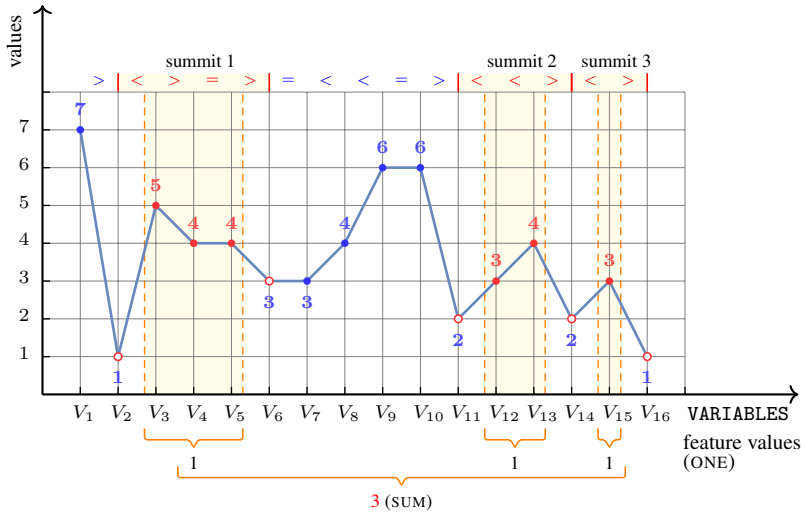


Figure 4.1070: Illustrating the NB_SUMMIT constraint of the **Example** slot

Automaton Figures 4.1071 and 4.1072 respectively depict the automaton associated with the constraint NB_SUMMIT and its simplified form.

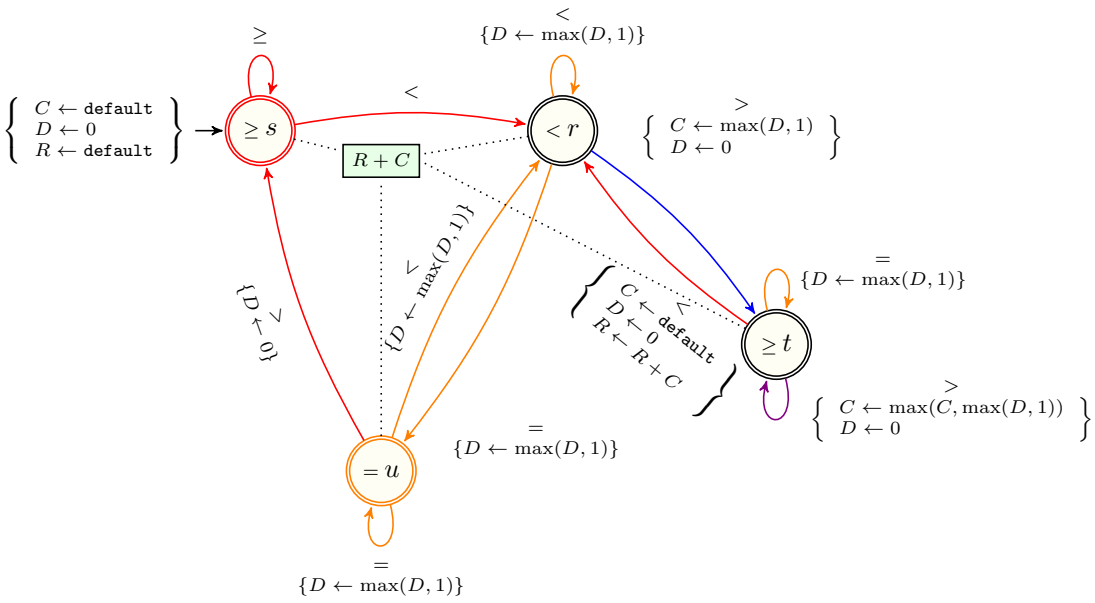


Figure 4.1071: Automaton for the NB_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

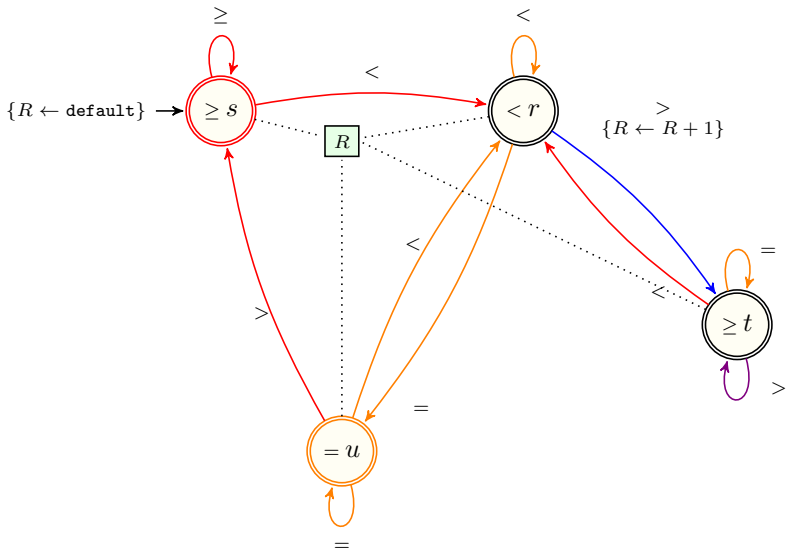


Figure 4.1072: Simplified automaton for the NB_SUMMIT constraint obtained by applying decoration Table 3.39 to the seed transducer of the SUMMIT pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	1 L	$\overrightarrow{C} + \overleftarrow{C}$	1 L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	1 R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.259: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the NB_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	1 C	0 R	0
t	0	0 L	0	0 L
u	0	0	0 R	0

Table 4.260: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the NB_SUMMIT constraint defined as the composition of the SUMMIT pattern , the feature ONE , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

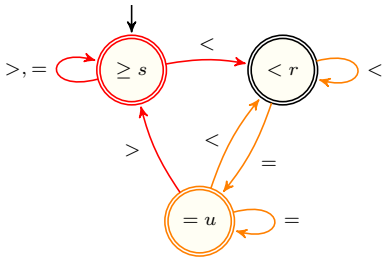


Figure 4.1073: Automaton without registers for the NB_SUMMIT_EQ_0 constraint; it describes all sequences containing no occurrence of the SUMMIT pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the SUMMIT pattern by removing the register R, the **found** transition from state r to t that increments R, and the state t that becomes unreachable after removing transition $r \to t$.

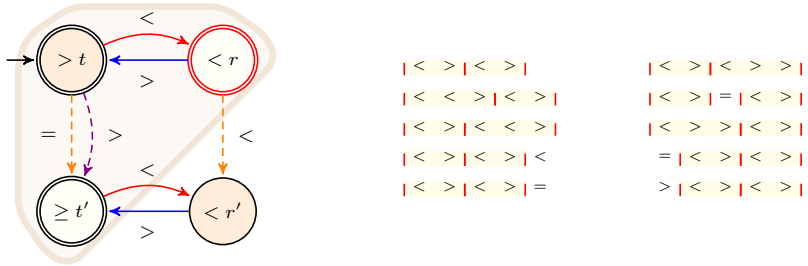


Figure 4.1074: (**left**) Automaton without registers for the NB_SUMMIT_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the SUMMIT pattern on a sequence of sv variables, i.e. $\lfloor \frac{\mathtt{sv}-1}{2} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the SUMMIT pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

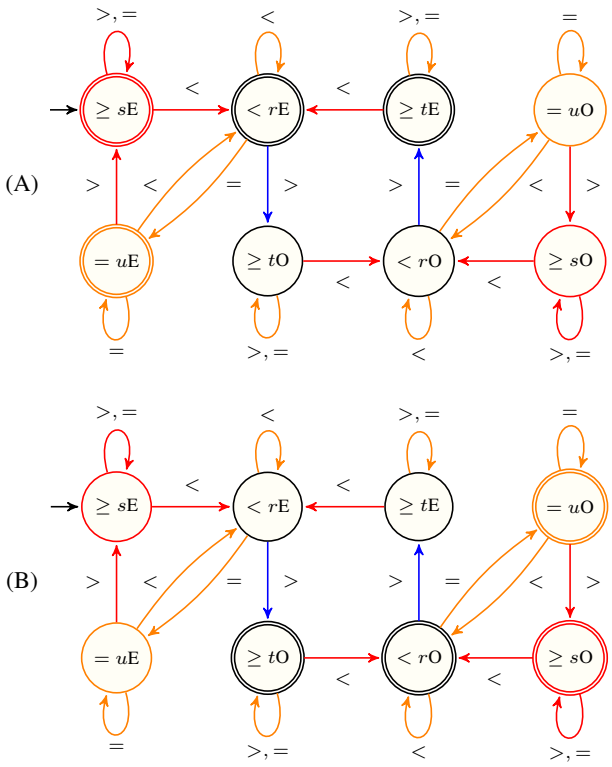
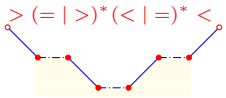


Figure 4.1075: Automata without registers for the (A) NB_SUMMIT_IS_EVEN and the (B) NB_SUMMIT_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the SUMMIT pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the SUMMIT pattern.

AGGREGATOR FEATURE PATTERN

NB_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

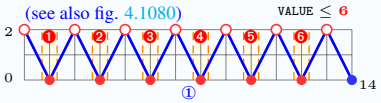
Constraint NB_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose VALUE is the number of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.

Example $(3, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7\rangle)$

Figure 4.1076 provides an example where the NB_VALLEY $(3, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

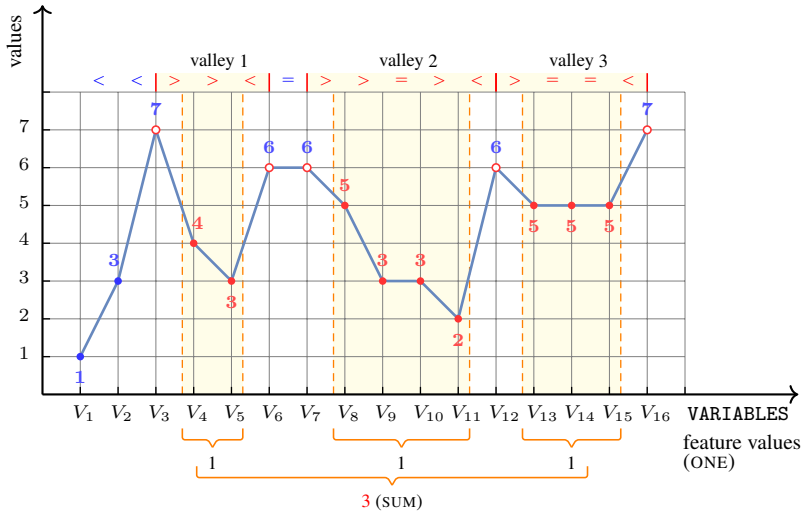


Figure 4.1076: Illustrating the NB_VALLEY constraint of the **Example** slot

Automaton

Figures 4.1077 and 4.1078 respectively depict the automaton associated with the constraint NB_VALLEY and its simplified form.

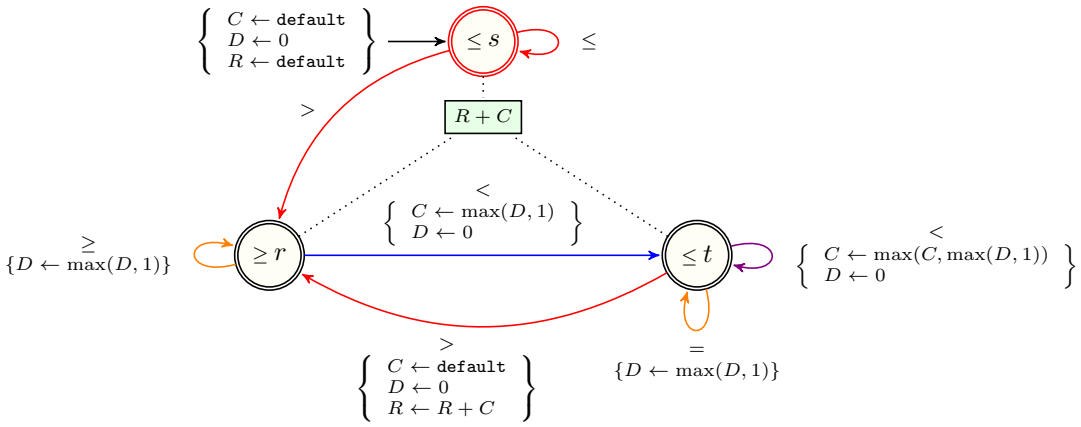


Figure 4.1077: Automaton for the NB_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is 0

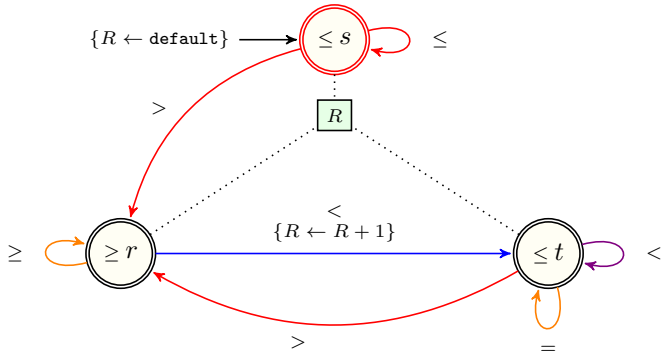


Figure 4.1078: Simplified automaton for the NB_VALLEY constraint obtained by applying decoration Table 3.39 to the seed transducer of the VALLEY pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-2} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	1 C	1 R
t	$\overrightarrow{C} + \overleftarrow{C}$	1 L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.261: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the NB_VALLEY constraint defined as the composition of the VALLEY pattern , the feature ONE , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	1 C	0 R
t	0	0 L	0

Table 4.262: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the NB_VALLEY constraint defined as the composition of the VALLEY pattern , the feature ONE , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

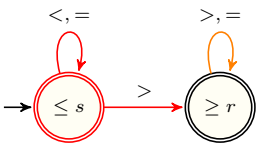


Figure 4.1079: Automaton without registers for the NB_VALLEY_EQ_0 constraint; it describes all sequences containing no occurrence of the VALLEY pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the VALLEY pattern by removing the register R, the **found** transition from state r to t that increments R, and the state t that becomes unreachable after removing transition $r \to t$.

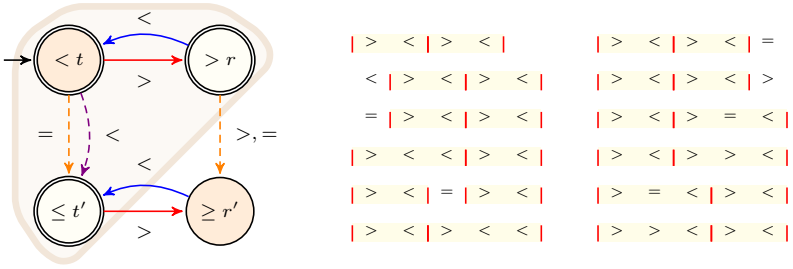


Figure 4.1080: (**left**) Automaton without registers for the NB_VALLEY_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the VALLEY pattern on a sequence of sv variables, i.e. $\max(0, \lfloor \frac{\mathtt{sv}-1}{2} \rfloor)$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the VALLEY pattern, dashed transitions to slack, and accepting states have a light-brown background; state t is accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

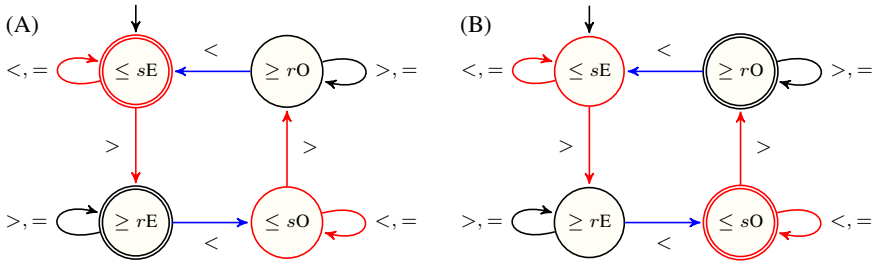
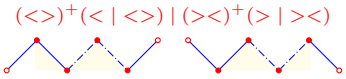


Figure 4.1081: Automata without registers for the (A) NB_VALLEY_IS_EVEN and the (B) NB_VALLEY_IS_ODD constraints; they respectively achieve an even/odd number of occurrences of the VALLEY pattern on a sequence of n variables; transitions in blue correspond to a new occurrence of the VALLEY pattern.

AGGREGATOR FEATURE PATTERN

NB_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint NB_ZIGZAG(VALUE, VARIABLES)

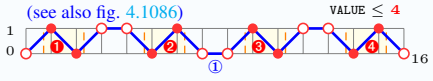
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

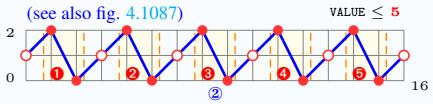
Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \lfloor \texttt{sv}/4 \rfloor$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(0, \lfloor (\texttt{sv} - 1)/3 \rfloor)$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)

(see also fig. 4.1086)



(see also fig. 4.1087)



Purpose VALUE is the number of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.

Example $(3, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.1082 provides an example where the NB_ZIGZAG $(3, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

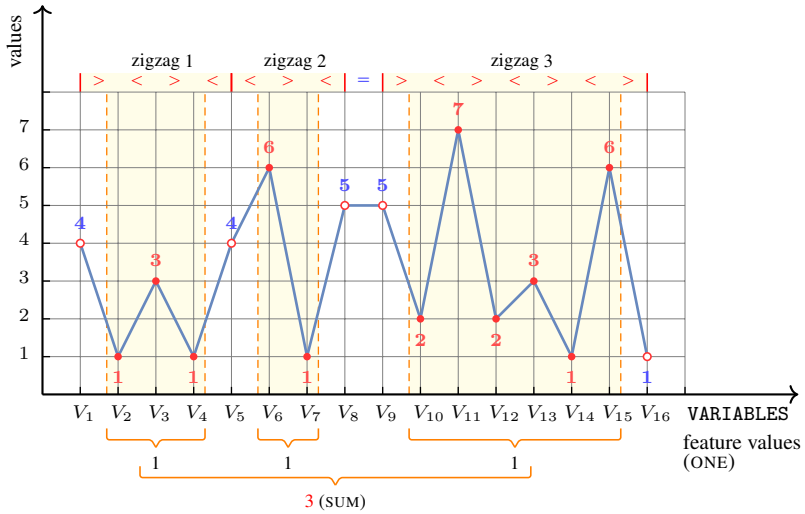


Figure 4.1082: Illustrating the NB_ZIGZAG constraint of the **Example** slot

Automaton

Figures 4.1083 and 4.1084 respectively depict the automaton associated with the constraint NB_ZIGZAG and its simplified form.

	s	a	b	c	d	e	f
s	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$
a	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	1 R	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$
b	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$	1 R
c	$\overrightarrow{C}+\overleftarrow{C}$	1 L	$\overrightarrow{C}+\overleftarrow{C}$	1 M	$\overrightarrow{C}+\overleftarrow{C}$	1 L	$\overrightarrow{C}+\overleftarrow{C}$
d	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	1 R
e	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$	1 R	$\overrightarrow{C}+\overleftarrow{C}$	1 C	$\overrightarrow{C}+\overleftarrow{C}$
f	$\overrightarrow{C}+\overleftarrow{C}$	$\overrightarrow{C}+\overleftarrow{C}$	1 L	$\overrightarrow{C}+\overleftarrow{C}$	1 L	$\overrightarrow{C}+\overleftarrow{C}$	1 M

Table 4.263: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the NB_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	a	b	c	d	e	f
s	0	0	0	0	0	0	0
a	0	0	0	0 R	0	1 C	0
b	0	0	1 C	0	1 C	0	0 R
c	0	0 L	0	−1 M	0	0 L	0
d	0	0	1 C	0	0	0	0 R
e	0	1 C	0	0 R	0	1 C	0
f	0	0	0 L	0	0 L	0	−1 M

Table 4.264: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the NB_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature ONE, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

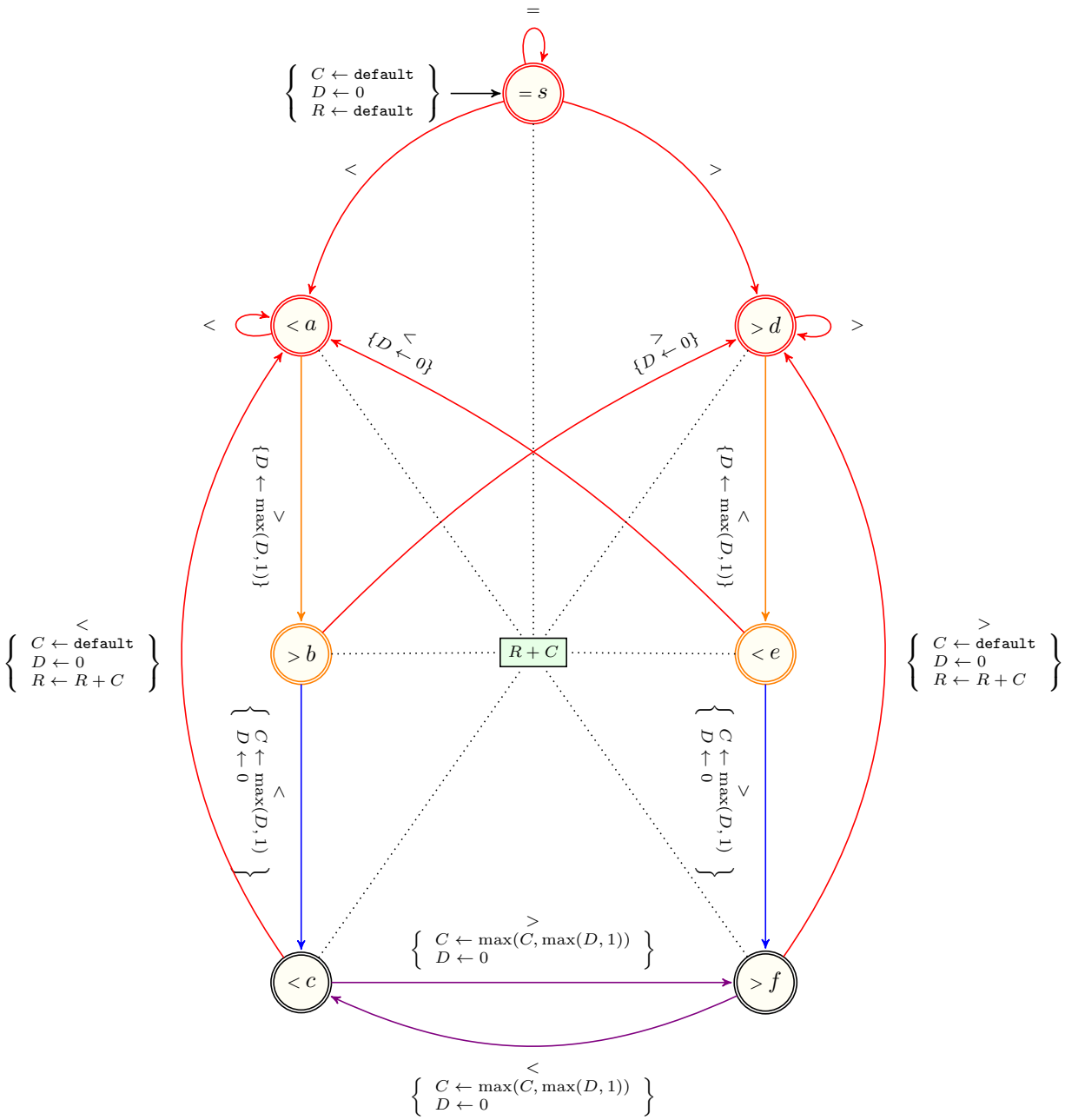


Figure 4.1083: Automaton for the NB_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

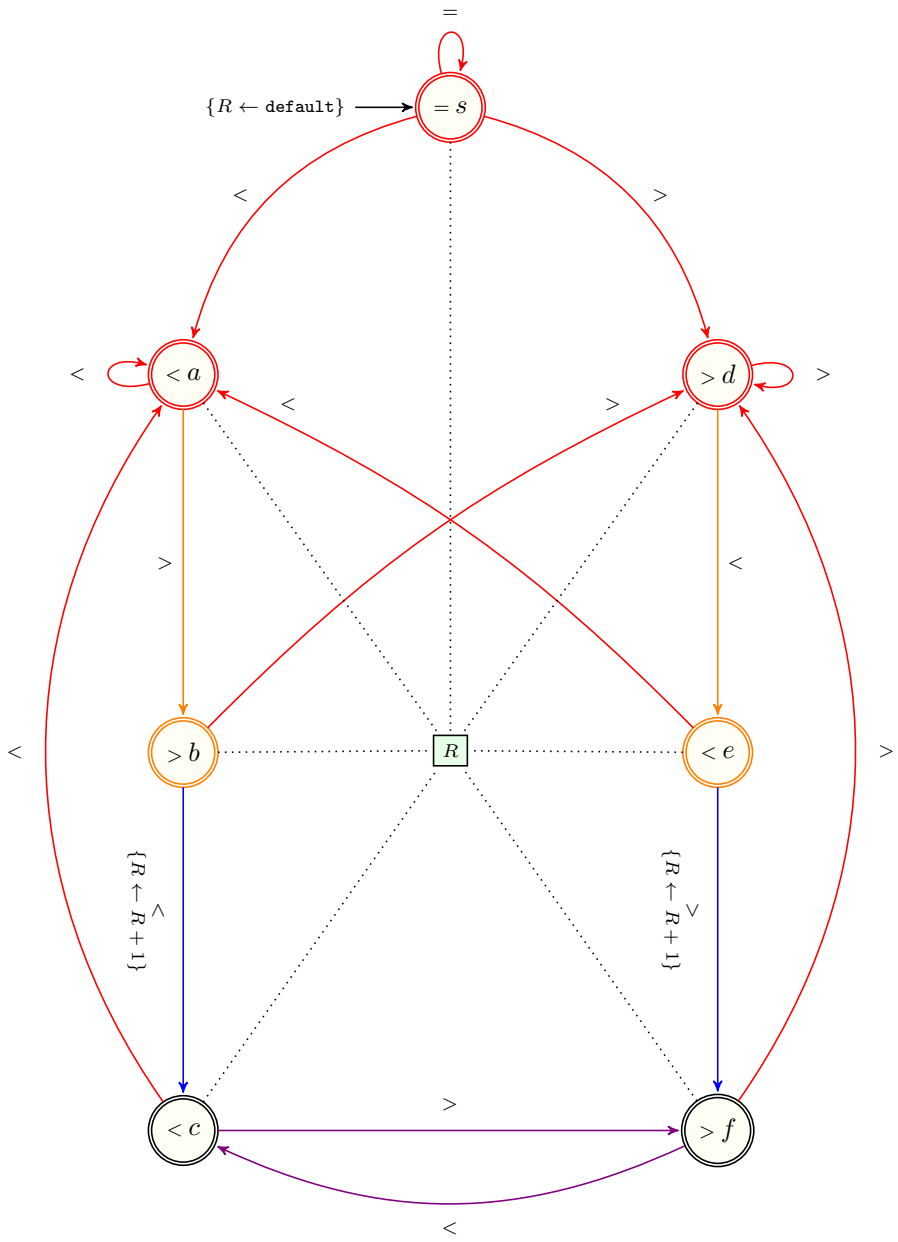


Figure 4.1084: Simplified automaton for the NB_ZIGZAG constraint obtained by applying decoration Table 3.39 to the seed transducer of the ZIGZAG pattern where `default` is 0; missing transitions from a, b, c, d, e, f to s are labelled by $=$; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-3} + 1 \geq 0$ are linear invariants.

Specialisation

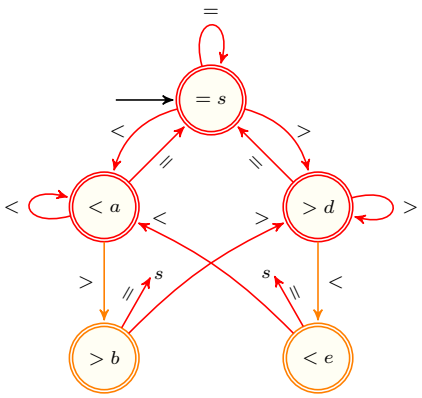


Figure 4.1085: Automaton without registers for the NB_ZIGZAG_EQ_0 constraint; it describes all sequences containing no occurrence of the ZIGZAG pattern on a sequence of variables; it is derived from the automaton that counts the number of occurrences of the ZIGZAG pattern by removing (1) the register R, (2) the transitions labelled by **found**, i.e. transitions $b \rightarrow c$ and $e \rightarrow f$ that increment R, and (3) the states c and f that become unreachable after removing transitions $b \rightarrow c$ and $e \rightarrow f$.

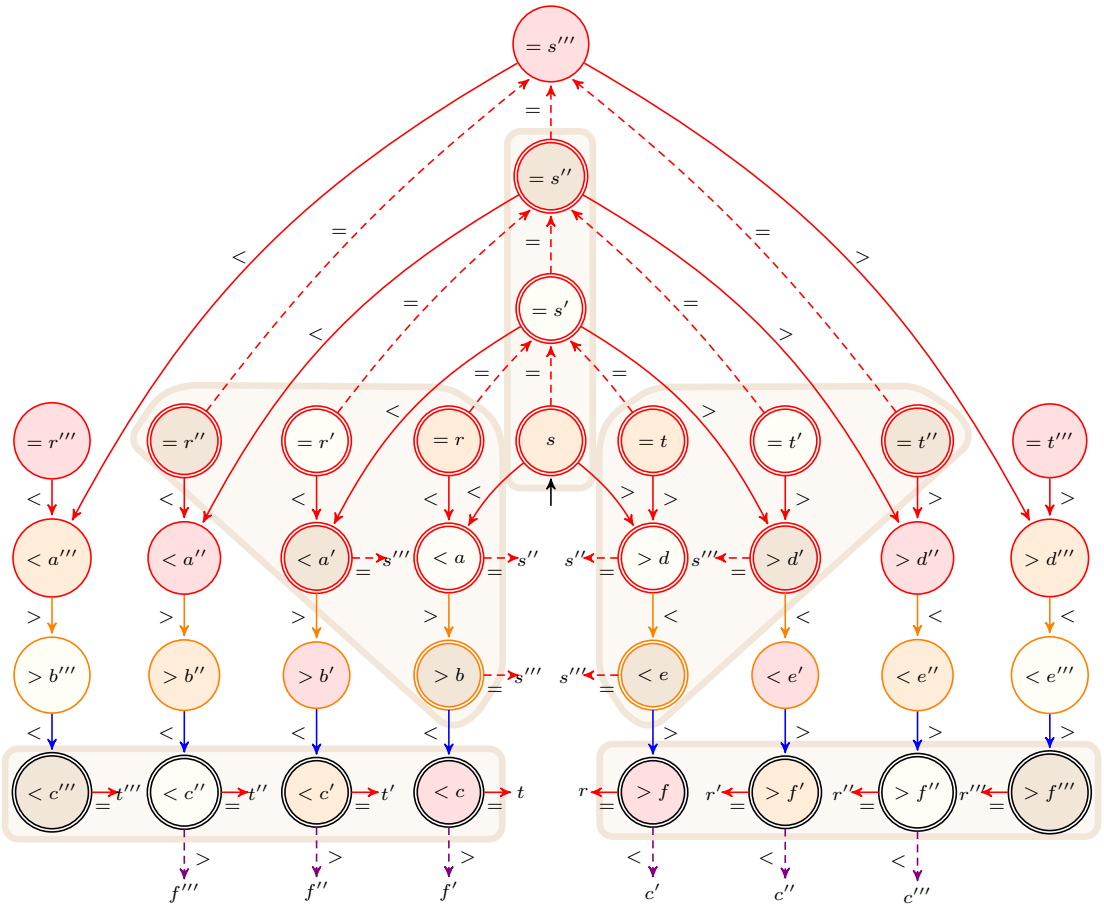
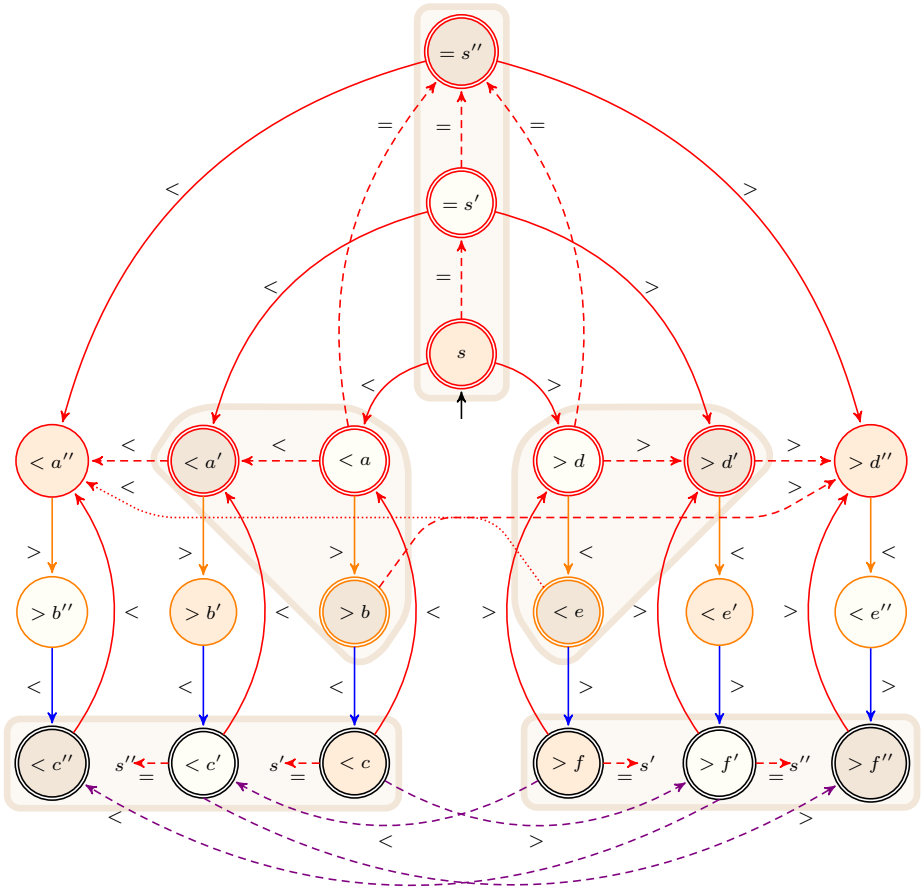


Figure 4.1086: Automaton without registers for the NB_ZIGZAG_EQ_UP_WHEN_RANGE_EQ_2 constraint; it describes all sequences containing the maximum number of occurrences of the ZIGZAG pattern when the difference between the maximum and the minimum of the variables plus one of the sequence of sv variables is equal to 2, i.e. $\lfloor \frac{\mathtt{sv}}{4} \rfloor$ of the **Restrictions** slot (see ①); transitions in blue correspond to a new occurrence of the ZIGZAG pattern, dashed transitions to slack, and accepting states have a light-brown background; states c and f are accepting when $\mathtt{sv} \bmod 4 = 0$, states s, r, t, c' and f' are accepting when $\mathtt{sv} \bmod 4 = 1$, states a, d, s', r', t', c'' and f'' are accepting when $\mathtt{sv} \bmod 4 = 2$, states b, e, a', d', s'', r'', t'', c''' and f''' are accepting when $\mathtt{sv} \bmod 4 = 3$.



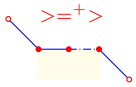
|< > <|	>|> < >|	|< > <|= =	= =|> < >|	|> < >|= >
|> < >|	< <|< > <|	|< > <|= >	=|> < > <|	|> < >|> <
<|< > <|	<|< > <|<	|< > < > <|	=|> < >|=	|> < >|> =
|< > <|<	<|< > <|=	|< > < >|=	=|> < >|>	|> < >|> >
|< > <|=	<|< > < >|	|< > < >|>	= >|> < >|	> =|< > <|
|< > < >|	< =|< > <|	< >|> < >|	> <|< > <|	> =|> < >|
=|< > <|	< =|> < >|	= <|< > <|	|> < > <|<	>|> < > <|
=|> < >|	|< > <|< <	=|< > <|<	|> < > <|=	>|> < >|=
|> < > <|	|< > <|< =	=|< > <|=	|> < > < >|	>|> < >|>
|> < >|=	|< > <|< >	=|< > < >|	|> < >|= <	> >|> < >|
|> < >|>	|< > <|= <	= =|< > <|	|> < >|= =	

Figure 4.1087: (**top**) Automaton without registers for the NB_ZIGZAG_EQ_UP constraint; it describes all sequences containing the maximum number of occurrences of the ZIGZAG pattern when the difference between the maximum and the minimum of the variables plus one of the sequence of sv variables is greater than or equal to 3, i.e. the upper bound $\max(0, \lfloor \frac{\mathtt{sv}-1}{3} \rfloor)$ of the **Restrictions** slot (see ②); transitions in blue correspond to a new occurrence of the ZIGZAG pattern, dashed transitions to slack, and accepting states have a light-brown background; states s, c and f are accepting when $\mathtt{sv} \bmod 3 = 1$, states s', a, d, c' and f' are accepting when $\mathtt{sv} \bmod 3 = 2$, states s'', b, e, a', d', c'' and f'' are accepting when $\mathtt{sv} \bmod 3 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{3, 4, 5\}$.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_HEIGHT_DECREASING_TERRACE.

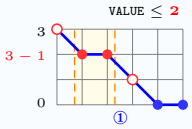
Constraint POS_MAX_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 4, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1088 provides an example where the POS_MAX_HEIGHT_DECREASING_TERRACE $(4, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3], [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0])$ constraint holds.

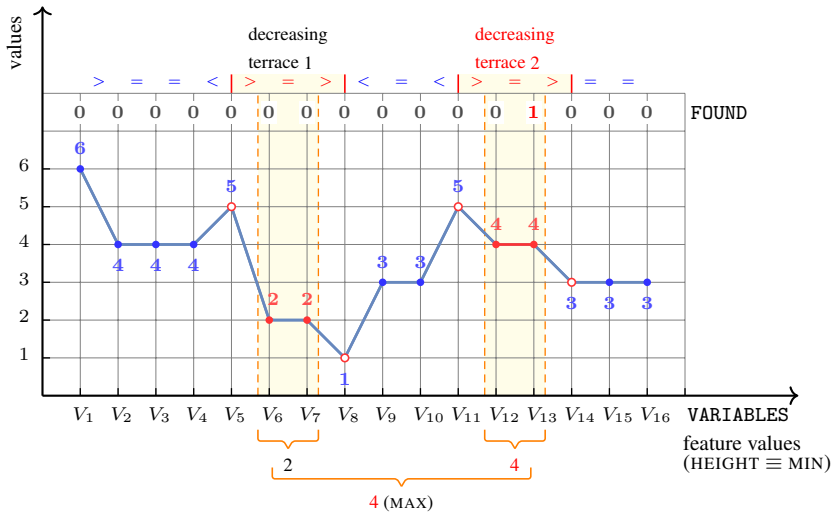


Figure 4.1088: Illustrating the POS_MAX_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

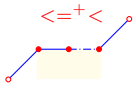
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_HEIGHT_INCREASING_TERRACE.

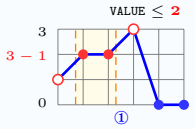
Constraint POS_MAX_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1089 provides an example where the POS_MAX_HEIGHT_INCREASING_TERRACE $(5, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

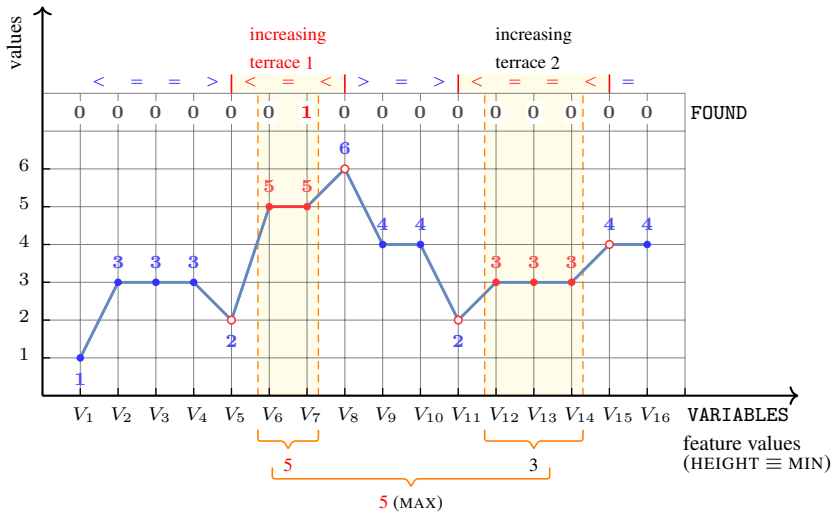


Figure 4.1089: Illustrating the POS_MAX_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_HEIGHT_PLAIN.

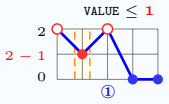
Constraint POS_MAX_HEIGHT_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1090 provides an example where the POS_MAX_HEIGHT_PLAIN $(5, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3], [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

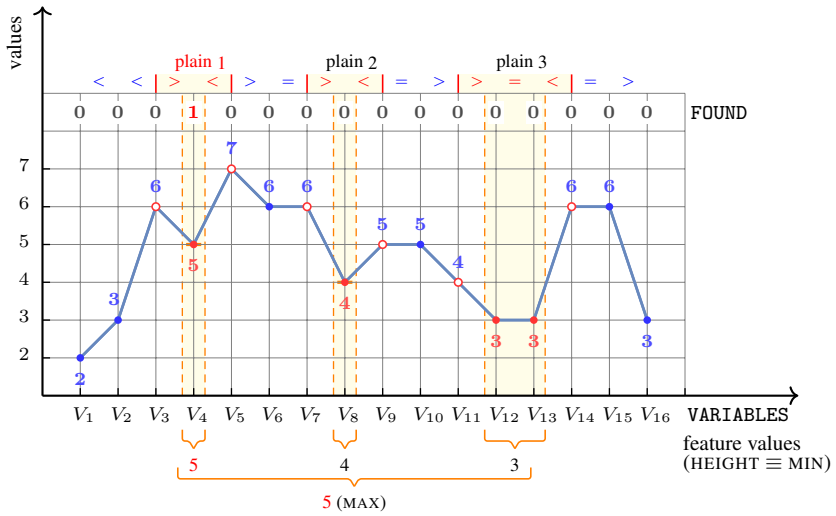


Figure 4.1090: Illustrating the POS_MAX_HEIGHT_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_HEIGHT_PLATEAU.

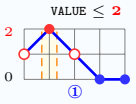
Constraint POS_MAX_HEIGHT_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1091 provides an example where the POS_MAX_HEIGHT_PLATEAU $(5, [7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5], [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0])$ constraint holds.

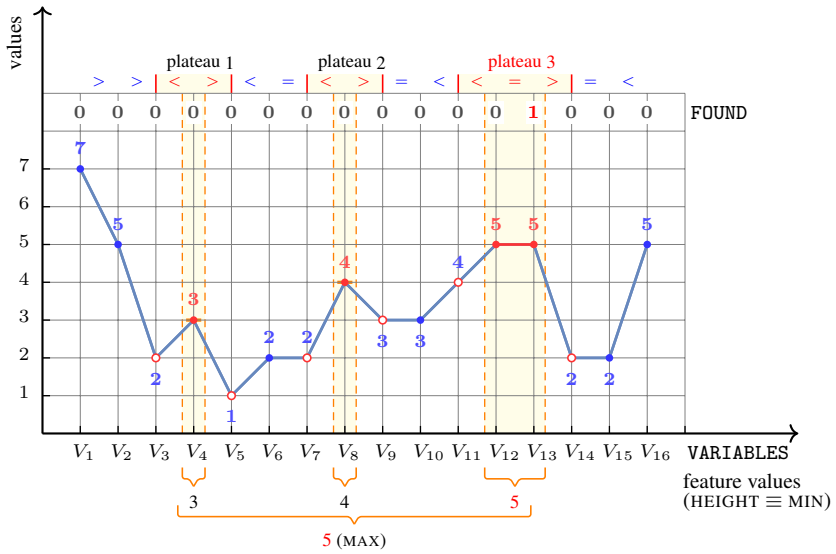


Figure 4.1091: Illustrating the POS_MAX_HEIGHT_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

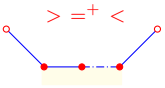
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_PROPER_PLAIN

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_HEIGHT_PROPER_PLAIN.

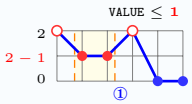
Constraint POS_MAX_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^+<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1092 provides an example where the POS_MAX_HEIGHT_PROPER_PLAIN $(5, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5], [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

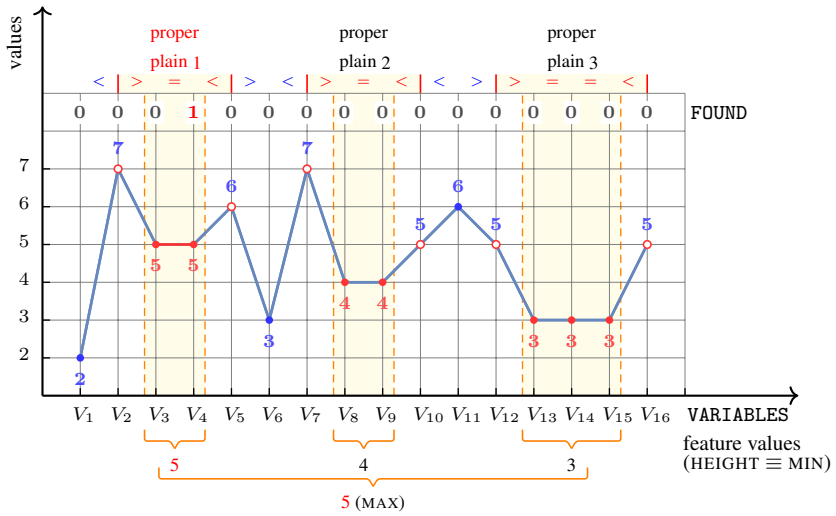


Figure 4.1092: Illustrating the POS_MAX_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

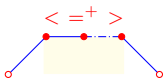
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_PROPER_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_HEIGHT_PROPER_PLATEAU.

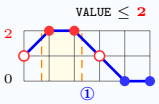
Constraint POS_MAX_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}_{①}$
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MAX_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$$

Figure 4.1093 provides an example where the POS_MAX_HEIGHT_PROPER_PLATEAU $(5, [7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

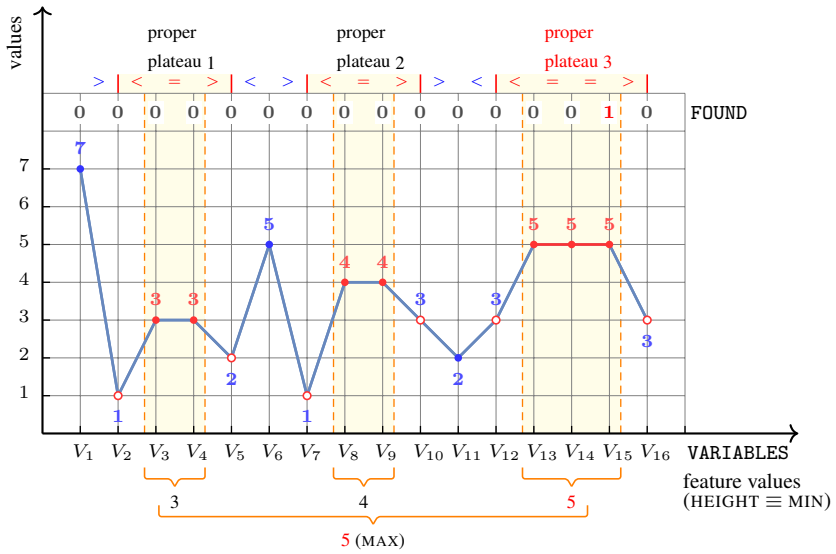


Figure 4.1093: Illustrating the POS_MAX_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_HEIGHT_STEADY.

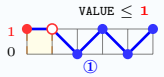
Constraint POS_MAX_HEIGHT_STEADY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose The constraint MAX_HEIGHT_STEADY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $\left(\begin{array}{l} 6, \langle 1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$

Figure 4.1094 provides an example where the POS_MAX_HEIGHT_STEADY $(6, [1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

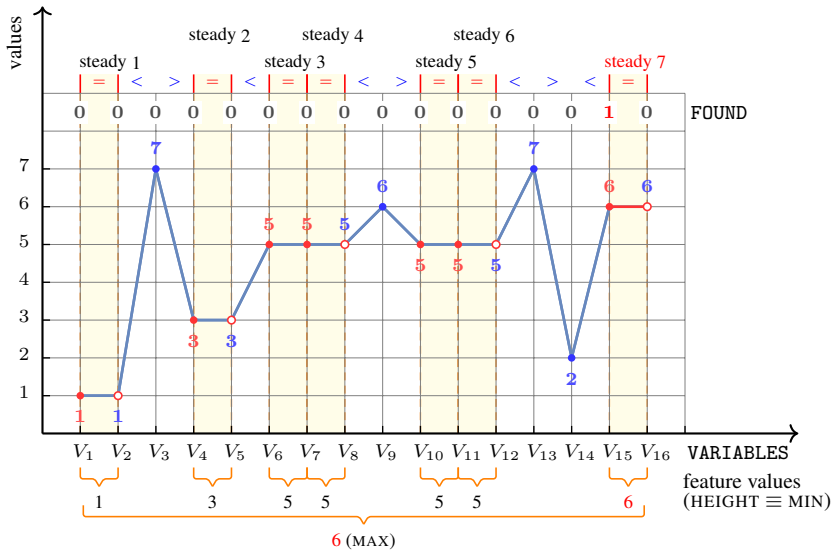


Figure 4.1094: Illustrating the POS_MAX_HEIGHT_STEADY constraint of the **Example** slot

Arg. properties

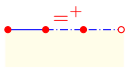
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_STEADY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_HEIGHT_STEADY_SEQUENCE.

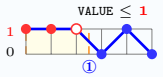
Constraint POS_MAX_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

The constraint MAX_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 5, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1095 provides an example where the POS_MAX_HEIGHT_STEADY_SEQUENCE $(5, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

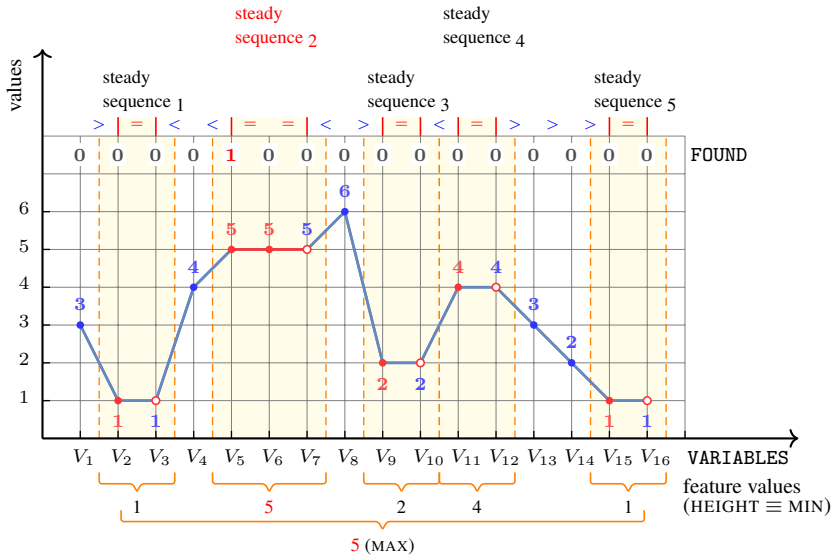


Figure 4.1095: Illustrating the POS_MAX_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

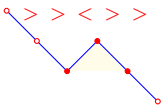
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_HEIGHT_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_BUMP_ON_DECREASING_SEQUENCE.

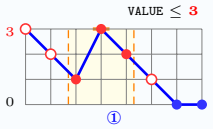
Constraint POS_MAX_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 2$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3 \rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1096 provides an example where the POS_MAX_MAX_BUMP_ON_DECREASING_SEQUENCE $(6, [7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

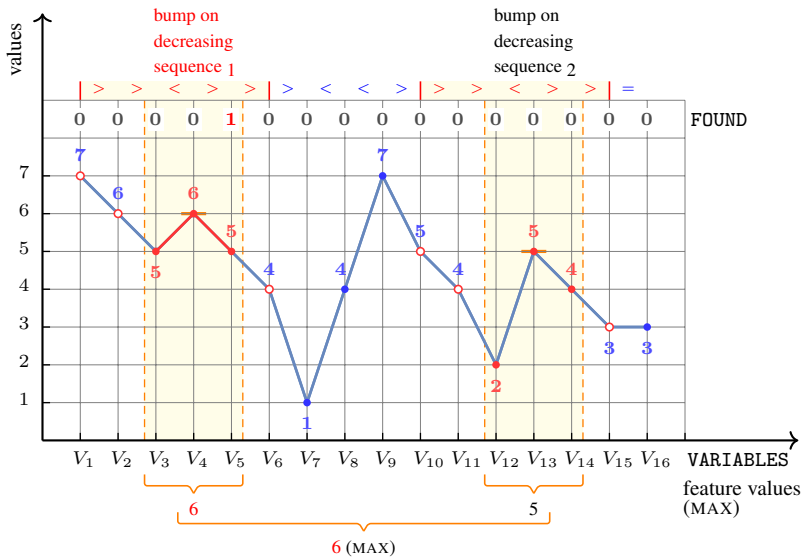


Figure 4.1096: Illustrating the POS_MAX_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_DECREASING.

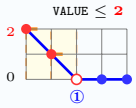
Constraint POS_MAX_MAX_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 6, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1097 provides an example where the POS_MAX_MAX_DECREASING $(6, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

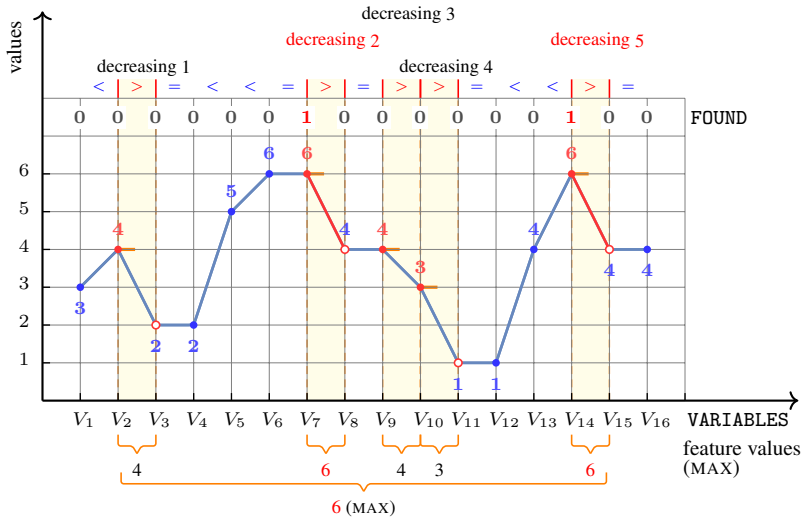


Figure 4.1097: Illustrating the POS_MAX_MAX_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

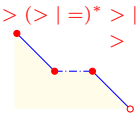
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_DECREASING_SEQUENCE.

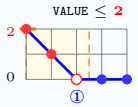
Constraint POS_MAX_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 6, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1098 provides an example where the POS_MAX_MAX_DECREASING_SEQUENCE $(6, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

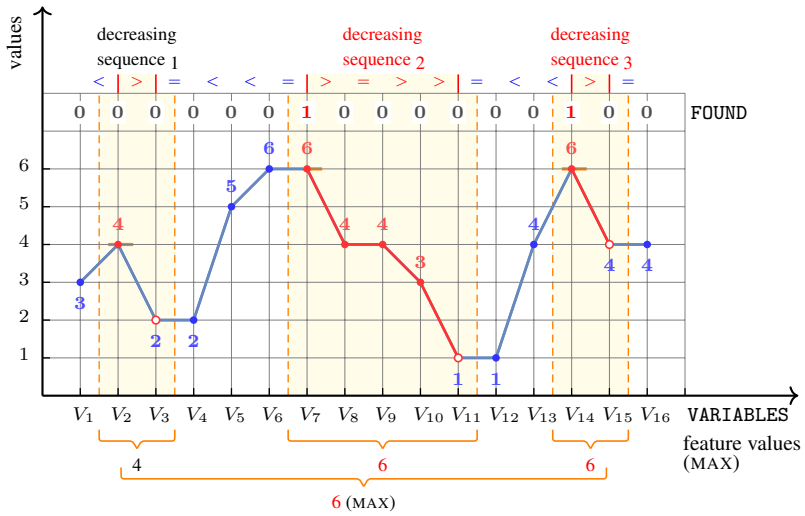


Figure 4.1098: Illustrating the POS_MAX_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

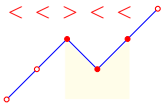
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_DIP_ON_INCREASING_SEQUENCE.

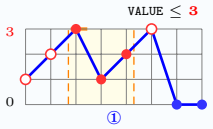
Constraint POS_MAX_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv} + 2$
$\mathtt{VALUE} \leq \mathtt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1099 provides an example where the POS_MAX_MAX_DIP_ON_INCREASING_SEQUENCE $(6, [1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

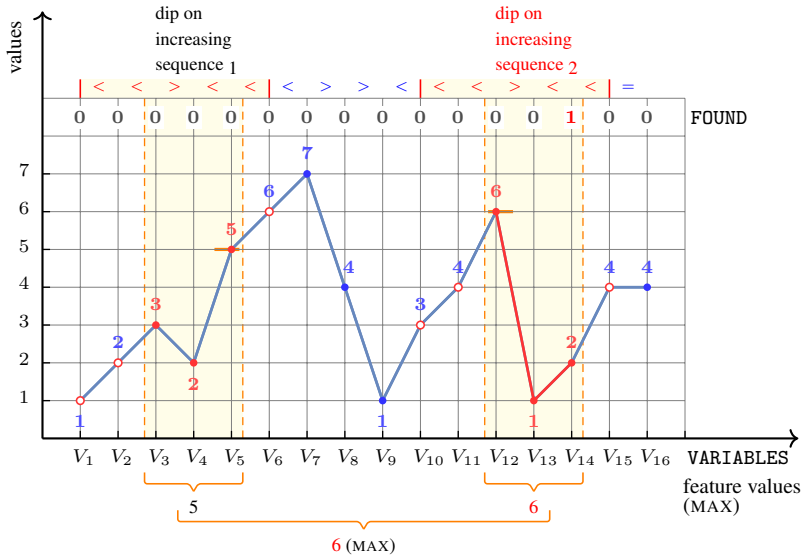


Figure 4.1099: Illustrating the POS_MAX_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_INCREASING.

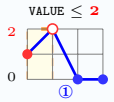
Constraint POS_MAX_MAX_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 6, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1100 provides an example where the POS_MAX_MAX_INCREASING $(6, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0])$ constraint holds.

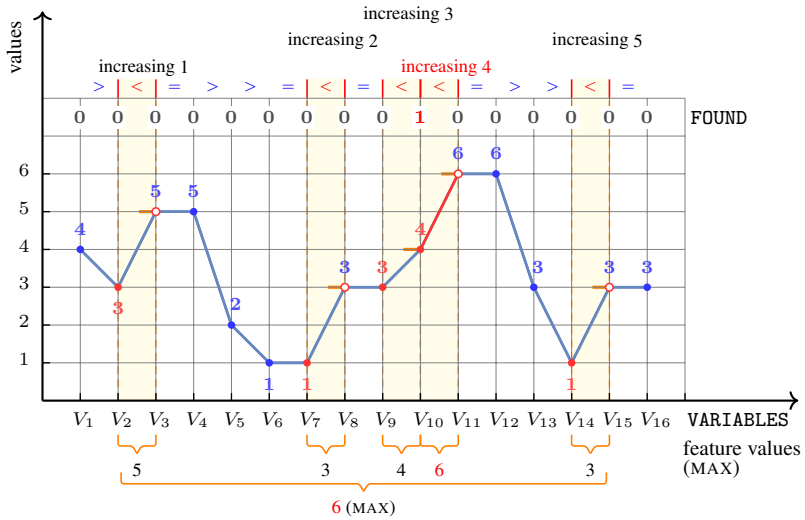


Figure 4.1100: Illustrating the POS_MAX_MAX_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

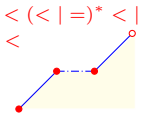
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_INCREASING_SEQUENCE.

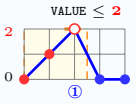
Constraint POS_MAX_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 6, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1101 provides an example where the POS_MAX_MAX_INCREASING_SEQUENCE $(6, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

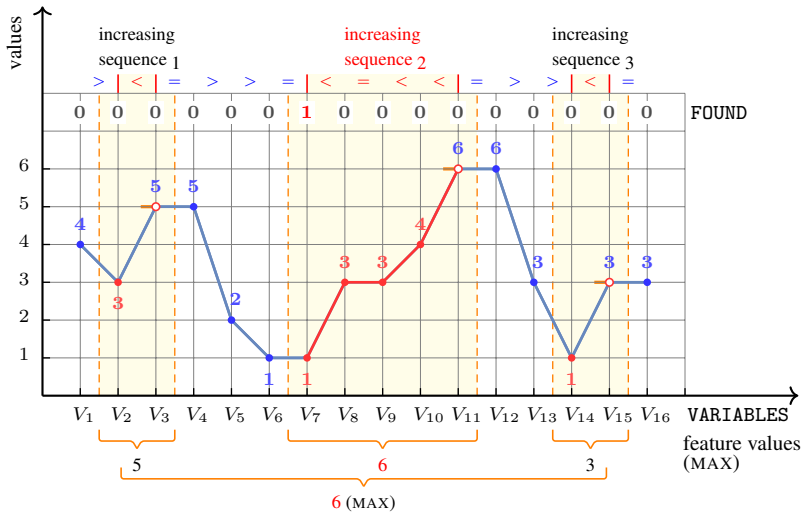


Figure 4.1101: Illustrating the POS_MAX_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

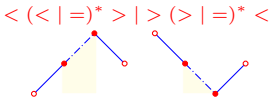
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_MAX_INFLEXION.

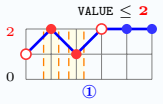
Constraint POS_MAX_MAX_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1102 provides an example where the POS_MAX_MAX_INFLEXION $(6, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

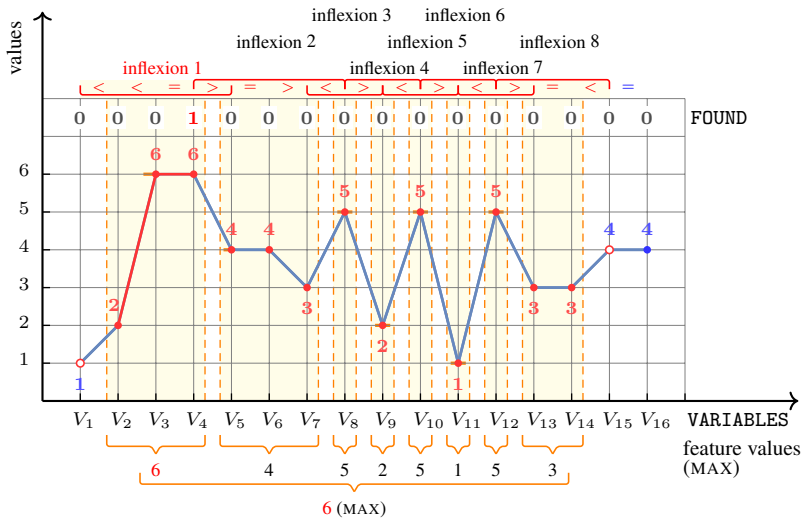


Figure 4.1102: Illustrating the POS_MAX_MAX_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

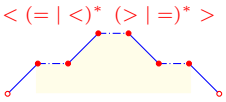
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_PEAK



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_MAX_PEAK.

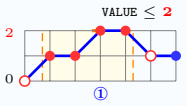
Constraint POS_MAX_MAX_PEAK(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1103 provides an example where the POS_MAX_MAX_PEAK $(6, [7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

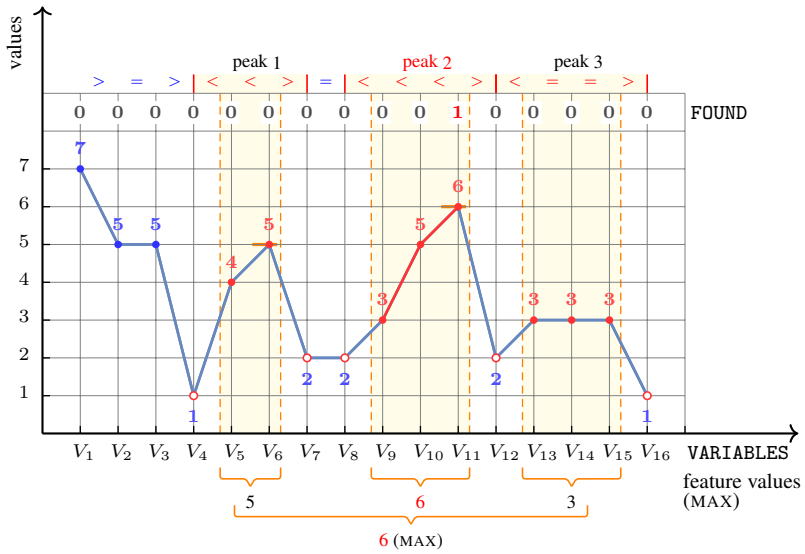


Figure 4.1103: Illustrating the POS_MAX_MAX_PEAK constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_MAX_STRICTLY_DECREASING_SEQUENCE.

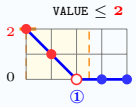
Constraint POS_MAX_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 6, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1104 provides an example where the POS_MAX_MAX_STRICTLY_DECREASING_SEQUENCE $(6, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

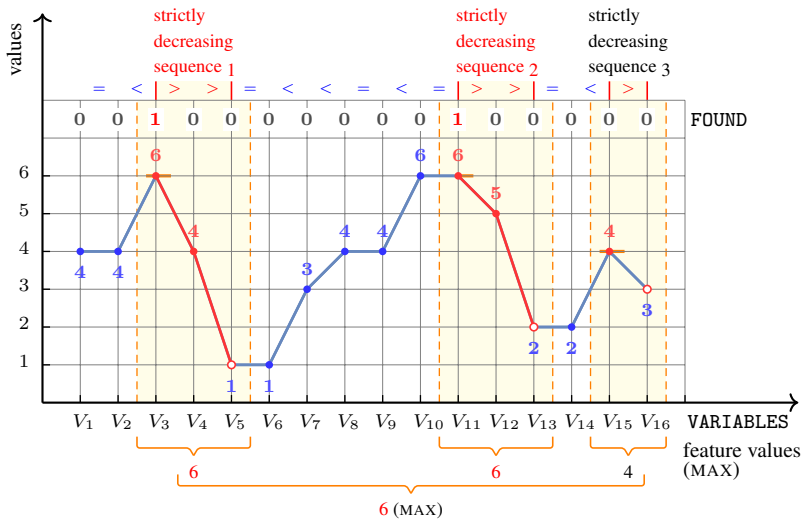


Figure 4.1104: Illustrating the POS_MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MAX_STRICTLY_INCREASING_SEQUENCE.

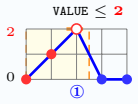
Constraint POS_MAX_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 6, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1105 provides an example where the POS_MAX_MAX_STRICTLY_INCREASING_SEQUENCE $(6, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

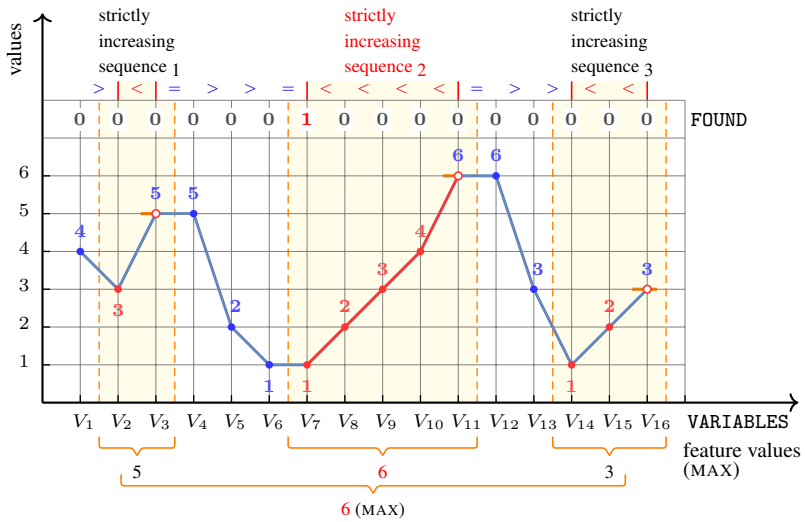


Figure 4.1105: Illustrating the POS_MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

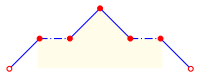
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_MAX_SUMMIT.

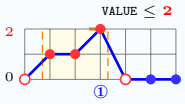
Constraint POS_MAX_MAX_SUMMIT(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv} + 1$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1106 provides an example where the POS_MAX_MAX_SUMMIT $(5, [7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

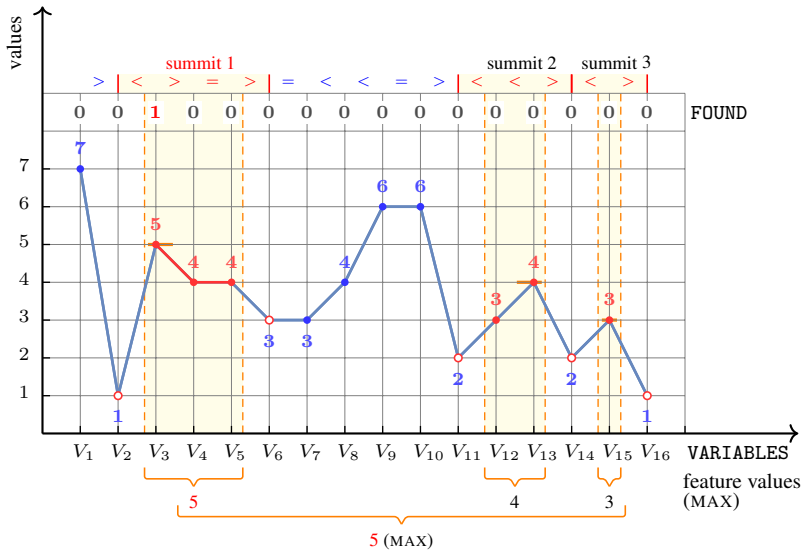


Figure 4.1106: Illustrating the POS_MAX_MAX_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

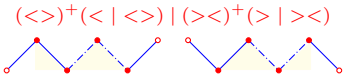
Automaton Similar to the automaton of the MAX_MAX_SUMMIT constraint but use the decoration table 3.35.

POS_MAX_MAX_SUMMIT 2167

AGGREGATOR FEATURE PATTERN

POS_MAX_MAX_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_MAX_ZIGZAG.

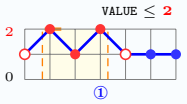
Constraint POS_MAX_MAX_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv} + 1$
$\mathtt{VALUE} \leq \mathtt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MAX_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 7, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1107 provides an example where the POS_MAX_MAX_ZIGZAG $(7, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

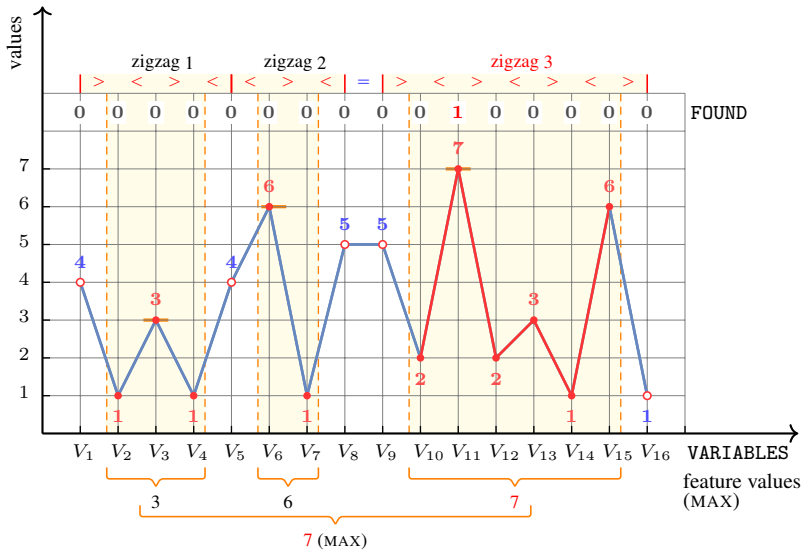


Figure 4.1107: Illustrating the POS_MAX_MAX_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

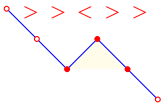
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MAX_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_BUMP_ON_DECREASING_SEQUENCE.

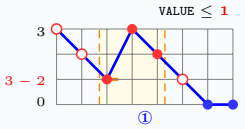
Constraint POS_MAX_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv}$
$\mathtt{VALUE} \leq \mathtt{maxv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+2$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3\rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1108 provides an example where the POS_MAX_MIN_BUMP_ON_DECREASING_SEQUENCE $(5, [7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

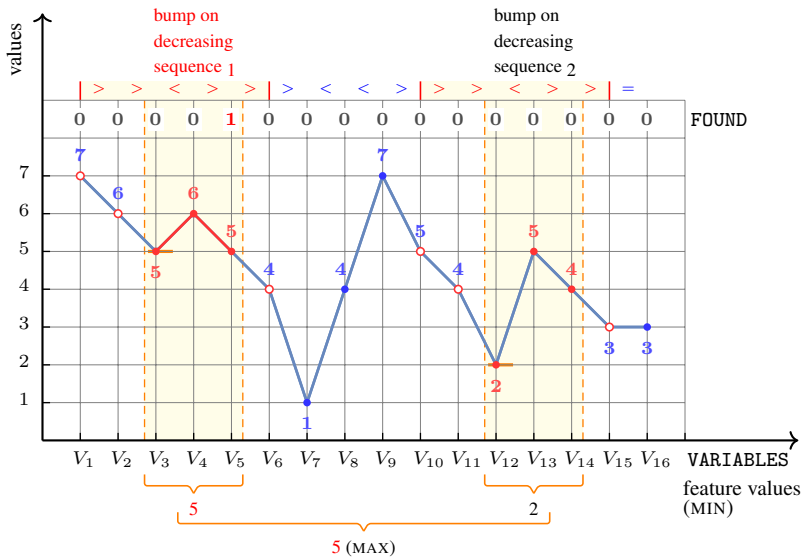


Figure 4.1108: Illustrating the POS_MAX_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_DECREASING.

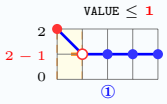
Constraint POS_MAX_MIN_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 4, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1109 provides an example where the POS_MAX_MIN_DECREASING $(4, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

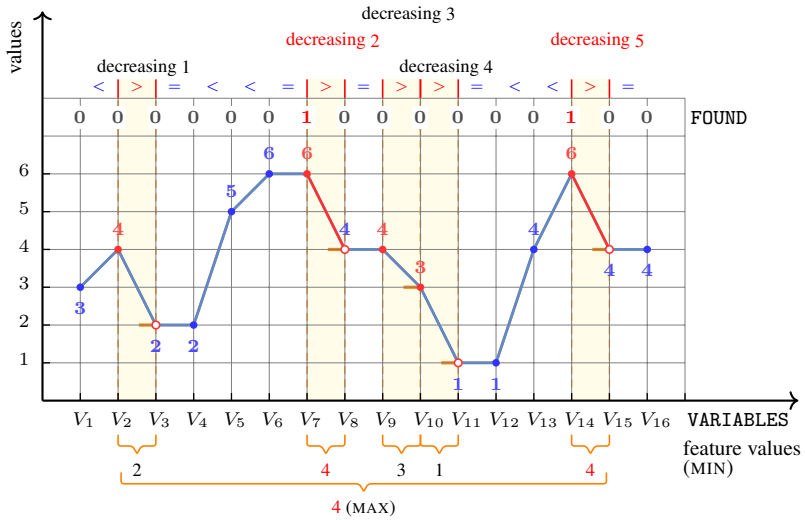


Figure 4.1109: Illustrating the POS_MAX_MIN_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

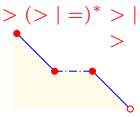
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_DECREASING_SEQUENCE.

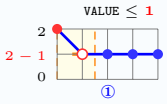
Constraint POS_MAX_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 4, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1110 provides an example where the POS_MAX_MIN_DECREASING_SEQUENCE $(4, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

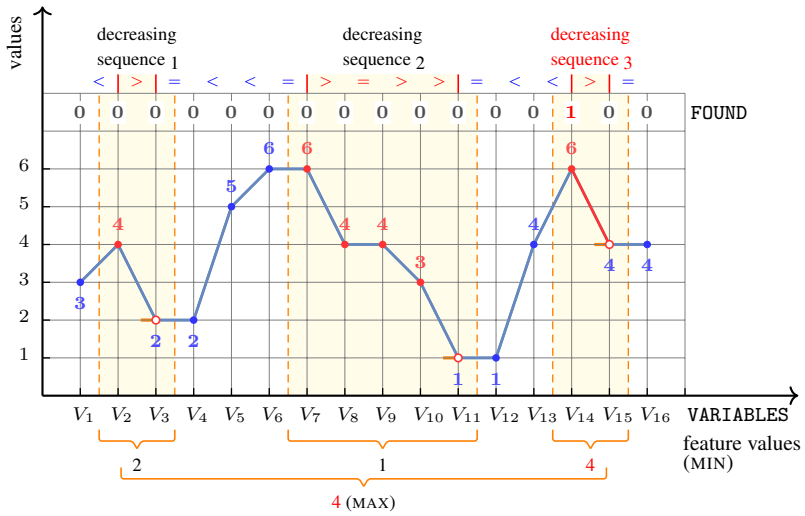


Figure 4.1110: Illustrating the POS_MAX_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

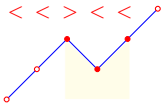
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_DIP_ON_INCREASING_SEQUENCE.

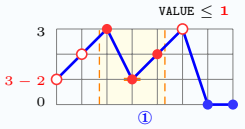
Constraint POS_MAX_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 2, \langle 1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4\rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1111 provides an example where the POS_MAX_MIN_DIP_ON_INCREASING_SEQUENCE $(2, [1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

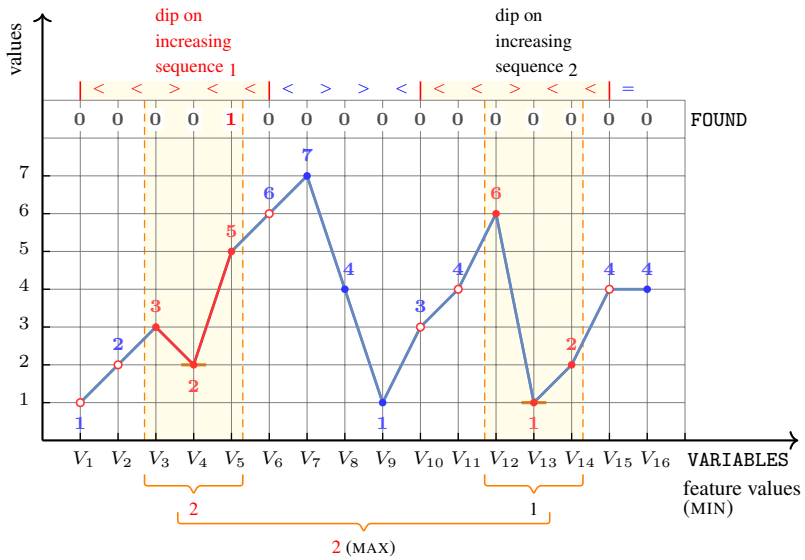


Figure 4.1111: Illustrating the POS_MAX_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

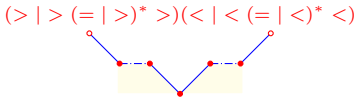
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_GORGE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_MIN_GORGE.

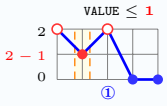
Constraint POS_MAX_MIN_GORGE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MAX_MIN_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $\left(\begin{array}{l} 5, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$

Figure 4.1112 provides an example where the POS_MAX_MIN_GORGE $(5, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

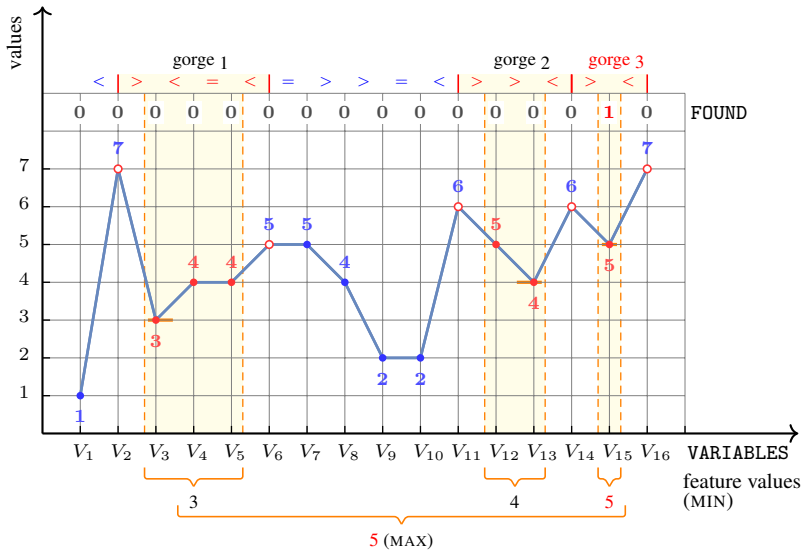


Figure 4.1112: Illustrating the POS_MAX_MIN_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_GORGE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_INCREASING.

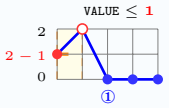
Constraint POS_MAX_MIN_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '$<$'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 4, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1113 provides an example where the POS_MAX_MIN_INCREASING $(4, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0])$ constraint holds.

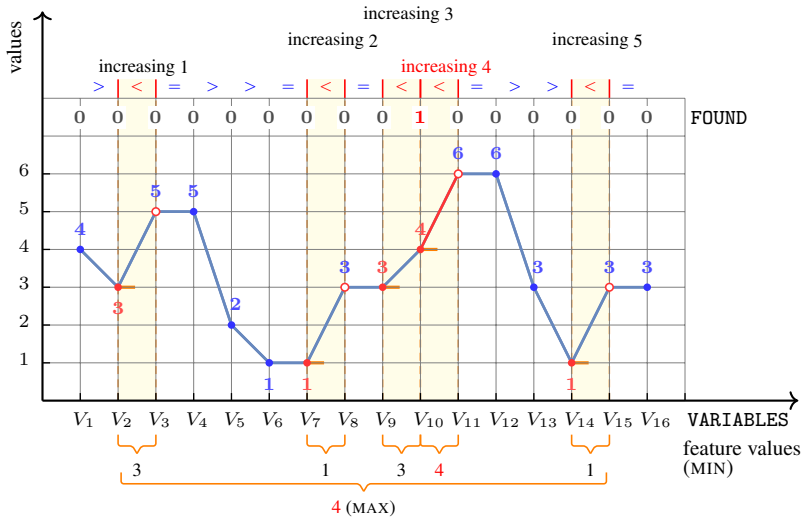


Figure 4.1113: Illustrating the POS_MAX_MIN_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

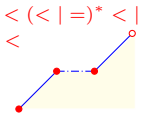
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_MIN_INCREASING_SEQUENCE.

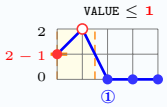
Constraint POS_MAX_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1114 provides an example where the POS_MAX_MIN_INCREASING_SEQUENCE $(3, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

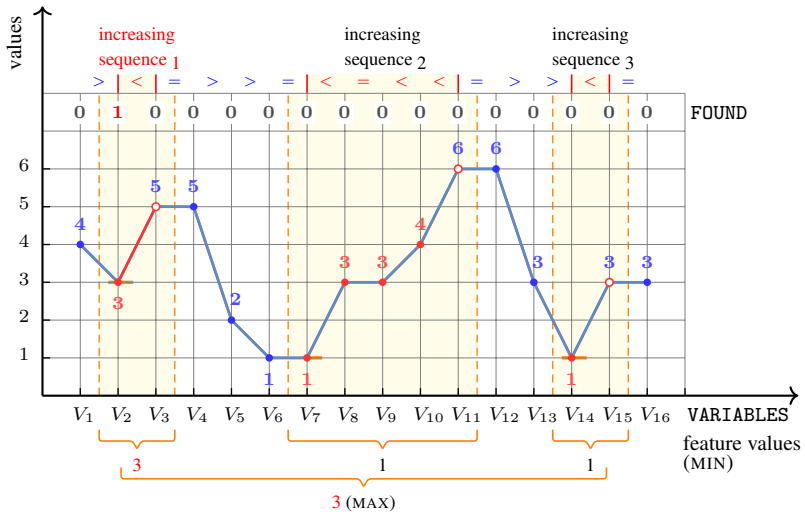


Figure 4.1114: Illustrating the POS_MAX_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

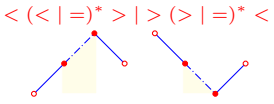
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_MIN_INFLEXION.

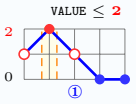
Constraint POS_MAX_MIN_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MIN_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1115 provides an example where the POS_MAX_MIN_INFLEXION $(5, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,0])$ constraint holds.

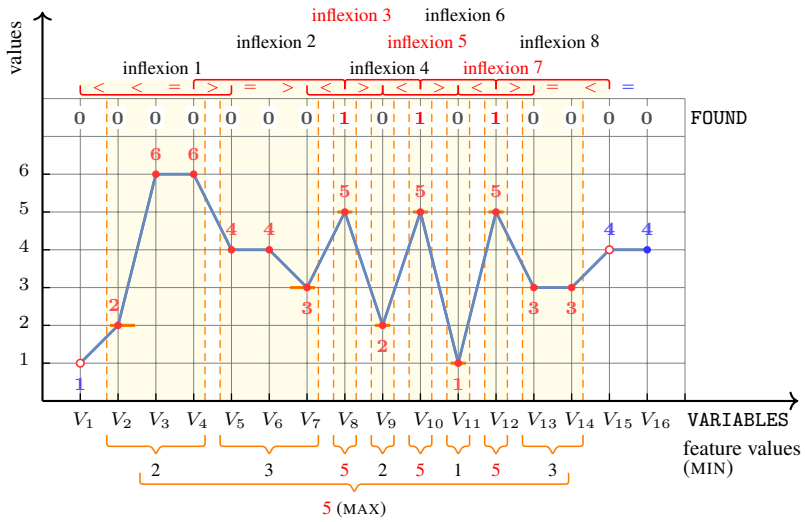


Figure 4.1115: Illustrating the POS_MAX_MIN_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_STRICTLY_DECREASING_SEQUENCE.

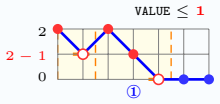
Constraint POS_MAX_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1116 provides an example where the POS_MAX_MIN_STRICTLY_DECREASING_SEQUENCE $(3, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

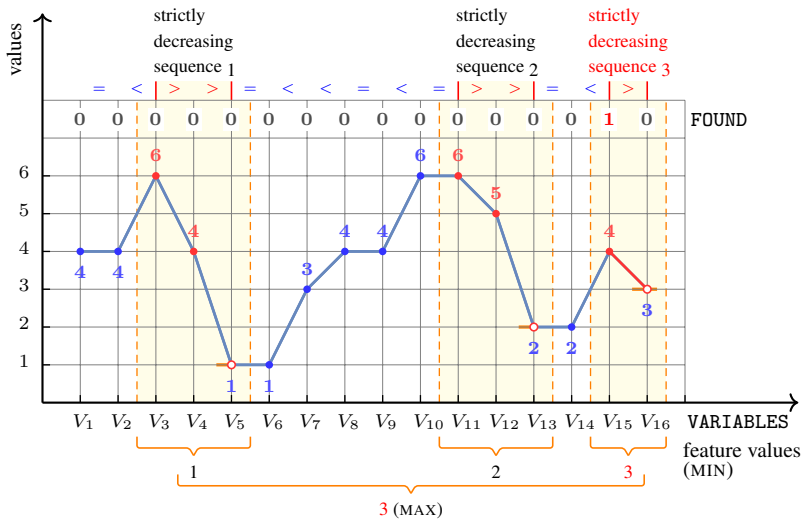


Figure 4.1116: Illustrating the POS_MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_MIN_STRICTLY_INCREASING_SEQUENCE.

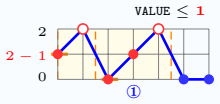
Constraint POS_MAX_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1117 provides an example where the POS_MAX_MIN_STRICTLY_INCREASING_SEQUENCE $(3, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

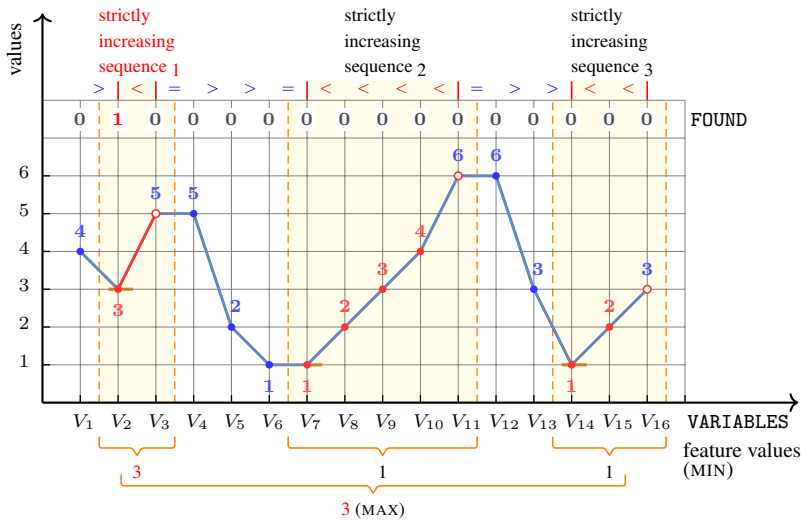


Figure 4.1117: Illustrating the POS_MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

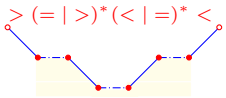
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_MIN_VALLEY.

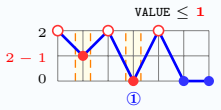
Constraint POS_MAX_MIN_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MIN_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1118 provides an example where the POS_MAX_MIN_VALLEY $(5, [1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

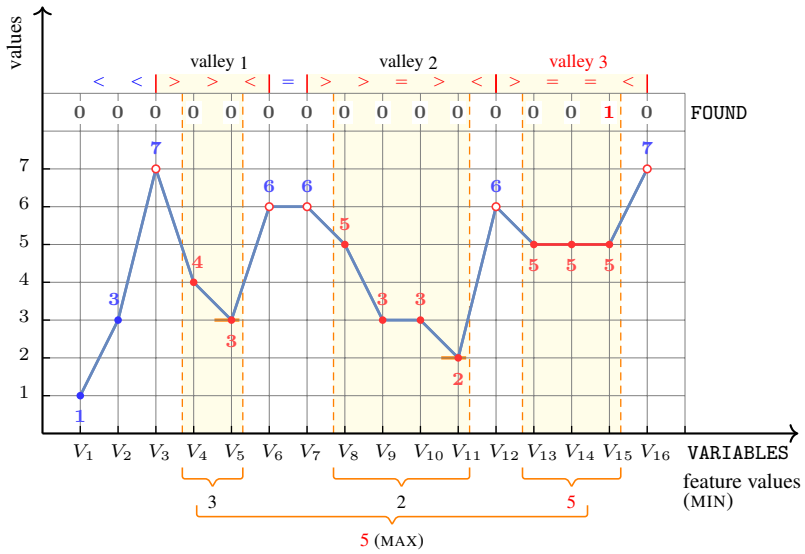


Figure 4.1118: Illustrating the POS_MAX_MIN_VALLEY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

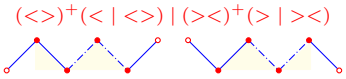
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_MIN_ZIGZAG

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_MIN_ZIGZAG.

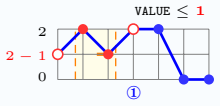
Constraint POS_MAX_MIN_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{minv}$
$\texttt{VALUE} \leq \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_MIN_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1 \rangle, \\ \langle 0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1119 provides an example where the POS_MAX_MIN_ZIGZAG $(1, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0])$ constraint holds.

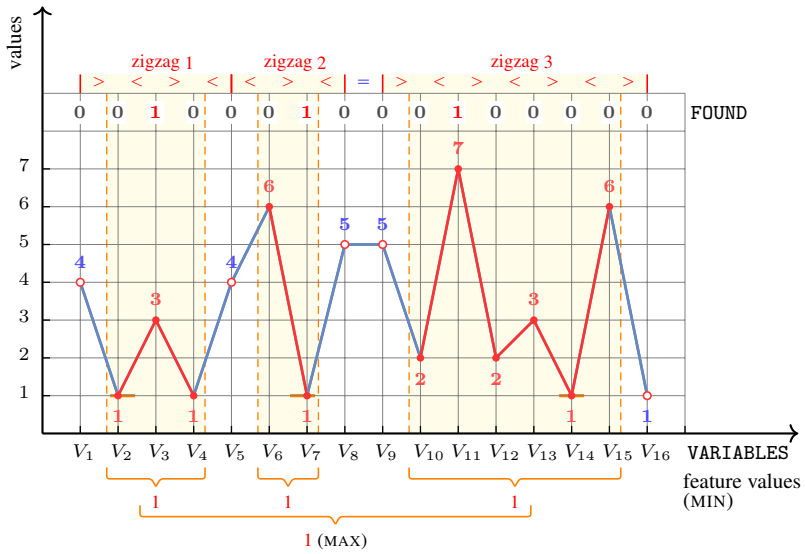


Figure 4.1119: Illustrating the POS_MAX_MIN_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

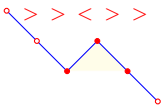
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_MIN_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_BUMP_ON_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_BUMP_ON_DECREASING_SEQUENCE.

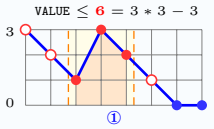
Constraint POS_MAX_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq 3 * \mathtt{minv} + 3$
$\mathtt{VALUE} \leq 3 * \mathtt{maxv} - 3$①
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[3, \mathtt{sv} - 1], \langle \mathtt{maxv} - 2, \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$
$\mathtt{n1} \geq \mathtt{VALUE} - 3 - \max(\mathtt{maxv} - 3, 0)$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[3, \mathtt{sv} - 1], \langle \mathtt{minv}, \mathtt{minv} + 1, \mathtt{minv} + 2 \rangle)$
$\mathtt{n2} \geq \min(\mathtt{minv} + 3, 0) - 3 - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example

$\left(16, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3 \rangle, \langle 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle \right)$

Figure 4.1120 provides an example where the POS_MAX_SURF_BUMP_ON_DECREASING_SEQUENCE $(16, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

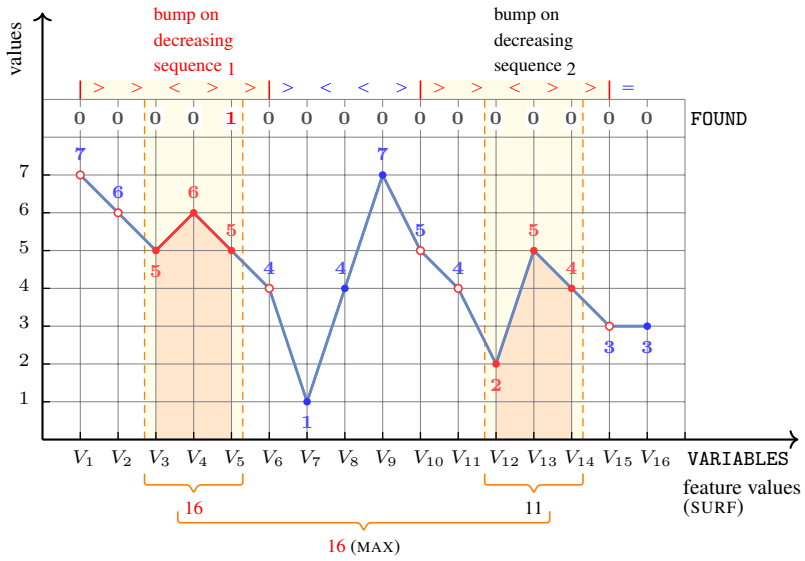


Figure 4.1120: Illustrating the POS_MAX_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_DECREASING.

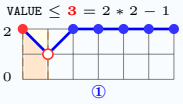
Constraint POS_MAX_SURF_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_SURF_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 10, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1121 provides an example where the POS_MAX_SURF_DECREASING $(10, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

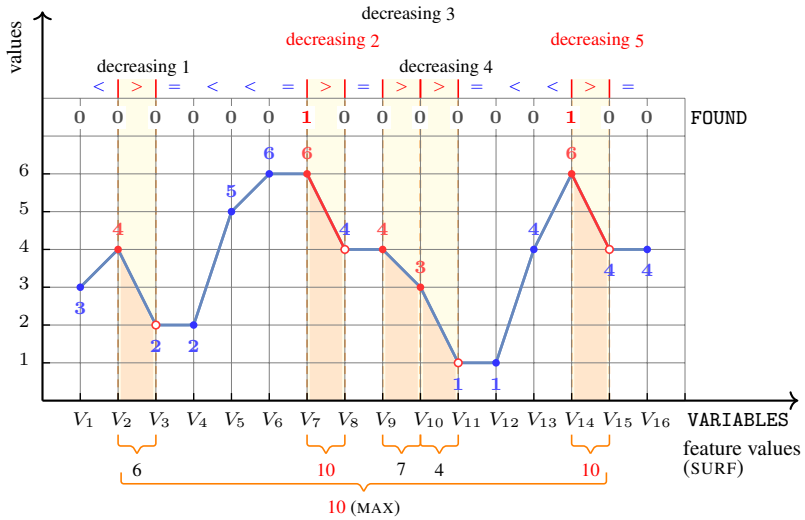


Figure 4.1121: Illustrating the POS_MAX_SURF_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

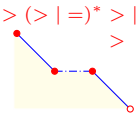
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_SURF_DECREASING_SEQUENCE.

Constraint POS_MAX_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

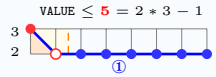
Arguments

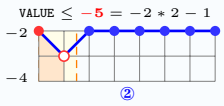
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

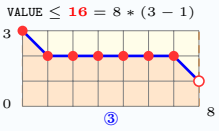
Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq 2 * \mathtt{minv} + 1$
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{min}(2 * \mathtt{minv} + 1, \mathtt{sv} * (\mathtt{minv} + 1))$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \leq 2 * \mathtt{maxv} - 1$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \leq \mathtt{max}(2 * \mathtt{maxv} - 1$②$, \mathtt{sv} * (\mathtt{maxv} - 1)$③$)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$
$\mathtt{rv} = 2 \vee \mathtt{maxv} = 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, 2 * \mathtt{maxv} - 3)$
$\mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{sv} * (\mathtt{maxv} - 2) - 1$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{minv}, \mathtt{minv} + 1 \rangle)$
$\mathtt{rv} = 2 \vee \mathtt{minv} = -1 \Rightarrow \mathtt{n2} \geq \mathtt{min}(0, 2 * \mathtt{minv} + 3) - \mathtt{VALUE}$
$\mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq \mathtt{sv} * (\mathtt{minv} + 2) - 1 - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$







Purpose

The constraint MAX_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '> (> | =)* > | >'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(18, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1122 provides an example where the POS_MAX_SURF_DECREASING_SEQUENCE $(18, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

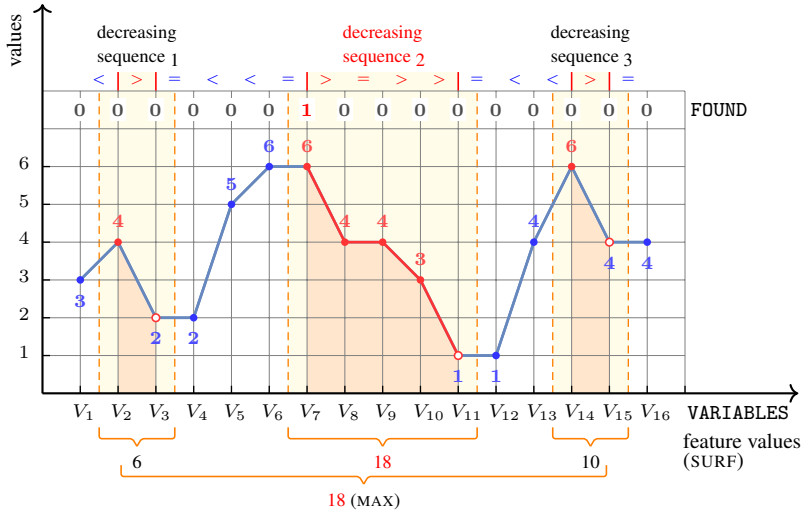


Figure 4.1122: Illustrating the POS_MAX_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

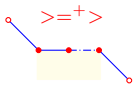
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_DECREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_DECREASING_TERRACE.

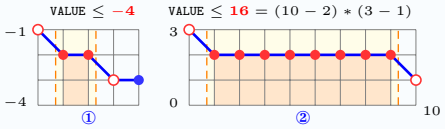
Constraint POS_MAX_SURF_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv}+1), (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv}-1)①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 8, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0 \rangle \end{array} \right)$$

Figure 4.1123 provides an example where the POS_MAX_SURF_DECREASING_TERRACE $(8, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0])$ constraint holds.

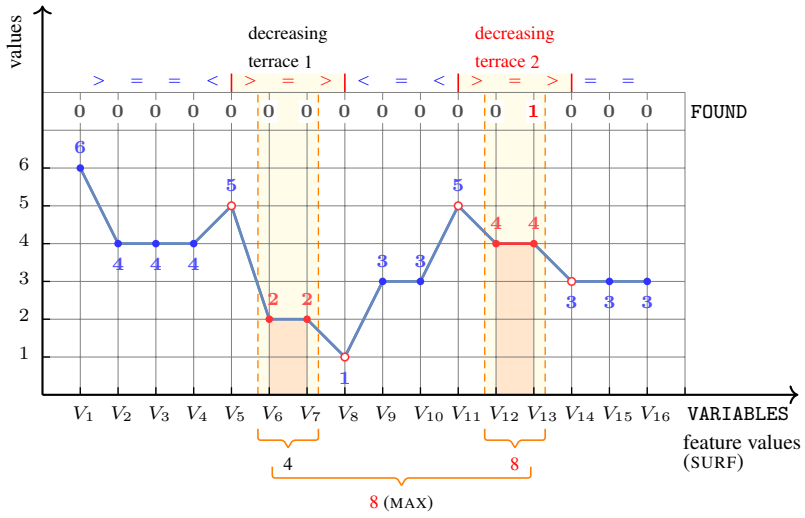


Figure 4.1123: Illustrating the POS_MAX_SURF_DECREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

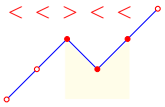
Arg. properties

- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MAX_SURF_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_DIP_ON_INCREASING_SEQUENCE.

Constraint POS_MAX_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

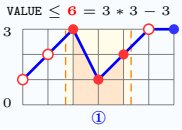
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 3 * \texttt{minv} + 3$
$\texttt{VALUE} \leq 3 * \texttt{maxv} - 3$①
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - 3 - \max(\texttt{maxv} - 3, 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{n2} \geq \min(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 10, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle, \\ \langle 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle \end{array} \right)$$

Figure 4.1124 provides an example where the POS_MAX_SURF_DIP_ON_INCREASING_SEQUENCE $(10, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4], [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

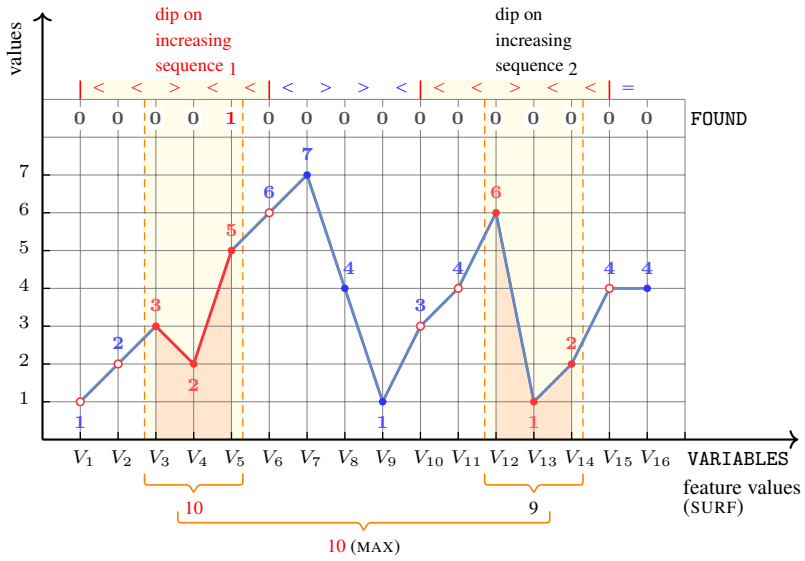


Figure 4.1124: Illustrating the POS_MAX_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

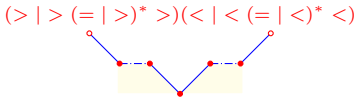
- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MAX_SURF_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_GORGE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_SURF_GORGE.

Constraint POS_MAX_SURF_GORGE(VALUE, VARIABLES, FOUND)

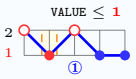
Arguments

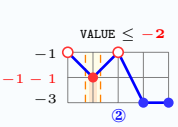
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

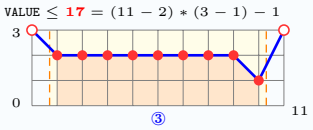
Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv}$
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{min}(\mathtt{minv}, (\mathtt{sv}-2) * (\mathtt{minv}+1) - 1)$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \leq \mathtt{maxv} - 1$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \leq \mathtt{max}(\mathtt{maxv} - 1$②$, (\mathtt{sv}-2) * (\mathtt{maxv}-1) - 1$③$)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1 \rangle)$
$\mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - (\mathtt{sv}-2) * (\mathtt{maxv}-2)$
$\mathtt{rv} = 2 \vee \mathtt{maxv} = 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, \mathtt{maxv}-2)$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}, \mathtt{minv}+1 \rangle)$
$\mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq (\mathtt{sv}-2) * (\mathtt{minv}+2) - 1 - \mathtt{VALUE}$
$\mathtt{rv} = 2 \vee \mathtt{minv} = -1 \Rightarrow \mathtt{n2} \geq \mathtt{min}(0, \mathtt{minv}+1) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$







Purpose

The constraint MAX_SURF_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(11, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7\rangle, \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1125 provides an example where the POS_MAX_SURF_GORGE $(11, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

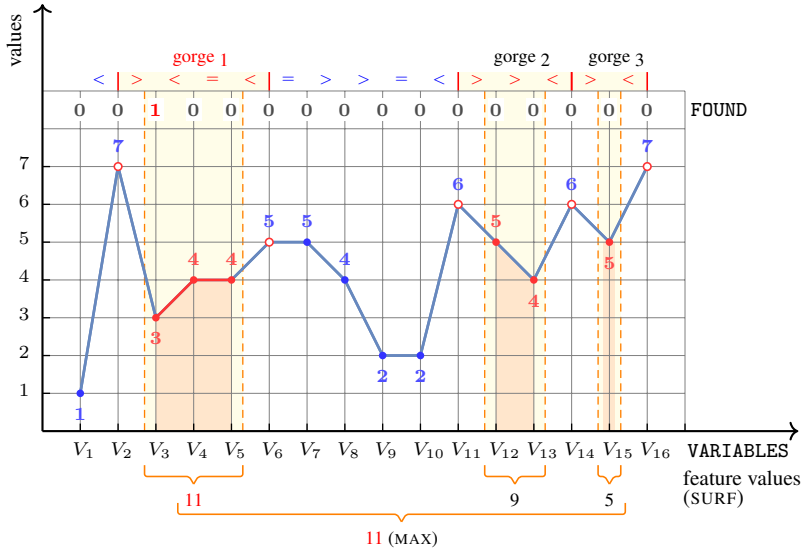


Figure 4.1125: Illustrating the POS_MAX_SURF_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_GORGE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_INCREASING.

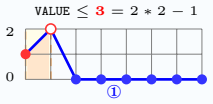
Constraint POS_MAX_SURF_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
maxv =maxval(VARIABLES.var)
minv =minval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MAX_SURF_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 10, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1126 provides an example where the POS_MAX_SURF_INCREASING $(10, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0])$ constraint holds.

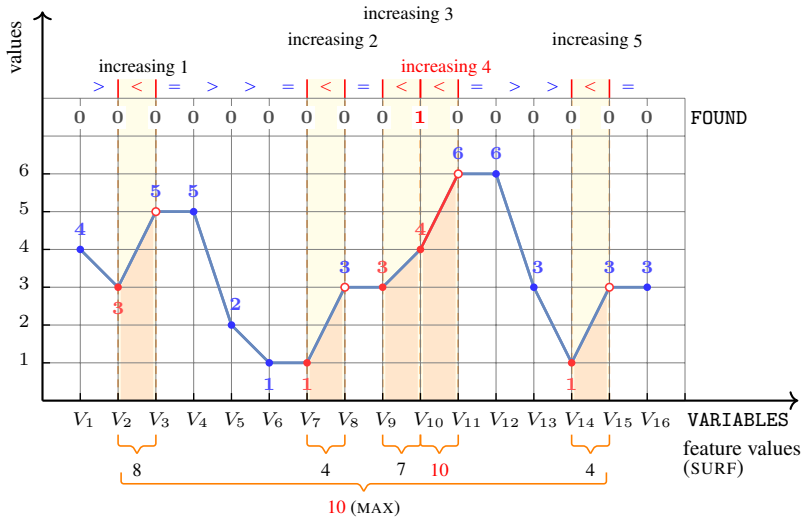


Figure 4.1126: Illustrating the POS_MAX_SURF_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

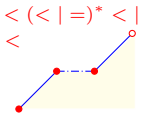
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_SURF_INCREASING_SEQUENCE.

Constraint POS_MAX_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

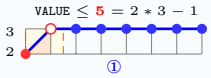
Arguments

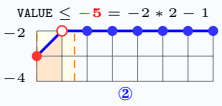
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

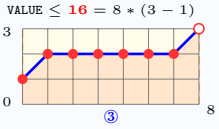
Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv} + 1, \texttt{sv} * (\texttt{minv} + 1))$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv} - 1$②$, \texttt{sv} * (\texttt{maxv} - 1)$③$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, 2 * \texttt{maxv} - 3)$
$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{sv} * (\texttt{maxv} - 2) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \texttt{min}(0, 2 * \texttt{minv} + 3) - \texttt{VALUE}$
$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - 1 - \texttt{VALUE}$
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|

where
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)
sv = |VARIABLES|
minv =minval(VARIABLES.var)







Purpose

The constraint MAX_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(17, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle , \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \right)$$

Figure 4.1127 provides an example where the POS_MAX_SURF_INCREASING_SEQUENCE $(17, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

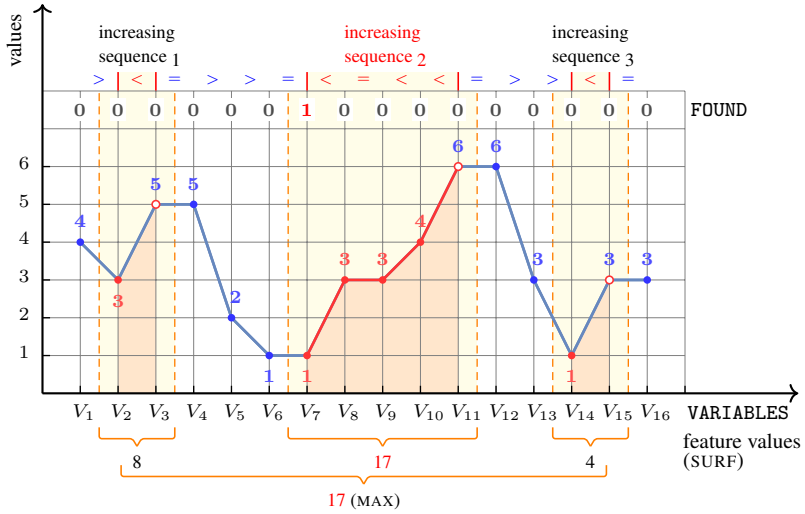


Figure 4.1127: Illustrating the POS_MAX_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

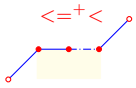
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_INCREASING_TERRACE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_SURF_INCREASING_TERRACE.

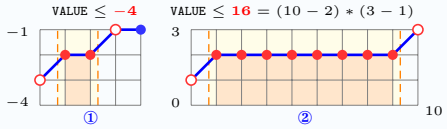
Constraint POS_MAX_SURF_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv}+1), (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv}-1)①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(10, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4 \rangle, \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \right)$$

Figure 4.1128 provides an example where the POS_MAX_SURF_INCREASING_TERRACE $(10, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4], [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

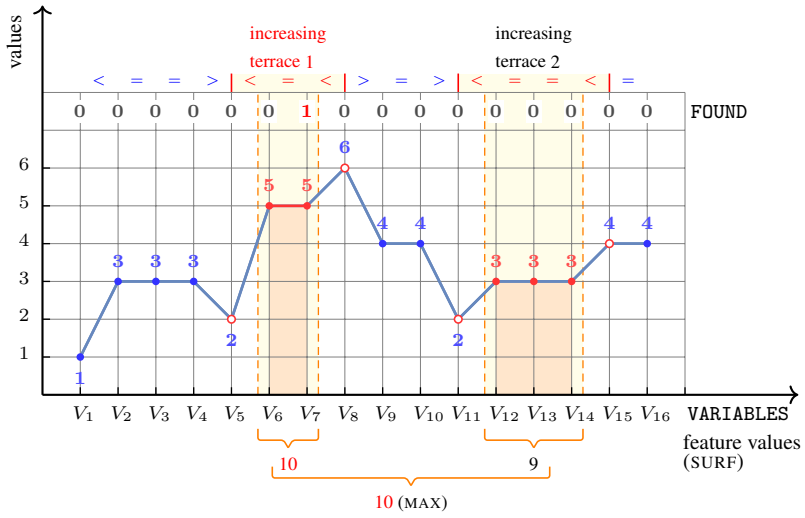


Figure 4.1128: Illustrating the POS_MAX_SURF_INCREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

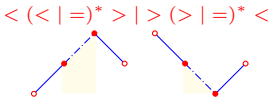
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_INFLEXION.

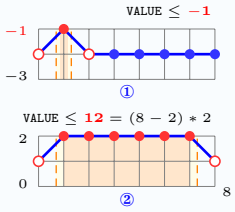
Constraint POS_MAX_SURF_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \min(\mathtt{minv}, (\mathtt{sv}-2) * \mathtt{minv})$
$\mathtt{VALUE} \leq \max(\mathtt{maxv}①, (\mathtt{sv}-2) * \mathtt{maxv}②)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv} \rangle)$
$\mathtt{n1} \geq \mathtt{VALUE} - \max(0, (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv} \rangle)$
$\mathtt{n2} \geq \min(0, (\mathtt{sv}-2) * (\mathtt{minv}+1)) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 14, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4 \rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1129 provides an example where the POS_MAX_SURF_INFLEXION $(14, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

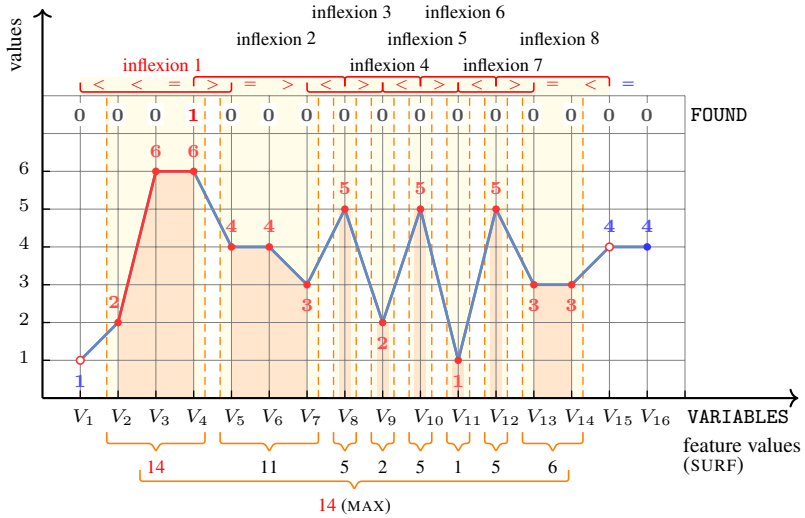


Figure 4.1129: Illustrating the POS_MAX_SURF_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

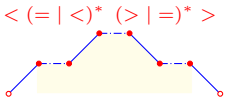
- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MAX_SURF_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_PEAK

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_PEAK.

Constraint POS_MAX_SURF_PEAK(VALUE, VARIABLES, FOUND)

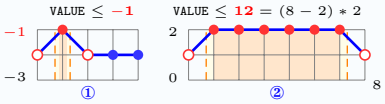
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}+1, (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \texttt{max}(\texttt{maxv}①, (\texttt{sv}-2) * \texttt{maxv}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(14, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0 \rangle \right)$$

Figure 4.1130 provides an example where the POS_MAX_SURF_PEAK $(14, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0])$ constraint holds.

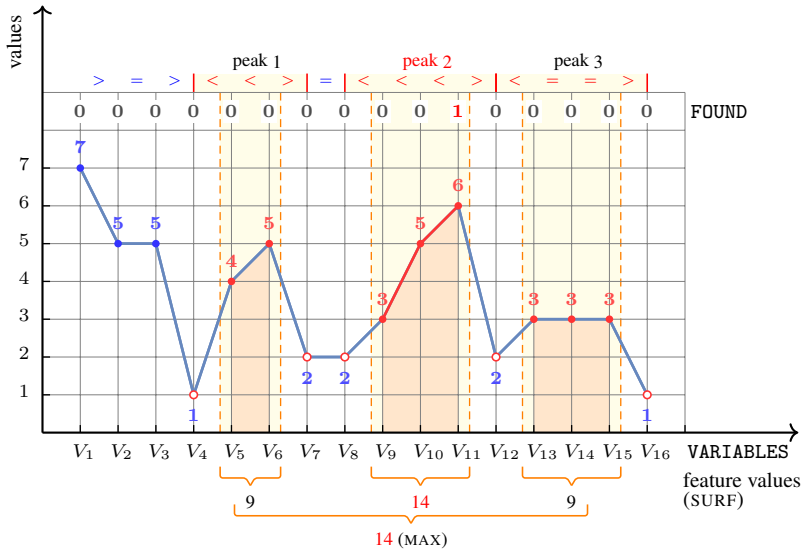


Figure 4.1130: Illustrating the POS_MAX_SURF_PEAK constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_PLAIN

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_SURF_PLAIN.

Constraint POS_MAX_SURF_PLAIN(VALUE, VARIABLES, FOUND)

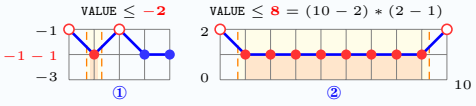
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(\texttt{minv}, (\texttt{sv}-2) * \texttt{minv})$
$\texttt{VALUE} \leq \max(\texttt{maxv} - 1①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0 \rangle \end{array} \right)$$

Figure 4.1131 provides an example where the POS_MAX_SURF_PLAIN $(6, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0])$ constraint holds.

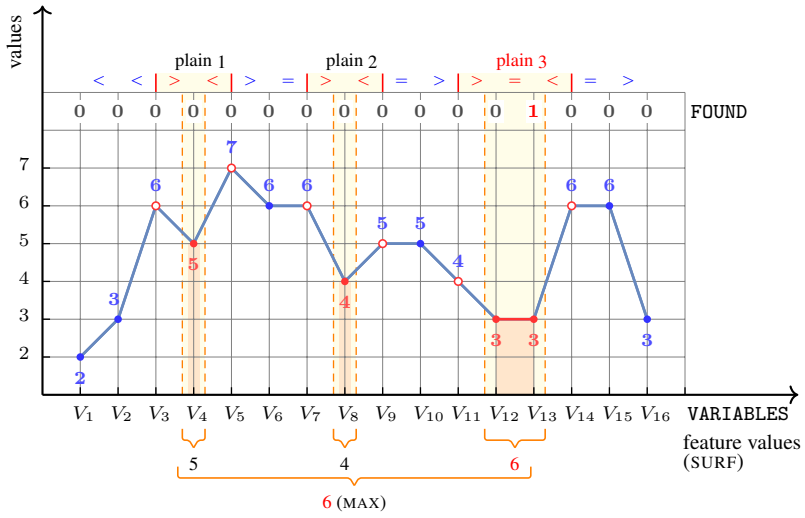


Figure 4.1131: Illustrating the POS_MAX_SURF_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_PLATEAU.

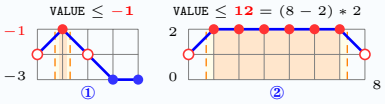
Constraint POS_MAX_SURF_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{min}(\mathtt{minv}+1, (\mathtt{sv}-2) * (\mathtt{minv}+1))$
$\mathtt{VALUE} \leq \mathtt{max}(\mathtt{maxv}①, (\mathtt{sv}-2) * \mathtt{maxv}②)$
among(n1, VARIABLES[2, sv − 1], ⟨maxv⟩)
$\mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
among(n2, VARIABLES[2, sv − 1], ⟨minv + 1⟩)
$\mathtt{n2} \geq \mathtt{min}(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \mathtt{VALUE}$
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
minv =minval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose

The constraint MAX_SURF_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '<=* >'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 10, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1132 provides an example where the POS_MAX_SURF_PLATEAU $(10, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0])$ constraint holds.

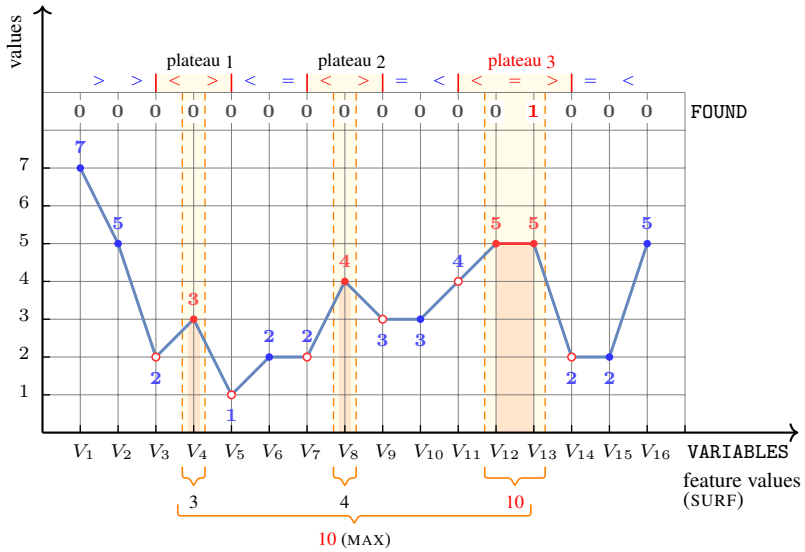


Figure 4.1132: Illustrating the POS_MAX_SURF_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

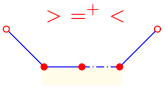
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_PROPER_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_PROPER_PLAIN.

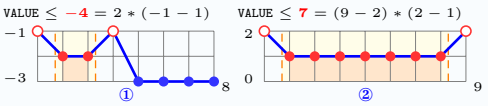
Constraint POS_MAX_SURF_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * \texttt{minv}, (\texttt{sv}-2) * \texttt{minv})$
$\texttt{VALUE} \leq \max(2 * (\texttt{maxv}-1)①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 10, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5 \rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1133 provides an example where the POS_MAX_SURF_PROPER_PLAIN $(10, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

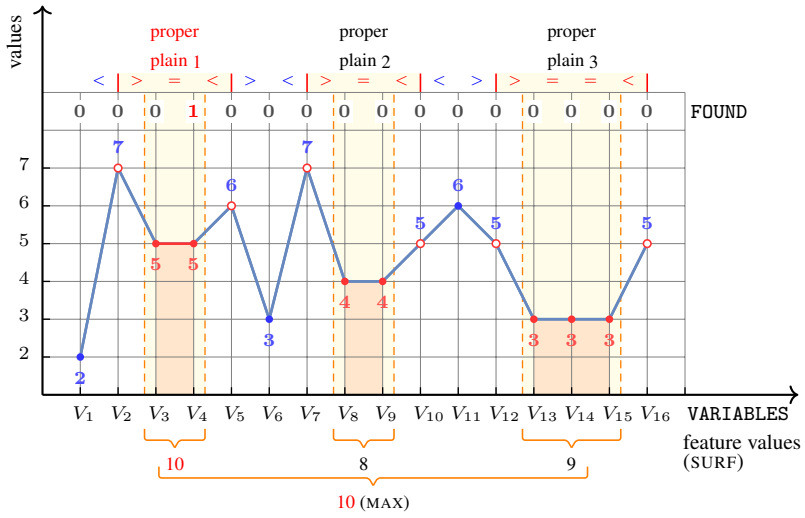


Figure 4.1133: Illustrating the POS_MAX_SURF_PROPER_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

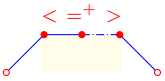
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_PROPER_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_SURF_PROPER_PLATEAU.

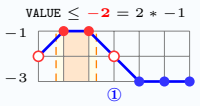
Constraint POS_MAX_SURF_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

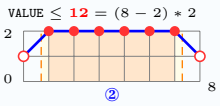
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv}+1), (\texttt{sv}-2) * (\texttt{minv}+1))$
$\texttt{VALUE} \leq \max(2 * \texttt{maxv}①, (\texttt{sv}-2) * \texttt{maxv}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}+1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$





Purpose

The constraint MAX_SURF_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^+>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(15, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \right)$$

Figure 4.1134 provides an example where the POS_MAX_SURF_PROPER_PLATEAU $(15, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])$ constraint holds.

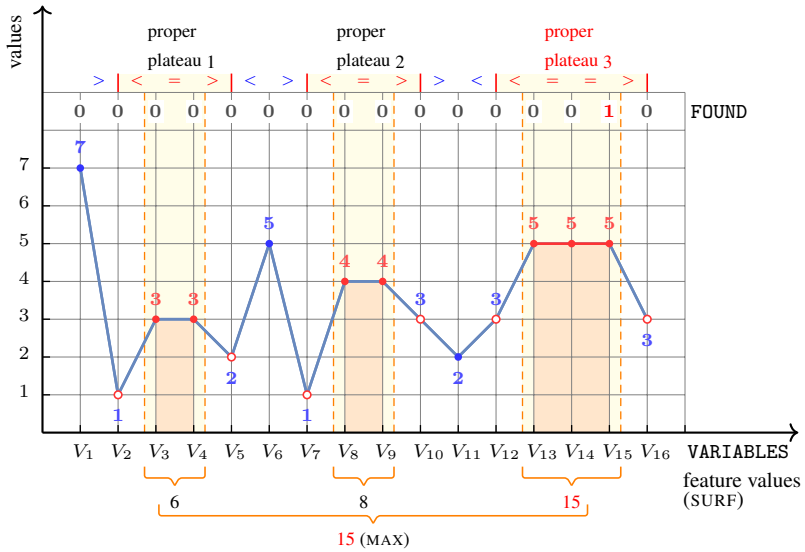


Figure 4.1134: Illustrating the POS_MAX_SURF_PROPER_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_STEADY.

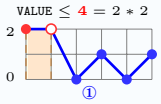
Constraint POS_MAX_SURF_STEADY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv}$
$\texttt{VALUE} \leq 2 * \texttt{maxv}$①
among(n1, VARIABLES[1, sv], ⟨maxv⟩)
$\texttt{n1} \geq \texttt{VALUE} - 2 * (\texttt{maxv} - 1)$
among(n2, VARIABLES[1, sv], ⟨minv⟩)
$\texttt{n2} \geq 2 * (\texttt{minv} + 1) - \texttt{VALUE}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \text{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \text{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose The constraint MAX_SURF_STEADY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $\left(\begin{array}{l} 12, \langle 1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$

Figure 4.1135 provides an example where the POS_MAX_SURF_STEADY $(12, [1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

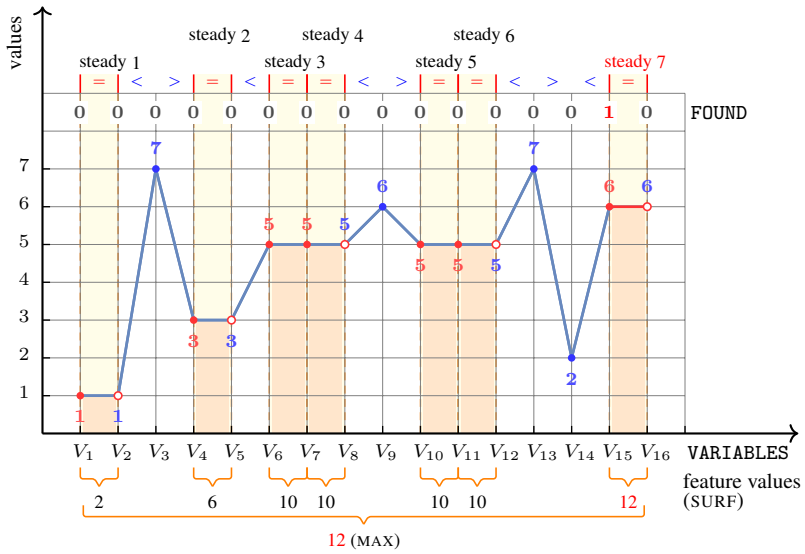


Figure 4.1135: Illustrating the POS_MAX_SURF_STEADY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

Arg. properties

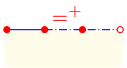
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_STEADY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_SURF_STEADY_SEQUENCE.

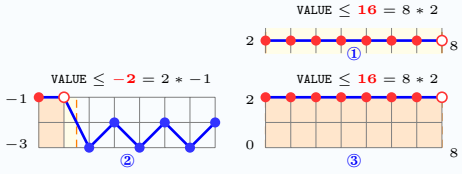
Constraint POS_MAX_SURF_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{sv} * \texttt{minv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}, \texttt{sv} * \texttt{minv})$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \leq \texttt{sv} * \texttt{maxv}①$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}②, \texttt{sv} * \texttt{maxv}③)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{sv} * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, \texttt{sv} * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(15, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle, \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1136 provides an example where the POS_MAX_SURF_STEADY_SEQUENCE $(15, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

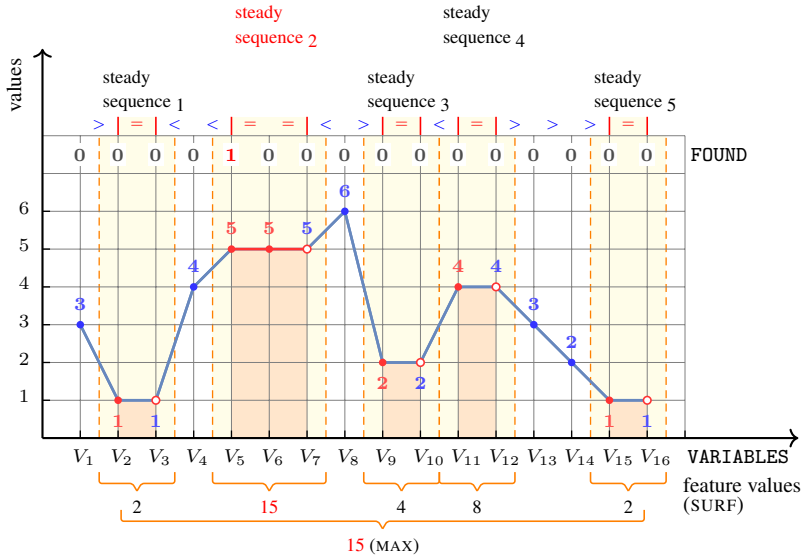


Figure 4.1136: Illustrating the POS_MAX_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_STRICTLY_DECREASING_SEQUENCE.

Constraint POS_MAX_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

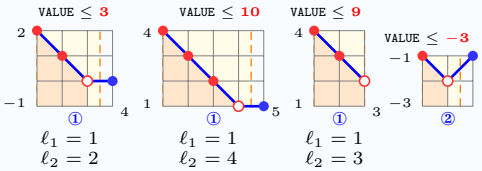
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$①
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$②
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(13, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0 \rangle \right)$$

Figure 4.1137 provides an example where the POS_MAX_SURF_STRICTLY_DECREASING_SEQUENCE $(13, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

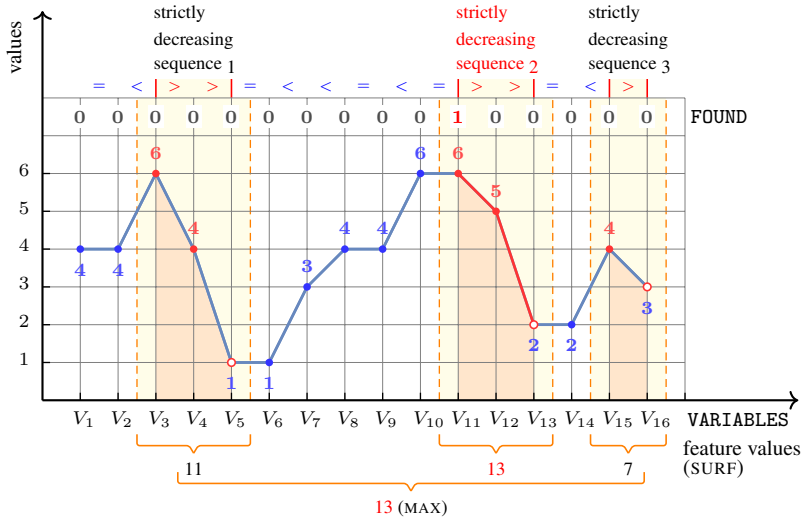


Figure 4.1137: Illustrating the POS_MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_SURF_STRICTLY_INCREASING_SEQUENCE.

Constraint POS_MAX_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

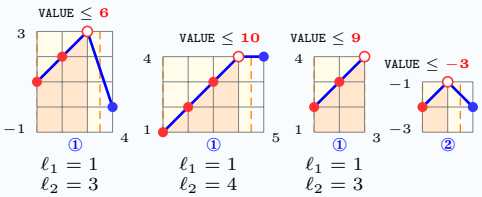
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq 2 * \texttt{minv} + 1$
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$①
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$②
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(16, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3 \rangle, \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \right)$$

Figure 4.1138 provides an example where the POS_MAX_SURF_STRICTLY_INCREASING_SEQUENCE $(16, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

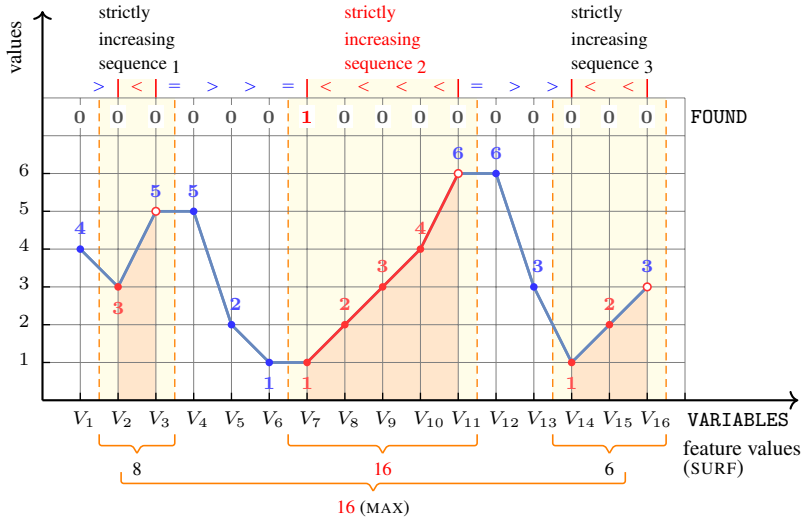


Figure 4.1138: Illustrating the POS_MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

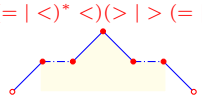
Automaton Similar to the automaton of the MAX_SURF_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_SUMMIT.

Constraint POS_MAX_SURF_SUMMIT(VALUE, VARIABLES, FOUND)

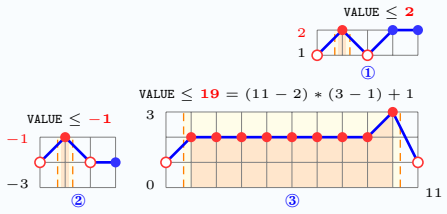
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = -\infty$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{minv} + 1$
$\mathtt{rv} \geq 3 \Rightarrow$
$\mathtt{VALUE} = -\infty \vee \mathtt{VALUE} \geq \mathtt{min}(\mathtt{minv} + 1, (\mathtt{sv} - 2) * (\mathtt{minv} + 1) + 1)$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \leq \mathtt{maxv}$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \leq \mathtt{max}(\mathtt{maxv}$②$, (\mathtt{sv} - 2) * (\mathtt{maxv} - 1) + 1$③$)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv} - 1], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$
$\mathtt{rv} = 2 \vee \mathtt{maxv} = 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, \mathtt{maxv} - 1)$
$\mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - (\mathtt{sv} - 2) * (\mathtt{maxv} - 2) - 1$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv} - 1], \langle \mathtt{minv} + 1 \rangle)$
$\mathtt{rv} = 2 \vee \mathtt{minv} = -1 \Rightarrow \mathtt{n2} \geq \mathtt{min}(0, \mathtt{minv} + 2) - \mathtt{VALUE}$
$\mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq (\mathtt{sv} - 2) * (\mathtt{minv} + 2) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$



Purpose

The constraint MAX_SURF_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example

$$\left(13, \langle 7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1\rangle, \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1139 provides an example where the POS_MAX_SURF_SUMMIT $(13, [7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

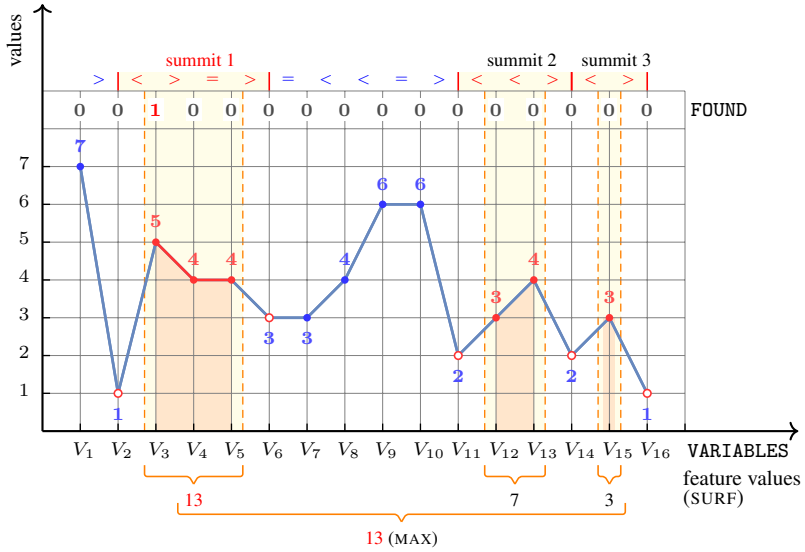


Figure 4.1139: Illustrating the POS_MAX_SURF_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

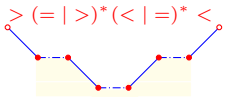
Automaton Similar to the automaton of the MAX_SURF_SUMMIT constraint but use the decoration table 3.35.

POS_MAX_SURF_SUMMIT 2299

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_SURF_VALLEY.

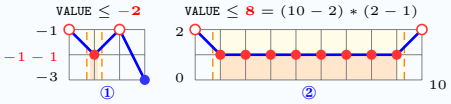
Constraint POS_MAX_SURF_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$
$\texttt{VALUE} = -\infty \vee \texttt{VALUE} \geq \min(\texttt{minv}, (\texttt{sv}-2) * \texttt{minv})$
$\texttt{VALUE} \leq \max(\texttt{maxv} - 1①, (\texttt{sv}-2) * (\texttt{maxv}-1)②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$



where
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

Purpose

The constraint MAX_SURF_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(15, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \right)$$

Figure 4.1140 provides an example where the POS_MAX_SURF_VALLEY $(15, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])$ constraint holds.

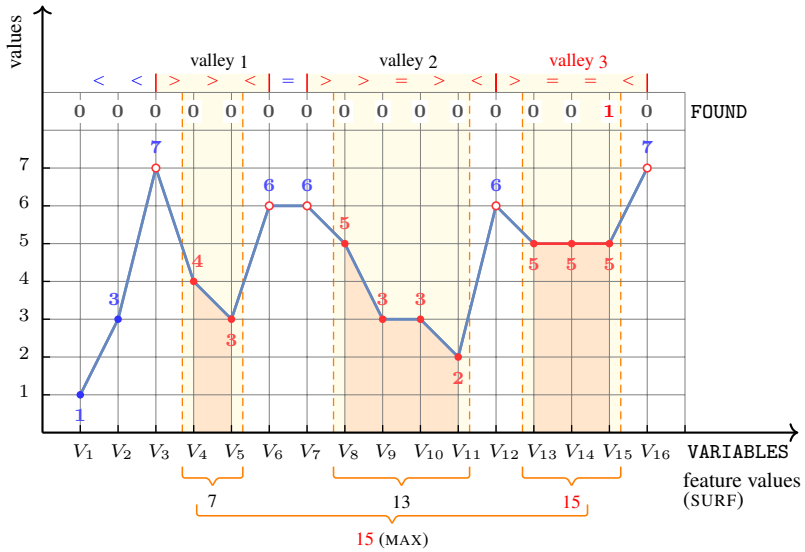


Figure 4.1140: Illustrating the POS_MAX_SURF_VALLEY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

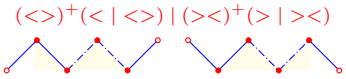
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_SURF_ZIGZAG

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_SURF_ZIGZAG.

Constraint POS_MAX_SURF_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = -\infty$$

$$\bigvee \left(\begin{array}{l} \texttt{VALUE} = -\infty, \\ \texttt{VALUE} \geq \min \left(\begin{array}{l} 2 * \texttt{minv} + 1, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{minv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{minv}+1) \end{array} \right) \end{array} \right)$$

$$\texttt{VALUE} \leq \max \left(\begin{array}{l} 2 * \texttt{maxv} - 1 ①, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{maxv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{maxv}-1) ② \end{array} \right)$$

$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1, \texttt{maxv} \rangle)$$

$$\texttt{n1} \geq \texttt{VALUE} - \lfloor (\texttt{sv}-1)/2 \rfloor - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$$

$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}, \texttt{minv}+1 \rangle)$$

$$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \lfloor (\texttt{sv}-1)/2 \rfloor - \texttt{VALUE}$$

$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

$$\texttt{required}(\texttt{FOUND}, \texttt{var})$$

$$|\texttt{VARIABLES}| = |\texttt{FOUND}|$$

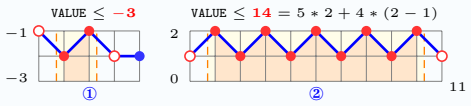
where

$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$

$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$

$$\texttt{sv} = |\texttt{VARIABLES}|$$

$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose

The constraint MAX_SURF_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(21, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \right)$$

Figure 4.1141 provides an example where the POS_MAX_SURF_ZIGZAG $(21, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0])$ constraint holds.

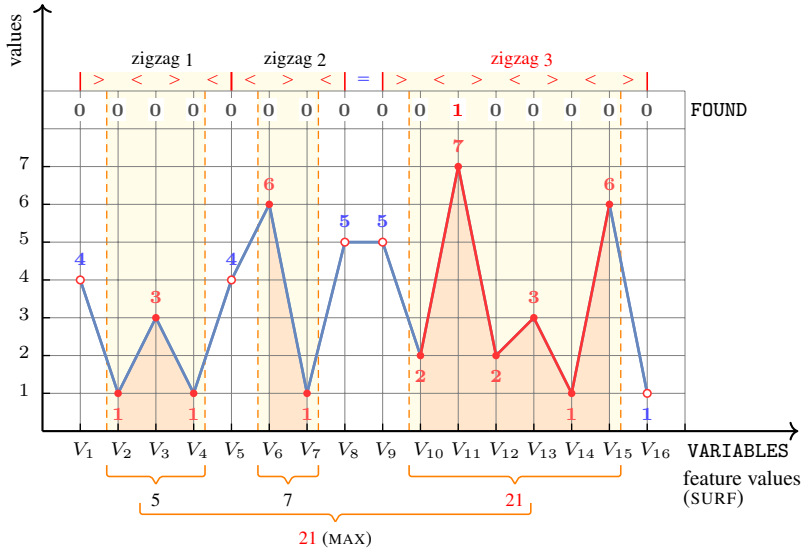


Figure 4.1141: Illustrating the POS_MAX_SURF_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

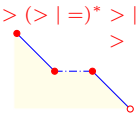
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_SURF_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_WIDTH_DECREASING_SEQUENCE.

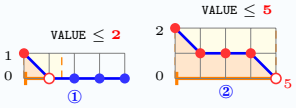
Constraint POS_MAX_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '> (> | =)* > | >'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 5, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1142 provides an example where the POS_MAX_WIDTH_DECREASING_SEQUENCE $(5, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

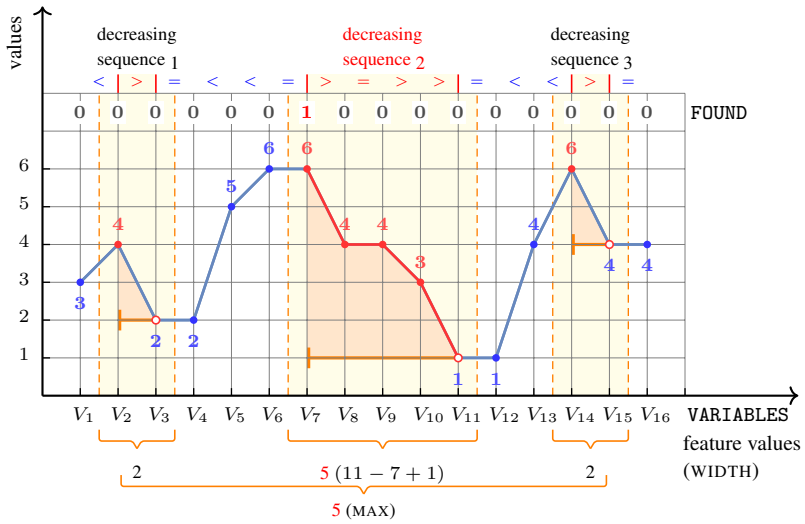


Figure 4.1142: Illustrating the POS_MAX_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

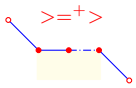
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_DECREASING_TERRACE.

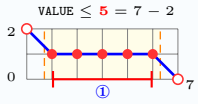
Constraint POS_MAX_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = 0$
$\mathtt{VALUE} = 0 \vee \mathtt{VALUE} \geq 2$
$\mathtt{VALUE} \leq \max(0, \mathtt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1143 provides an example where the POS_MAX_WIDTH_DECREASING_TERRACE $(2, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

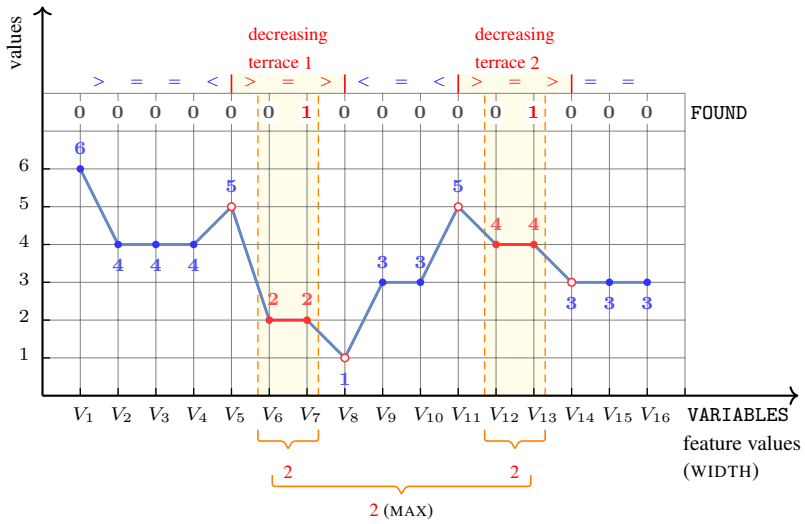


Figure 4.1143: Illustrating the POS_MAX_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Arg. properties

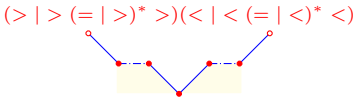
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_WIDTH_GORGE.

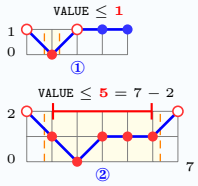
Constraint POS_MAX_WIDTH_GORGE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1144 provides an example where the POS_MAX_WIDTH_GORGE $(3, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

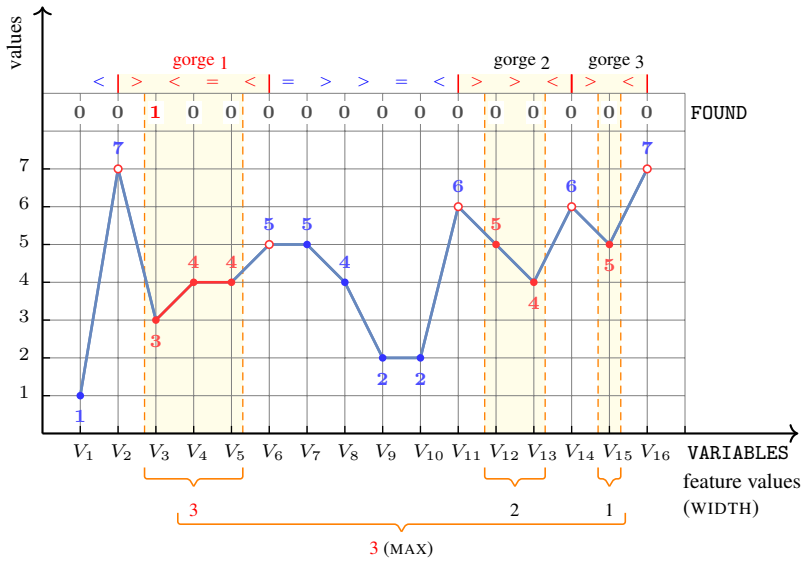


Figure 4.1144: Illustrating the POS_MAX_WIDTH_GORGE constraint of the **Example** slot

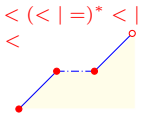
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_GORGE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_WIDTH_INCREASING_SEQUENCE.

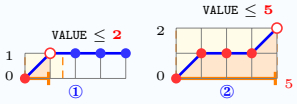
Constraint POS_MAX_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 5, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1145 provides an example where the POS_MAX_WIDTH_INCREASING_SEQUENCE $(5, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

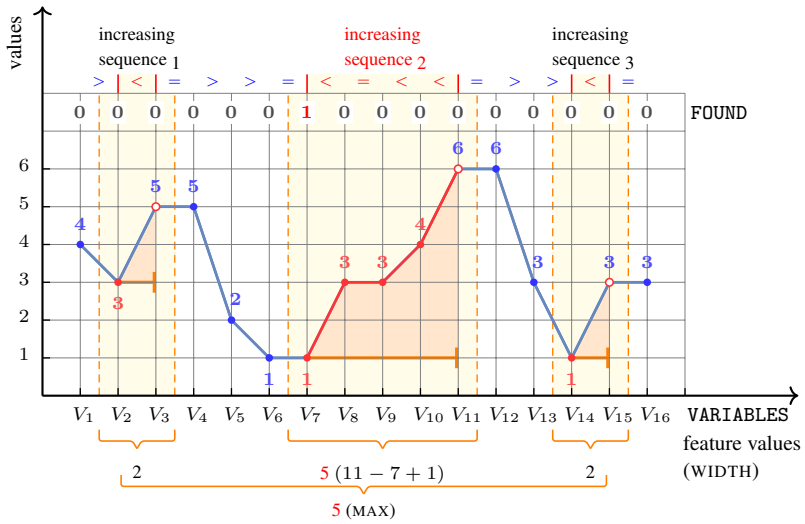


Figure 4.1145: Illustrating the POS_MAX_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

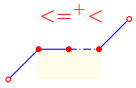
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_WIDTH_INCREASING_TERRACE.

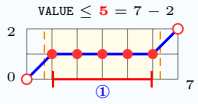
Constraint POS_MAX_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^+<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1146 provides an example where the POS_MAX_WIDTH_INCREASING_TERRACE $(3, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

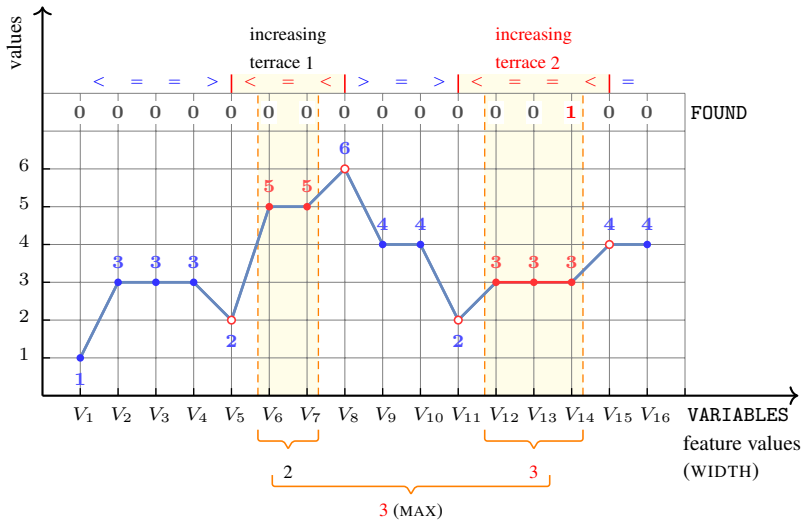


Figure 4.1146: Illustrating the POS_MAX_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

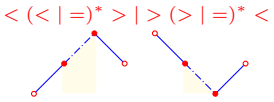
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_INFLEXION



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_WIDTH_INFLEXION.

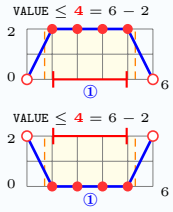
Constraint POS_MAX_WIDTH_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4 \rangle, \\ \langle 0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1147 provides an example where the POS_MAX_WIDTH_INFLEXION $(3, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

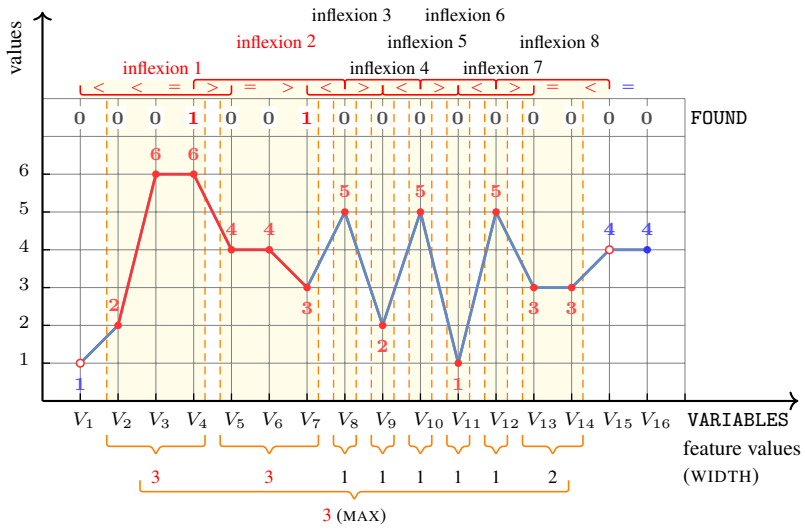


Figure 4.1147: Illustrating the POS_MAX_WIDTH_INFLEXION constraint of the **Example** slot

Arg. properties

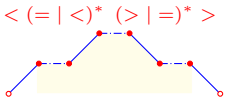
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_PEAK.

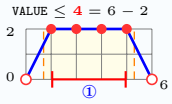
Constraint POS_MAX_WIDTH_PEAK(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1148 provides an example where the POS_MAX_WIDTH_PEAK $(3, [7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

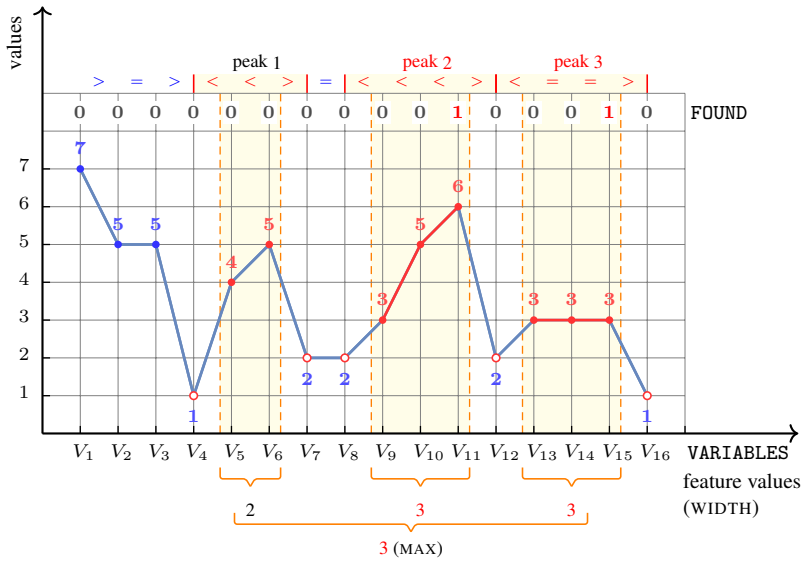


Figure 4.1148: Illustrating the POS_MAX_WIDTH_PEAK constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_WIDTH_PLAIN.

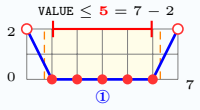
Constraint POS_MAX_WIDTH_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1149 provides an example where the POS_MAX_WIDTH_PLAIN $(2, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3], [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

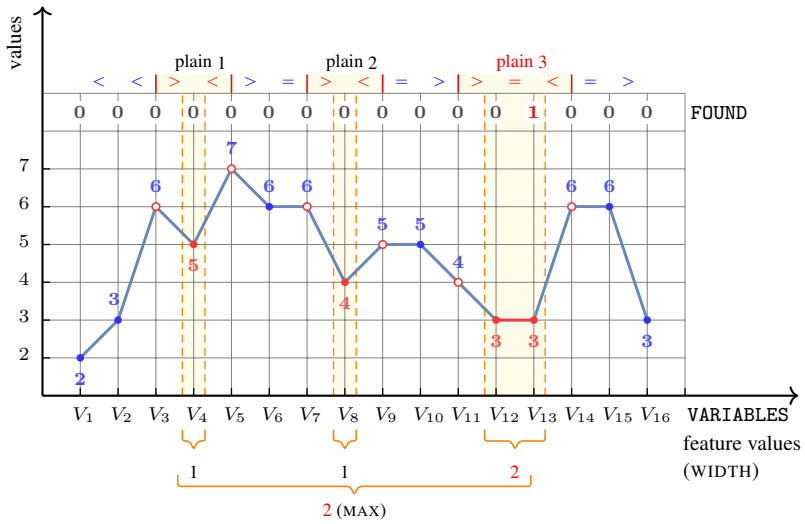


Figure 4.1149: Illustrating the POS_MAX_WIDTH_PLAIN constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_WIDTH_PLATEAU.

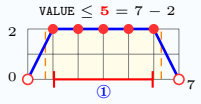
Constraint POS_MAX_WIDTH_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MAX_WIDTH_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $\left(\begin{array}{l} 4, \langle 1,3,3,5,5,5,5,2,4,4,4,3,3,1,5,5\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$

Figure 4.1150 provides an example where the POS_MAX_WIDTH_PLATEAU $(4, [1,3,3,5,5,5,5,2,4,4,4,3,3,1,5,5], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

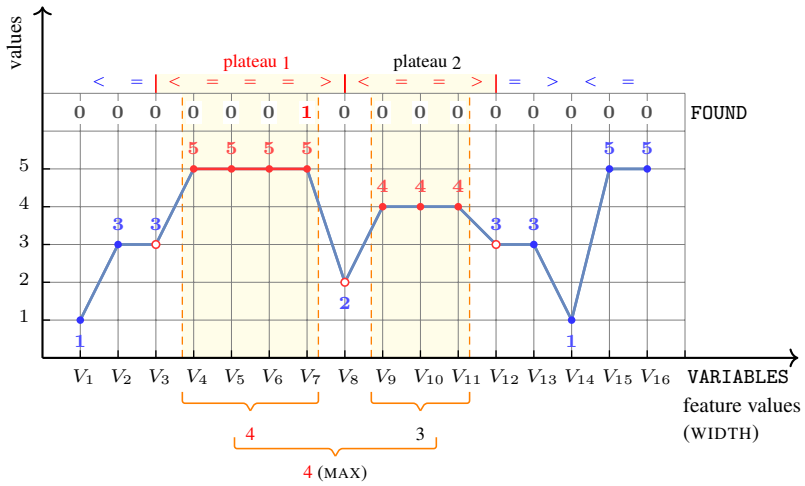


Figure 4.1150: Illustrating the POS_MAX_WIDTH_PLATEAU constraint of the **Example** slot

Arg. properties

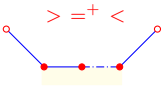
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_PROPER_PLAIN.

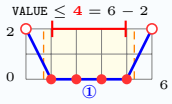
Constraint POS_MAX_WIDTH_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1151 provides an example where the POS_MAX_WIDTH_PROPER_PLAIN $(3, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

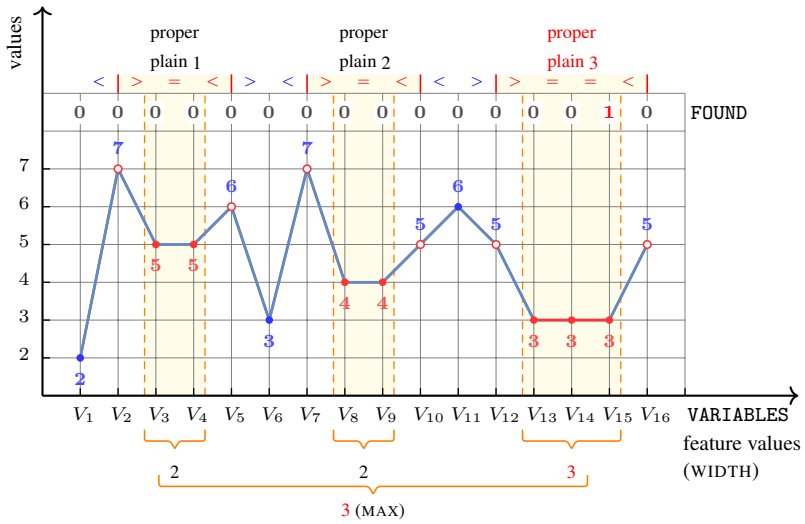


Figure 4.1151: Illustrating the POS_MAX_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Arg. properties

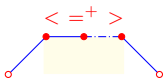
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_PROPER_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_PROPER_PLATEAU.

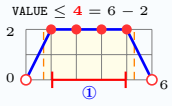
Constraint POS_MAX_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$< =^{+} >$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 3, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$$

Figure 4.1152 provides an example where the POS_MAX_WIDTH_PROPER_PLATEAU $(3, [7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

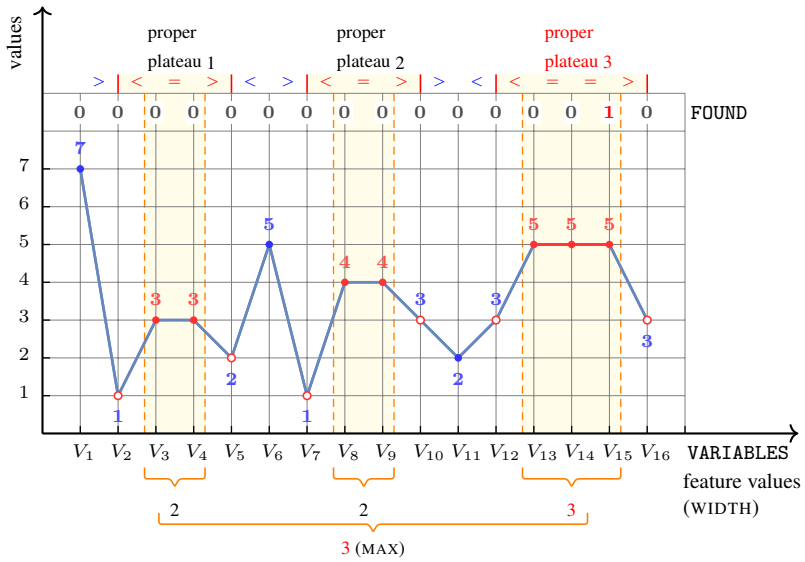


Figure 4.1152: Illustrating the POS_MAX_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Arg. properties

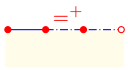
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_STEADY_SEQUENCE.

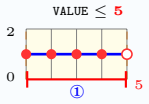
Constraint POS_MAX_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq \texttt{sv}$
$\texttt{rv} \geq 2 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \texttt{sv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 3, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1 \rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1153 provides an example where the POS_MAX_WIDTH_STEADY_SEQUENCE $(3, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

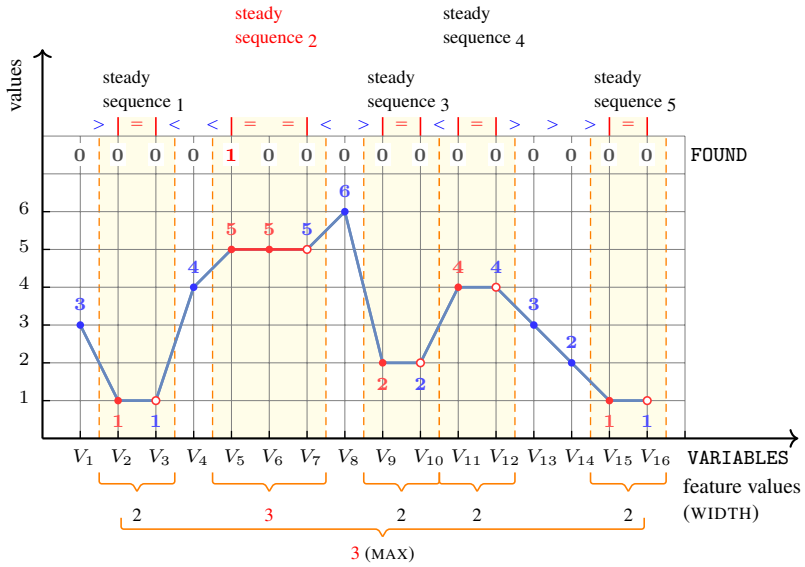


Figure 4.1153: Illustrating the POS_MAX_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE.

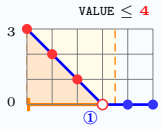
Constraint POS_MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \texttt{min}(\texttt{sv}, \texttt{rv}①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 3, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1154 provides an example where the POS_MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE $(3, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

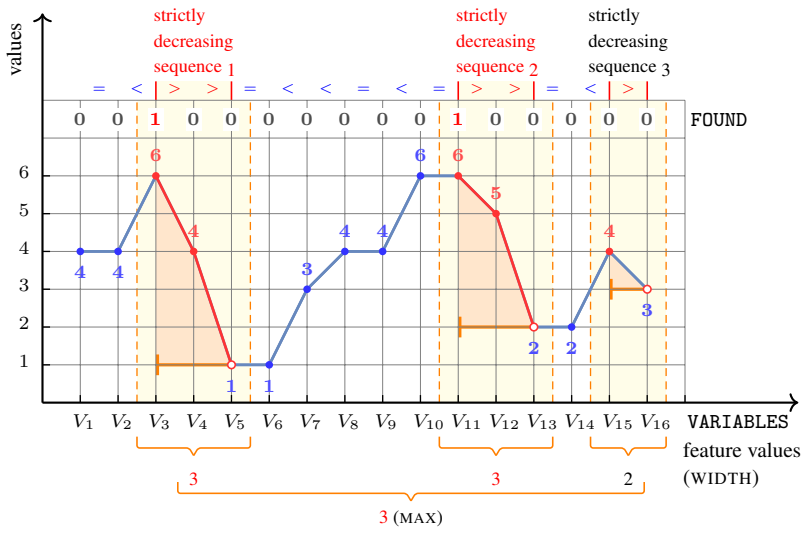


Figure 4.1154: Illustrating the POS_MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE.

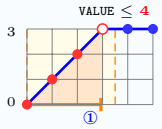
Constraint POS_MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}[1])$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 5, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1155 provides an example where the POS_MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE $(5, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

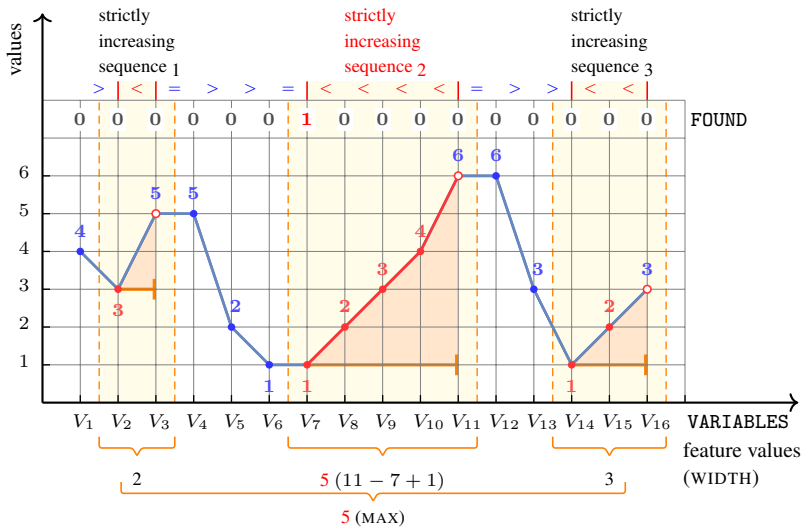


Figure 4.1155: Illustrating the POS_MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties

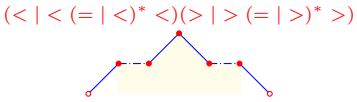
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_SUMMIT

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_WIDTH_SUMMIT.

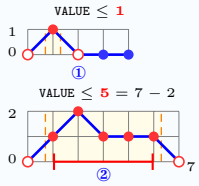
Constraint POS_MAX_WIDTH_SUMMIT(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose The constraint MAX_WIDTH_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $\left(3, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle, \langle 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle \right)$

Figure 4.1156 provides an example where the POS_MAX_WIDTH_SUMMIT $(3, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1], [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

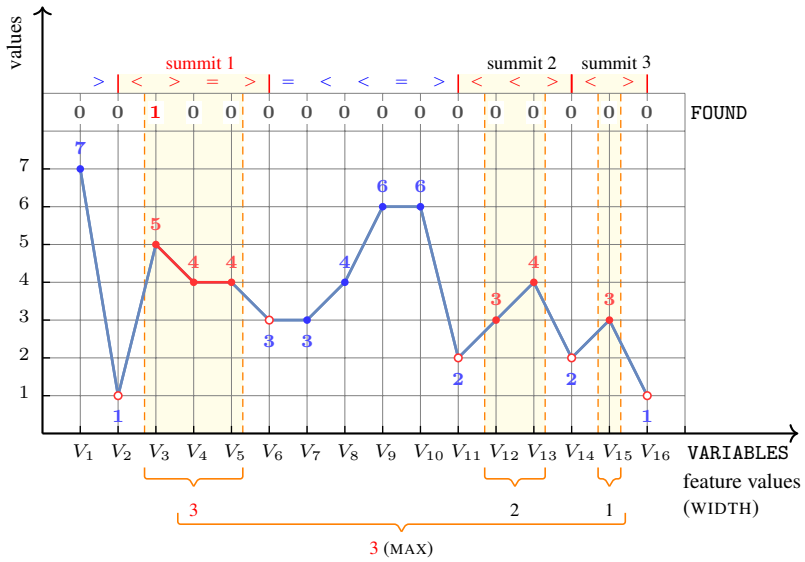


Figure 4.1156: Illustrating the POS_MAX_WIDTH_SUMMIT constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

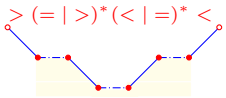
Automaton Similar to the automaton of the MAX_WIDTH_SUMMIT constraint but use the decoration table 3.35.

POS_MAX_WIDTH_SUMMIT 2367

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MAX_WIDTH_VALLEY.

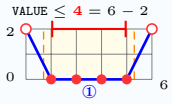
Constraint POS_MAX_WIDTH_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MAX_WIDTH_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 4, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1157 provides an example where the POS_MAX_WIDTH_VALLEY $(4, [1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

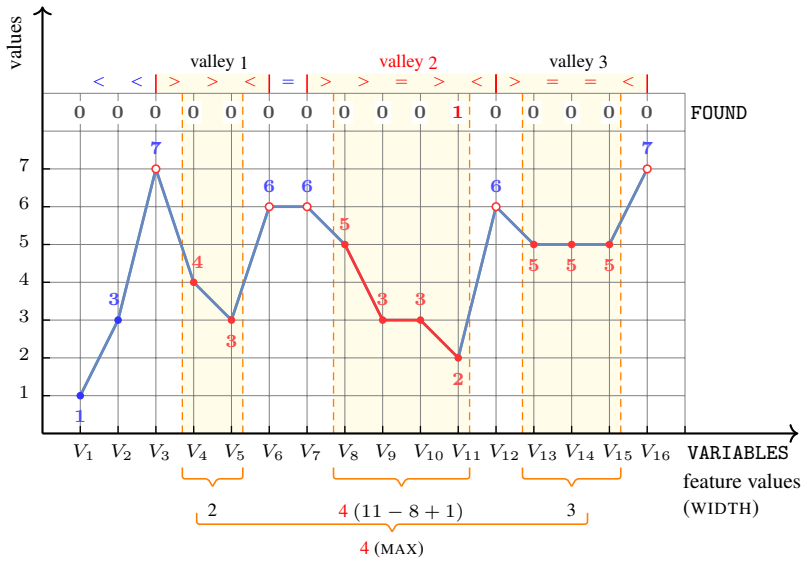


Figure 4.1157: Illustrating the POS_MAX_WIDTH_VALLEY constraint of the **Example** slot

Arg. properties

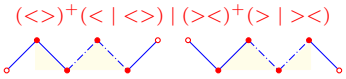
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MAX_WIDTH_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MAX_WIDTH_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MAX_WIDTH_ZIGZAG.

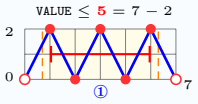
Constraint POS_MAX_WIDTH_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MAX_WIDTH_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 6, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1158 provides an example where the POS_MAX_WIDTH_ZIGZAG $(6, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

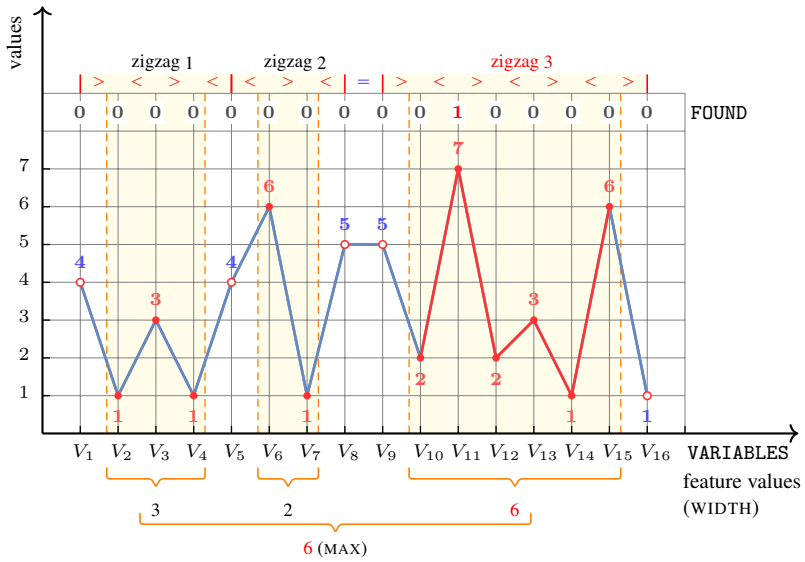


Figure 4.1158: Illustrating the POS_MAX_WIDTH_ZIGZAG constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

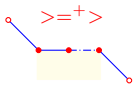
Automaton Similar to the automaton of the MAX_WIDTH_ZIGZAG constraint but use the decoration table 3.35.

POS_MAX_WIDTH_ZIGZAG 2375

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_HEIGHT_DECREASING_TERRACE.

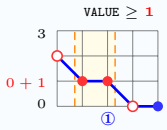
Constraint POS_MIN_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 2, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1159 provides an example where the POS_MIN_HEIGHT_DECREASING_TERRACE $(2, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

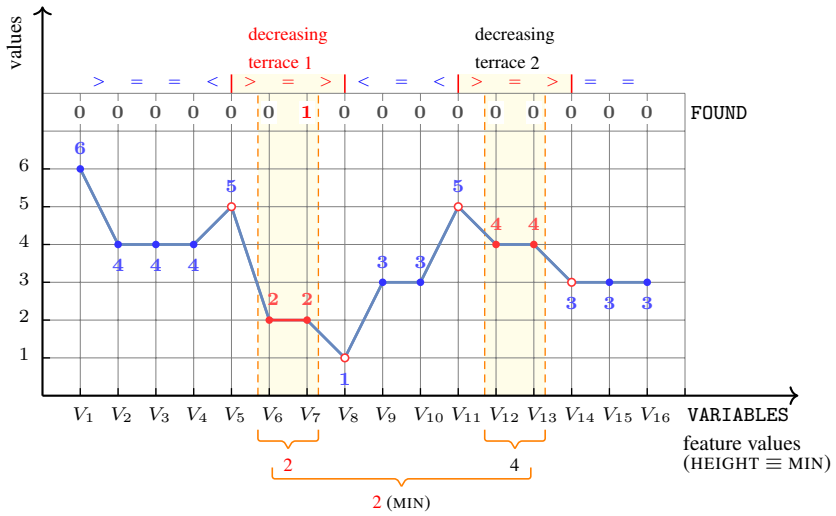


Figure 4.1159: Illustrating the POS_MIN_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

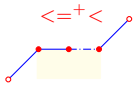
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_HEIGHT_INCREASING_TERRACE.

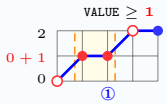
Constraint POS_MIN_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1160 provides an example where the POS_MIN_HEIGHT_INCREASING_TERRACE $(3, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

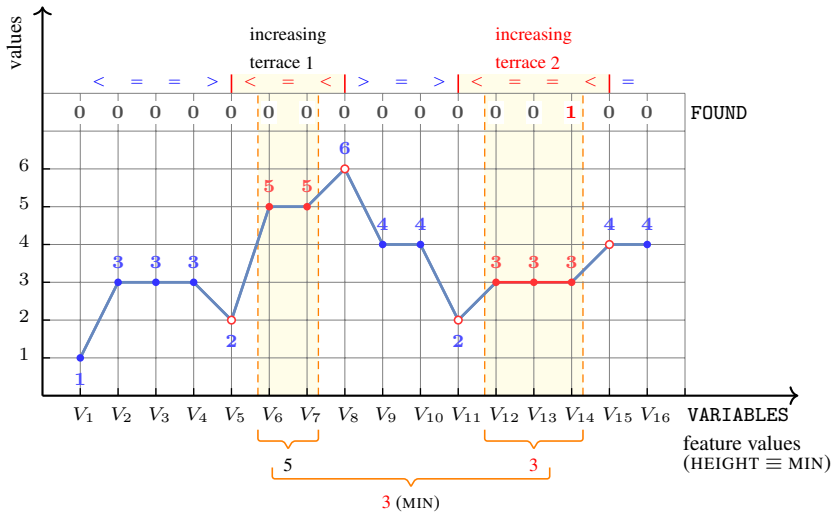


Figure 4.1160: Illustrating the POS_MIN_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_HEIGHT_PLAIN.

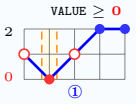
Constraint POS_MIN_HEIGHT_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1161 provides an example where the POS_MIN_HEIGHT_PLAIN $(3, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3], [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0])$ constraint holds.

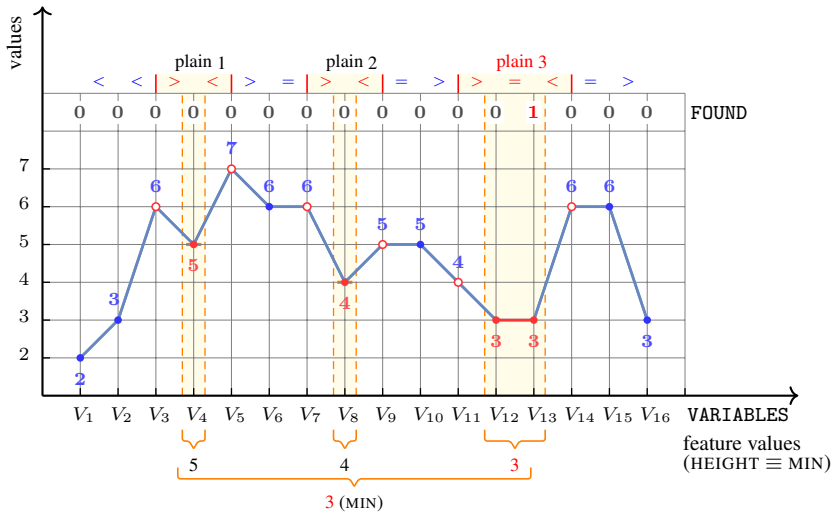


Figure 4.1161: Illustrating the POS_MIN_HEIGHT_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_HEIGHT_PLATEAU.

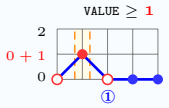
Constraint POS_MIN_HEIGHT_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1162 provides an example where the POS_MIN_HEIGHT_PLATEAU $(3, [7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5], [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

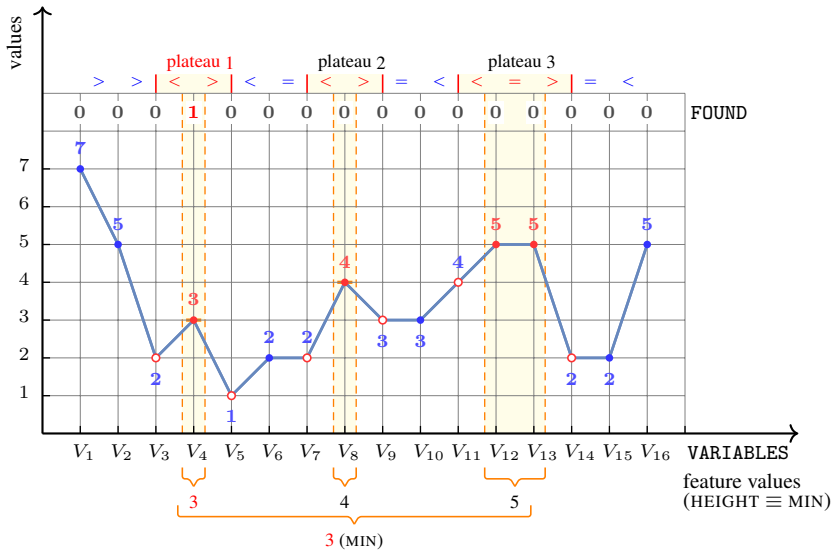


Figure 4.1162: Illustrating the POS_MIN_HEIGHT_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

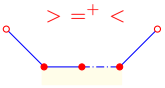
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_HEIGHT_PROPER_PLAIN.

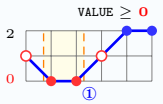
Constraint POS_MIN_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^+<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1163 provides an example where the POS_MIN_HEIGHT_PROPER_PLAIN $(3, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

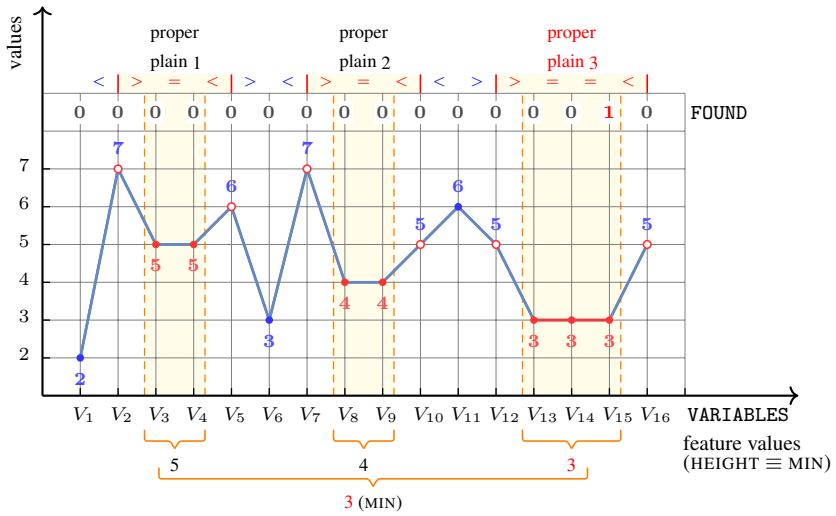


Figure 4.1163: Illustrating the POS_MIN_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

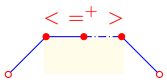
Automaton Similar to the automaton of the MIN_HEIGHT_PROPER_PLAIN constraint but use the decoration table 3.35.

POS_MIN_HEIGHT_PROPER_PLAIN 2395

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_HEIGHT_PROPER_PLATEAU.

Constraint POS_MIN_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

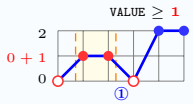
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

sv ≤ 3 ∨ rv ≤ 1 ⇒ VALUE = +∞
VALUE ≥ minv + 1①
VALUE = +∞ ∨ VALUE ≤ maxv
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|

where

minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MIN_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1164 provides an example where the POS_MIN_HEIGHT_PROPER_PLATEAU $(3, [7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3], [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

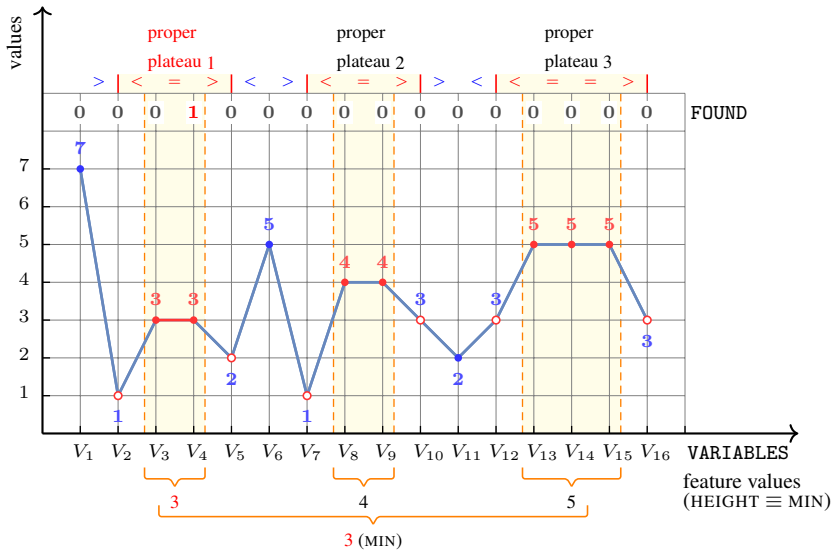


Figure 4.1164: Illustrating the POS_MIN_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_HEIGHT_STEADY.

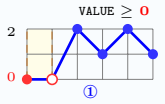
Constraint POS_MIN_HEIGHT_STEADY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose The constraint MIN_HEIGHT_STEADY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example $\left(\begin{array}{l} 1, \langle 1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6\rangle, \\ \langle 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$

Figure 4.1165 provides an example where the POS_MIN_HEIGHT_STEADY $(1, [1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6], [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

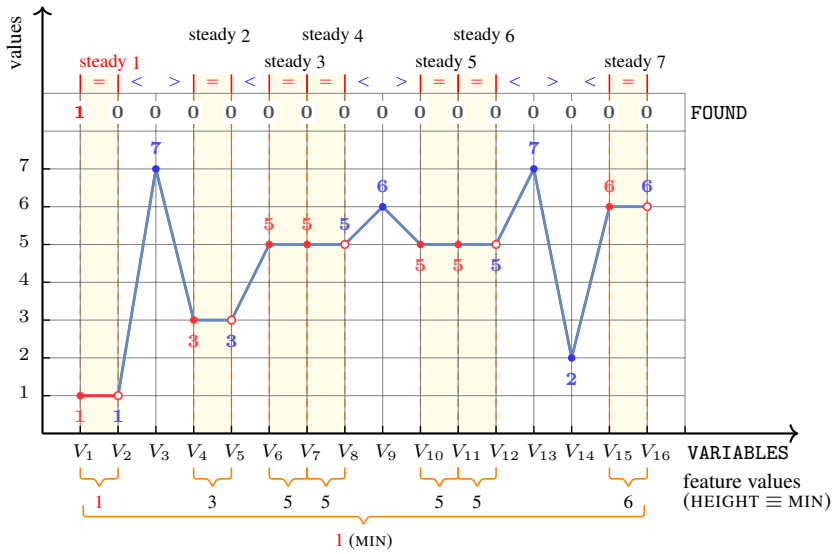


Figure 4.1165: Illustrating the POS_MIN_HEIGHT_STEADY constraint of the **Example** slot

Arg. properties

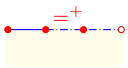
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_STEADY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_HEIGHT_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_HEIGHT_STEADY_SEQUENCE.

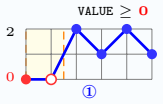
Constraint POS_MIN_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

The constraint MIN_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 1, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1166 provides an example where the POS_MIN_HEIGHT_STEADY_SEQUENCE $(1, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

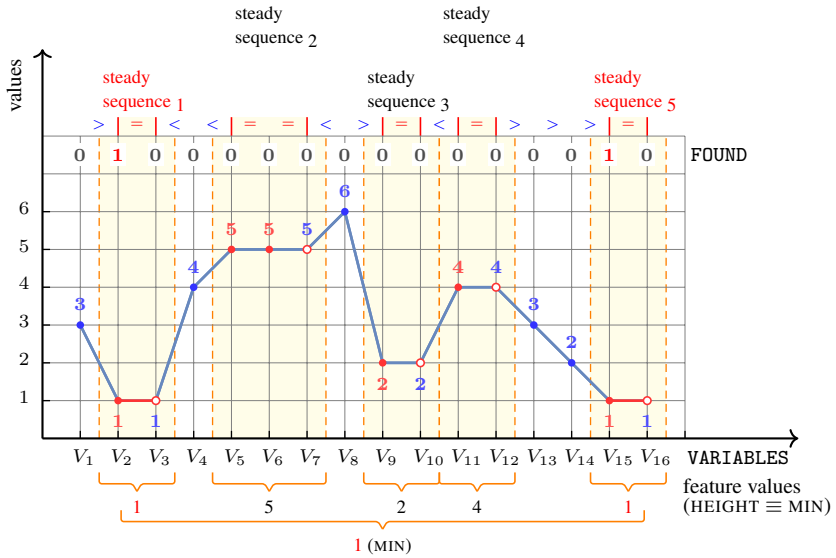


Figure 4.1166: Illustrating the POS_MIN_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Typical $|\texttt{VARIABLES}| > 1$

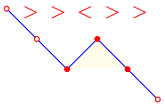
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_HEIGHT_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_BUMP_ON_DECREASING_SEQUENCE.

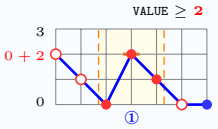
Constraint POS_MIN_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 2$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1167 provides an example where the POS_MIN_MAX_BUMP_ON_DECREASING_SEQUENCE $(5, [7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

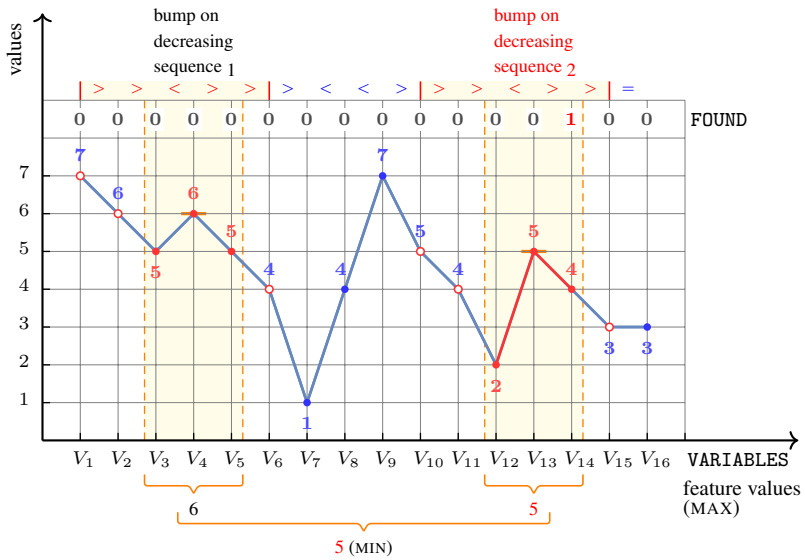


Figure 4.1167: Illustrating the POS_MIN_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_DECREASING.

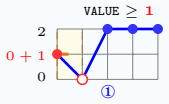
Constraint POS_MIN_MAX_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MAX_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 3, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1168 provides an example where the POS_MIN_MAX_DECREASING $(3, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0])$ constraint holds.

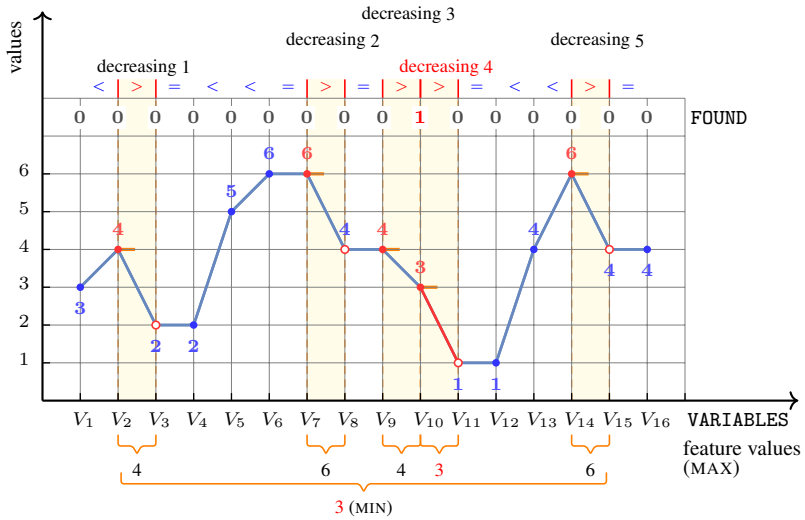


Figure 4.1168: Illustrating the POS_MIN_MAX_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

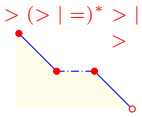
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_DECREASING_SEQUENCE.

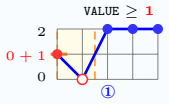
Constraint POS_MIN_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 1$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 4, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1169 provides an example where the POS_MIN_MAX_DECREASING_SEQUENCE $(4, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

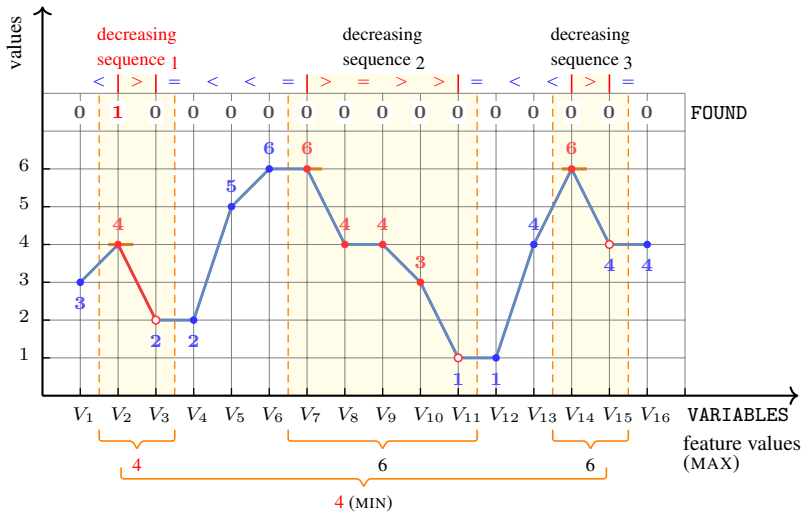


Figure 4.1169: Illustrating the POS_MIN_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

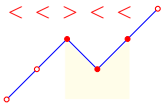
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_MAX_DIP_ON_INCREASING_SEQUENCE.

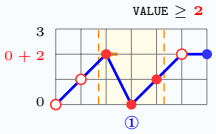
Constraint POS_MIN_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 2$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 5, \langle 1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4\rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1170 provides an example where the POS_MIN_MAX_DIP_ON_INCREASING_SEQUENCE $(5, [1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

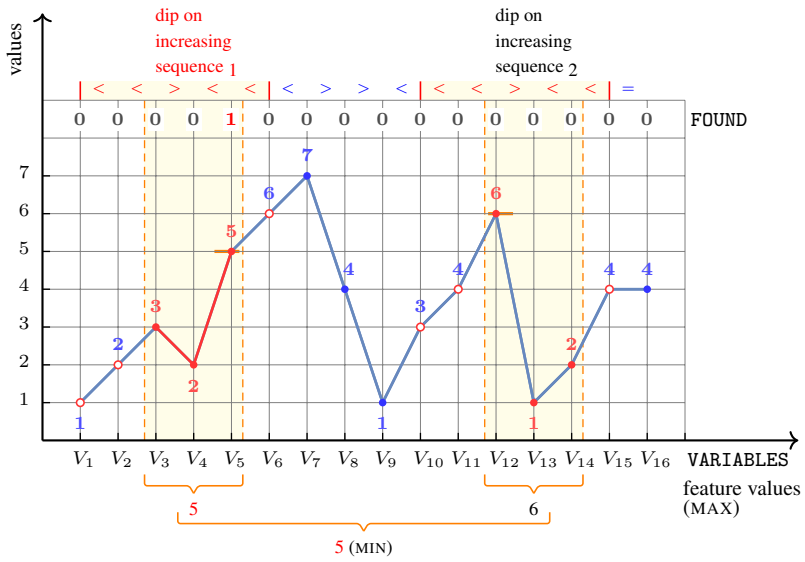


Figure 4.1170: Illustrating the POS_MIN_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_INCREASING.

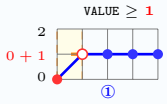
Constraint POS_MIN_MAX_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MAX_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1171 provides an example where the POS_MIN_MAX_INCREASING $(3, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

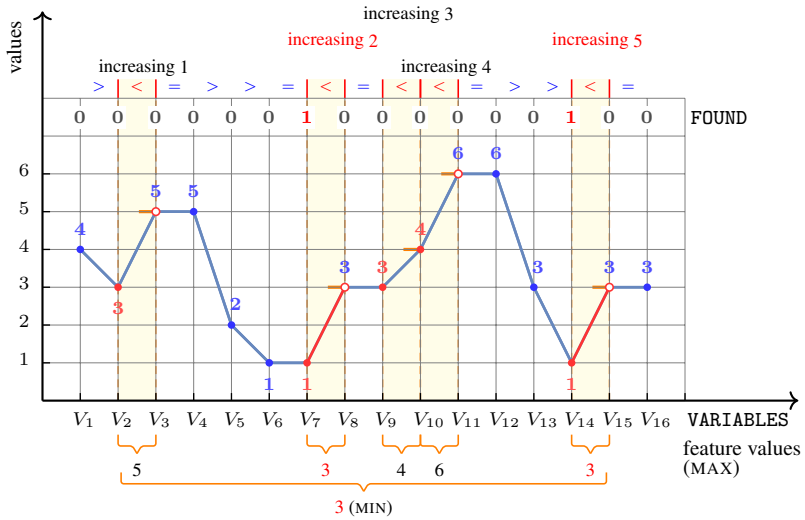


Figure 4.1171: Illustrating the POS_MIN_MAX_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

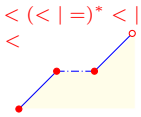
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_MAX_INCREASING_SEQUENCE.

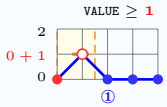
Constraint POS_MIN_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 1$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1172 provides an example where the POS_MIN_MAX_INCREASING_SEQUENCE $(3, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

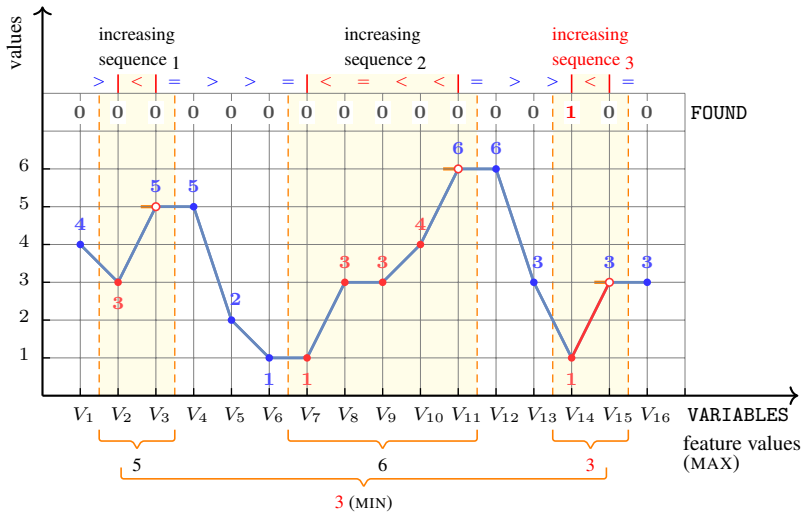


Figure 4.1172: Illustrating the POS_MIN_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

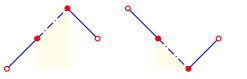
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_INFLEXION

< (< | =)* > | > (> | =)* <



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_INFLEXION.

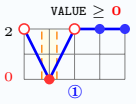
Constraint POS_MIN_MAX_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv}$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1173 provides an example where the POS_MIN_MAX_INFLEXION $(1, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

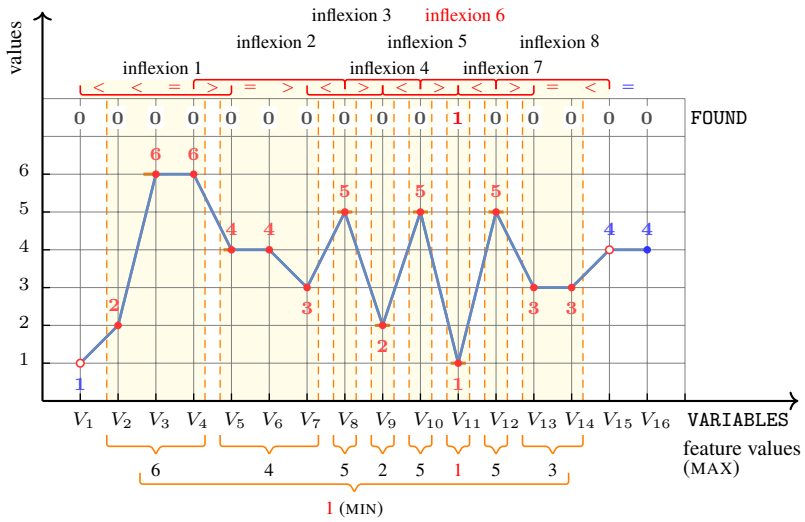


Figure 4.1173: Illustrating the POS_MIN_MAX_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

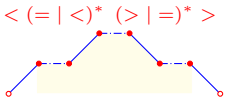
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_PEAK

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_MAX_PEAK.

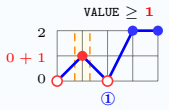
Constraint POS_MIN_MAX_PEAK(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1174 provides an example where the POS_MIN_MAX_PEAK $(3, [7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

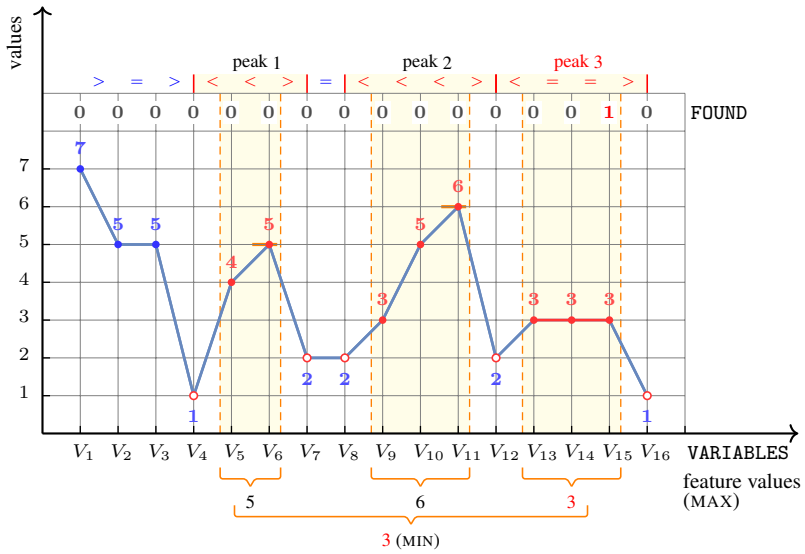


Figure 4.1174: Illustrating the POS_MIN_MAX_PEAK constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_MAX_STRICTLY_DECREASING_SEQUENCE.

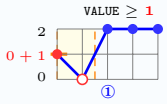
Constraint POS_MIN_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv} + 1$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 4, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1175 provides an example where the POS_MIN_MAX_STRICTLY_DECREASING_SEQUENCE $(4, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

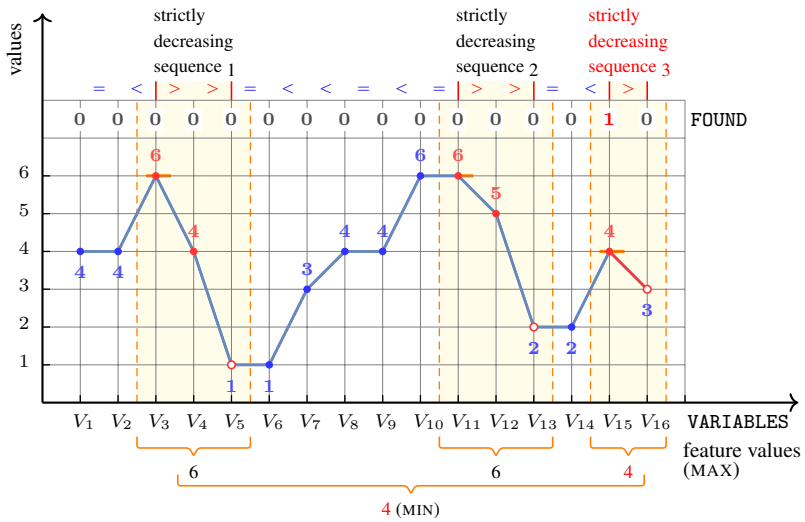


Figure 4.1175: Illustrating the POS_MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MAX_STRICTLY_INCREASING_SEQUENCE.

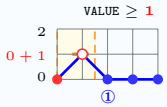
Constraint POS_MIN_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 3, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1176 provides an example where the POS_MIN_MAX_STRICTLY_INCREASING_SEQUENCE $(3, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

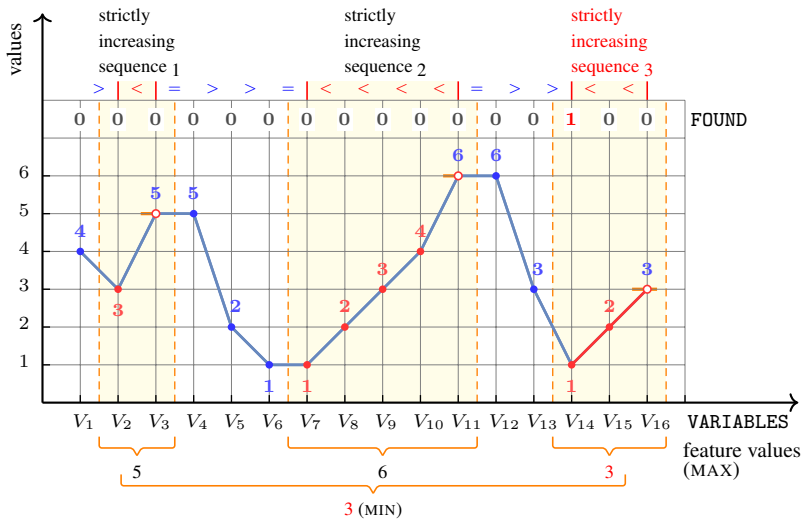


Figure 4.1176: Illustrating the POS_MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

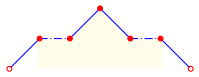
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_MAX_SUMMIT.

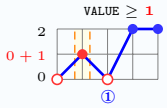
Constraint POS_MIN_MAX_SUMMIT(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MAX_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1177 provides an example where the POS_MIN_MAX_SUMMIT $(3, [7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

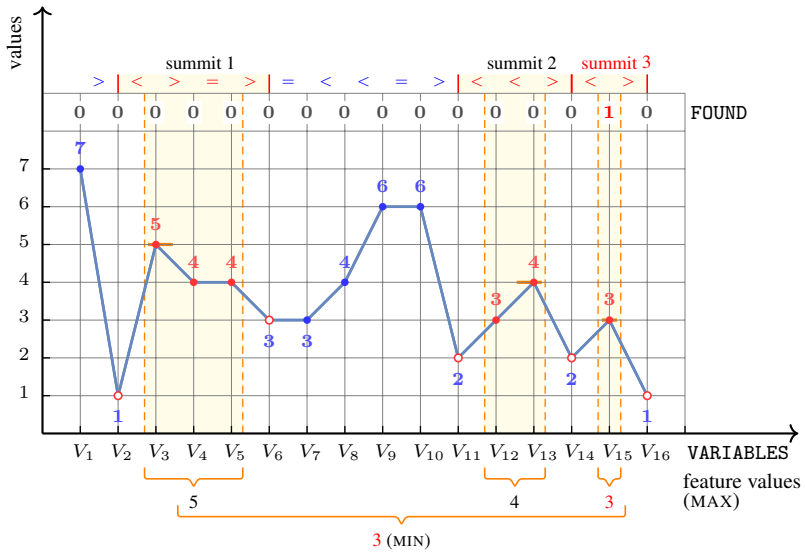


Figure 4.1177: Illustrating the POS_MIN_MAX_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

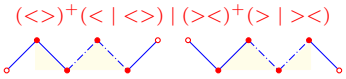
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_SUMMIT constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MAX_ZIGZAG

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_MAX_ZIGZAG.

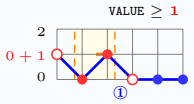
Constraint POS_MIN_MAX_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MIN_MAX_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $\left(\begin{array}{l} 3, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$

Figure 4.1178 provides an example where the POS_MIN_MAX_ZIGZAG $(3, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

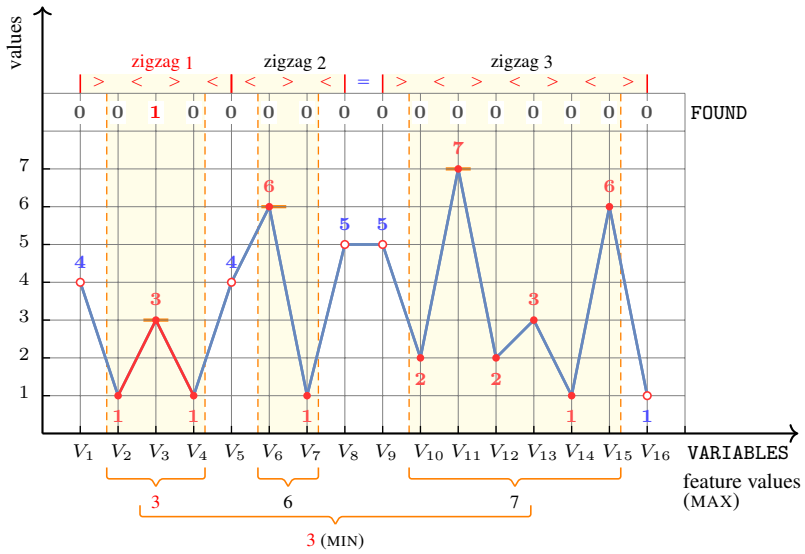


Figure 4.1178: Illustrating the POS_MIN_MAX_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

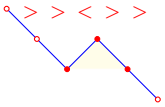
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MAX_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_BUMP_ON_DECREASING_SEQUENCE.

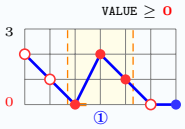
Constraint POS_MIN_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{minv}$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv} - 2$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 2, \langle 7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1179 provides an example where the POS_MIN_MIN_BUMP_ON_DECREASING_SEQUENCE $(2, [7,6,5,6,5,4,1,4,7,5,4,2,5,4,3,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

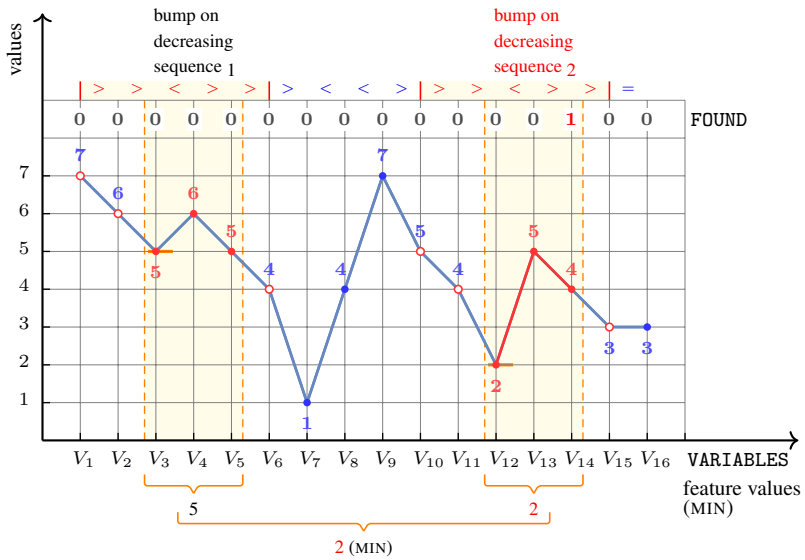


Figure 4.1179: Illustrating the POS_MIN_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_DECREASING.

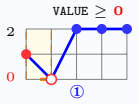
Constraint POS_MIN_MIN_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 1, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1180 provides an example where the POS_MIN_MIN_DECREASING $(1, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0])$ constraint holds.

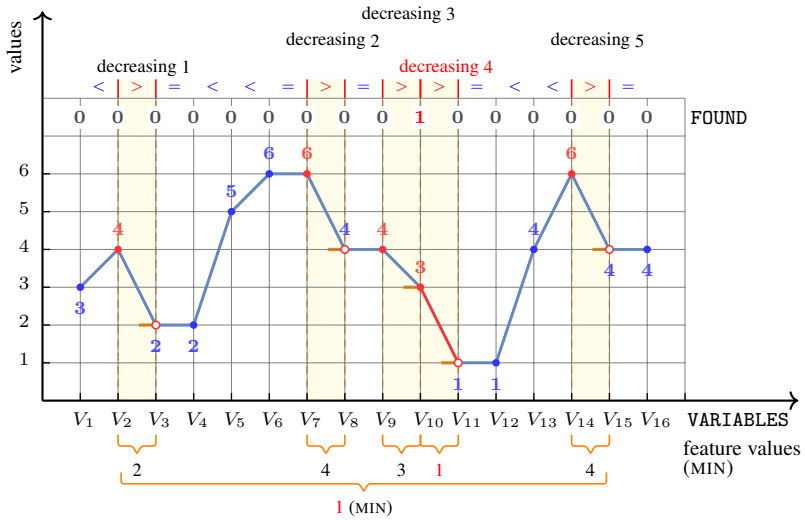


Figure 4.1180: Illustrating the POS_MIN_MIN_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

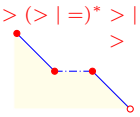
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_DECREASING_SEQUENCE.

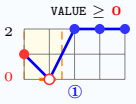
Constraint POS_MIN_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 1, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1181 provides an example where the POS_MIN_MIN_DECREASING_SEQUENCE $(1, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

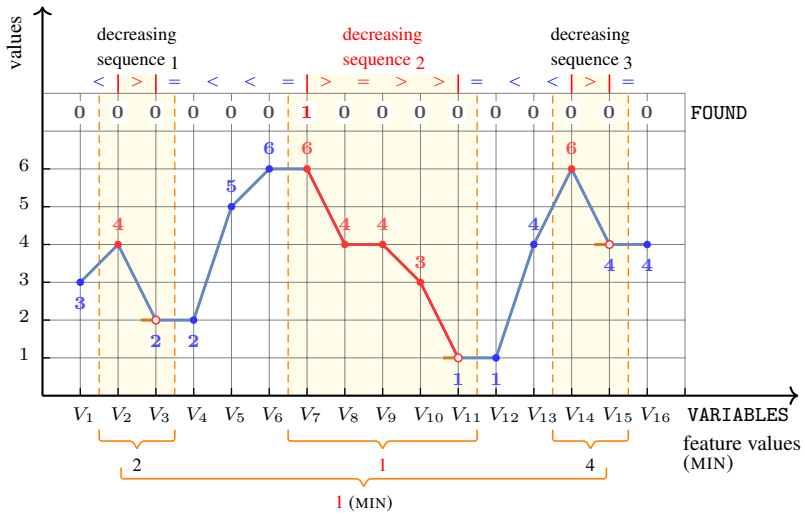


Figure 4.1181: Illustrating the POS_MIN_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

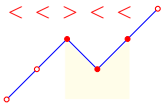
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_DIP_ON_INCREASING_SEQUENCE.

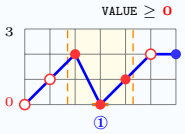
Constraint POS_MIN_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 2$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1182 provides an example where the POS_MIN_MIN_DIP_ON_INCREASING_SEQUENCE $(1, [1,2,3,2,5,6,7,4,1,3,4,6,1,2,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

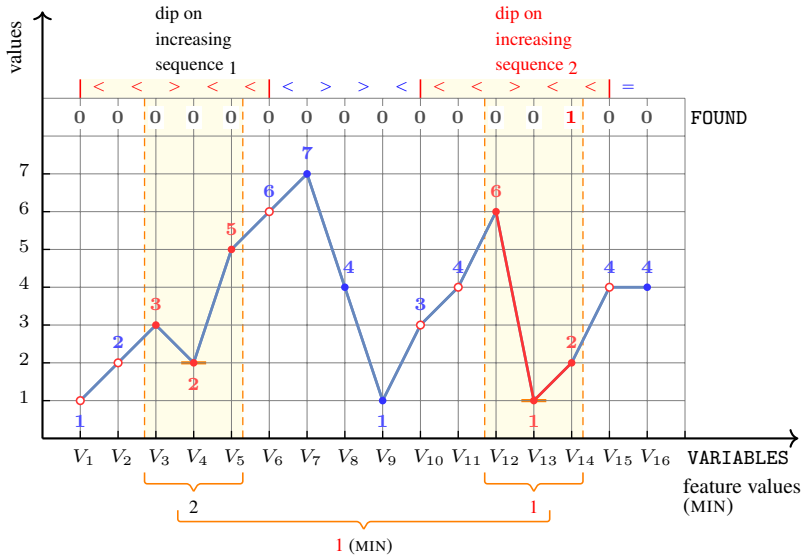


Figure 4.1182: Illustrating the POS_MIN_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

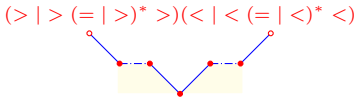
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MIN_MIN_GORGE.

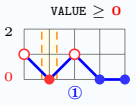
Constraint POS_MIN_MIN_GORGE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MIN_MIN_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $\left(\begin{array}{l} 3, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$

Figure 4.1183 provides an example where the POS_MIN_MIN_GORGE $(3, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

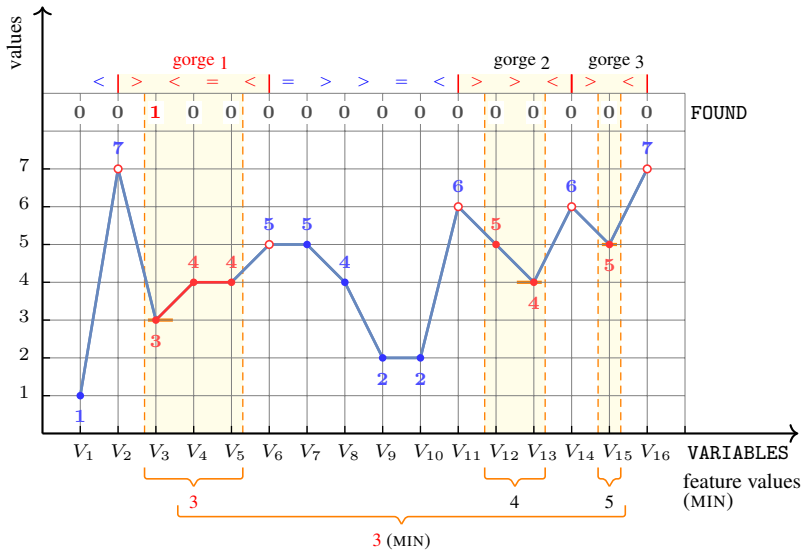


Figure 4.1183: Illustrating the POS_MIN_MIN_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_GORGE constraint but use the decoration table 3.35.

POS_MIN_MIN_GORGE 2475

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_INCREASING.

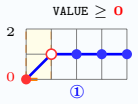
Constraint POS_MIN_MIN_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '$<$'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 1, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1184 provides an example where the POS_MIN_MIN_INCREASING $(1, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

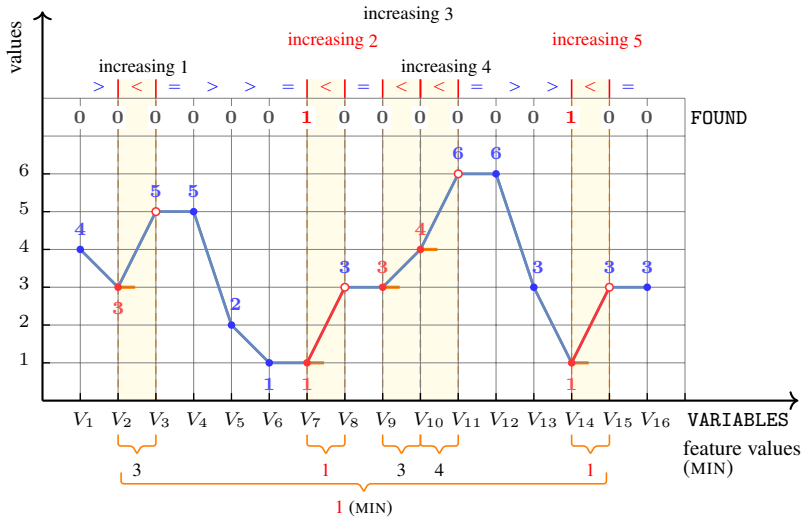


Figure 4.1184: Illustrating the POS_MIN_MIN_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

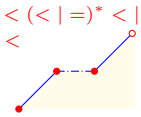
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_MIN_INCREASING_SEQUENCE.

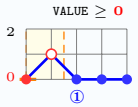
Constraint POS_MIN_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 1, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1185 provides an example where the POS_MIN_MIN_INCREASING_SEQUENCE $(1, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

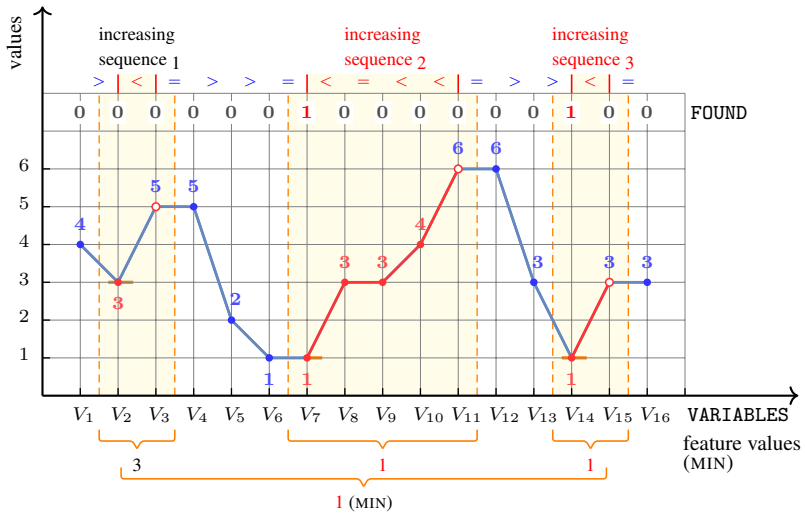


Figure 4.1185: Illustrating the POS_MIN_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

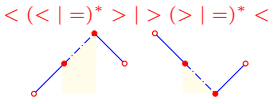
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_MIN_INFLEXION.

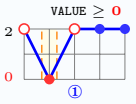
Constraint POS_MIN_MIN_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}①$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv}$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1186 provides an example where the POS_MIN_MIN_INFLEXION $(1, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

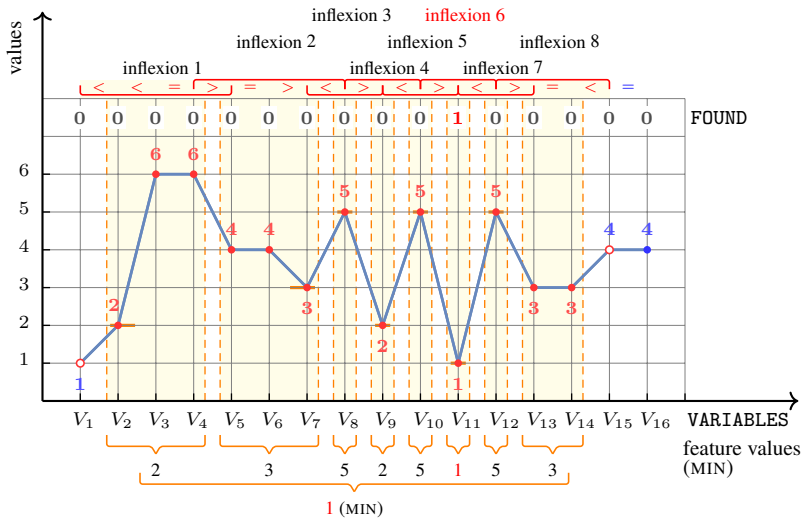


Figure 4.1186: Illustrating the POS_MIN_MIN_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_STRICTLY_DECREASING_SEQUENCE.

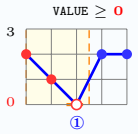
Constraint POS_MIN_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 1, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1187 provides an example where the POS_MIN_MIN_STRICTLY_DECREASING_SEQUENCE $(1, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

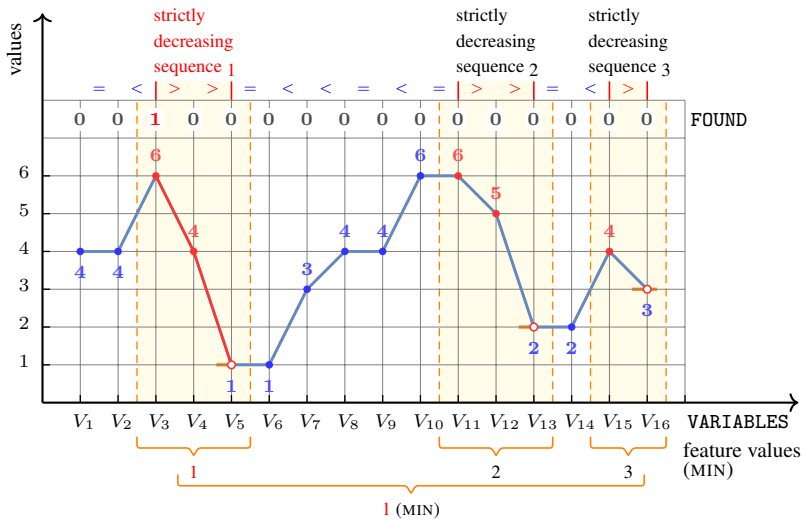


Figure 4.1187: Illustrating the POS_MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_MIN_STRICTLY_INCREASING_SEQUENCE.

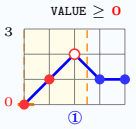
Constraint POS_MIN_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 1, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1188 provides an example where the POS_MIN_MIN_STRICTLY_INCREASING_SEQUENCE $(1, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

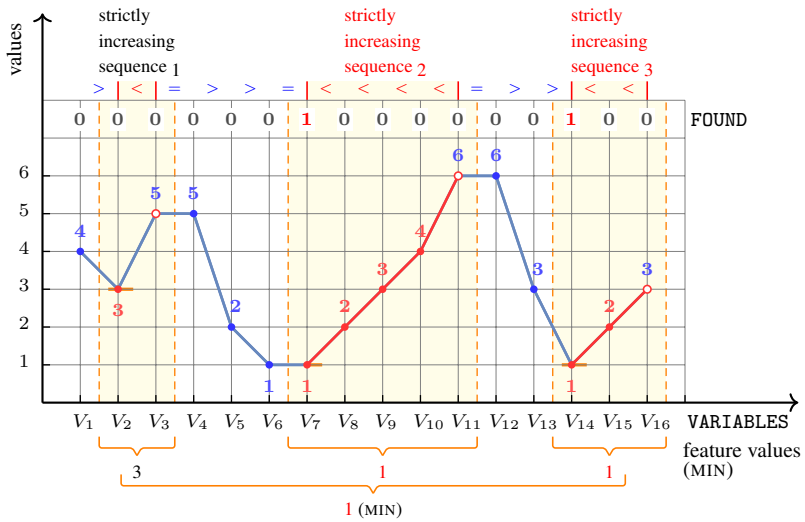


Figure 4.1188: Illustrating the POS_MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

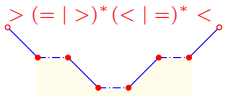
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_MIN_VALLEY.

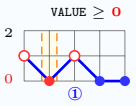
Constraint POS_MIN_MIN_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MIN_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 2, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1189 provides an example where the POS_MIN_MIN_VALLEY $(2, [1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

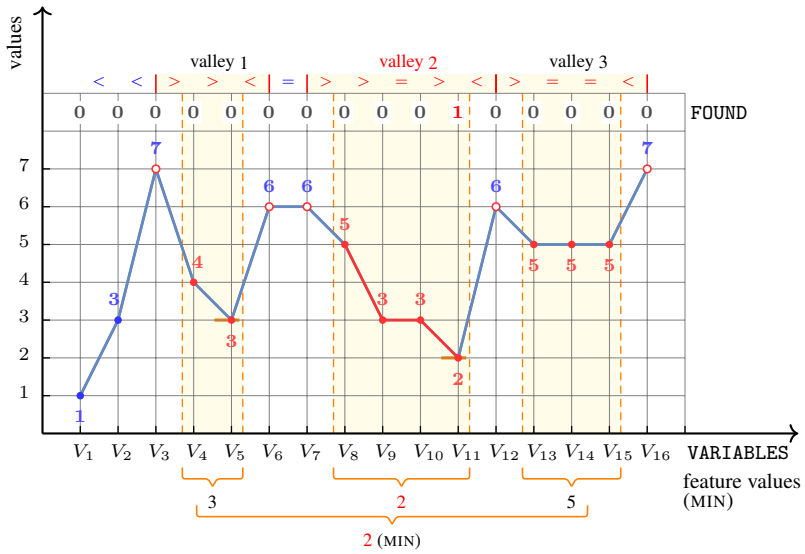


Figure 4.1189: Illustrating the POS_MIN_MIN_VALLEY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

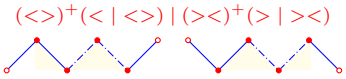
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_MIN_ZIGZAG

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_MIN_ZIGZAG.

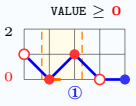
Constraint POS_MIN_MIN_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_MIN_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \\ \langle 0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1190 provides an example where the POS_MIN_MIN_ZIGZAG $(1, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0])$ constraint holds.

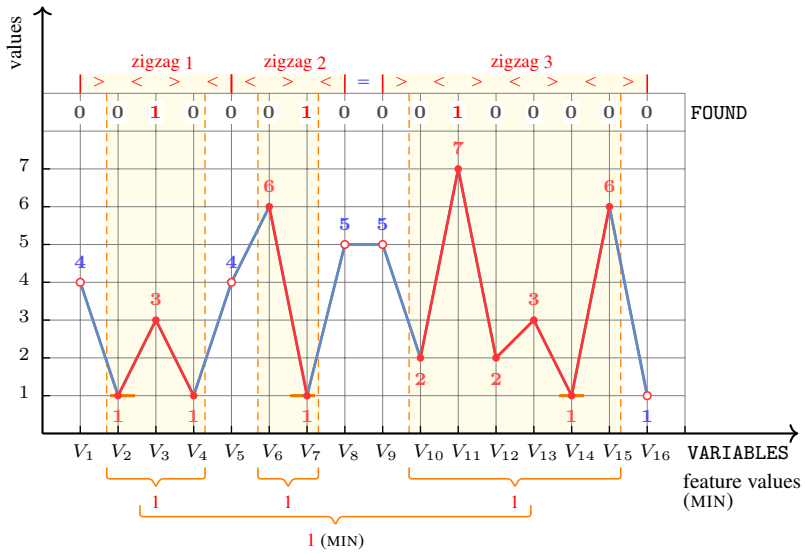


Figure 4.1190: Illustrating the POS_MIN_MIN_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

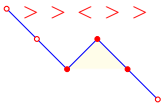
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_MIN_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_BUMP_ON_DECREASING_SEQUENCE.

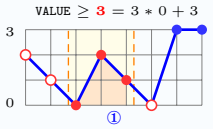
Constraint POS_MIN_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 5 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq 3 * \mathtt{minv} + 3$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq 3 * \mathtt{maxv} - 3$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[3, \mathtt{sv} - 1], \langle \mathtt{maxv} - 2, \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - 3 - \mathtt{max}(\mathtt{maxv} - 3, 0)$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[3, \mathtt{sv} - 1], \langle \mathtt{minv}, \mathtt{minv} + 1, \mathtt{minv} + 2 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \mathtt{min}(\mathtt{minv} + 3, 0) - 3 - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern BUMP_ON_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example

$\left(\begin{array}{l} 11, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3 \rangle, \\ \langle 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 \rangle \end{array} \right)$

Figure 4.1191 provides an example where the POS_MIN_SURF_BUMP_ON_DECREASING_SEQUENCE $(11, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0])$ constraint holds.

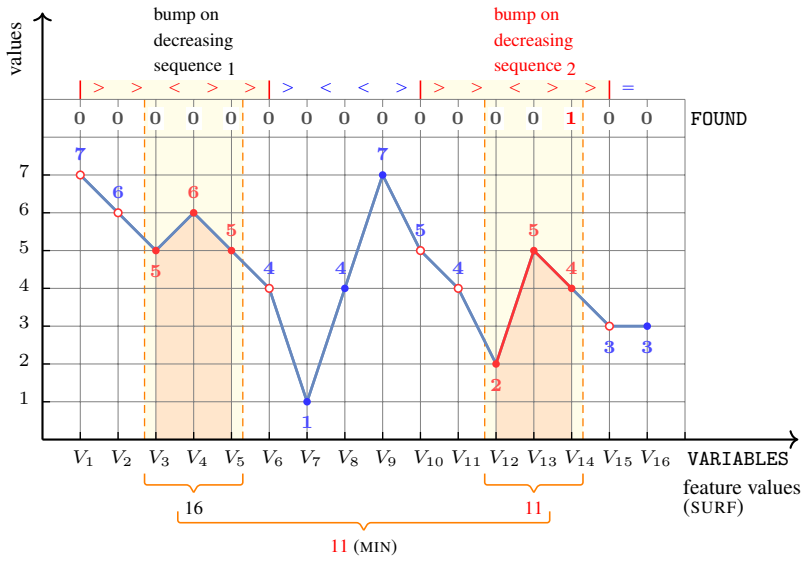


Figure 4.1191: Illustrating the POS_MIN_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_BUMP_ON_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_DECREASING.

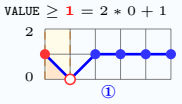
Constraint POS_MIN_SURF_DECREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq 2 * \mathtt{minv} + 1$①
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq 2 * \mathtt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} =$minval(VARIABLES.var)
$\mathtt{maxv} =$maxval(VARIABLES.var)
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_SURF_DECREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 4, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1192 provides an example where the POS_MIN_SURF_DECREASING $(4, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0])$ constraint holds.

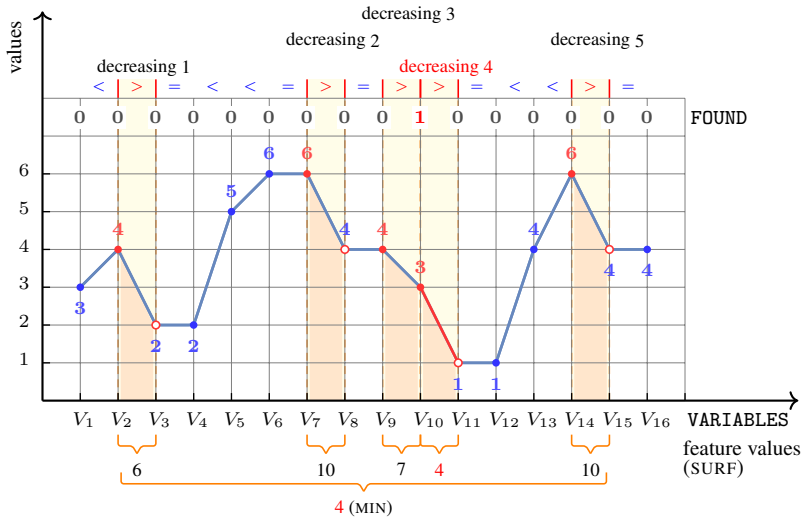


Figure 4.1192: Illustrating the POS_MIN_SURF_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

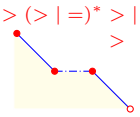
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_DECREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_DECREASING_SEQUENCE.

Constraint POS_MIN_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

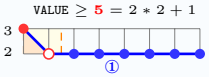
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \geq 2 * \mathtt{minv} + 1$①

$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \geq \mathtt{min}(2 * \mathtt{minv} + 1$②$, \mathtt{sv} * (\mathtt{minv} + 1)$③$)$



$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq 2 * \mathtt{maxv} - 1$

$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{max}(2 * \mathtt{maxv} - 1, \mathtt{sv} * (\mathtt{maxv} - 1))$

$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$

$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{maxv} = 1) \Rightarrow$
$\mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, 2 * \mathtt{maxv} - 3)$

$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - 1 - \mathtt{sv} * (\mathtt{maxv} - 2)$

$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{minv}, \mathtt{minv} + 1 \rangle)$

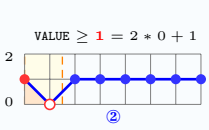
$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{minv} = -1) \Rightarrow$
$\mathtt{n2} \geq \mathtt{min}(0, 2 * \mathtt{minv} + 3) - \mathtt{VALUE}$

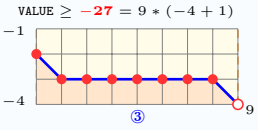
$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq \mathtt{sv} * (\mathtt{minv} + 2) - 1 - \mathtt{VALUE}$

$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$

$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$

$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$





where

$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$

$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$

$\mathtt{sv} = |\mathtt{VARIABLES}|$

$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$

Purpose

The constraint MIN_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.

The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.

An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.

Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(6, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4\rangle , \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1193 provides an example where the POS_MIN_SURF_DECREASING_SEQUENCE $(6, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

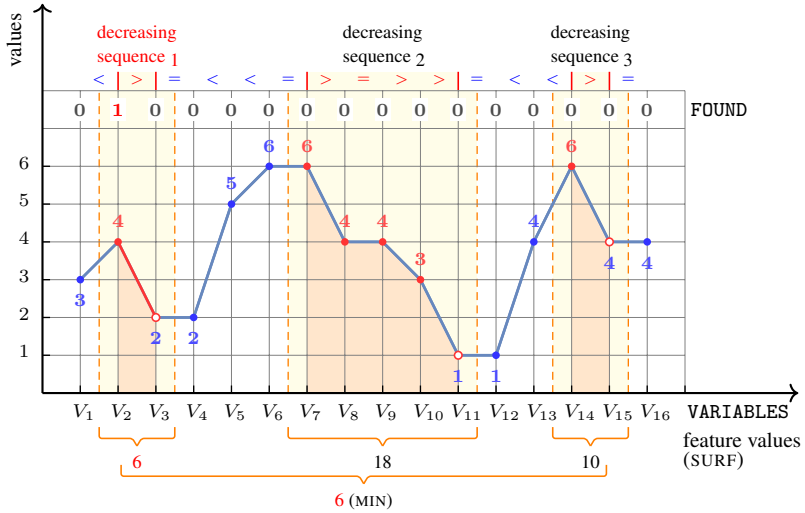


Figure 4.1193: Illustrating the POS_MIN_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

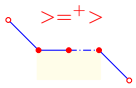
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_DECREASING_TERRACE.

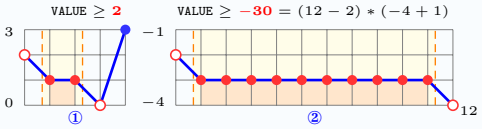
Constraint POS_MIN_SURF_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{min}(2 * (\mathtt{minv}+1)①, (\mathtt{sv}-2) * (\mathtt{minv}+1)②)$
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{max}(2 * (\mathtt{maxv}-1), (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, (\mathtt{sv}-2) * (\mathtt{maxv}-2))$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}+1 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \mathtt{min}(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 4, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3 \rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1194 provides an example where the POS_MIN_SURF_DECREASING_TERRACE $(4, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3], [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

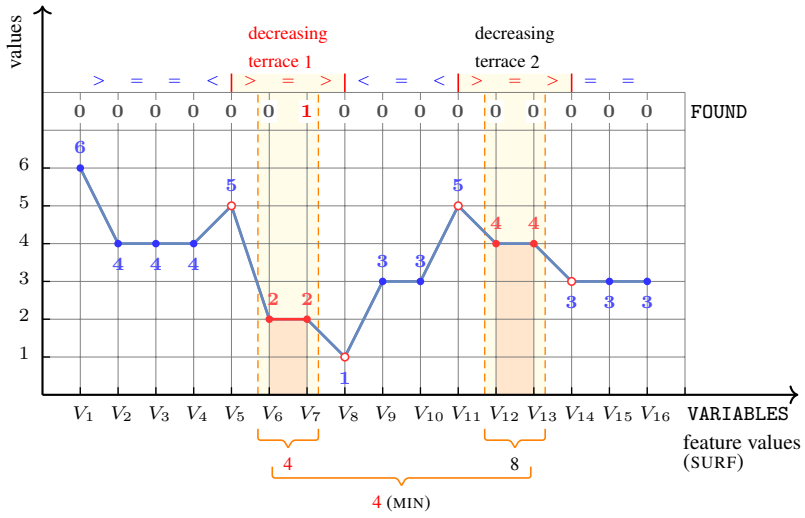


Figure 4.1194: Illustrating the POS_MIN_SURF_DECREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

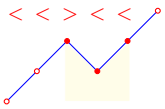
Arg. properties

- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MIN_SURF_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_DIP_ON_INCREASING_SEQUENCE.

Constraint POS_MIN_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

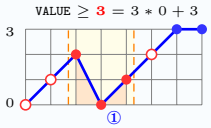
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 3 * \texttt{minv} + 3$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 3 * \texttt{maxv} - 3$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 3 - \texttt{max}(\texttt{maxv} - 3, 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(\texttt{minv} + 3, 0) - 3 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DIP_ON_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example

$$\left(\begin{array}{l} 9, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle, \\ \langle 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 \rangle \end{array} \right)$$

Figure 4.1195 provides an example where the POS_MIN_SURF_DIP_ON_INCREASING_SEQUENCE $(9, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0])$ constraint holds.

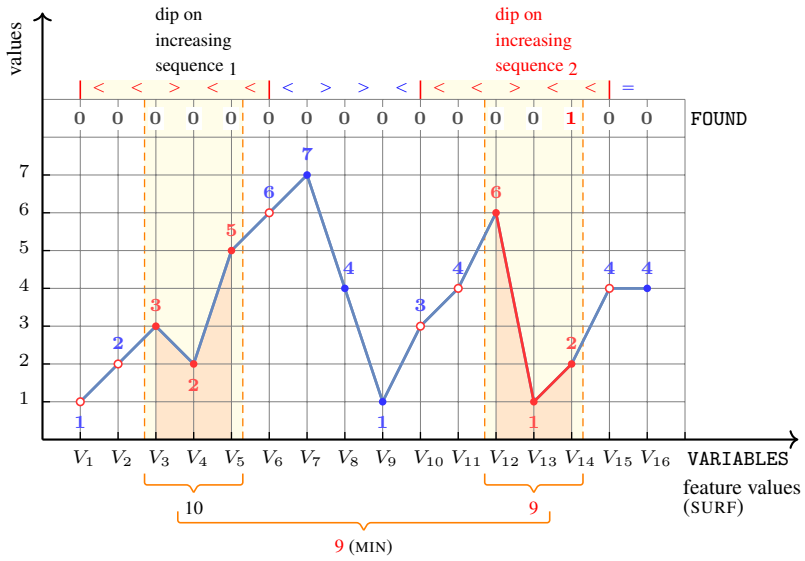


Figure 4.1195: Illustrating the POS_MIN_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

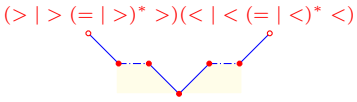
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_DIP_ON_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_GORGE.

Constraint POS_MIN_SURF_GORGE(VALUE, VARIABLES, FOUND)

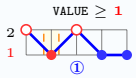
Arguments

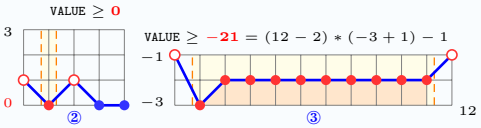
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \geq \mathtt{minv}$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \geq \min(\mathtt{minv}$②$, (\mathtt{sv}-2) * (\mathtt{minv}+1) - 1$③$)$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv} - 1$
$\mathtt{rv} \geq 3 \Rightarrow$
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \max(\mathtt{maxv}-1, (\mathtt{sv}-2) * (\mathtt{maxv}-1) - 1)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1 \rangle)$
$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{maxv} = 1) \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \max(0, \mathtt{maxv}-2)$
$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - (\mathtt{sv}-2) * (\mathtt{maxv}-2)$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}, \mathtt{minv}+1 \rangle)$
$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{minv} = -1) \Rightarrow \mathtt{n2} \geq \min(0, \mathtt{minv}+1) - \mathtt{VALUE}$
$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow$
$\mathtt{n2} \geq (\mathtt{sv}-2) * (\mathtt{minv}+2) - 1 - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$





Purpose

The constraint MIN_SURF_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(5, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7 \rangle , \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \right)$$

Figure 4.1196 provides an example where the POS_MIN_SURF_GORGE $(5, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

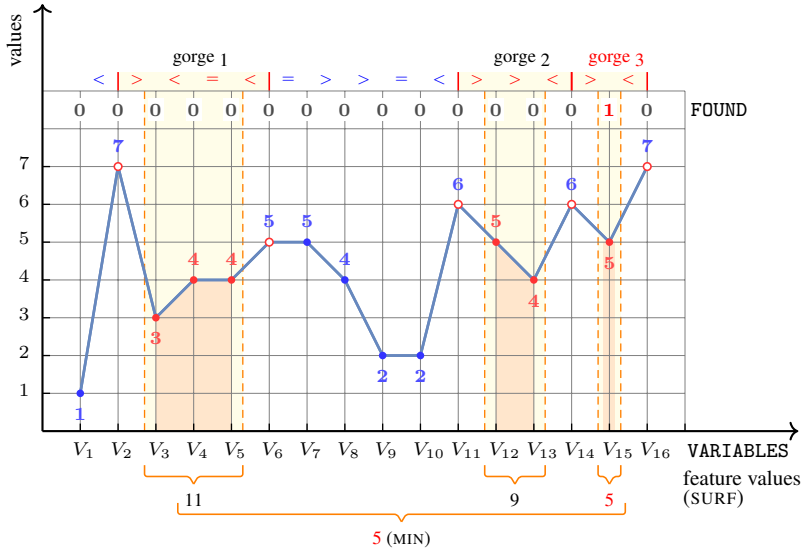


Figure 4.1196: Illustrating the POS_MIN_SURF_GORGE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_GORGE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_INCREASING.

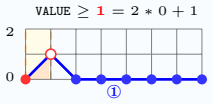
Constraint POS_MIN_SURF_INCREASING(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 2 * \texttt{minv} + 1$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_SURF_INCREASING(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$$\left(\begin{array}{l} 4, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0\rangle \end{array}\right)$$

Figure 4.1197 provides an example where the POS_MIN_SURF_INCREASING $(4, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0])$ constraint holds.

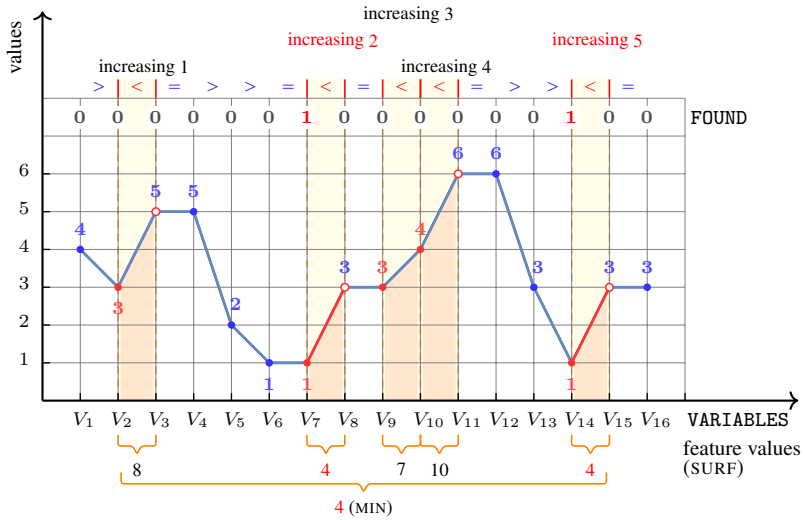


Figure 4.1197: Illustrating the POS_MIN_SURF_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

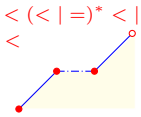
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_INCREASING constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_SURF_INCREASING_SEQUENCE.

Constraint POS_MIN_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

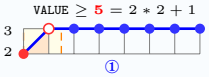
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$

$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$①

$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \geq \min(2 * \texttt{minv} + 1$②$, \texttt{sv} * (\texttt{minv} + 1)$③$)$



$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$

$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv} - 1, \texttt{sv} * (\texttt{maxv} - 1))$

$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$

$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{maxv} = 1) \Rightarrow$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, 2 * \texttt{maxv} - 3)$

$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 1 - \texttt{sv} * (\texttt{maxv} - 2)$

$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$

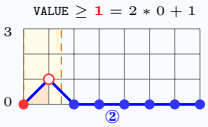
$\texttt{VALUE} < +\infty \wedge (\texttt{rv} = 2 \vee \texttt{minv} = -1) \Rightarrow$
$\texttt{n2} \geq \min(0, 2 * \texttt{minv} + 3) - \texttt{VALUE}$

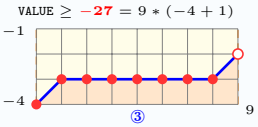
$\texttt{VALUE} < +\infty \wedge \texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - 1 - \texttt{VALUE}$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

$\texttt{required}(\texttt{FOUND}, \texttt{var})$

$|\texttt{VARIABLES}| = |\texttt{FOUND}|$





where

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

Purpose

The constraint MIN_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.

The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.

An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.

Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 4, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle , \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1198 provides an example where the POS_MIN_SURF_INCREASING_SEQUENCE $(4, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

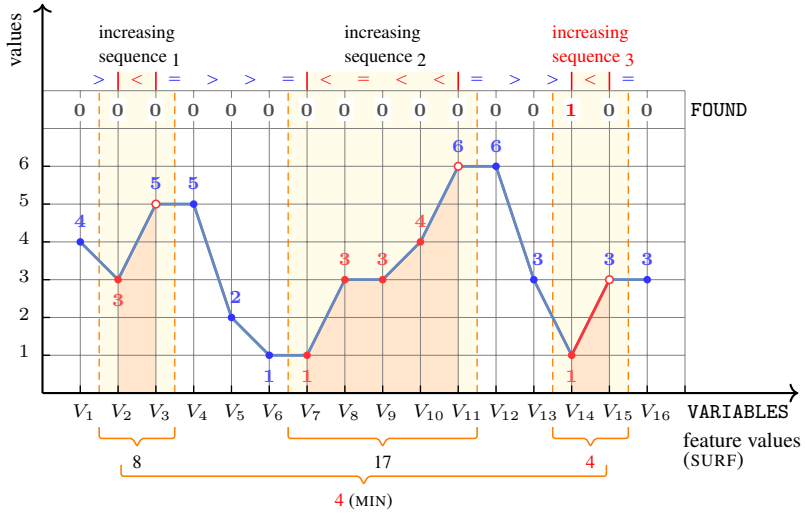


Figure 4.1198: Illustrating the POS_MIN_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

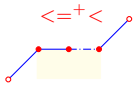
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_INCREASING_TERRACE.

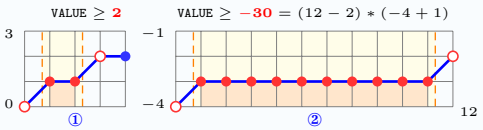
Constraint POS_MIN_SURF_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \min(2 * (\mathtt{minv}+1)①, (\mathtt{sv}-2) * (\mathtt{minv}+1)②)$
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \max(2 * (\mathtt{maxv}-1), (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \max(0, (\mathtt{sv}-2) * (\mathtt{maxv}-2))$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}+1 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \min(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 9, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1199 provides an example where the POS_MIN_SURF_INCREASING_TERRACE $(9, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

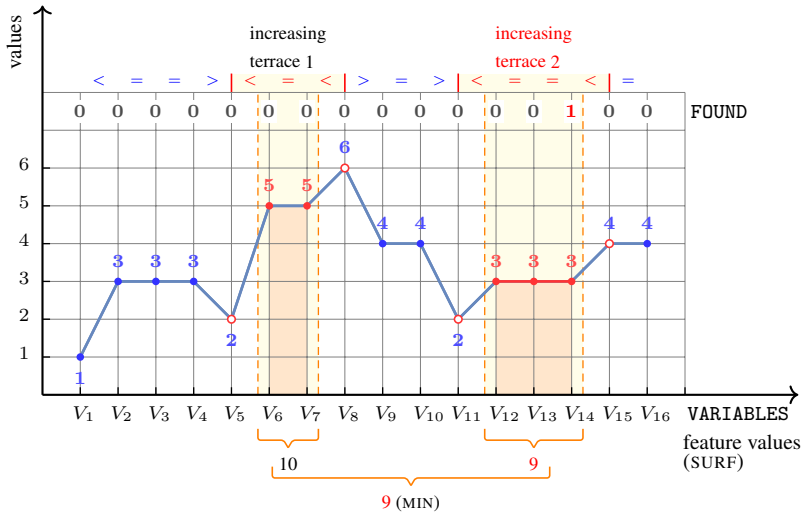


Figure 4.1199: Illustrating the POS_MIN_SURF_INCREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

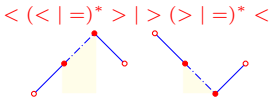
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_INFLEXION.

Constraint POS_MIN_SURF_INFLEXION(VALUE, VARIABLES, FOUND)

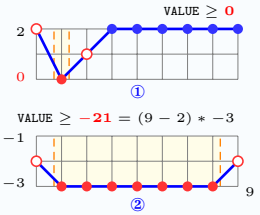
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \min(\mathtt{minv}①, (\mathtt{sv}-2) * \mathtt{minv}②)$
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \max(\mathtt{maxv}, (\mathtt{sv}-2) * \mathtt{maxv})$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv} \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \max(0, (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \min(0, (\mathtt{sv}-2) * (\mathtt{minv}+1)) - \mathtt{VALUE}$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv} \rangle)$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 1, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1200 provides an example where the POS_MIN_SURF_INFLEXION $(1, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0])$ constraint holds.

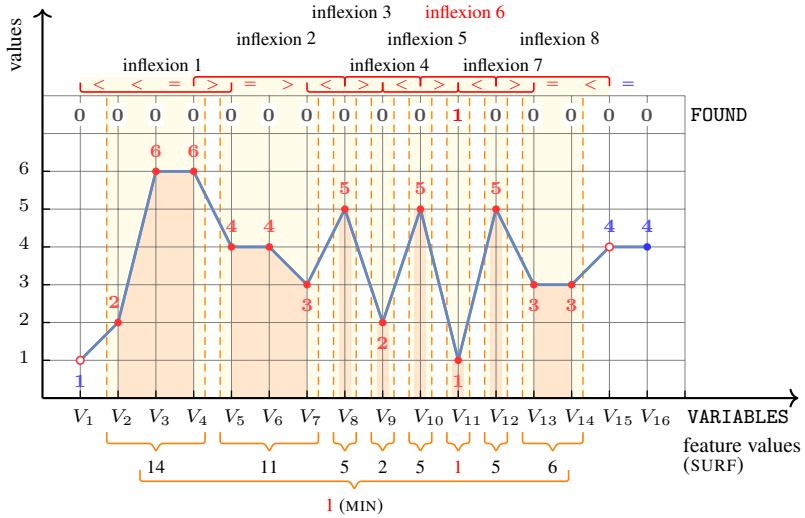


Figure 4.1200: Illustrating the POS_MIN_SURF_INFLEXION constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

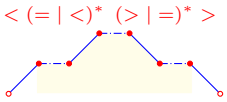
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_PEAK

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_PEAK.

Constraint POS_MIN_SURF_PEAK(VALUE, VARIABLES, FOUND)

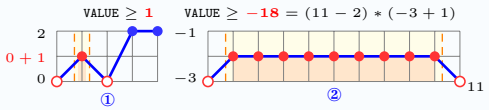
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{VALUE} \geq \mathtt{min}(\mathtt{minv}+1①, (\mathtt{sv}-2) * (\mathtt{minv}+1)②)$
$\mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{max}(\mathtt{maxv}, (\mathtt{sv}-2) * \mathtt{maxv})$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv} \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{max}(0, (\mathtt{sv}-2) * (\mathtt{maxv}-1))$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}+1 \rangle)$
$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \mathtt{min}(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose The constraint MIN_SURF_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 9, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1 \rangle, \\ \langle 0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$$

Figure 4.1201 provides an example where the POS_MIN_SURF_PEAK $(9, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1], [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])$ constraint holds.

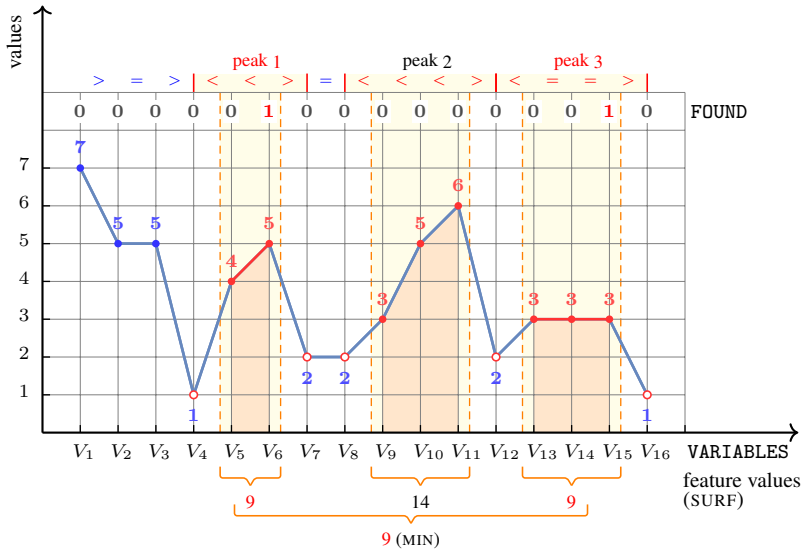


Figure 4.1201: Illustrating the POS_MIN_SURF_PEAK constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_PLAIN.

Constraint POS_MIN_SURF_PLAIN(VALUE, VARIABLES, FOUND)

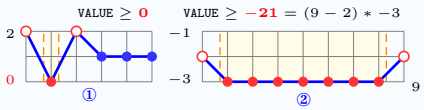
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(\texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}-1, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 4, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3 \rangle, \\ \langle 0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1202 provides an example where the POS_MIN_SURF_PLAIN $(4, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3], [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

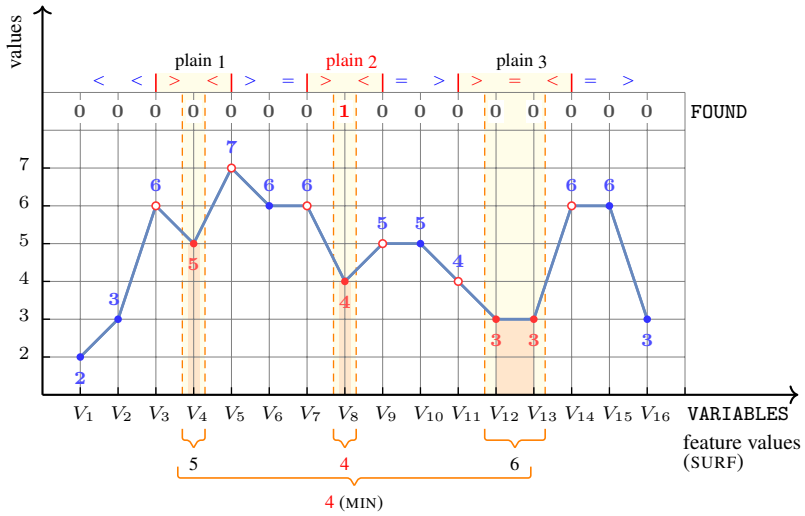


Figure 4.1202: Illustrating the POS_MIN_SURF_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_PLATEAU.

Constraint POS_MIN_SURF_PLATEAU(VALUE, VARIABLES, FOUND)

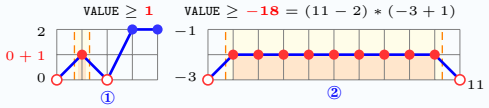
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(\texttt{minv} + 1①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}, (\texttt{sv} - 2) * \texttt{maxv})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example

$$\left(\begin{array}{l} 3, \langle 7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5 \rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1203 provides an example where the POS_MIN_SURF_PLATEAU $(3, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

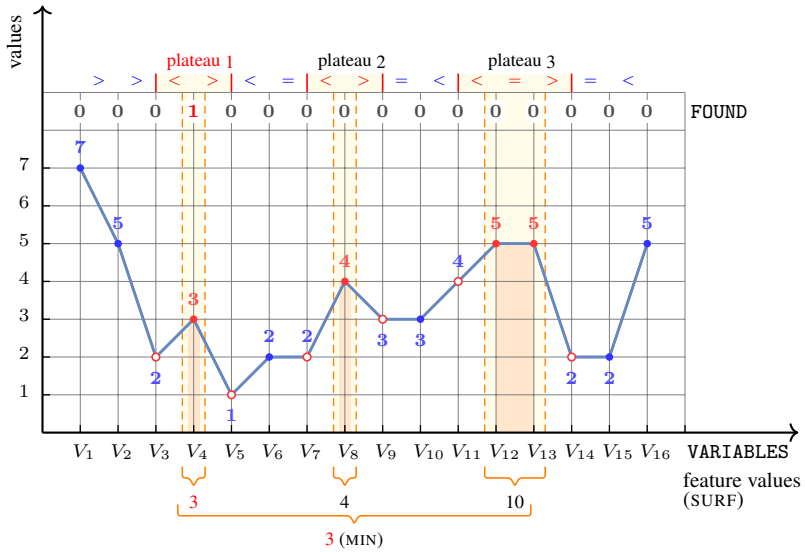


Figure 4.1203: Illustrating the POS_MIN_SURF_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

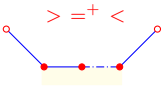
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_PROPER_PLAIN.

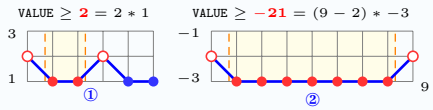
Constraint POS_MIN_SURF_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(2 * \texttt{minv①}, (\texttt{sv} - 2) * \texttt{minv②})$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1), (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose The constraint MIN_SURF_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^+<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $\left(8, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5 \rangle , \langle 0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0 \rangle \right)$

Figure 4.1204 provides an example where the POS_MIN_SURF_PROPER_PLAIN $(8, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5], [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

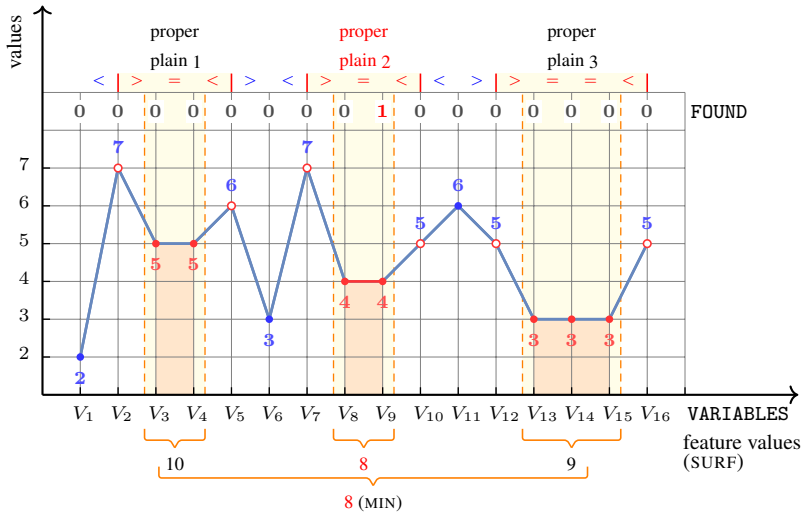


Figure 4.1204: Illustrating the POS_MIN_SURF_PROPER_PLAIN constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

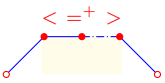
Automaton Similar to the automaton of the MIN_SURF_PROPER_PLAIN constraint but use the decoration table 3.35.

POS_MIN_SURF_PROPER_PLAIN 2559

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_PROPER_PLATEAU.

Constraint POS_MIN_SURF_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

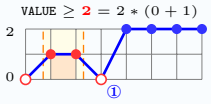
Arguments

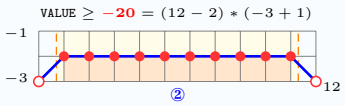
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \texttt{min}(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}, (\texttt{sv} - 2) * \texttt{maxv})$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
among(n1, VARIABLES[2, sv − 1], ⟨maxv⟩)
among(n2, VARIABLES[2, sv − 1], ⟨minv + 1⟩)
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|

where
minv =minval(VARIABLES.var)
sv = |VARIABLES|
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)





Purpose

The constraint MIN_SURF_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 6, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3\rangle, \\ \langle 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1205 provides an example where the POS_MIN_SURF_PROPER_PLATEAU $(6, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3], [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

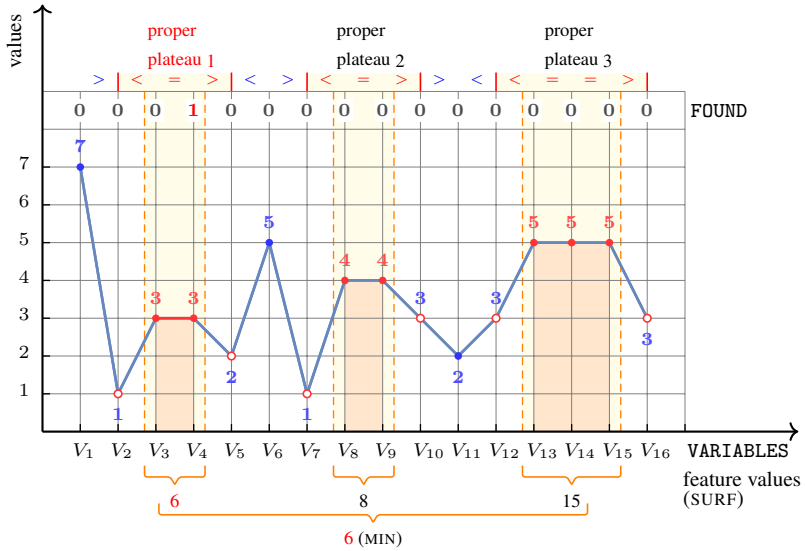


Figure 4.1205: Illustrating the POS_MIN_SURF_PROPER_PLATEAU constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MIN_SURF_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_STEADY.

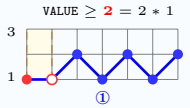
Constraint POS_MIN_SURF_STEADY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq 2 * \texttt{minv}$①
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv}$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - 2 * (\texttt{maxv} - 1)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq 2 * (\texttt{minv} + 1) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose The constraint MIN_SURF_STEADY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $\left(\begin{array}{l} 2, \langle 1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6 \rangle, \\ \langle 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$

Figure 4.1206 provides an example where the POS_MIN_SURF_STEADY $(2, [1,1,7,3,3,5,5,5,6,5,5,5,7,2,6,6], [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

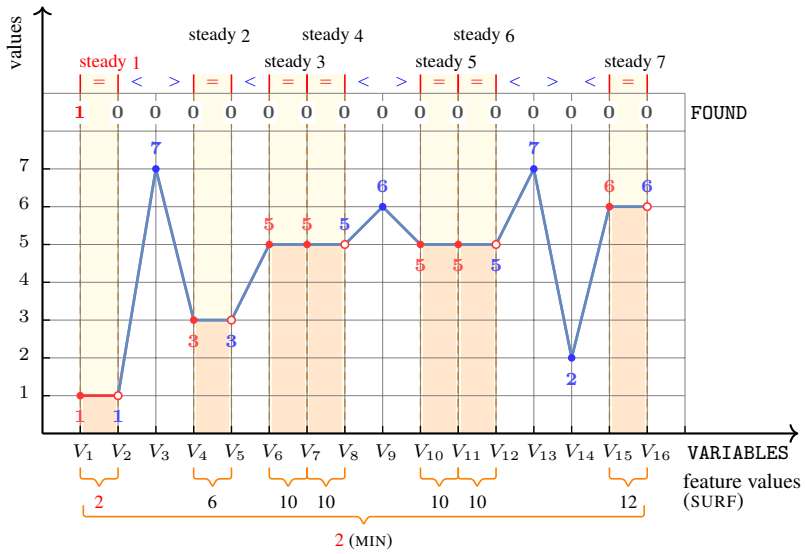


Figure 4.1206: Illustrating the POS_MIN_SURF_STEADY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

Arg. properties

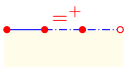
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_STEADY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_STEADY_SEQUENCE.

Constraint POS_MIN_SURF_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

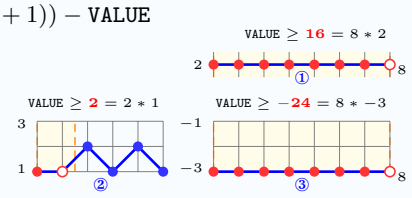
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv} * \texttt{minv}$①
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq \min(2 * \texttt{minv}$②$, \texttt{sv} * \texttt{minv}$③$)$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \texttt{sv} * \texttt{maxv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv}, \texttt{sv} * \texttt{maxv})$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{sv} * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, \texttt{sv} * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(\begin{array}{l} 2, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1 \rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$$

Figure 4.1207 provides an example where the POS_MIN_SURF_STEADY_SEQUENCE $(2, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

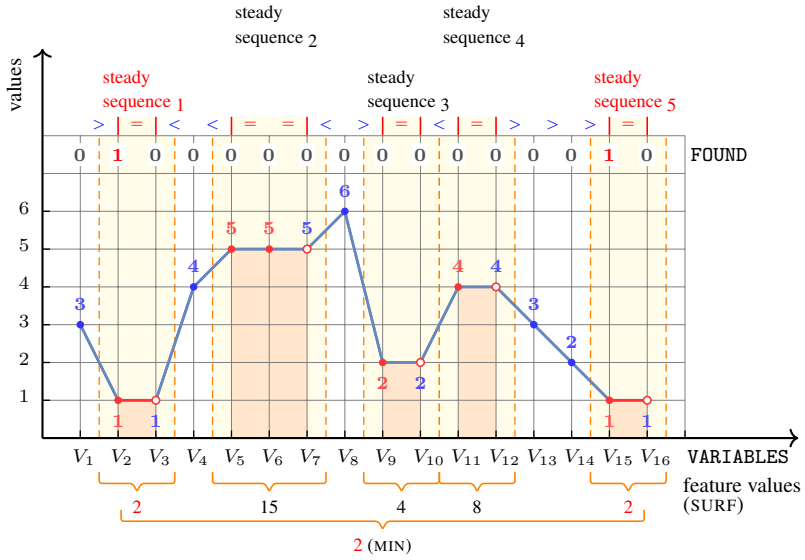


Figure 4.1207: Illustrating the POS_MIN_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$

Arg. properties

- Functional dependency: VALUE determined by VARIABLES.
- Functional dependency: FOUND determined by VARIABLES.

Automaton Similar to the automaton of the MIN_SURF_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_STRICTLY_DECREASING_SEQUENCE.

Constraint POS_MIN_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

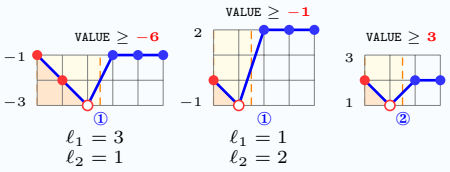
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$ ①
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$ ②
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
among(n1, VARIABLES[1, sv], ⟨maxv − 1, maxv⟩)
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
among(n2, VARIABLES[1, sv], ⟨minv, minv + 1⟩)
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|

where
minv = minval(VARIABLES.var)
rv = range(VARIABLES.var)
sv = |VARIABLES|
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
maxv = maxval(VARIABLES.var)



Purpose

The constraint MIN_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(7, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \right)$$

Figure 4.1208 provides an example where the POS_MIN_SURF_STRICTLY_DECREASING_SEQUENCE $(7, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

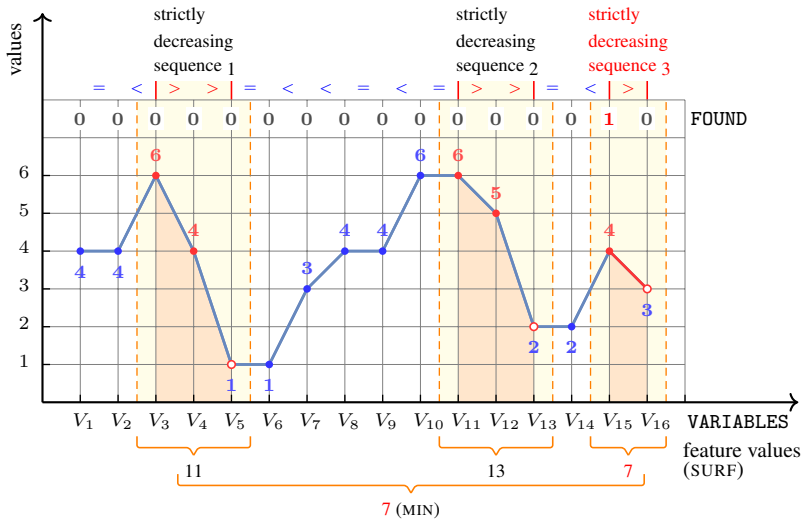


Figure 4.1208: Illustrating the POS_MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_SURF_STRICTLY_INCREASING_SEQUENCE.

Constraint POS_MIN_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

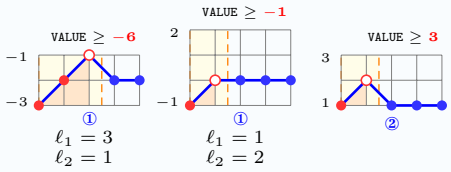
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{minv} < 0 \Rightarrow \texttt{VALUE} \geq \ell_1 * \texttt{minv} + \lfloor \ell_1 * (\ell_1 - 1)/2 \rfloor$①
$\texttt{minv} \geq 0 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} + 1$②
$\texttt{maxv} > 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \ell_2 * \texttt{maxv} - \lfloor \ell_2 * (\ell_2 - 1)/2 \rfloor$
$\texttt{maxv} \leq 0 \Rightarrow \texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq 2 * \texttt{maxv} - 1$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{maxv} - 2)) - 1$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \texttt{min}(0, \texttt{min}(\texttt{sv}, \texttt{rv}) * (\texttt{minv} + 2)) - 1 - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$

where

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\ell_1 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}|)$
$\ell_2 = \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}|)$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example

$$\left(6, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3 \rangle, \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \right)$$

Figure 4.1209 provides an example where the POS_MIN_SURF_STRICTLY_INCREASING_SEQUENCE $(6, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

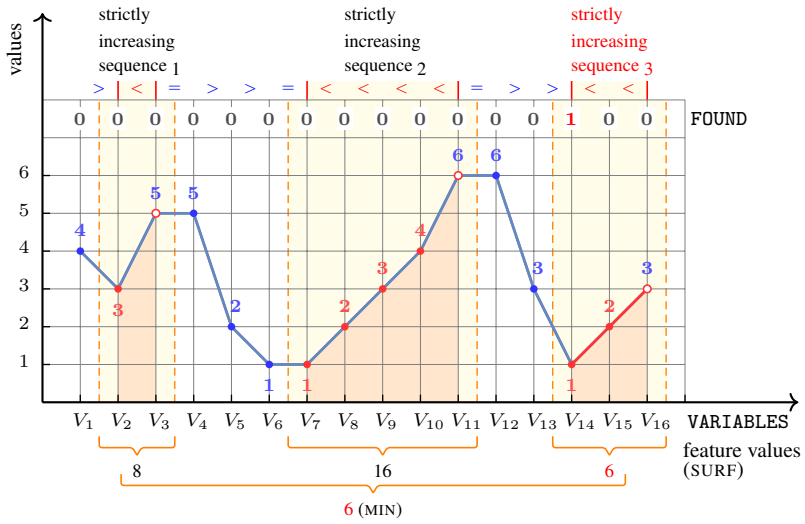


Figure 4.1209: Illustrating the POS_MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

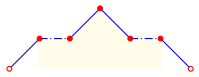
Automaton Similar to the automaton of the MIN_SURF_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MIN_SURF_SUMMIT.

Constraint POS_MIN_SURF_SUMMIT(VALUE, VARIABLES, FOUND)

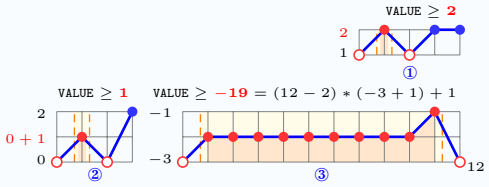
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} \geq \mathtt{minv} + 1$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} \geq \min(\mathtt{minv} + 1$①$, (\mathtt{sv} - 2) * (\mathtt{minv} + 1) + 1$②$)$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \mathtt{maxv}$
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = +\infty \vee \mathtt{VALUE} \leq \max(\mathtt{maxv}, (\mathtt{sv} - 2) * (\mathtt{maxv} - 1) + 1)$
$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv} - 1], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$
$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{maxv} = 1) \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \max(0, \mathtt{maxv} - 1)$
$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow$
$\mathtt{n1} \geq \mathtt{VALUE} - (\mathtt{sv} - 2) * (\mathtt{maxv} - 2) - 1$
$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv} - 1], \langle \mathtt{minv} + 1 \rangle)$
$\mathtt{VALUE} < +\infty \wedge (\mathtt{rv} = 2 \vee \mathtt{minv} = -1) \Rightarrow \mathtt{n2} \geq \min(0, \mathtt{minv} + 2) - \mathtt{VALUE}$
$\mathtt{VALUE} < +\infty \wedge \mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq (\mathtt{sv} - 2) * (\mathtt{minv} + 2) - \mathtt{VALUE}$
$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$
$\mathtt{required}(\mathtt{FOUND}, \mathtt{var})$
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$

where
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$



Purpose

The constraint MIN_SURF_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '(< | < (= | <)* <)(> | > (= | >)* >)'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example

$$\left(3, \langle 7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1\rangle, \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \right)$$

Figure 4.1210 provides an example where the POS_MIN_SURF_SUMMIT $(3, [7,1,5,4,4,3,3,4,6,6,2,3,4,2,3,1], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

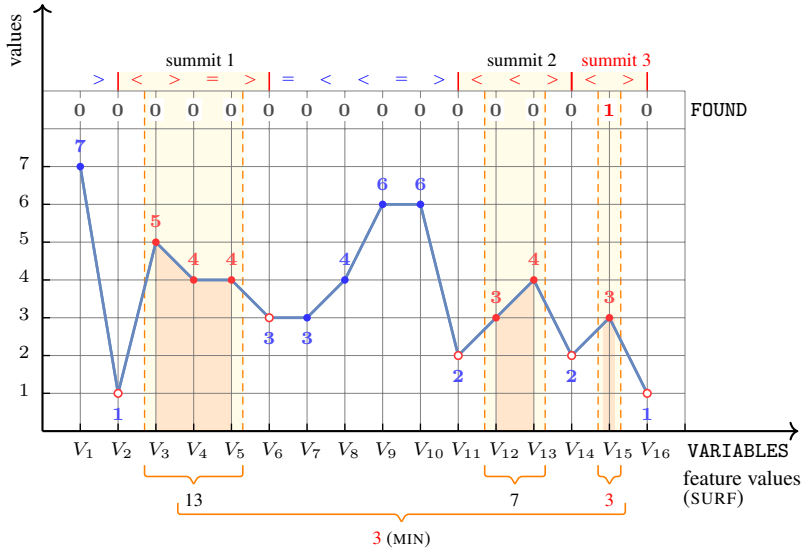


Figure 4.1210: Illustrating the POS_MIN_SURF_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

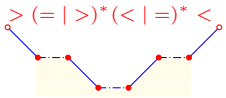
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_SUMMIT constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_SURF_VALLEY.

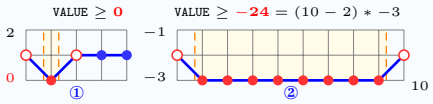
Constraint POS_MIN_SURF_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = +\infty$
$\texttt{VALUE} \geq \min(\texttt{minv}①, (\texttt{sv}-2) * \texttt{minv}②)$
$\texttt{VALUE} = +\infty \vee \texttt{VALUE} \leq \max(\texttt{maxv}-1, (\texttt{sv}-2) * (\texttt{maxv}-1))$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1 \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv} \rangle)$
$\texttt{VALUE} < +\infty \Rightarrow \texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
$\texttt{required}(\texttt{FOUND}, \texttt{var})$
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$



where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

Purpose

The constraint MIN_SURF_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(\begin{array}{l} 7, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7 \rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1211 provides an example where the POS_MIN_SURF_VALLEY $(7, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7], [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])$ constraint holds.

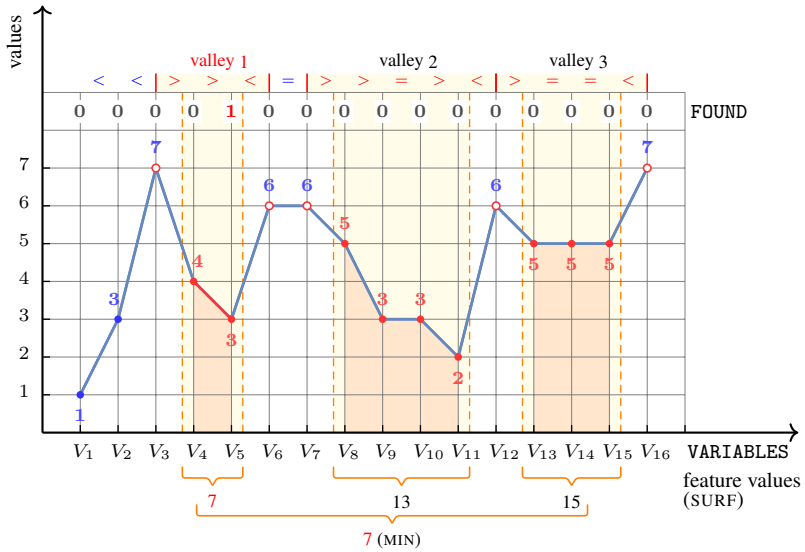


Figure 4.1211: Illustrating the POS_MIN_SURF_VALLEY constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

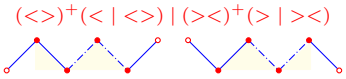
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_SURF_ZIGZAG

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MIN_SURF_ZIGZAG.

Constraint POS_MIN_SURF_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = +\infty$

$$\mathtt{VALUE} \geq \min \left(\begin{array}{l} 2 * \mathtt{minv} + 1①, \\ \lfloor (\mathtt{sv}-1)/2 \rfloor * \mathtt{minv} + \lfloor (\mathtt{sv}-2)/2 \rfloor * (\mathtt{minv}+1)② \end{array} \right)$$

$$\bigvee \left(\begin{array}{l} \mathtt{VALUE} = +\infty, \\ \mathtt{VALUE} \leq \max \left(\begin{array}{l} 2 * \mathtt{maxv} - 1, \\ \lfloor (\mathtt{sv}-1)/2 \rfloor * \mathtt{maxv} + \lfloor (\mathtt{sv}-2)/2 \rfloor * (\mathtt{maxv}-1) \end{array} \right) \end{array} \right)$$

among(n1, VARIABLES[2, sv − 1], ⟨maxv − 1, maxv⟩)

$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \lfloor (\mathtt{sv}-1)/2 \rfloor - \max(0, (\mathtt{sv}-2) * (\mathtt{maxv}-2))$

among(n2, VARIABLES[2, sv − 1], ⟨minv, minv + 1⟩)

$\mathtt{VALUE} < +\infty \Rightarrow \mathtt{n2} \geq \min(0, (\mathtt{sv}-2) * (\mathtt{minv}+2)) - \lfloor (\mathtt{sv}-1)/2 \rfloor - \mathtt{VALUE}$

required(VARIABLES, var)

required(FOUND, var)

|VARIABLES| = |FOUND|

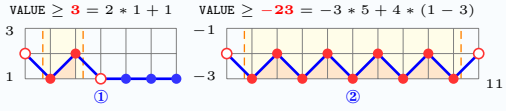
where

minv =minval(VARIABLES.var)

sv = |VARIABLES|

maxv =maxval(VARIABLES.var)

rv =range(VARIABLES.var)



Purpose

The constraint MIN_SURF_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example

$$\left(5, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \langle 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \right)$$

Figure 4.1212 provides an example where the POS_MIN_SURF_ZIGZAG $(5, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

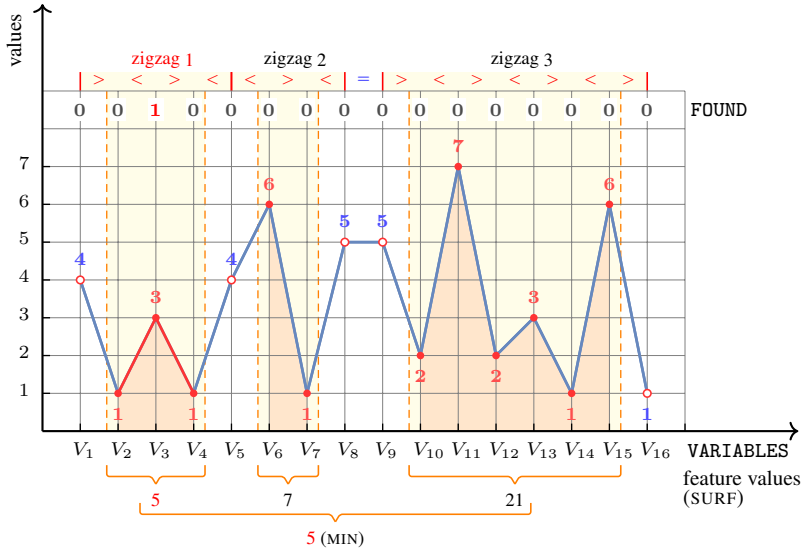


Figure 4.1212: Illustrating the POS_MIN_SURF_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

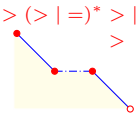
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_SURF_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_WIDTH_DECREASING_SEQUENCE.

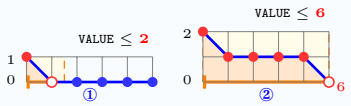
Constraint POS_MIN_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '> (> | =)* > | >'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 2, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1213 provides an example where the POS_MIN_WIDTH_DECREASING_SEQUENCE $(2, [3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4], [0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

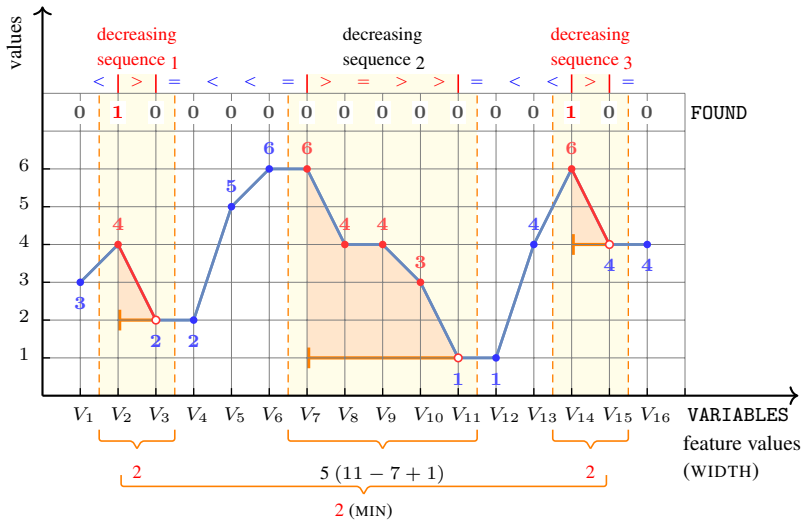


Figure 4.1213: Illustrating the POS_MIN_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

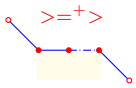
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_DECREASING_TERRACE.

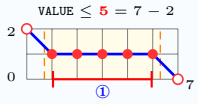
Constraint POS_MIN_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 2$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern DECREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^+>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0\rangle \end{array}\right)$$

Figure 4.1214 provides an example where the POS_MIN_WIDTH_DECREASING_TERRACE $(2, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3], [0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

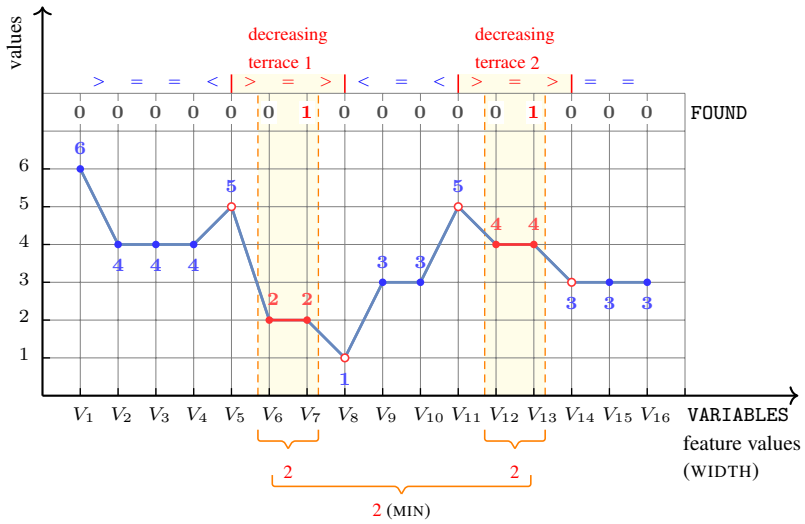


Figure 4.1214: Illustrating the POS_MIN_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Arg. properties

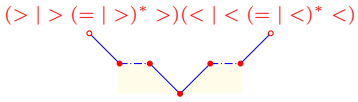
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_DECREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_GORGE.

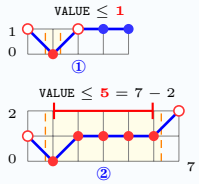
Constraint POS_MIN_WIDTH_GORGE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 1$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$②
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_GORGE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern GORGE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 1, \langle 1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7 \rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0 \rangle \end{array} \right)$$

Figure 4.1215 provides an example where the POS_MIN_WIDTH_GORGE $(1, [1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

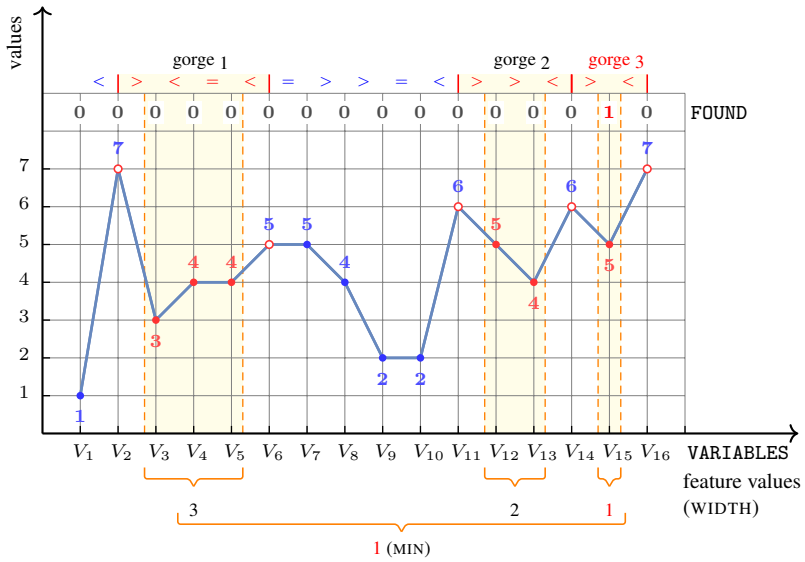


Figure 4.1215: Illustrating the POS_MIN_WIDTH_GORGE constraint of the **Example** slot

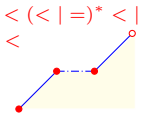
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_GORGE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_WIDTH_INCREASING_SEQUENCE.

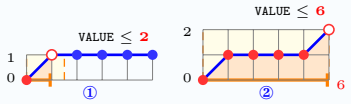
Constraint POS_MIN_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
required(FOUND, var)
|VARIABLES| = |FOUND|
where
sv = |VARIABLES|
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 2, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \rangle \end{array} \right)$$

Figure 4.1216 provides an example where the POS_MIN_WIDTH_INCREASING_SEQUENCE $(2, [4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3], [0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0])$ constraint holds.

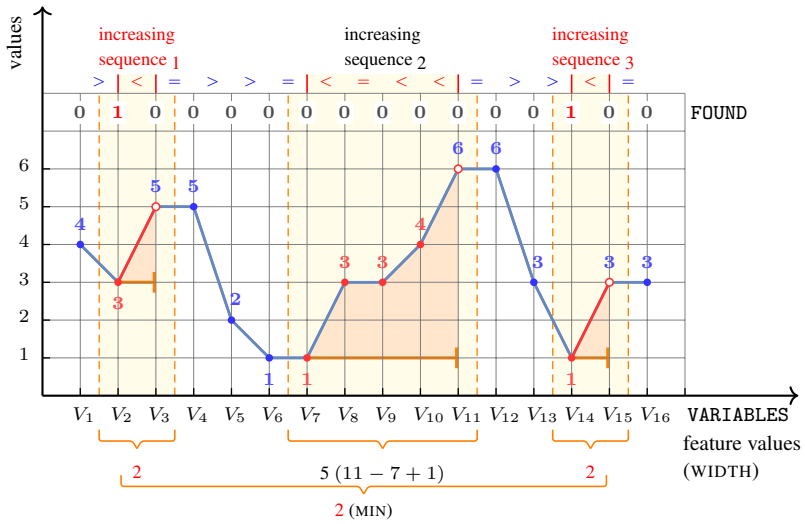


Figure 4.1216: Illustrating the POS_MIN_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

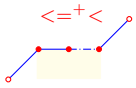
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_INCREASING_TERRACE.

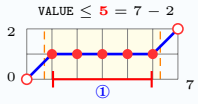
Constraint POS_MIN_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 2 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 2$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INCREASING_TERRACE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1217 provides an example where the POS_MIN_WIDTH_INCREASING_TERRACE $(2, [1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

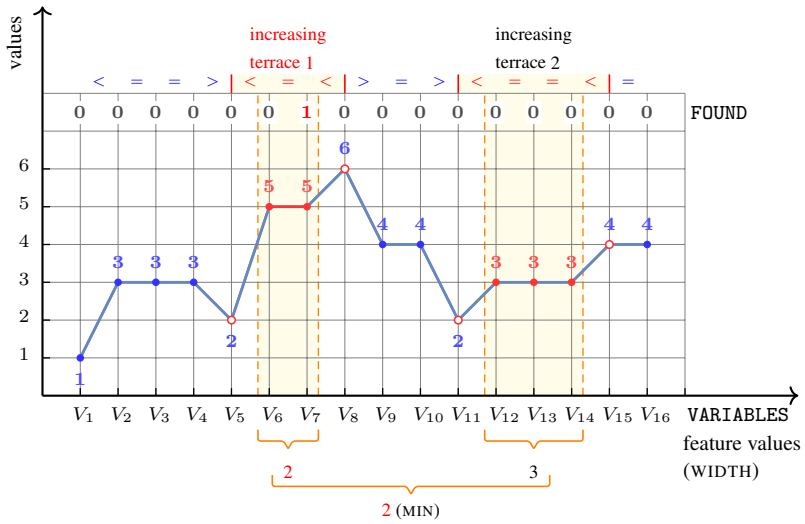


Figure 4.1217: Illustrating the POS_MIN_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Arg. properties

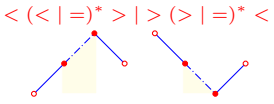
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_INCREASING_TERRACE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_INFLEXION.

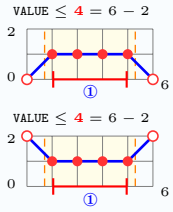
Constraint POS_MIN_WIDTH_INFLEXION(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 1$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MIN_WIDTH_INFLEXION(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern INFLEXION for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 1, \langle 1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4\rangle, \\ \langle 0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1218 provides an example where the POS_MIN_WIDTH_INFLEXION $(1, [1,2,6,6,4,4,3,5,2,5,1,5,3,3,4,4], [0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
$\mathtt{range}(\mathtt{VARIABLES.var}) > 1$

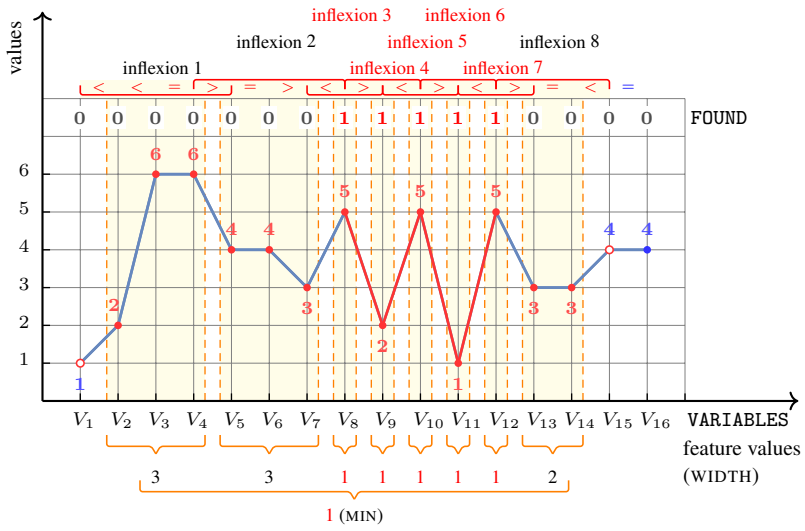


Figure 4.1218: Illustrating the POS_MIN_WIDTH_INFLEXION constraint of the **Example** slot

Arg. properties

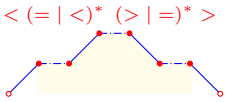
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_INFLEXION constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_PEAK.

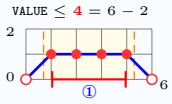
Constraint POS_MIN_WIDTH_PEAK(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 1$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_PEAK(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PEAK for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1\rangle, \\ \langle 0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1219 provides an example where the POS_MIN_WIDTH_PEAK $(2, [7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1], [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

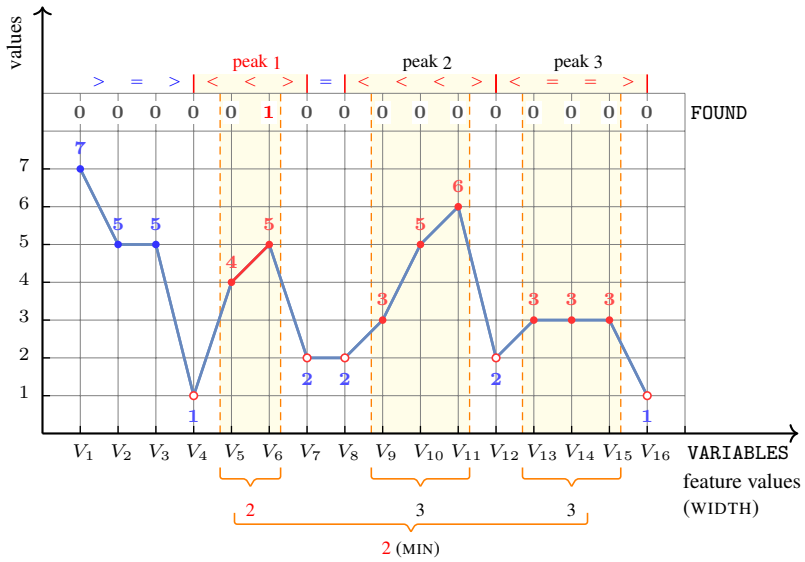


Figure 4.1219: Illustrating the POS_MIN_WIDTH_PEAK constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_PEAK constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_WIDTH_PLAIN.

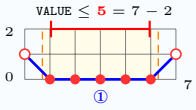
Constraint POS_MIN_WIDTH_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 1$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 1, \langle 2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3\rangle, \\ \langle 0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1220 provides an example where the POS_MIN_WIDTH_PLAIN $(1, [2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3], [0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

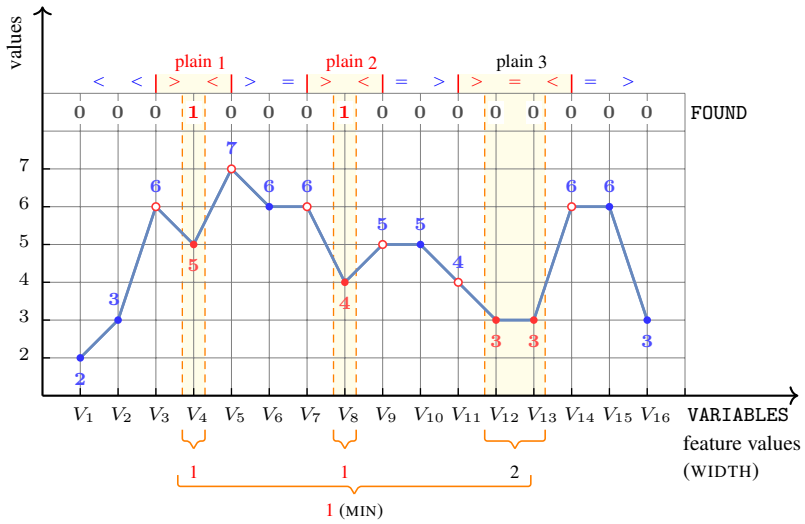


Figure 4.1220: Illustrating the POS_MIN_WIDTH_PLAIN constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_WIDTH_PLATEAU.

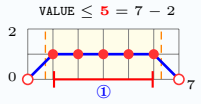
Constraint POS_MIN_WIDTH_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 1$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose The constraint MIN_WIDTH_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PLATEAU for which the feature value is VALUE. The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $\left(\begin{array}{l} 3, \langle 1,3,3,5,5,5,5,2,4,4,4,3,3,1,5,5\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0\rangle \end{array}\right)$

Figure 4.1221 provides an example where the POS_MIN_WIDTH_PLATEAU $(3, [1,3,3,5,5,5,5,2,4,4,4,3,3,1,5,5], [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

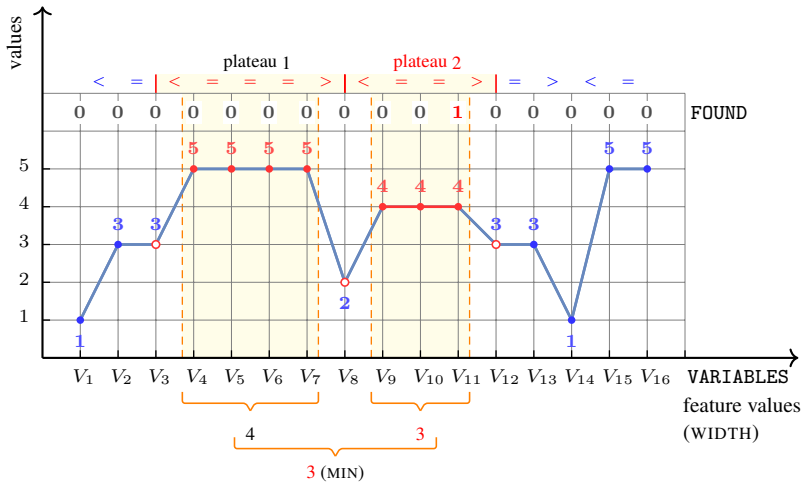


Figure 4.1221: Illustrating the POS_MIN_WIDTH_PLATEAU constraint of the **Example** slot

Arg. properties

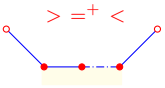
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_PROPER_PLAIN.

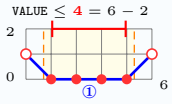
Constraint POS_MIN_WIDTH_PROPER_PLAIN(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_PROPER_PLAIN(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLAIN for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5 \rangle, \\ \langle 0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1222 provides an example where the POS_MIN_WIDTH_PROPER_PLAIN $(2, [2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5], [0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

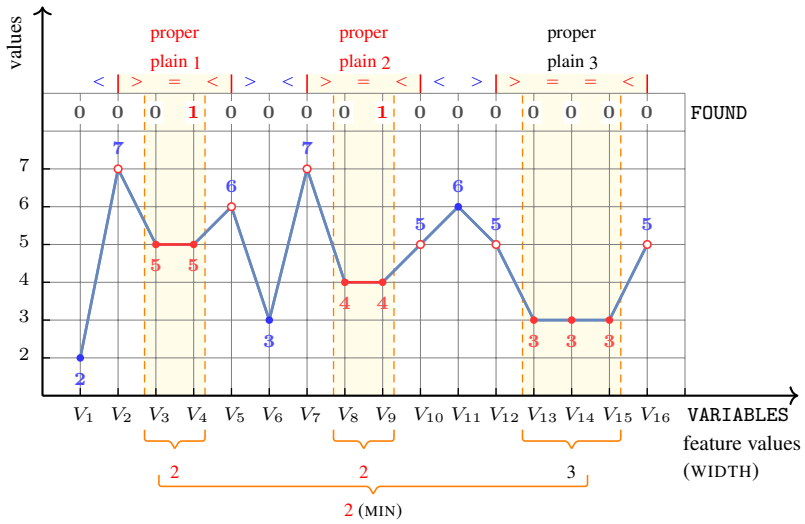


Figure 4.1222: Illustrating the POS_MIN_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Arg. properties

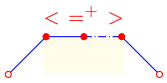
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_PROPER_PLAIN constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_PROPER_PLATEAU.

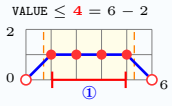
Constraint POS_MIN_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv} - 2$ ①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern PROPER_PLATEAU for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$< =^{+} >$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3 \rangle, \\ \langle 0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1223 provides an example where the POS_MIN_WIDTH_PROPER_PLATEAU $(2, [7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3], [0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

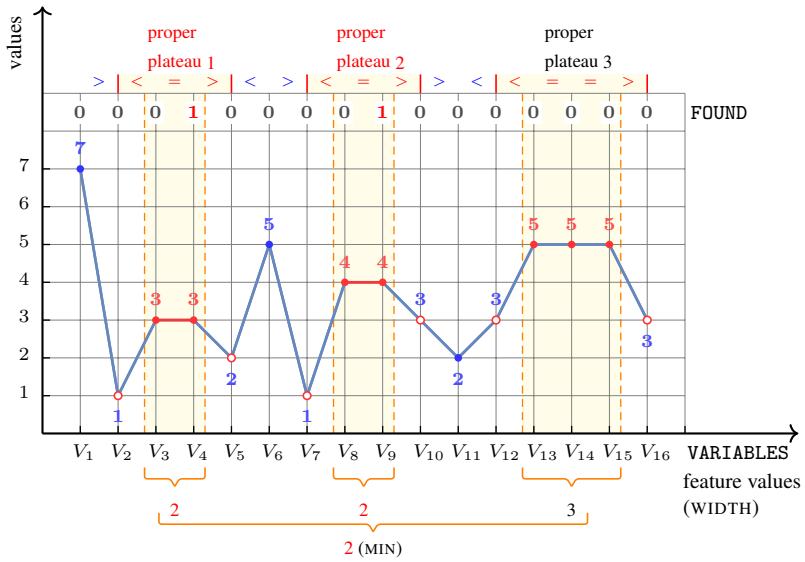


Figure 4.1223: Illustrating the POS_MIN_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Arg. properties

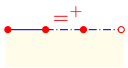
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_PROPER_PLATEAU constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_STEADY_SEQUENCE.

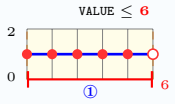
Constraint POS_MIN_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \texttt{sv}$①
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

The constraint MIN_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STEADY_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 2, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle, \\ \langle 0,1,0,0,0,0,0,0,1,0,1,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1224 provides an example where the POS_MIN_WIDTH_STEADY_SEQUENCE $(2, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1], [0,1,0,0,0,0,0,0,1,0,1,0,0,0,1,0])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

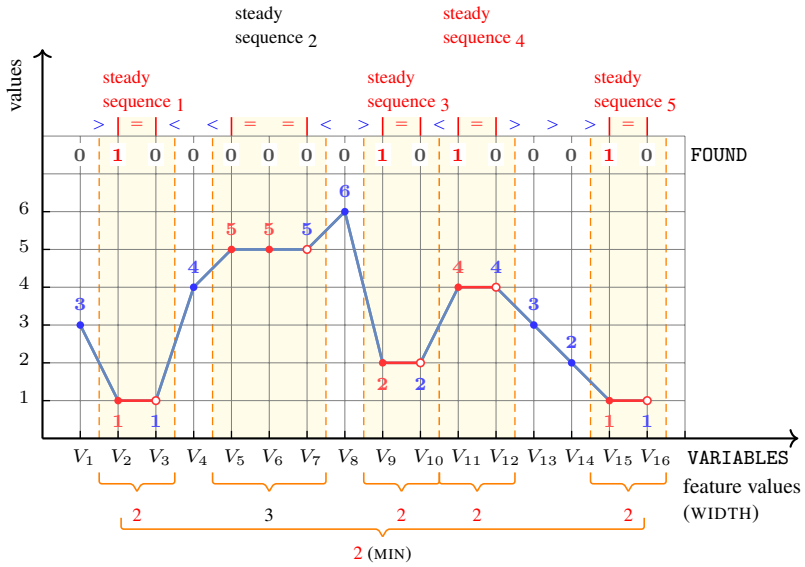


Figure 4.1224: Illustrating the POS_MIN_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_STEADY_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on constraint MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE.

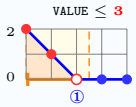
Constraint POS_MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_DECREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 2, \langle 4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3\rangle, \\ \langle 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0\rangle \end{array}\right)$$

Figure 4.1225 provides an example where the POS_MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE $(2, [4,4,6,4,1,1,3,4,4,6,6,5,2,2,4,3], [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

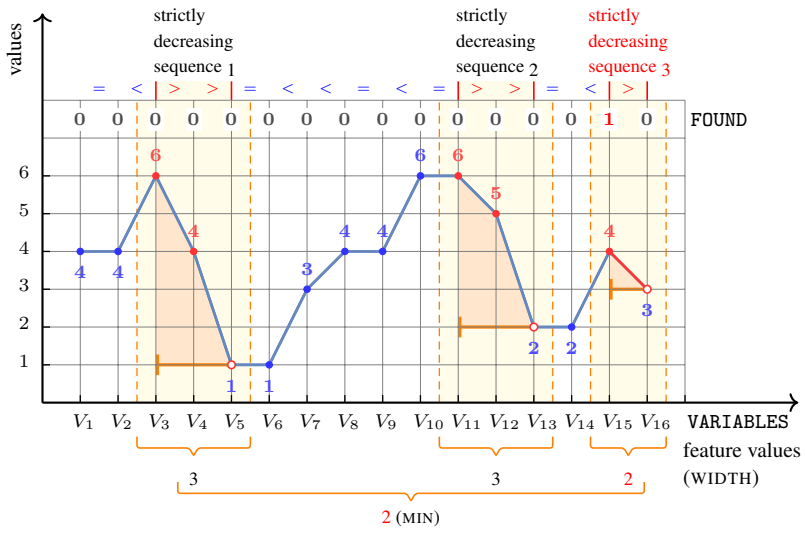


Figure 4.1225: Illustrating the POS_MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on constraint MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE.

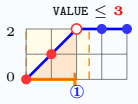
Constraint POS_MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = \texttt{sv} + 1$
$\texttt{VALUE} \geq 2$
$\texttt{VALUE} = \texttt{sv} + 1 \vee \texttt{VALUE} \leq \min(\texttt{sv}, \texttt{rv}①)$
required(VARIABLES, var)
required(FOUND, var)
$|\texttt{VARIABLES}| = |\texttt{FOUND}|$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern STRICTLY_INCREASING_SEQUENCE for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example

$$\left(\begin{array}{l} 2, \langle 4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3\rangle, \\ \langle 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1226 provides an example where the POS_MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE $(2, [4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3], [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

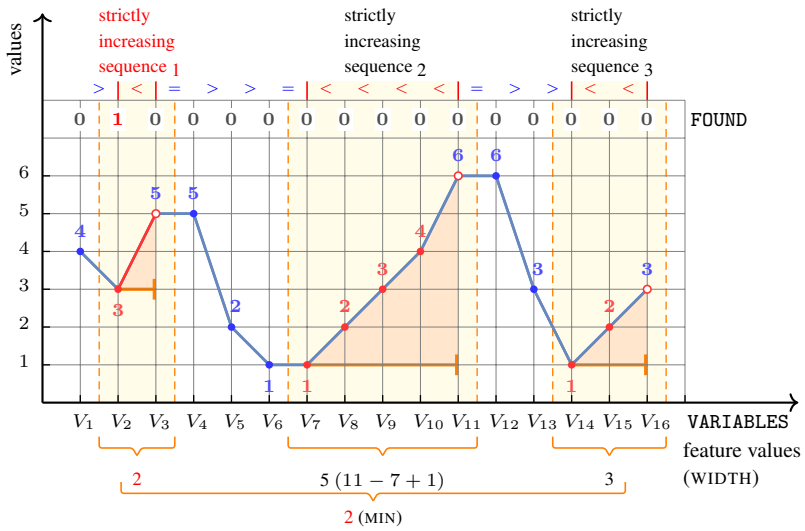


Figure 4.1226: Illustrating the POS_MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

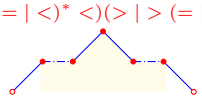
Automaton Similar to the automaton of the MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_SUMMIT.

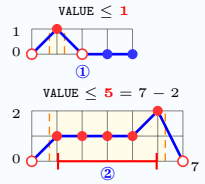
Constraint POS_MIN_WIDTH_SUMMIT(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 1$
$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq 1$①
$\mathtt{rv} \geq 3 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$②
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{rv} =$range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_SUMMIT(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern SUMMIT for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 1, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle, \\ \langle 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 \rangle \end{array} \right)$$

Figure 4.1227 provides an example where the POS_MIN_WIDTH_SUMMIT $(1, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

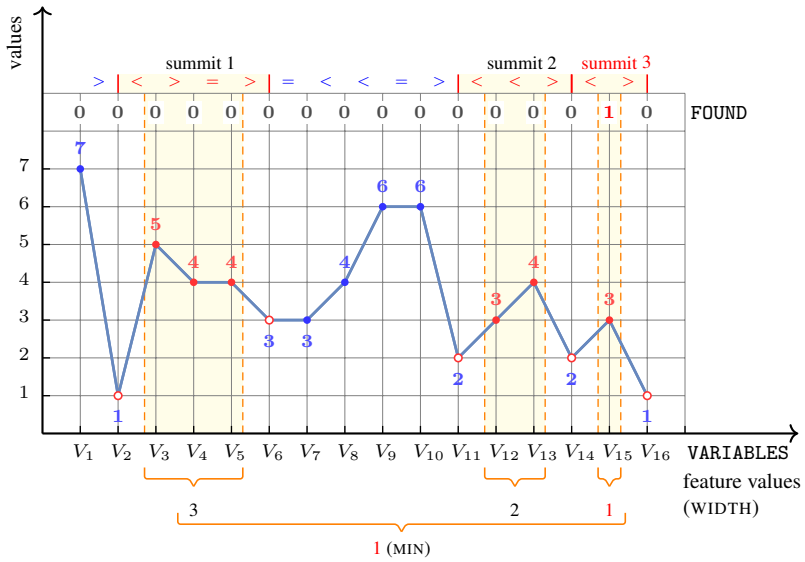


Figure 4.1227: Illustrating the POS_MIN_WIDTH_SUMMIT constraint of the **Example** slot

Arg. properties

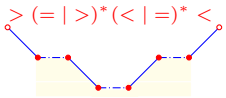
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_SUMMIT constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on constraint MIN_WIDTH_VALLEY.

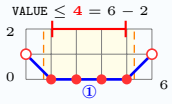
Constraint POS_MIN_WIDTH_VALLEY(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 1$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_VALLEY(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern VALLEY for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7 \rangle, \\ \langle 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0 \rangle \end{array} \right)$$

Figure 4.1228 provides an example where the POS_MIN_WIDTH_VALLEY $(2, [1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7], [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

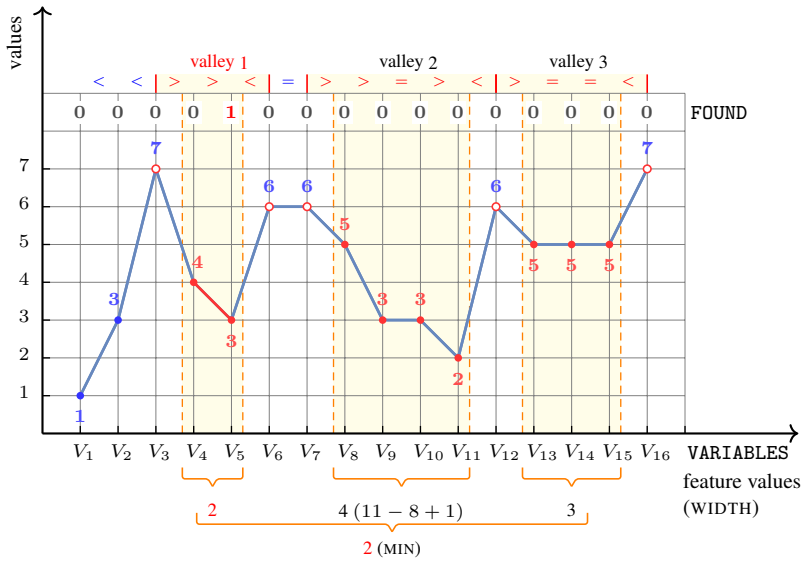


Figure 4.1228: Illustrating the POS_MIN_WIDTH_VALLEY constraint of the **Example** slot

Arg. properties

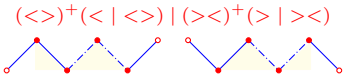
- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_VALLEY constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

POS_MIN_WIDTH_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on constraint MIN_WIDTH_ZIGZAG.

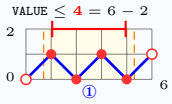
Constraint POS_MIN_WIDTH_ZIGZAG(VALUE, VARIABLES, FOUND)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)
FOUND	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 3 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = \mathtt{sv} + 1$
$\mathtt{VALUE} \geq 2$
$\mathtt{VALUE} = \mathtt{sv} + 1 \vee \mathtt{VALUE} \leq \mathtt{sv} - 2$①
required(VARIABLES, var)
required(FOUND, var)
$|\mathtt{VARIABLES}| = |\mathtt{FOUND}|$
where
$\mathtt{sv} = |\mathtt{VARIABLES}|$
rv =range(VARIABLES.var)



Purpose

The constraint MIN_WIDTH_ZIGZAG(VALUE, VARIABLES) holds. In addition, FOUND is a collection of 0/1 variables where the value 1 indicates the position of the **found** letter in those occurrences of the pattern ZIGZAG for which the feature value is VALUE.
The position of the **found** letter in an occurrence of a pattern is the first position where the occurrence of pattern is identified, even if the pattern is not complete.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example

$$\left(\begin{array}{l} 2, \langle 4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1\rangle, \\ \langle 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0\rangle \end{array}\right)$$

Figure 4.1229 provides an example where the POS_MIN_WIDTH_ZIGZAG $(2, [4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1], [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 3$
$\mathtt{range}(\mathtt{VARIABLES.var}) > 1$

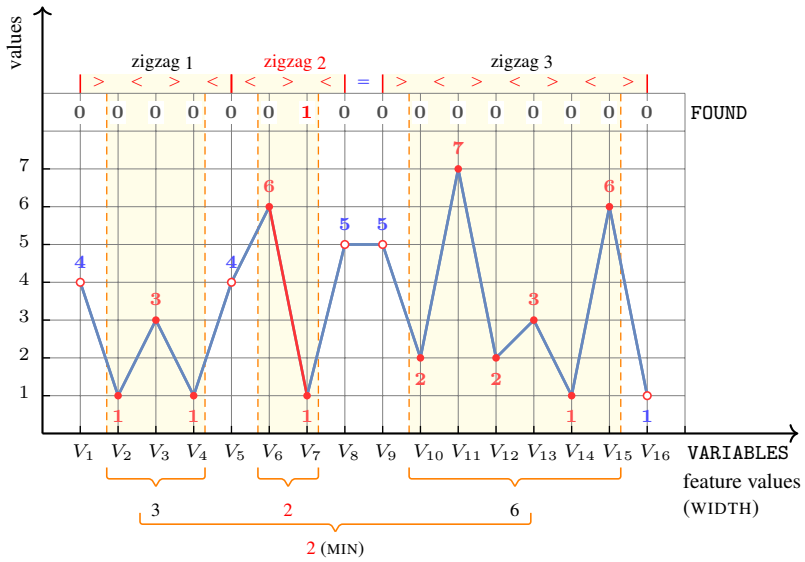


Figure 4.1229: Illustrating the POS_MIN_WIDTH_ZIGZAG constraint of the **Example** slot

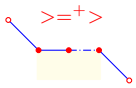
Arg. properties

- Functional dependency: `VALUE` determined by `VARIABLES`.
- Functional dependency: `FOUND` determined by `VARIABLES`.

Automaton Similar to the automaton of the MIN_WIDTH_ZIGZAG constraint but use the decoration table 3.35.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_DECREASING_TERRACE



DESCRIPTION AUTOMATON

Origin Based on the DECREASING_TERRACE pattern.

Constraint SUM_HEIGHT_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(0, \sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \texttt{rv} - 2 \ \right) \ \right), \\ \texttt{q}, \\ \sum \left(\begin{array}{l} \texttt{np1} * \texttt{minv}, \\ \lfloor (\texttt{np1}+1) * \texttt{np1}/2 \rfloor \end{array} \right) \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \left\lfloor \left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} \texttt{rv}, \\ |\texttt{minv}| + 1 \end{array} \right) - \\ \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \end{array} \right) / \right\rfloor - 2 \right) \ \right), \\ \lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \\ \texttt{q} * 2 * \texttt{np1} \end{array} \right) - \\ /2 \rfloor, \\ \texttt{minv} + \texttt{np1} + 1 \end{array} \right) \end{array} \right) \right)$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(0, \sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \texttt{rv} - 2 \ \right) \ \right), \\ \texttt{q}, \\ \texttt{np2} * \texttt{maxv} - \\ \lfloor (\texttt{np2}+1) * \texttt{np2}/2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \left\lfloor \left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} \texttt{rv}, \\ |\texttt{maxv}| + 1 \end{array} \right) - \\ \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \end{array} \right) / \right\rfloor - 2 \right) \ \right), \\ \lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \\ \texttt{q} * 2 * \texttt{np2} \end{array} \right) - \\ /2 \rfloor, \\ \texttt{maxv} - \texttt{np2} - 1 \end{array} \right) \end{array} \right) \right)$$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

$$\texttt{np1} = \texttt{max}\left(0, \sum \left(\begin{array}{l} \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor, \\ \texttt{min}\left(\ 0, \begin{array}{l} \texttt{min}(\texttt{rv}, |\texttt{minv}| + 1) - \\ \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor \end{array} - 2 \ \right) \end{array} \right) \right)$$

$$\texttt{np2} = \texttt{max}\left(0, \sum \left(\begin{array}{l} \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor, \\ \texttt{min}\left(\ 0, \begin{array}{l} \texttt{min}(\texttt{rv}, |\texttt{maxv}| + 1) - \\ \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor \end{array} - 2 \ \right) \end{array} \right) \right)$$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$$\texttt{lb1} = \texttt{min}\left(\begin{array}{l} \lfloor \texttt{sv}/4 \rfloor + 1, \\ \sum \left(\begin{array}{l} \left\lfloor \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2)/ \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(1, (\texttt{min}(|\texttt{minv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} \right\rfloor, \\ \texttt{max}\left(0, \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) \bmod \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(2, (\texttt{min}(|\texttt{minv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) - 3 \end{array} \right) \right) \end{array} \right) \end{array} \right)$$

$\texttt{ub1} = \lfloor \texttt{sv}/4 \rfloor + 1$

$$\texttt{lb2} = \texttt{min}\left(\begin{array}{l} \lfloor \texttt{sv}/4 \rfloor + 1, \\ \sum \left(\begin{array}{l} \left\lfloor \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2)/ \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(1, (\texttt{min}(|\texttt{maxv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} \right\rfloor, \\ \texttt{max}\left(0, \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) \bmod \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(2, (\texttt{min}(|\texttt{maxv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) - 3 \end{array} \right) \right) \end{array} \right) \end{array} \right)$$

$\texttt{ub2} = \lfloor \texttt{sv}/4 \rfloor + 1$

Purpose

VALUE is the sum of all minimum values in each occurrence of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$(6, \langle 6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3\rangle)$

Figure 4.1230 provides an example where the SUM_HEIGHT_DECREASING_TERRACE $(6, [6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3])$ constraint holds.

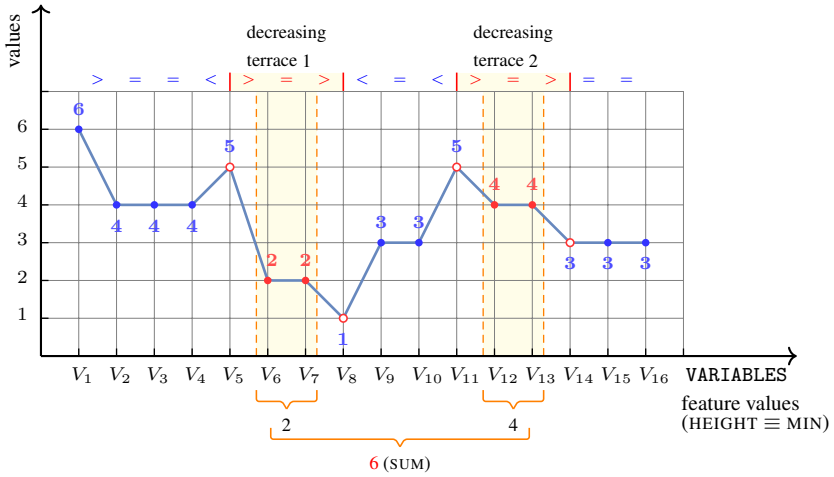


Figure 4.1230: Illustrating the SUM_HEIGHT_DECREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1231 and 4.1232 respectively depict the automaton associated with the constraint SUM_HEIGHT_DECREASING_TERRACE and its simplified form.

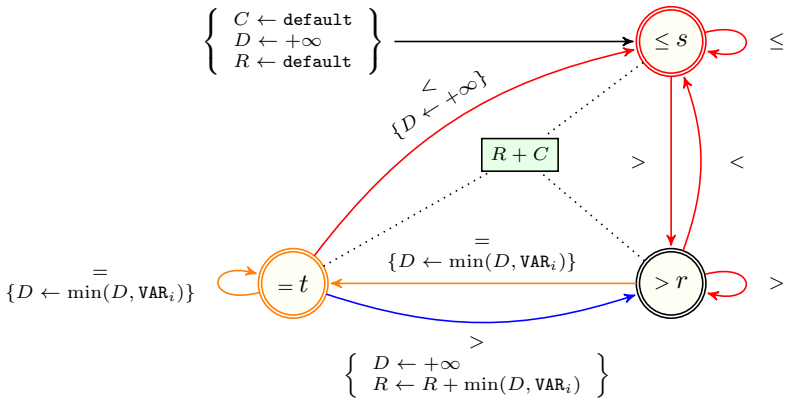


Figure 4.1231: Automaton for the SUM_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where default is 0

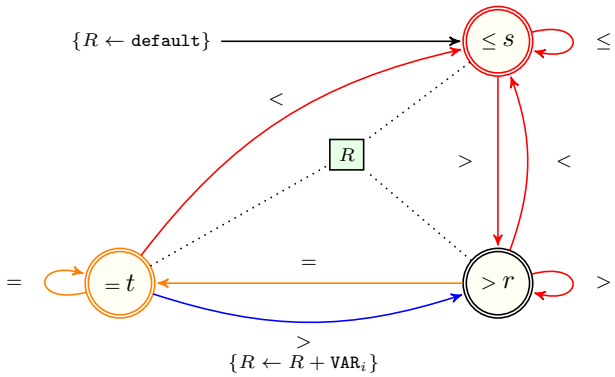


Figure 4.1232: Simplified automaton for the SUM_HEIGHT_DECREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_TERRACE pattern where default is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

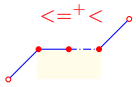
Table 4.265: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the SUM_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\mathtt{VAR}_{i+1}$ C
t	0	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.266: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the SUM_HEIGHT_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_INCREASING_TERRACE



DESCRIPTION AUTOMATON

Origin Based on the INCREASING_TERRACE pattern.

Constraint SUM_HEIGHT_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \texttt{rv} - 2 \ \right) \ \right), \\ \texttt{q}, \\ \sum \left(\begin{array}{l} \texttt{np1} * \texttt{minv}, \\ \lfloor (\texttt{np1}+1) * \texttt{np1}/2 \rfloor \end{array} \right) \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} 1, \texttt{max}\left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} \texttt{rv}, \\ |\texttt{minv}| + 1 \end{array} \right) - \\ 0, \left\lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \end{array} \right) / \right. \\ \left. 2 * \texttt{q} \right\rfloor - 2 \end{array} \right) \end{array} \right) \end{array} \right), \\ \lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \\ \texttt{q} * 2 * \texttt{np1} \end{array} \right) - \\ \texttt{minv} + \texttt{np1} + 1 \end{array} \right) \end{array} \right)$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{min}\left(\ 1, \texttt{max}\left(\ 0, \texttt{rv} - 2 \ \right) \ \right), \\ \texttt{q}, \\ \texttt{np2} * \texttt{maxv} - \\ \lfloor (\texttt{np2}+1) * \texttt{np2}/2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} 1, \texttt{max}\left(\begin{array}{l} \texttt{min}\left(\begin{array}{l} \texttt{rv}, \\ |\texttt{maxv}| + 1 \end{array} \right) - \\ 0, \left\lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \end{array} \right) / \right. \\ \left. 2 * \texttt{q} \right\rfloor - 2 \end{array} \right) \end{array} \right) \end{array} \right), \\ \lfloor \left(\begin{array}{l} \texttt{max}\left(\ 1, \texttt{sv} - \texttt{sv} \bmod 2 \ \right) - \\ 2 * \texttt{q} \\ \texttt{q} * 2 * \texttt{np2} \end{array} \right) - \\ \texttt{maxv} - \texttt{np2} - 1 \end{array} \right) \end{array} \right)$$

$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$$\texttt{sv} = |\texttt{VARIABLES}|$$

$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$

$$\texttt{np1} = \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor, \\ \texttt{min}\left(\begin{array}{l} 0, \ \texttt{min}(\texttt{rv}, |\texttt{minv}| + 1) - \\ \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor \end{array} - 2 \right) \end{array} \right) \end{array} \right)$$

$$\texttt{np2} = \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor, \\ \texttt{min}\left(\begin{array}{l} 0, \ \texttt{min}(\texttt{rv}, |\texttt{maxv}| + 1) - \\ \lfloor (\texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) - 2 * \texttt{q})/2 * \texttt{q} \rfloor \end{array} - 2 \right) \end{array} \right) \end{array} \right)$$

$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$

$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$

$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/4 \rfloor + 1, \\ \sum \left(\begin{array}{l} \left\lfloor \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2)/ \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(1, (\texttt{min}(|\texttt{minv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} \right\rfloor, \\ \texttt{max}\left(\begin{array}{l} 0, \texttt{min}\left(\begin{array}{l} 1, \ \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) \bmod \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(2, (\texttt{min}(|\texttt{minv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} - 3 \end{array} \right) \end{array} \right) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/4 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/4 \rfloor + 1, \\ \sum \left(\begin{array}{l} \left\lfloor \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2)/ \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(1, (\texttt{min}(|\texttt{maxv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} \right\rfloor, \\ \texttt{max}\left(\begin{array}{l} 0, \texttt{min}\left(\begin{array}{l} 1, \ \begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2) \bmod \\ \texttt{min}\left(\begin{array}{l} \texttt{max}(1, \texttt{sv} - \texttt{sv} \bmod 2), \\ \texttt{max}(2, (\texttt{min}(|\texttt{maxv}| + 1, \texttt{rv}) - 2) * 2 + 2) \end{array} \right) \end{array} - 3 \end{array} \right) \end{array} \right) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/4 \rfloor + 1$$

Purpose

VALUE is the sum of all minimum values in each occurrence of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example

$(8, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.1233 provides an example where the SUM_HEIGHT_INCREASING_TERRACE $(8, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

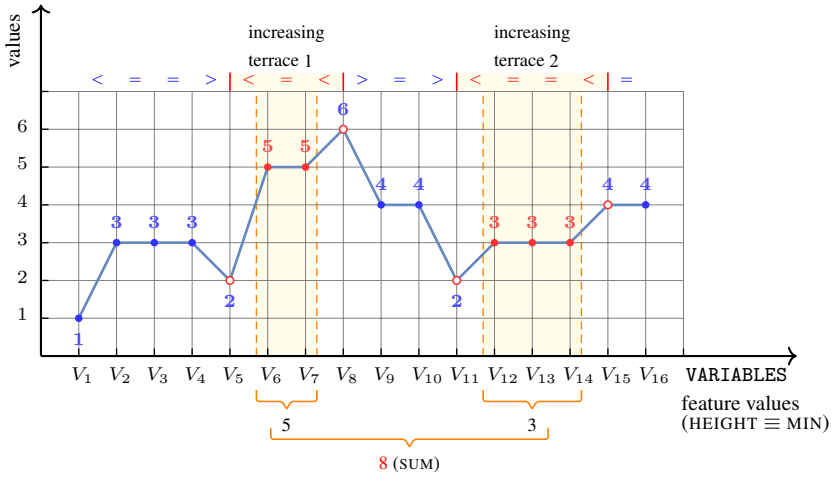


Figure 4.1233: Illustrating the SUM_HEIGHT_INCREASING_TERRACE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1234 and 4.1235 respectively depict the automaton associated with the constraint SUM_HEIGHT_INCREASING_TERRACE and its simplified form.

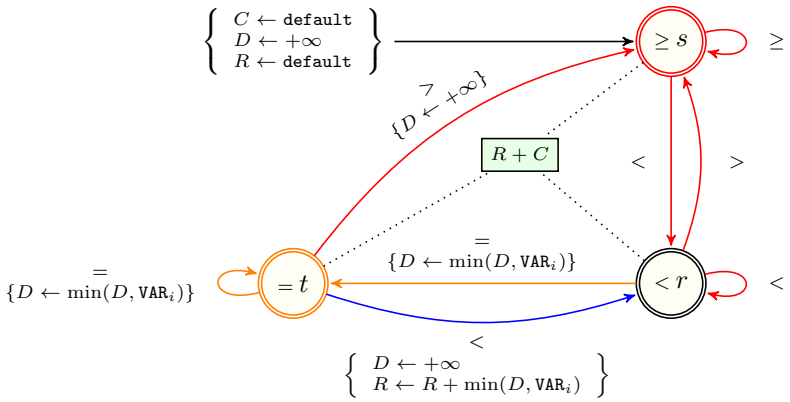


Figure 4.1234: Automaton for the SUM_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where default is 0

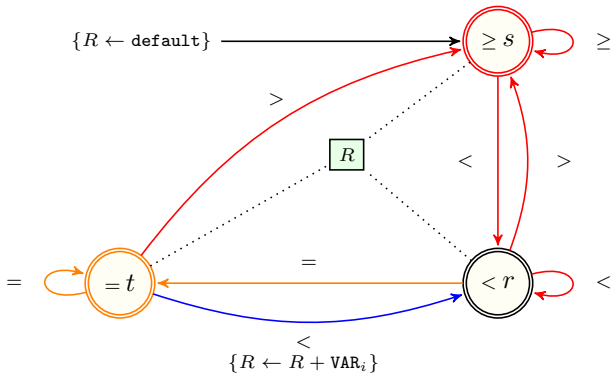


Figure 4.1235: Simplified automaton for the SUM_HEIGHT_INCREASING_TERRACE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_TERRACE pattern where default is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.267: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the SUM_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\texttt{VAR}_{i+1}$ C
t	0	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.268: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the SUM_HEIGHT_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

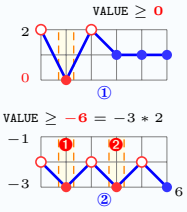
Constraint SUM_HEIGHT_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(\texttt{maxv} - 1, (\texttt{maxv} - 1) * \texttt{np})$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor (\texttt{sv} - 1)/2 \rfloor)$
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{rv} =$range(VARIABLES.var)



Purpose

VALUE is the sum of all minimum values in each occurrence of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(12, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3 \rangle)$

Figure 4.1236 provides an example where the SUM_HEIGHT_PLAIN $(12, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

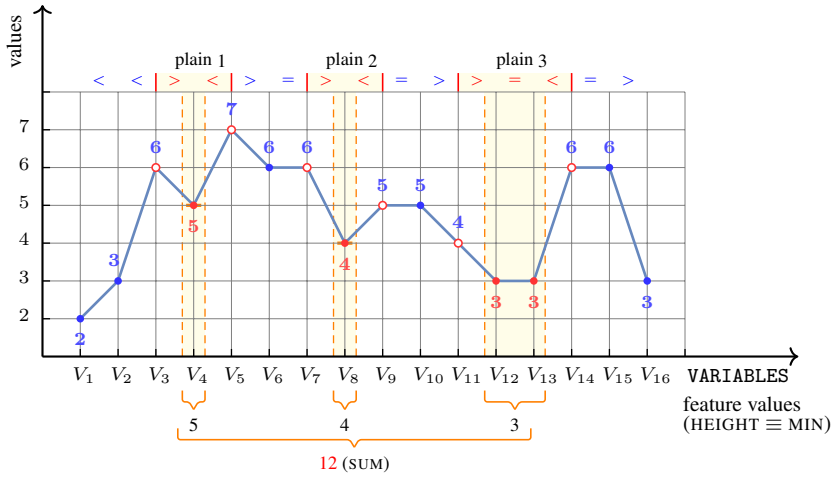


Figure 4.1236: Illustrating the SUM_HEIGHT_PLAIN constraint of the **Example** slot

Automaton Figures 4.1237 and 4.1238 respectively depict the automaton associated with the constraint SUM_HEIGHT_PLAIN and its simplified form.

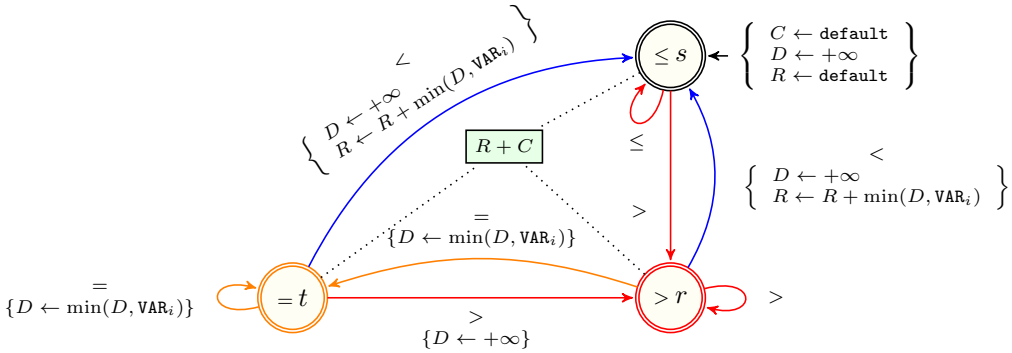


Figure 4.1237: Automaton for the SUM_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is 0

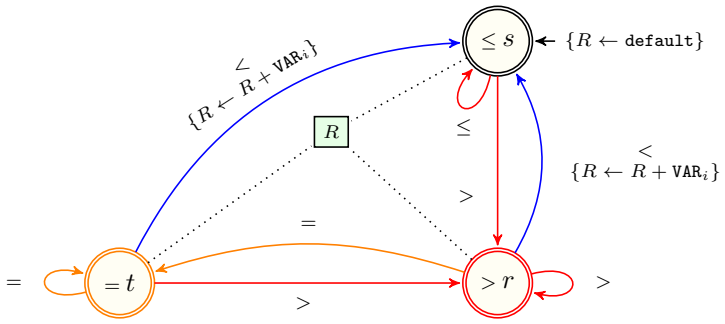


Figure 4.1238: Simplified automaton for the SUM_HEIGHT_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLAIN pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.269: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the SUM_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C
t	0	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.270: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the SUM_HEIGHT_PLAIN constraint defined as the composition of the PLAIN pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_PLATEAU

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLATEAU pattern.

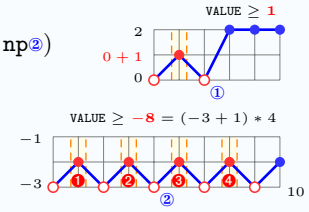
Constraint SUM_HEIGHT_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(\texttt{maxv}, \texttt{maxv} * \texttt{np})$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all minimum values in each occurrence of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(12, \langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5\rangle)$

Figure 4.1239 provides an example where the SUM_HEIGHT_PLATEAU $(12, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

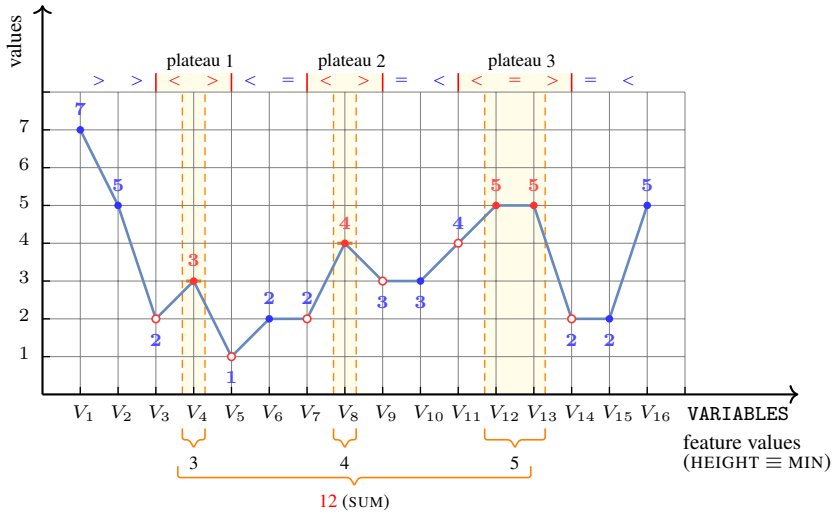


Figure 4.1239: Illustrating the SUM_HEIGHT_PLATEAU constraint of the **Example** slot

Automaton Figures 4.1240 and 4.1241 respectively depict the automaton associated with the constraint SUM_HEIGHT_PLATEAU and its simplified form.

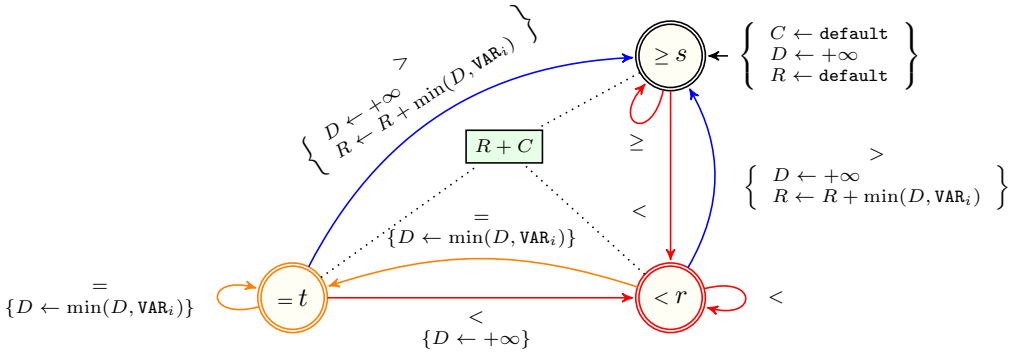


Figure 4.1240: Automaton for the SUM_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is 0

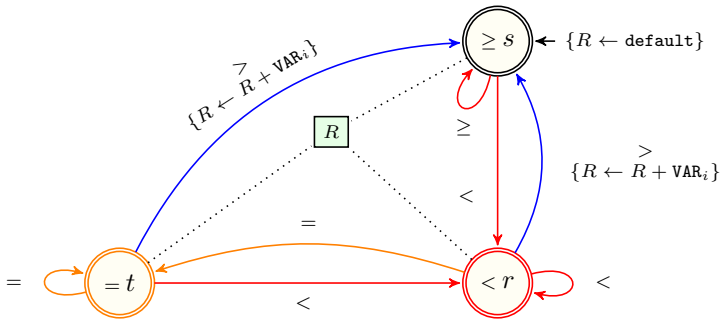


Figure 4.1241: Simplified automaton for the SUM_HEIGHT_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PLATEAU pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C

Table 4.271: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the SUM_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

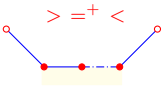
	s	r	t
s	0	0	0
r	0	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C
t	0	$\mathtt{VAR}_{i+1}$ C	$\mathtt{VAR}_{i+1}$ C

Table 4.272: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the SUM_HEIGHT_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_PROPER_PLAIN

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

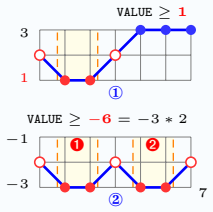
Constraint SUM_HEIGHT_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1, (\texttt{maxv} - 1) * \texttt{np})$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor (\texttt{sv} - 1)/3 \rfloor)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose

VALUE is the sum of all minimum values in each occurrence of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(12, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5 \rangle)$

Figure 4.1242 provides an example where the SUM_HEIGHT_PROPER_PLAIN $(12, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

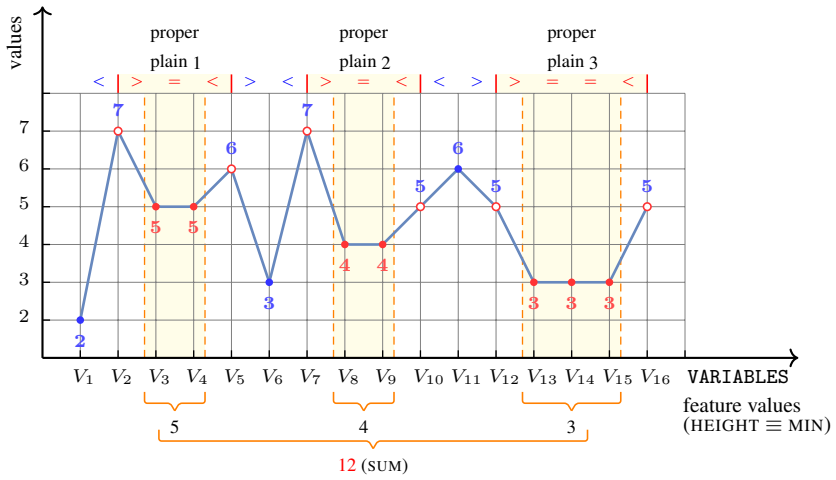


Figure 4.1242: Illustrating the SUM_HEIGHT_PROPER_PLAIN constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton Figures 4.1243 and 4.1244 respectively depict the automaton associated with the constraint SUM_HEIGHT_PROPER_PLAIN and its simplified form.

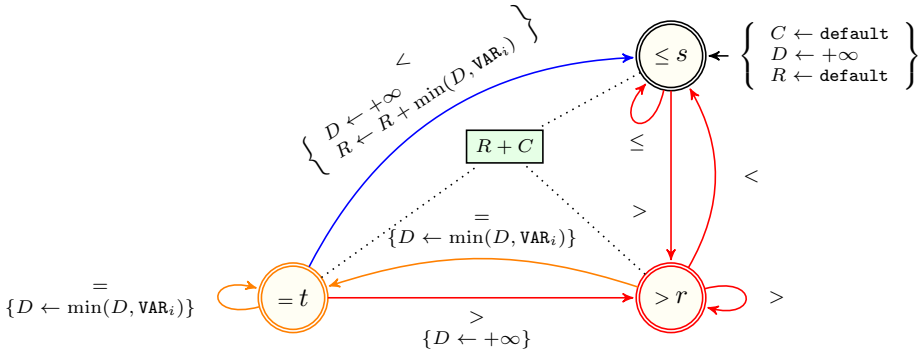


Figure 4.1243: Automaton for the SUM_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

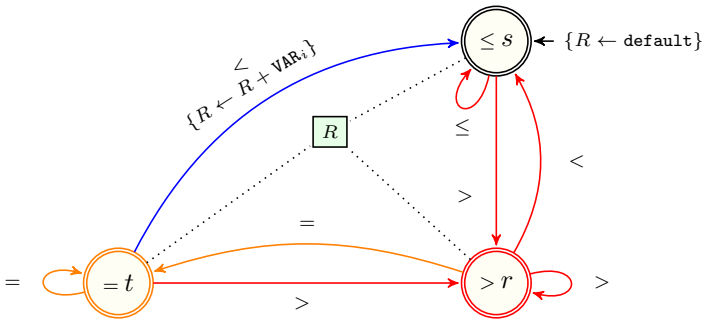


Figure 4.1244: Simplified automaton for the SUM_HEIGHT_PROPER_PLAIN constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.273: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the SUM_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

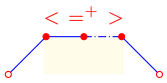
	s	r	t
s	0	0	0
r	0	0	$\texttt{VAR}_{i+1}$ C
t	0	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.274: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the SUM_HEIGHT_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

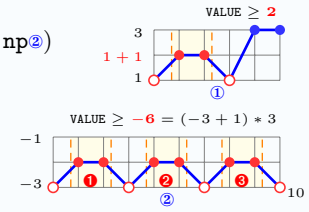
Constraint SUM_HEIGHT_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, \texttt{maxv} * \texttt{np})$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 1)/3\rfloor)$
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{rv} =$range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index $i+1$ to index j.

Example $(12, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3\rangle)$

Figure 4.1245 provides an example where the SUM_HEIGHT_PROPER_PLATEAU $(12, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetry Items of VARIABLES can be reversed.

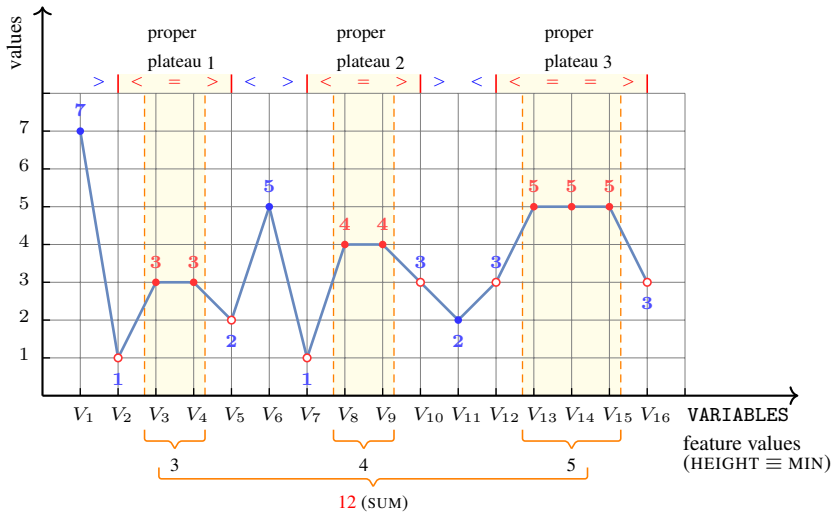


Figure 4.1245: Illustrating the SUM_HEIGHT_PROPER_PLATEAU constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1246 and 4.1247 respectively depict the automaton associated with the constraint SUM_HEIGHT_PROPER_PLATEAU and its simplified form.

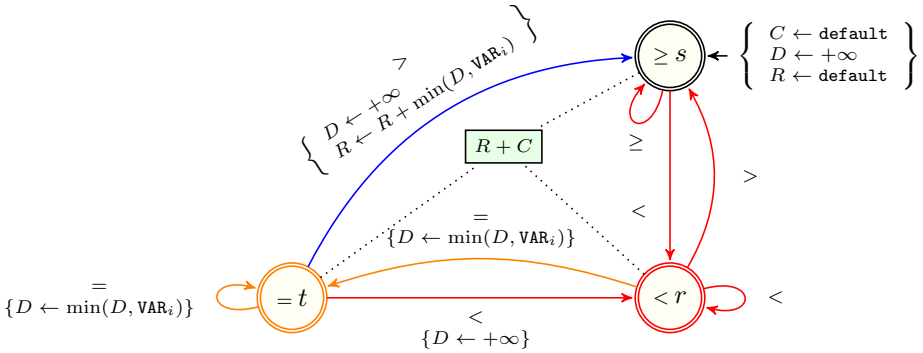


Figure 4.1246: Automaton for the SUM_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

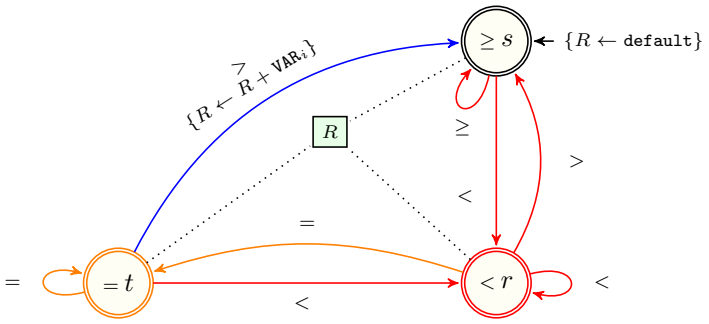


Figure 4.1247: Simplified automaton for the SUM_HEIGHT_PROPER_PLATEAU constraint obtained by applying decoration Table 3.39 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C

Table 4.275: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the SUM_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\texttt{VAR}_{i+1}$ C
t	0	$\texttt{VAR}_{i+1}$ C	$\texttt{VAR}_{i+1}$ C

Table 4.276: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the SUM_HEIGHT_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_STEADY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

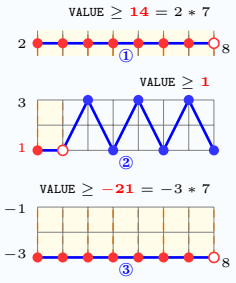
Constraint SUM_HEIGHT_STEADY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{minv} * \texttt{np}$①
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv}$②$, \texttt{minv} * \texttt{np}$③$)$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \leq \texttt{maxv} * \texttt{np}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(\texttt{maxv}, \texttt{maxv} * \texttt{np})$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)
$\texttt{np} = \max(0, \texttt{sv} - 1)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j + 1$.

Example $(30, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6 \rangle)$

Figure 4.1248 provides an example where the SUM_HEIGHT_STEADY $(30, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetry Items of VARIABLES can be reversed.

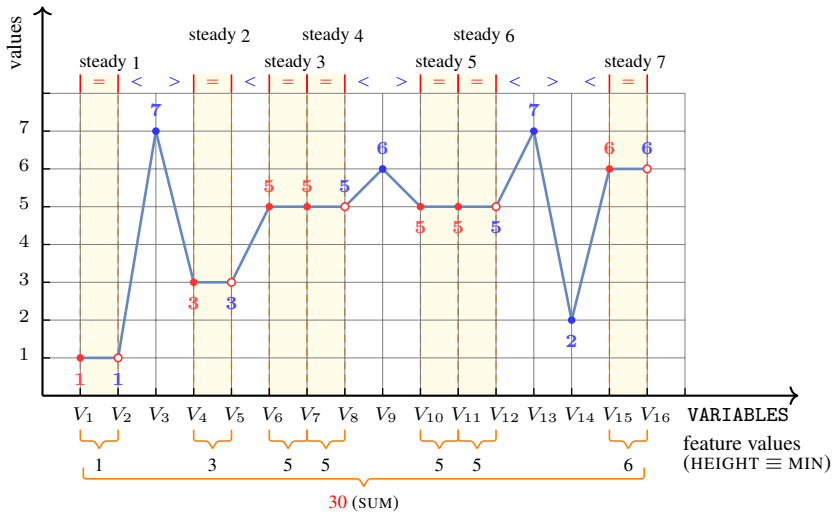


Figure 4.1248: Illustrating the SUM_HEIGHT_STEADY constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1249 and 4.1250 respectively depict the automaton associated with the constraint SUM_HEIGHT_STEADY and its simplified form.

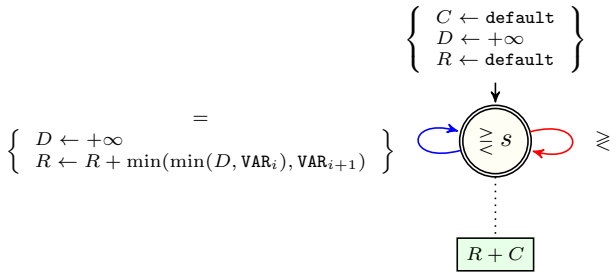


Figure 4.1249: Automaton for the SUM_HEIGHT_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is 0

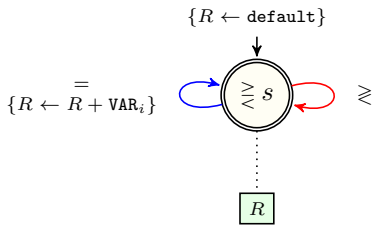


Figure 4.1250: Simplified automaton for the SUM_HEIGHT_STEADY constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.277: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the SUM_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

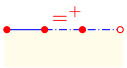
	s
s	0

Table 4.278: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the SUM_HEIGHT_STEADY constraint defined as the composition of the STEADY pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_HEIGHT_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint SUM_HEIGHT_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

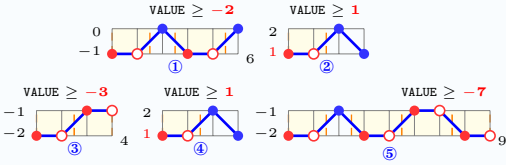
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$
$$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{minv}$$
$$\texttt{rv} \geq 2 \wedge \texttt{minv} = -1 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq -1 * \lfloor (\texttt{sv}-2)/3 \rfloor - 1 ①$$
$$\texttt{rv} \geq 2 \wedge \texttt{minv} \neq -1 \wedge \texttt{sv} \leq 4 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min(\texttt{minv} ②, \texttt{minv} * \lfloor (\texttt{np}+1)/2 \rfloor + (\texttt{minv}+1) * \lfloor \texttt{np}/2 \rfloor ③) \end{array} \right)$$
$$\texttt{rv} \geq 2 \wedge \texttt{minv} \neq -1 \wedge \texttt{sv} \geq 5 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min \left(\begin{array}{l} \texttt{minv} ④, \\ \texttt{minv} * \lfloor (\texttt{np}+1)/2 \rfloor + (\texttt{minv}+1) * \lfloor \texttt{np}/2 \rfloor - \\ (\texttt{np}+1) \bmod 2 * \texttt{sv} \bmod 2 \end{array} \right) ⑤ \end{array} \right)$$
$$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{maxv}$$
$$\texttt{rv} \geq 2 \wedge \texttt{maxv} = 1 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \lfloor (\texttt{sv}-2)/3 \rfloor + 1$$
$$\texttt{rv} \geq 2 \wedge \texttt{maxv} \neq 1 \wedge \texttt{sv} \leq 4 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max(\texttt{maxv}, \texttt{maxv} * \lfloor (\texttt{np}+1)/2 \rfloor + (\texttt{maxv}-1) * \lfloor \texttt{np}/2 \rfloor) \end{array} \right)$$
$$\texttt{rv} \geq 2 \wedge \texttt{maxv} \neq 1 \wedge \texttt{sv} \geq 5 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max \left(\begin{array}{l} \texttt{maxv}, \\ \sum \left(\begin{array}{l} \texttt{maxv} * \lfloor (\texttt{np}+1)/2 \rfloor, \\ (\texttt{maxv}-1) * \lfloor \texttt{np}/2 \rfloor, \\ (\texttt{np}+1) \bmod 2 * \texttt{sv} \bmod 2 \end{array} \right) \end{array} \right) \end{array} \right)$$
$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$



Purpose

VALUE is the sum of all minimum values in each occurrence of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^{+}$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature MIN, called HEIGHT in the name of the constraint since all feature values are identical, computes the minimum of the values from index i to index $j+1$.

Example

$(13, \langle 3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1\rangle)$

Figure 4.1251 provides an example where the SUM_HEIGHT_STEADY_SEQUENCE $(13, [3,1,1,4,5,5,5,6,2,2,4,4,3,2,1,1])$ constraint holds.

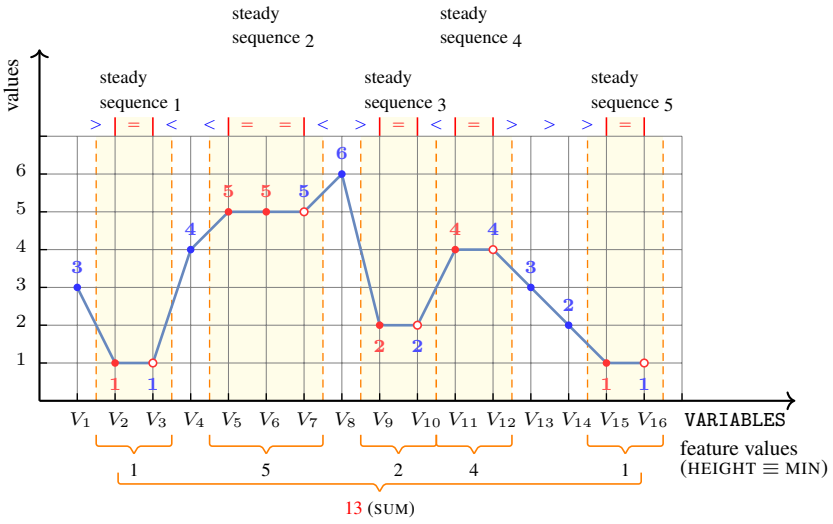


Figure 4.1251: Illustrating the SUM_HEIGHT_STEADY_SEQUENCE constraint of the **Example** slot

Typical

|VARIABLES| > 1

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1252 and 4.1253 respectively depict the automaton associated with the constraint SUM_HEIGHT_STEADY_SEQUENCE and its simplified form.

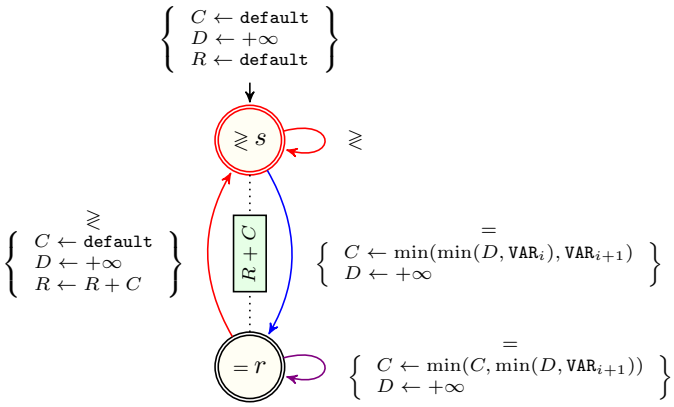


Figure 4.1252: Automaton for the SUM_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

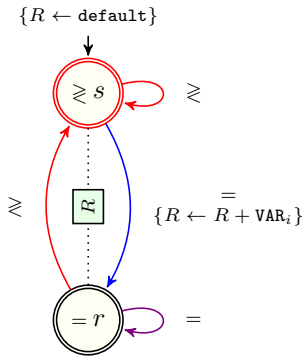


Figure 4.1253: Simplified automaton for the SUM_HEIGHT_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

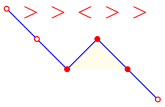
Table 4.279: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the SUM_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	$-\mathtt{VAR}_{i+1}$ M

Table 4.280: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the SUM_HEIGHT_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

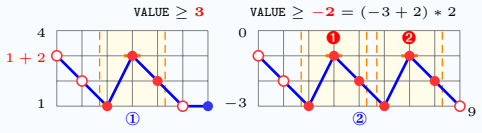
Constraint SUM_MAX_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv} + 2①, (\texttt{minv} + 2) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(\texttt{maxv}, \texttt{maxv} * \texttt{np})$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor(\texttt{sv} - 3)/3\rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all maximum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(11, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.1254 provides an example where the SUM_MAX_BUMP_ON_DECREASING_SEQUENCE $(11, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

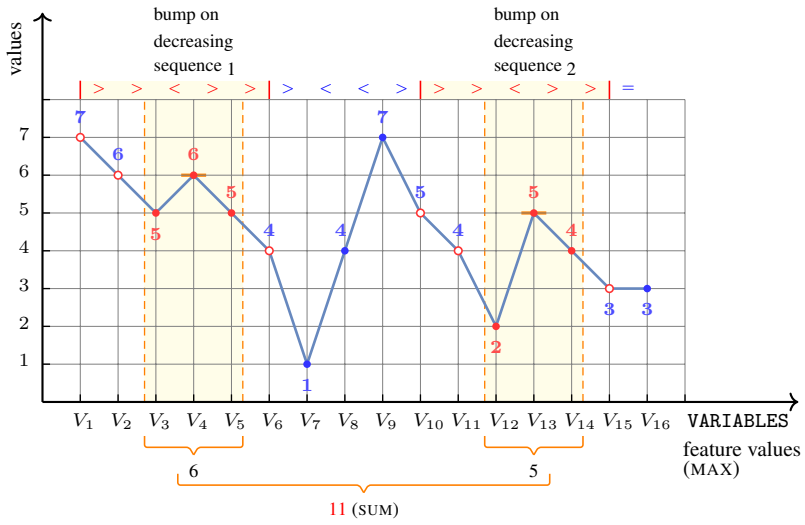


Figure 4.1254: Illustrating the SUM_MAX_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1255 and 4.1256 respectively depict the automaton associated with the constraint SUM_MAX_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

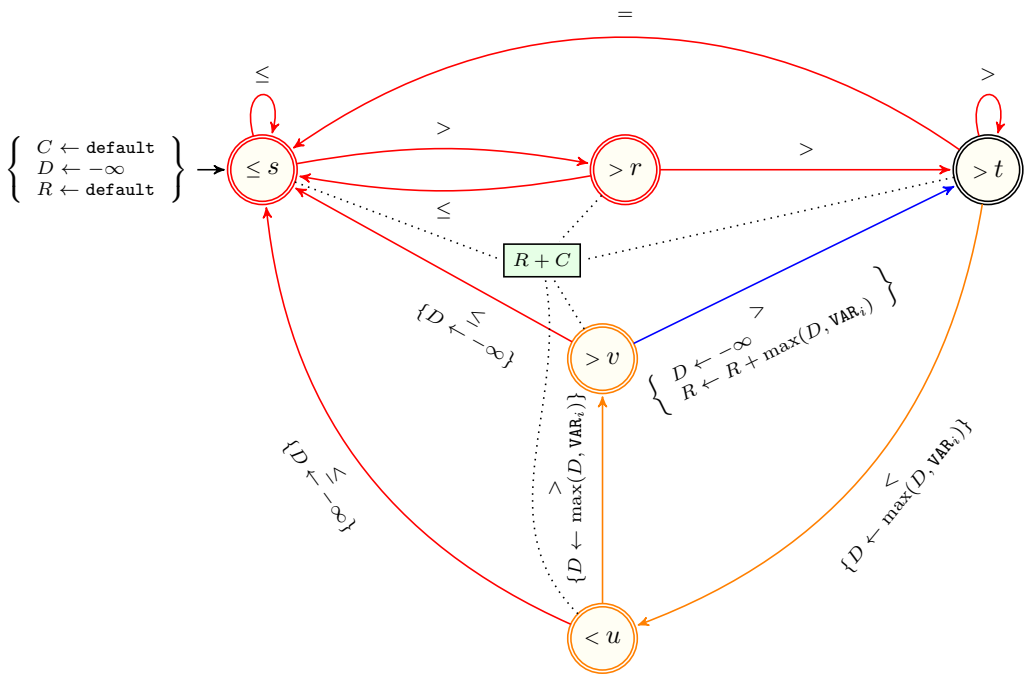


Figure 4.1255: Automaton for the SUM_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

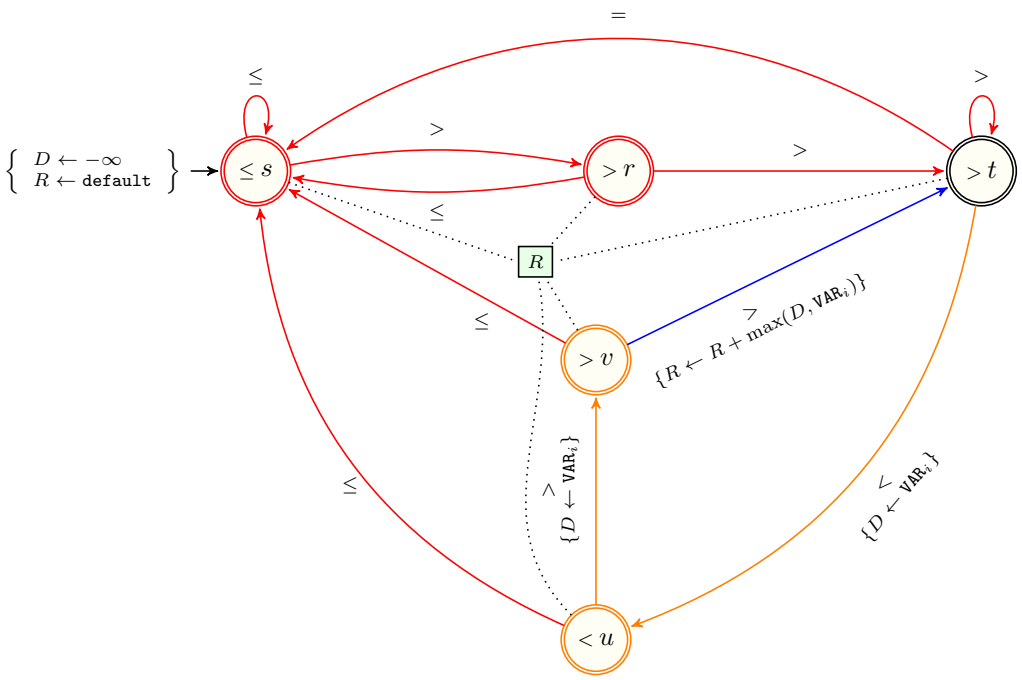


Figure 4.1256: Simplified automaton for the SUM_MAX_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_MAX_DECREASING

DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint SUM_MAX_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{minv}+1 \end{array} \right), \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) /2 \rfloor \end{array} \right) \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{minv}+1+\texttt{np}-1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{minv}+1, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\begin{array}{l} (\texttt{np}-1) * \texttt{maxv} - \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) /2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{maxv}-\texttt{np}+1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{maxv}, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right)$$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

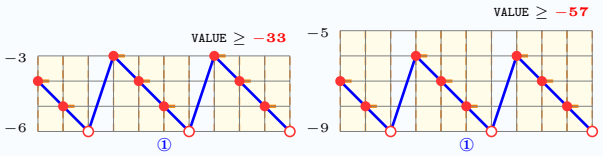
$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$$

Purpose

VALUE is the sum of all maximum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.

Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example

$(23, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle)$

Figure 4.1257 provides an example where the SUM_MAX_DECREASING $(23, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

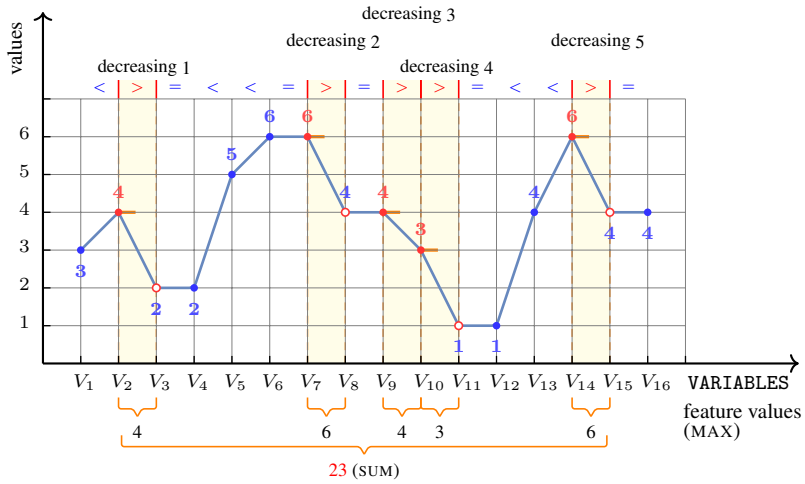


Figure 4.1257: Illustrating the SUM_MAX_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1258 and 4.1259 respectively depict the automaton associated with the constraint SUM_MAX_DECREASING and its simplified form.

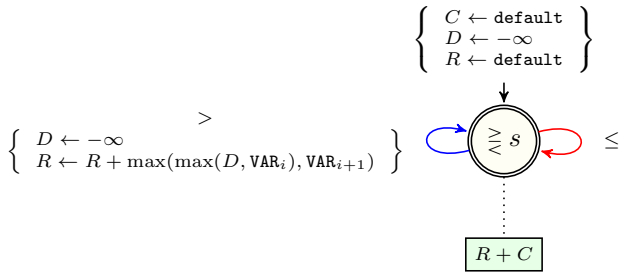


Figure 4.1258: Automaton for the SUM_MAX_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is 0

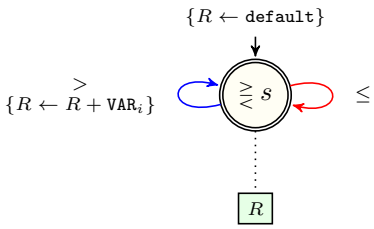


Figure 4.1259: Simplified automaton for the SUM_MAX_DECREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

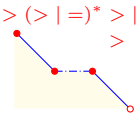
Table 4.281: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the SUM_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.282: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the SUM_MAX_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

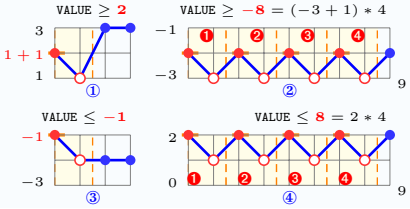
Constraint SUM_MAX_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all maximum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '> (> | =)* > | >'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(16, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1260 provides an example where the SUM_MAX_DECREASING_SEQUENCE $(16, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

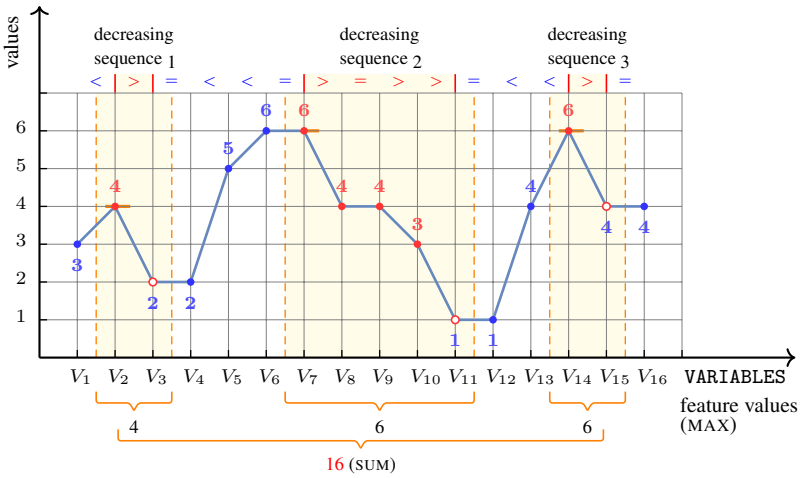


Figure 4.1260: Illustrating the SUM_MAX_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1261 and 4.1262 respectively depict the automaton associated with the constraint SUM_MAX_DECREASING_SEQUENCE and its simplified form.

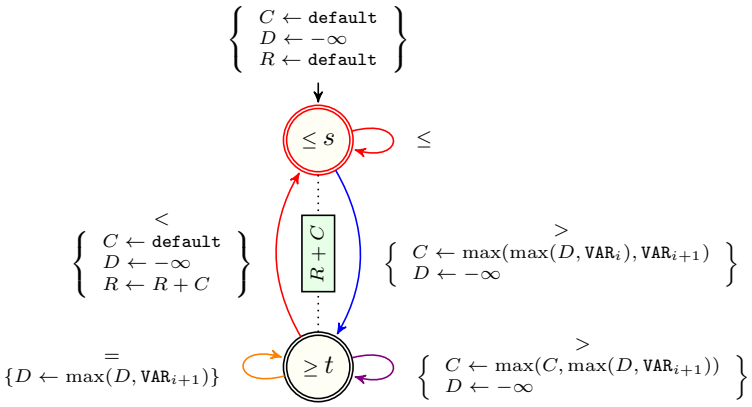


Figure 4.1261: Automaton for the SUM_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

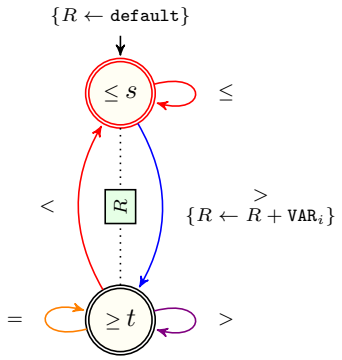


Figure 4.1262: Simplified automaton for the SUM_MAX_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

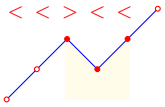
Table 4.283: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the SUM_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	0	$\overleftarrow{C}$
t	0	0 M

Table 4.284: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the SUM_MAX_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

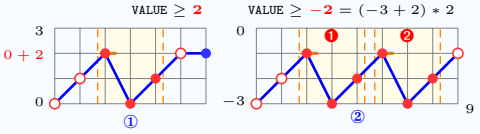
Constraint SUM_MAX_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 2①, (\texttt{minv} + 2) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, \texttt{maxv} * \texttt{np})$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor (\texttt{sv} - 3)/3 \rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all maximum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 2$ to index j.

Example $(11, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle)$

Figure 4.1263 provides an example where the SUM_MAX_DIP_ON_INCREASING_SEQUENCE $(11, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

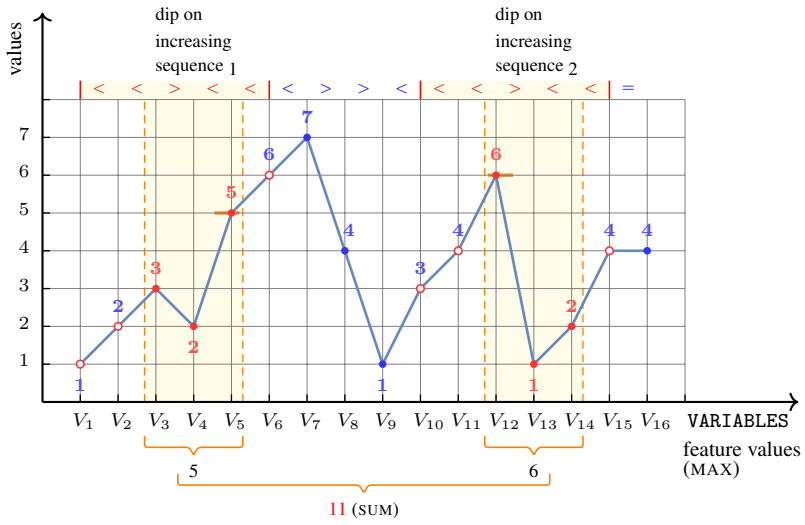


Figure 4.1263: Illustrating the SUM_MAX_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1264 and 4.1265 respectively depict the automaton associated with the constraint SUM_MAX_DIP_ON_INCREASING_SEQUENCE and its simplified form.

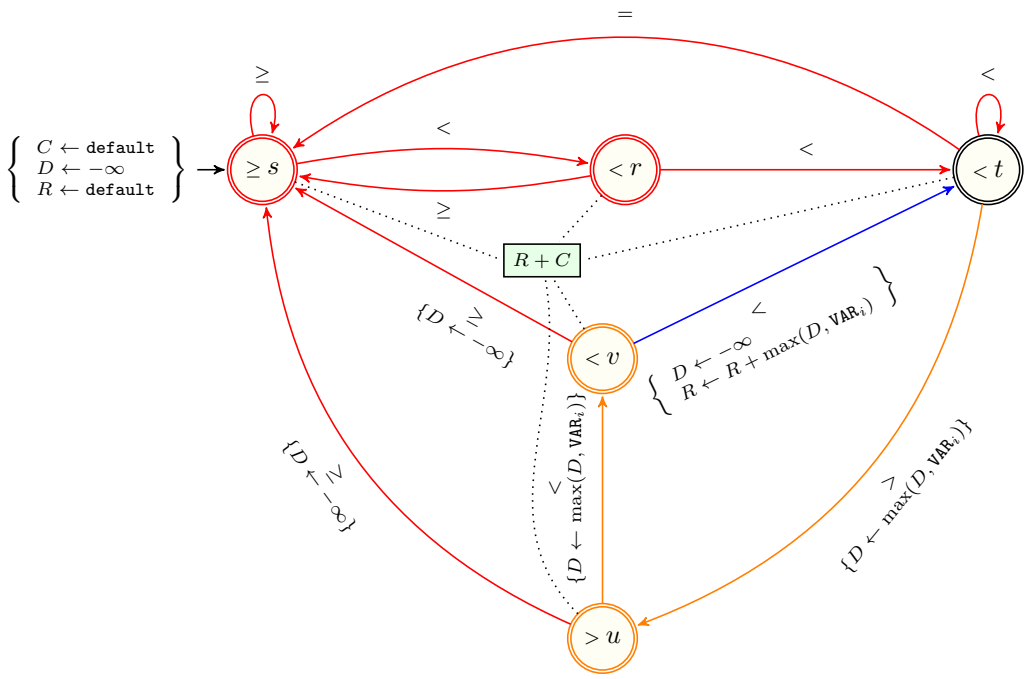


Figure 4.1264: Automaton for the SUM_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

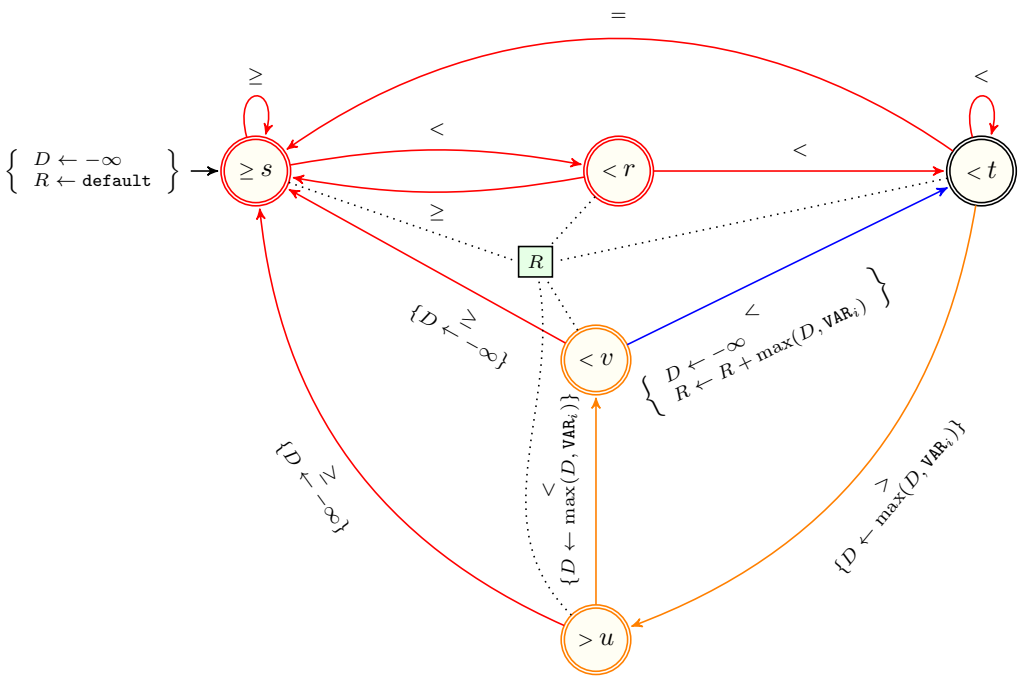


Figure 4.1265: Simplified automaton for the SUM_MAX_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_MAX_INCREASING

DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint SUM_MAX_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{minv}+1 \end{array} \right), \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) /2 \rfloor \end{array} \right) \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{minv}+1+\texttt{np}-1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{minv}+1, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\begin{array}{l} (\texttt{np}-1) * \texttt{maxv} - \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) /2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{maxv}-\texttt{np}+1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{maxv}, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right)$$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

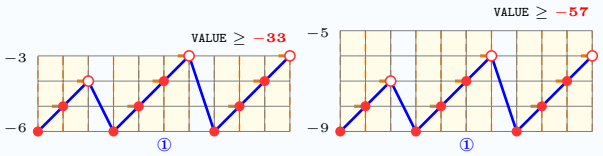
$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$$

Purpose

VALUE is the sum of all maximum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.

Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j+1$.

Example

$(21, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1266 provides an example where the SUM_MAX_INCREASING $(21, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

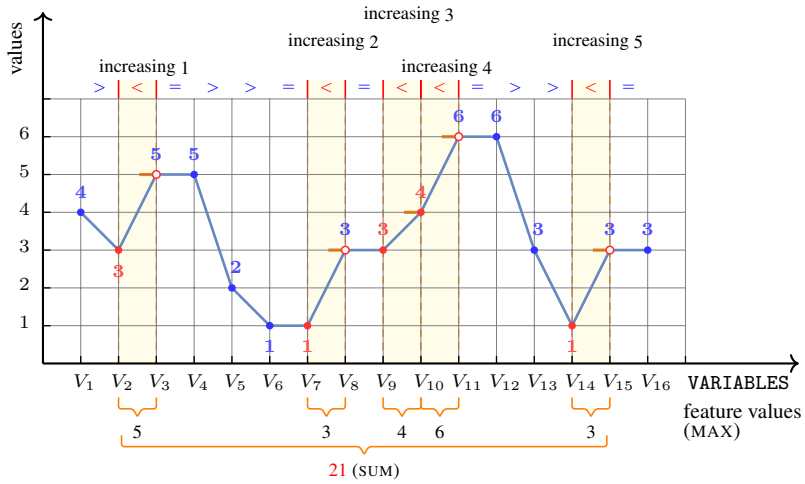


Figure 4.1266: Illustrating the SUM_MAX_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1267 and 4.1268 respectively depict the automaton associated with the constraint SUM_MAX_INCREASING and its simplified form.

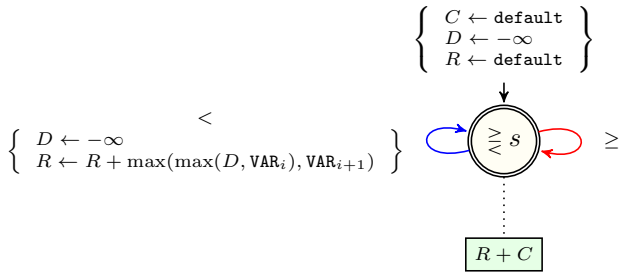


Figure 4.1267: Automaton for the SUM_MAX_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is 0

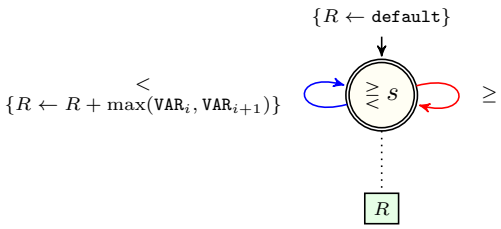


Figure 4.1268: Simplified automaton for the SUM_MAX_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

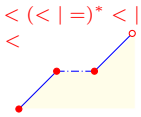
Table 4.285: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the SUM_MAX_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.286: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the SUM_MAX_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

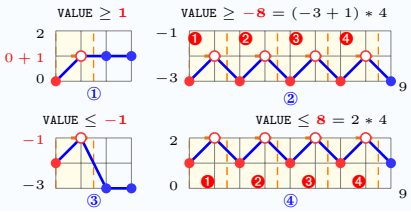
Constraint SUM_MAX_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all maximum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(14, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1269 provides an example where the SUM_MAX_INCREASING_SEQUENCE $(14, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

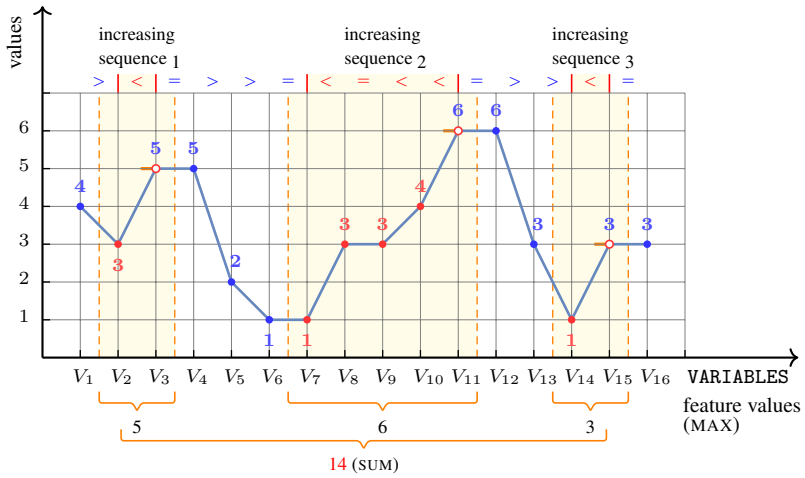


Figure 4.1269: Illustrating the SUM_MAX_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1270 and 4.1271 respectively depict the automaton associated with the constraint SUM_MAX_INCREASING_SEQUENCE and its simplified form.

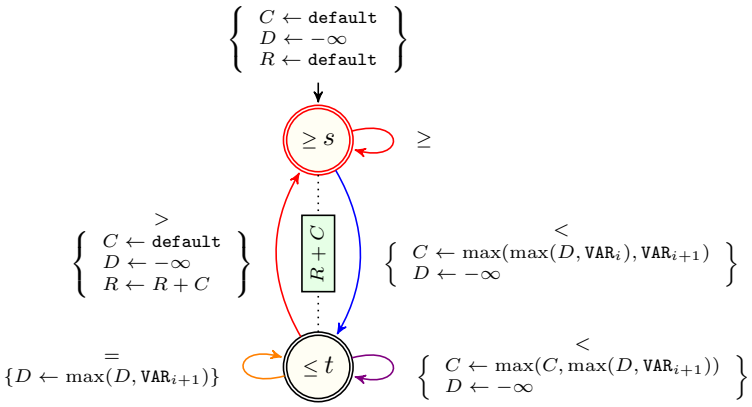


Figure 4.1270: Automaton for the SUM_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

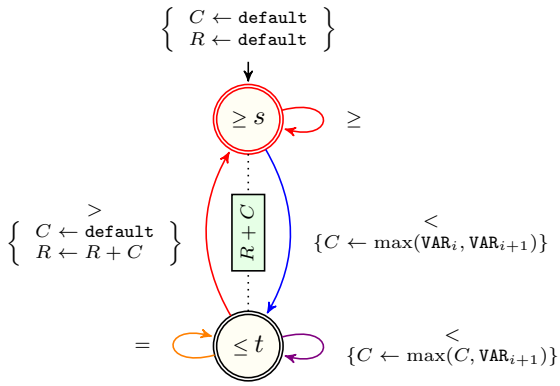


Figure 4.1271: Simplified automaton for the SUM_MAX_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.287: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the SUM_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

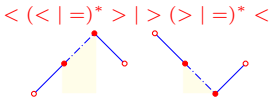
	s	t
s	0	0
t	$\overrightarrow{C}$	0 M

Table 4.288: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the SUM_MAX_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_INFLEXION

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint SUM_MAX_INFLEXION(VALUE, VARIABLES)

Arguments

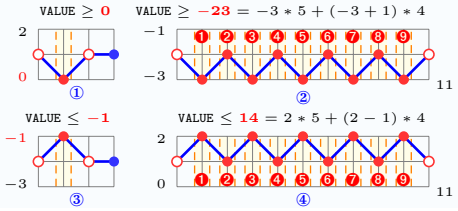
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\begin{array}{l} \mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = 0 \\ \bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \geq \mathtt{min}(\mathtt{minv}①, \mathtt{minv} * \lfloor (\mathtt{np}+1)/2 \rfloor + (\mathtt{minv}+1) * \lfloor \mathtt{np}/2 \rfloor②) \end{array} \right) \\ \bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \leq \mathtt{max}(\mathtt{maxv}③, \mathtt{maxv} * \lfloor (\mathtt{np}+1)/2 \rfloor + (\mathtt{maxv}-1) * \lfloor \mathtt{np}/2 \rfloor④) \end{array} \right) \\ \mathtt{required}(\mathtt{VARIABLES}, \mathtt{var}) \end{array}$$

where

$\mathtt{sv} = |\mathtt{VARIABLES}|$
$\mathtt{np} = \mathtt{max}(0, \mathtt{sv} - 2)$
$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$
$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$
$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose VALUE is the sum of all maximum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(31, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.1272 provides an example where the SUM_MAX_INFLEXION $(31, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical

$|\mathtt{VARIABLES}| > 2$
$\mathtt{range}(\mathtt{VARIABLES.var}) > 1$

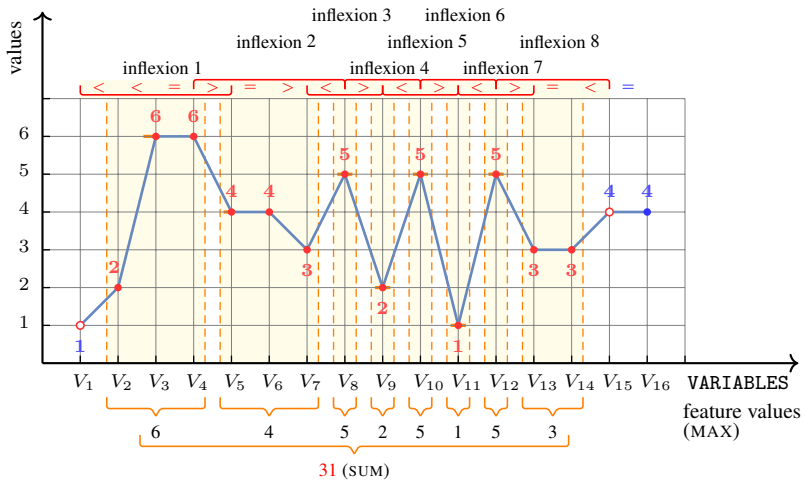


Figure 4.1272: Illustrating the SUM_MAX_INFLEXION constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1273 and 4.1274 respectively depict the automaton associated with the constraint SUM_MAX_INFLEXION and its simplified form.

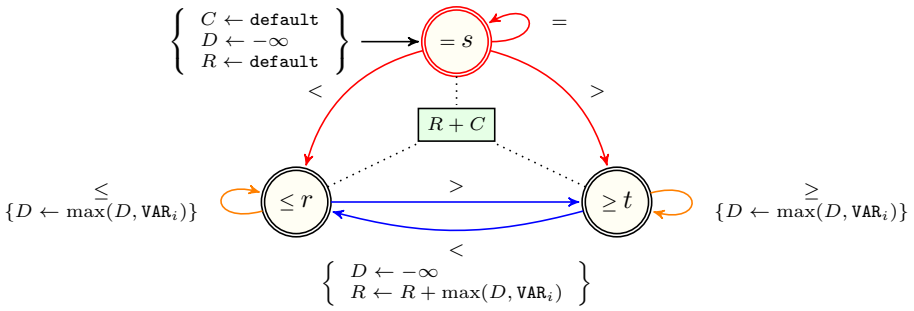


Figure 4.1273: Automaton for the SUM_MAX_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

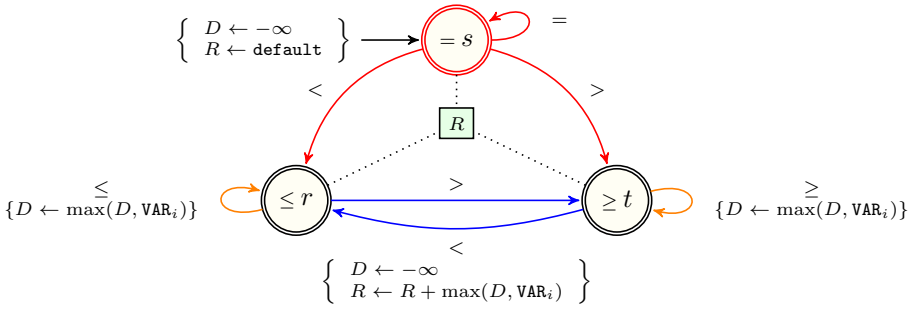
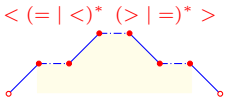


Figure 4.1274: Simplified automaton for the SUM_MAX_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

AGGREGATOR FEATURE PATTERN

SUM_MAX_PEAK

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

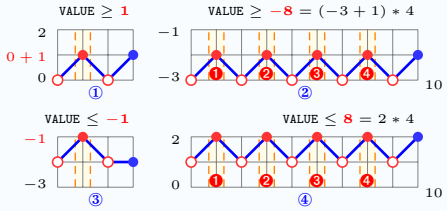
Constraint SUM_MAX_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all maximum values in each occurrence of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i + 1$ to index j.

Example $(14, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1\rangle)$

Figure 4.1275 provides an example where the SUM_MAX_PEAK $(14, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

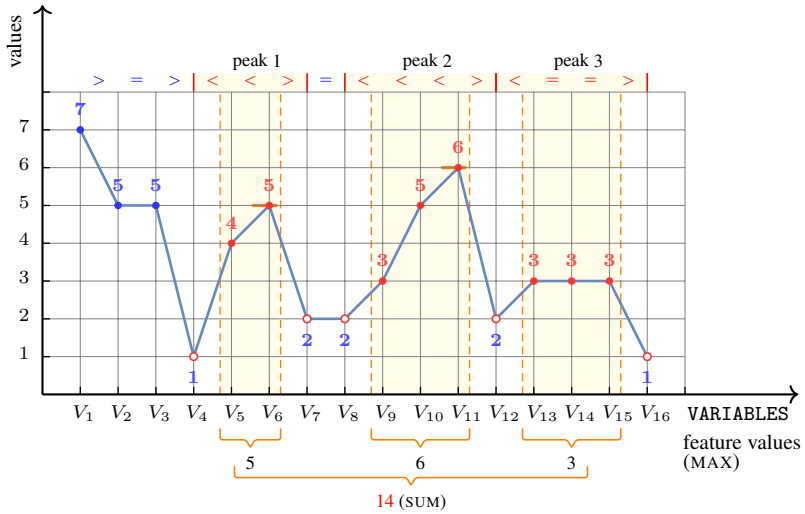


Figure 4.1275: Illustrating the SUM_MAX_PEAK constraint of the **Example** slot

Automaton

Figures 4.1276 and 4.1277 respectively depict the automaton associated with the constraint SUM_MAX_PEAK and its simplified form.

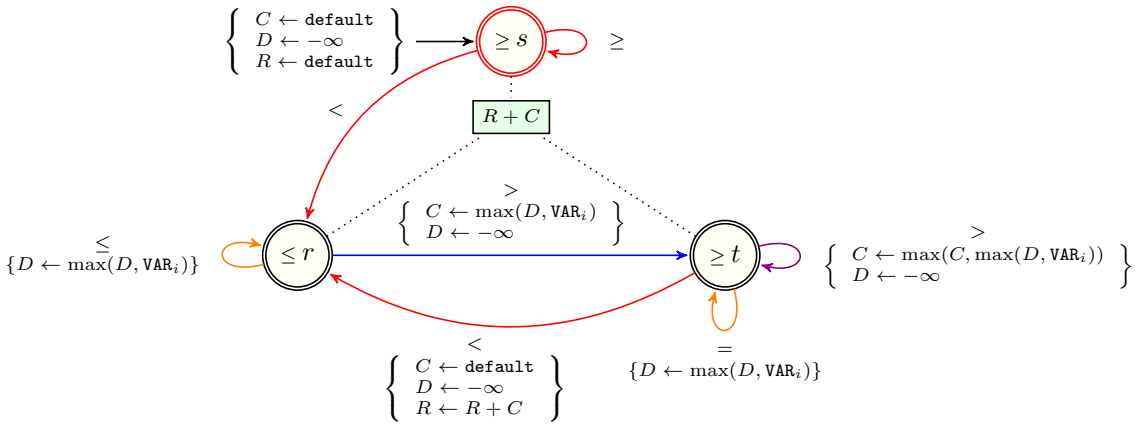


Figure 4.1276: Automaton for the SUM_MAX_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is 0

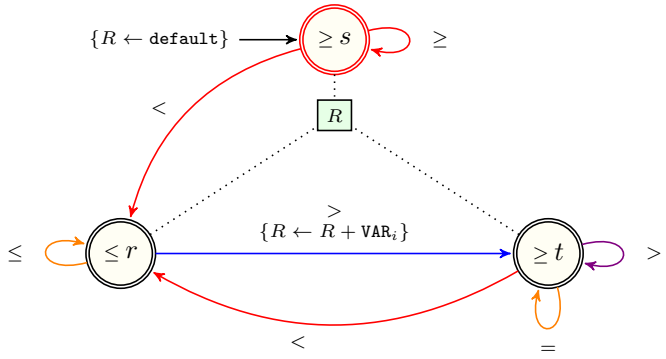


Figure 4.1277: Simplified automaton for the SUM_MAX_PEAK constraint obtained by applying decoration Table 3.39 to the seed transducer of the PEAK pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.289: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the SUM_MAX_PEAK constraint defined as the composition of the PEAK pattern , the feature MAX , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\texttt{VAR}_{i+1}$ C	0 R
t	0	0 L	0

Table 4.290: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the SUM_MAX_PEAK constraint defined as the composition of the PEAK pattern , the feature MAX , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint SUM_MAX_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$

$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$

$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$

required(VARIABLES, var)

where

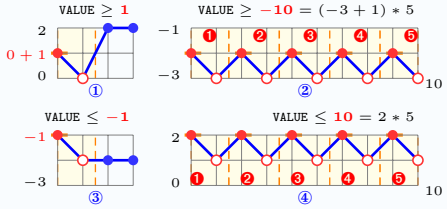
$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$

minv =minval(VARIABLES.var)

maxv =maxval(VARIABLES.var)

rv =range(VARIABLES.var)



Purpose VALUE is the sum of all maximum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.

Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(16, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1278 provides an example where the SUM_MAX_STRICTLY_DECREASING_SEQUENCE $(16, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$

$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

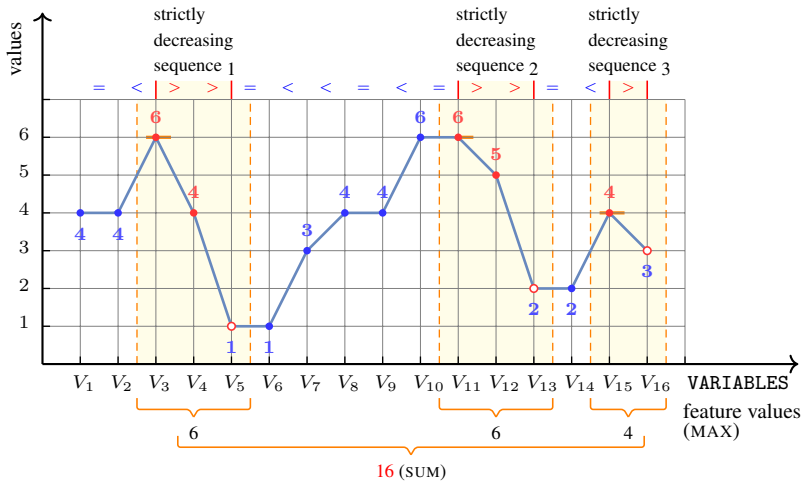


Figure 4.1278: Illustrating the SUM_MAX_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1279 and 4.1280 respectively depict the automaton associated with the constraint SUM_MAX_STRICTLY_DECREASING_SEQUENCE and its simplified form.

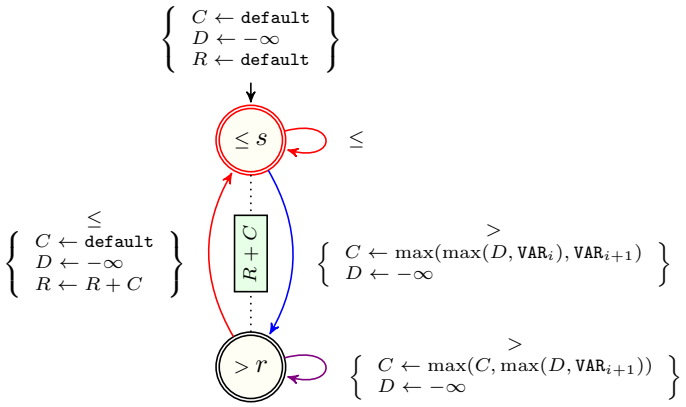


Figure 4.1279: Automaton for the SUM_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

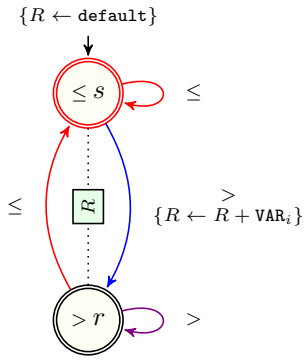


Figure 4.1280: Simplified automaton for the SUM_MAX_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.291: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the SUM_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	$\overleftarrow{C}$
r	0	0 M

Table 4.292: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the SUM_MAX_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

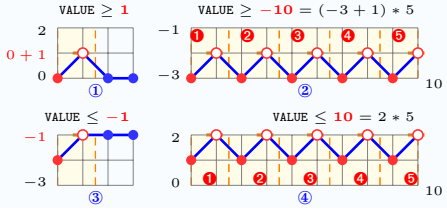
Constraint SUM_MAX_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all maximum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MAX computes the maximum of the values from index i to index $j + 1$.

Example $(14, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1281 provides an example where the SUM_MAX_STRICTLY_INCREASING_SEQUENCE $(14, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

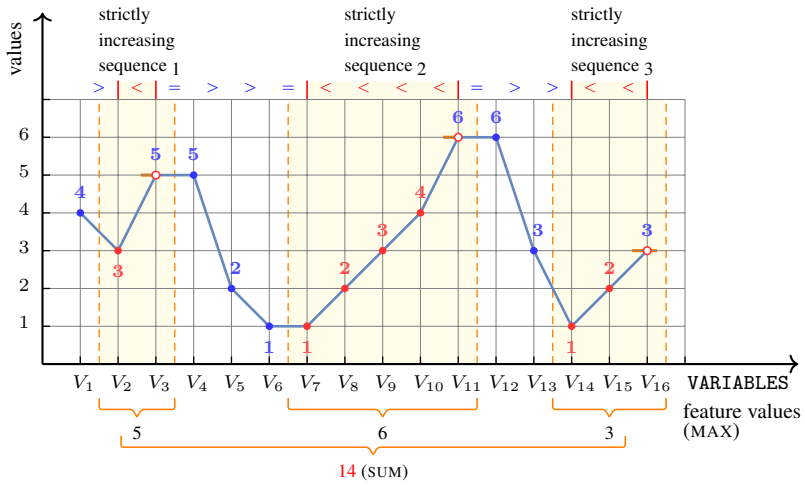


Figure 4.1281: Illustrating the SUM_MAX_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1282 and 4.1283 respectively depict the automaton associated with the constraint SUM_MAX_STRICTLY_INCREASING_SEQUENCE and its simplified form.

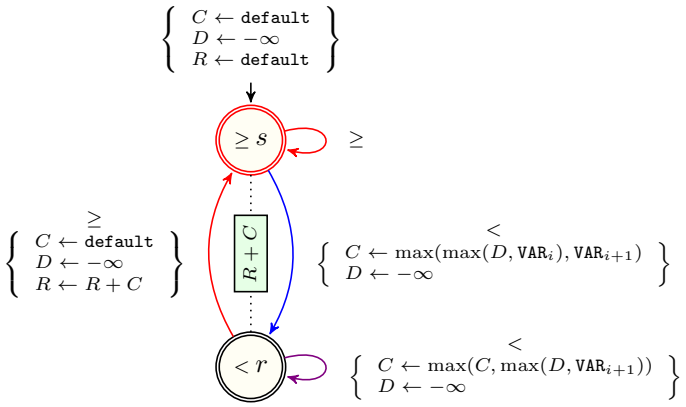


Figure 4.1282: Automaton for the SUM_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

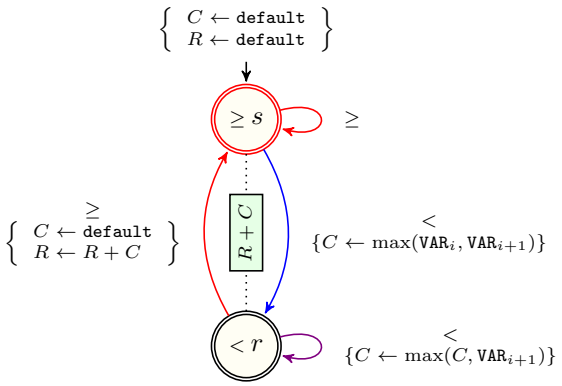


Figure 4.1283: Simplified automaton for the SUM_MAX_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.293: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the SUM_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

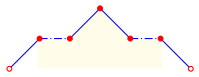
	s	r
s	0	0
r	$\overrightarrow{C}$	0 M

Table 4.294: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the SUM_MAX_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

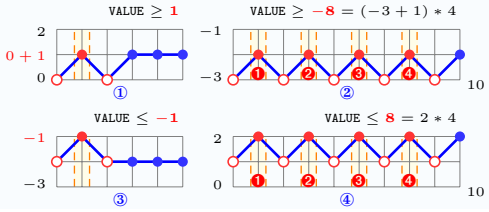
Constraint SUM_MAX_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor (\texttt{sv} - 1)/2 \rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all maximum values in each occurrence of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(12, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.1284 provides an example where the SUM_MAX_SUMMIT $(12, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

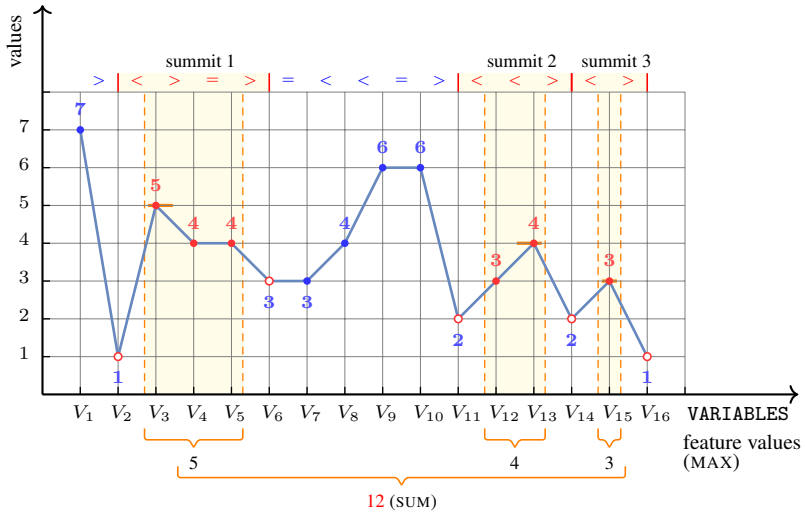


Figure 4.1284: Illustrating the SUM_MAX_SUMMIT constraint of the **Example** slot

Automaton Figures 4.1285 and 4.1286 respectively depict the automaton associated with the constraint SUM_MAX_SUMMIT and its simplified form.

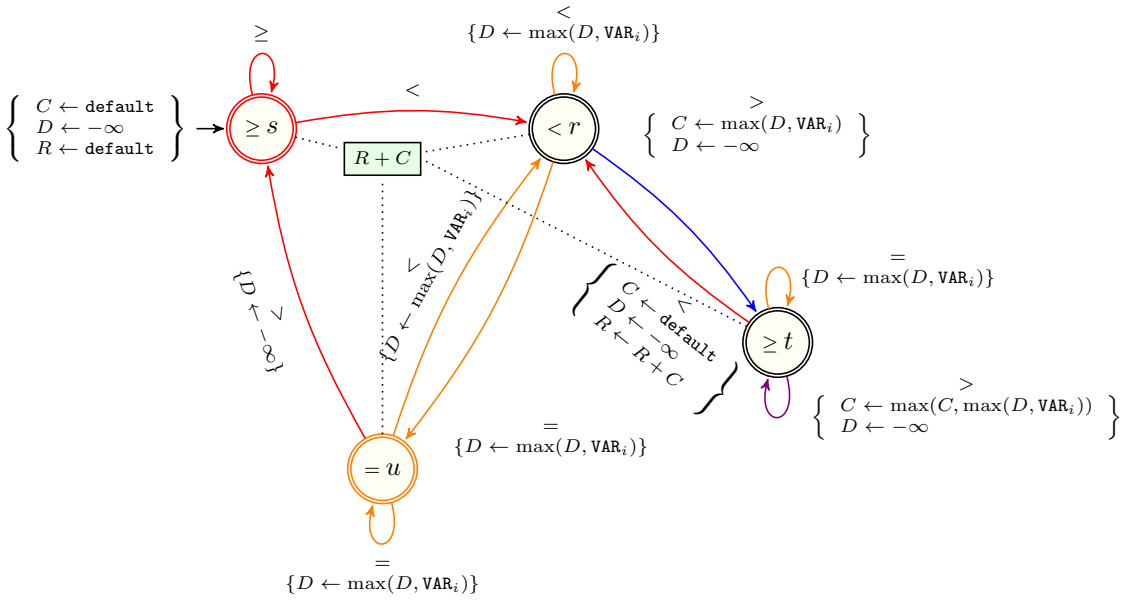


Figure 4.1285: Automaton for the SUM_MAX_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

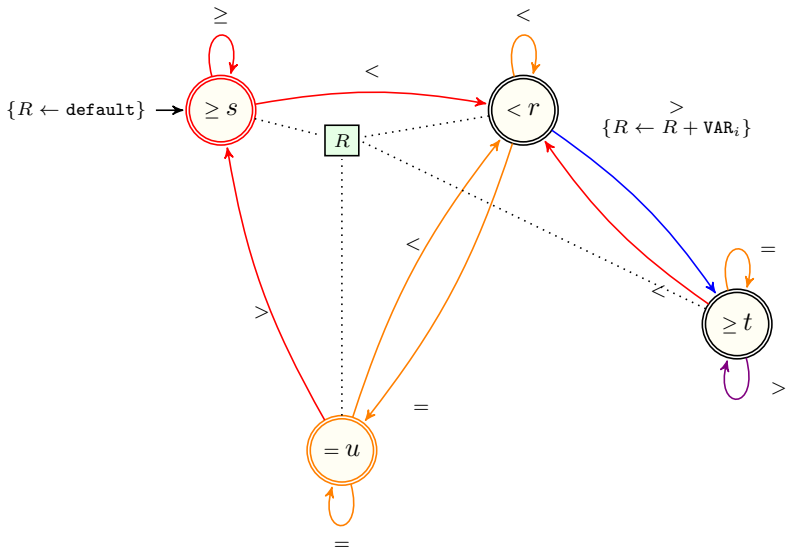


Figure 4.1286: Simplified automaton for the SUM_MAX_SUMMIT constraint obtained by applying decoration Table 3.39 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \to r$ has the same register update as transition $r \to u$)

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\max(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.295: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the SUM_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

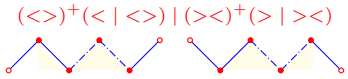
	s	r	t	u
s	0	0	0	0
r	0	$\texttt{VAR}_{i+1}$ C	0 R	0
t	0	0 L	0	0 L
u	0	0	0 R	0

Table 4.296: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the SUM_MAX_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MAX_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint SUM_MAX_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [0,\texttt{qub}]} \texttt{min} \left(0, \sum \left(\prod \begin{pmatrix} \lfloor (\texttt{sv} - 3 * \texttt{q})/4 \rfloor, \\ \texttt{minv} + 1 \end{pmatrix}, \begin{matrix} \texttt{q} * (\texttt{minv} + 2), \\ \prod \begin{pmatrix} \texttt{min}\left(1, \texttt{max}\left(\texttt{maxv} - \texttt{minv} - 1, 0 \right) \right), \\ \texttt{max}\left(-1, \texttt{min}\left(0, \texttt{q} * (\texttt{minv} + 2) \right) \right), \\ \texttt{min}\left(\texttt{q}, \left((\texttt{sv} - \lfloor (\texttt{sv} - 3 * \texttt{q})/4 \rfloor) * 4 - \\ \texttt{q} * 3 \end{pmatrix} \bmod 3 \right) \end{pmatrix} \end{matrix} \right) \right)$$

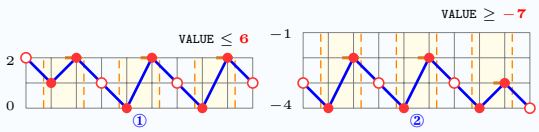
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, \texttt{maxv} * \texttt{np1})$$

$$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}, \texttt{max}(\texttt{maxv} * \texttt{np1}, \texttt{maxv} * \texttt{np2}))$$

required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{np1} = \lfloor \texttt{sv}/4 \rfloor$
$\texttt{np2} = \texttt{max}(0, \lfloor (\texttt{sv} - 1)/3 \rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{qub} = \texttt{min}(1, \texttt{max}(\texttt{maxv} - \texttt{minv} - 1, 0)) * \texttt{np2}$



Purpose VALUE is the sum of all maximum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MAX computes the maximum of the values from index $i+1$ to index j.

Example $(16, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.1287 provides an example where the SUM_MAX_ZIGZAG $(16, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

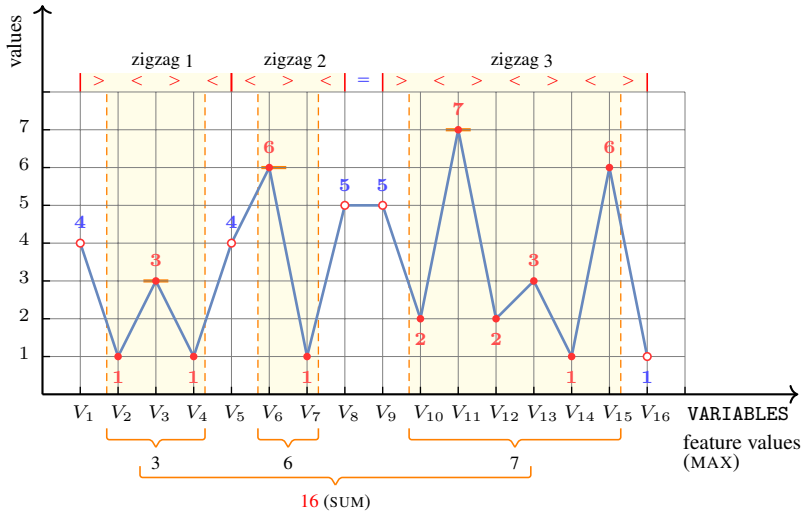


Figure 4.1287: Illustrating the SUM_MAX_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1288 and 4.1289 respectively depict the automaton associated with the constraint SUM_MAX_ZIGZAG and its simplified form.

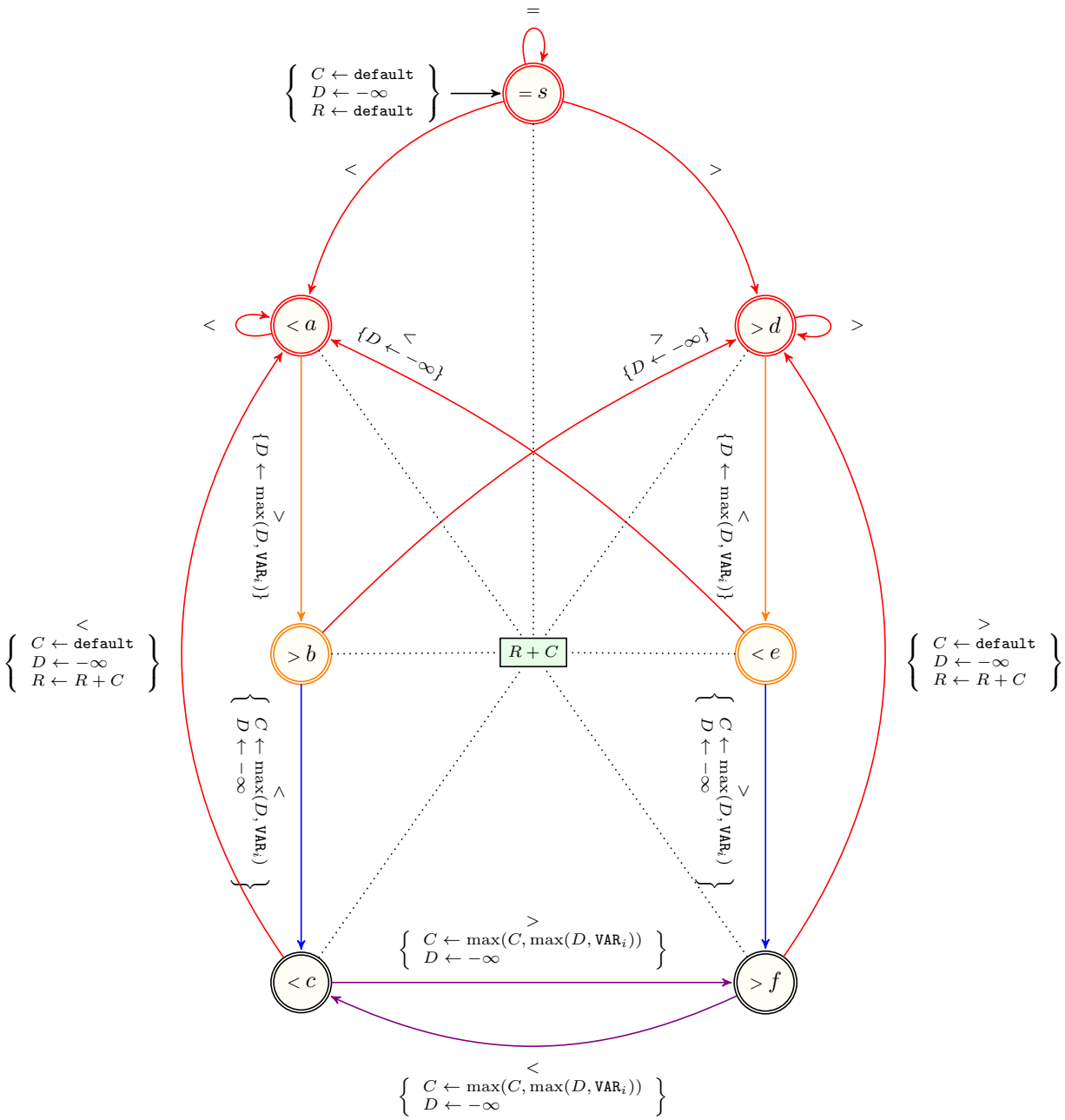


Figure 4.1288: Automaton for the SUM_MAX_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

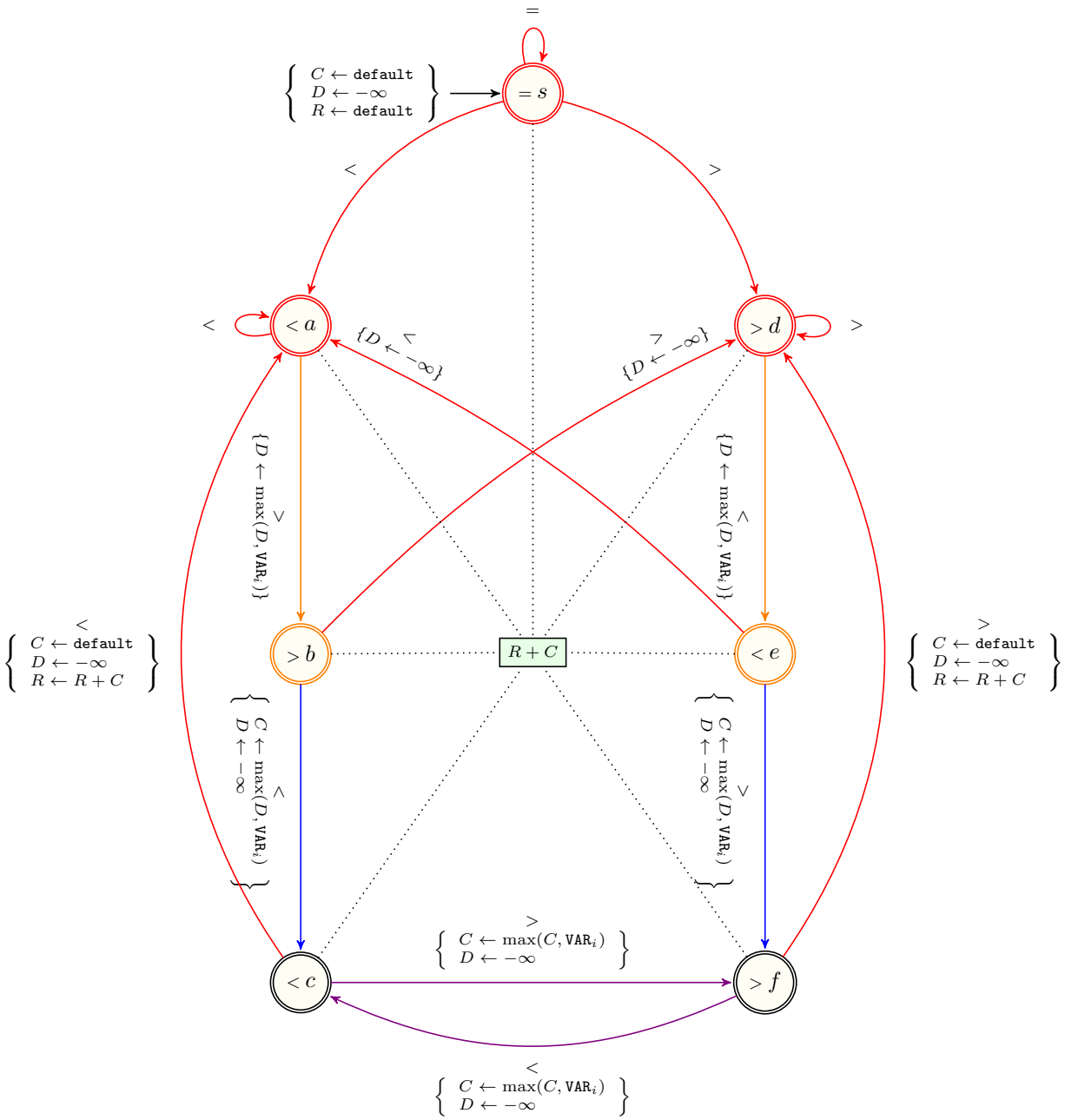


Figure 4.1289: Simplified automaton for the SUM_MAX_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is 0; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.

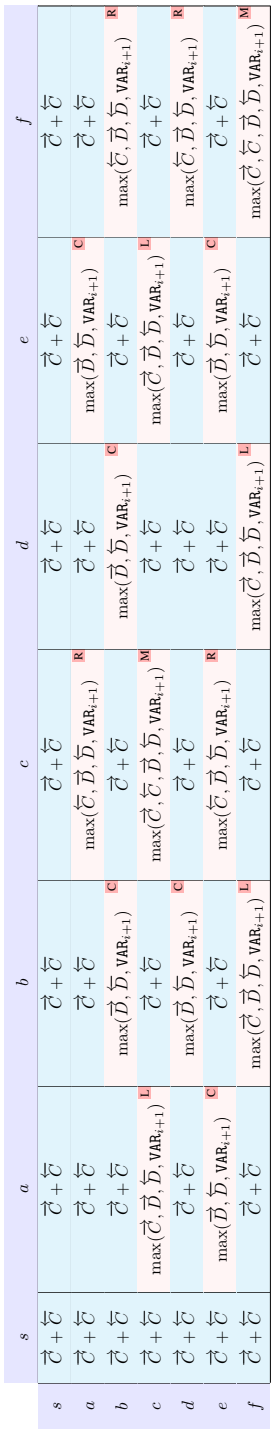


Table 4.297: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the SUM_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MAX, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

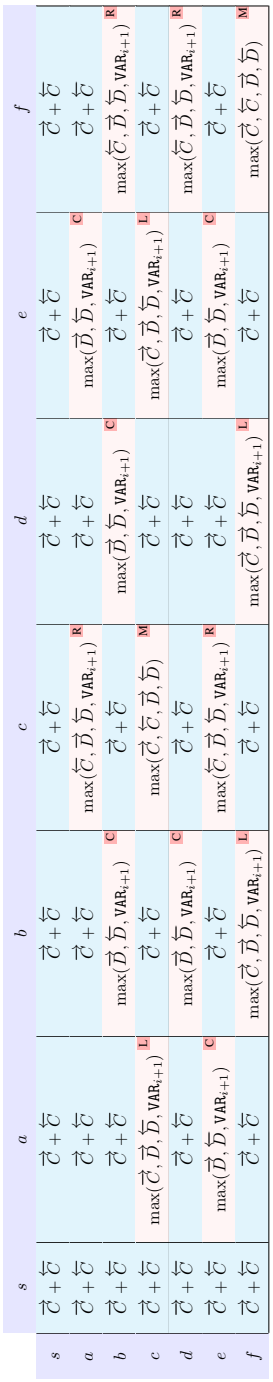
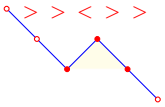


Table 4.298: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the SUM_MAX_ZIGZAG constraint defined as the composition of the ZIGZAG pattern , the feature MAX , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_BUMP_ON_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint SUM_MIN_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

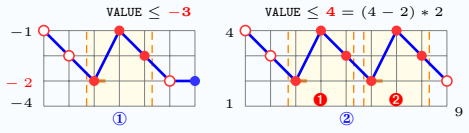
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}, \texttt{minv} * \texttt{np})$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 2①, (\texttt{maxv} - 2) * \texttt{np}②)$
required(VARIABLES, var)

where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 3)/3\rfloor)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(7, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3\rangle)$

Figure 4.1290 provides an example where the SUM_MIN_BUMP_ON_DECREASING_SEQUENCE $(7, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

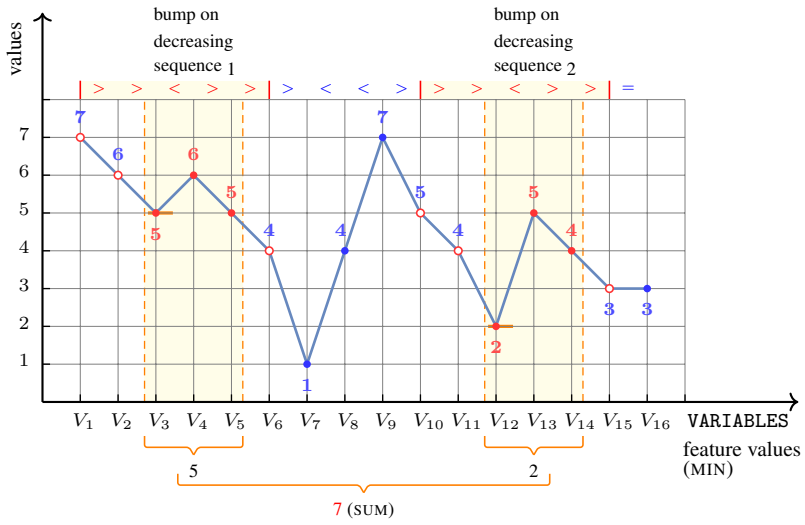


Figure 4.1290: Illustrating the SUM_MIN_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.1291 and 4.1292 respectively depict the automaton associated with the constraint SUM_MIN_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

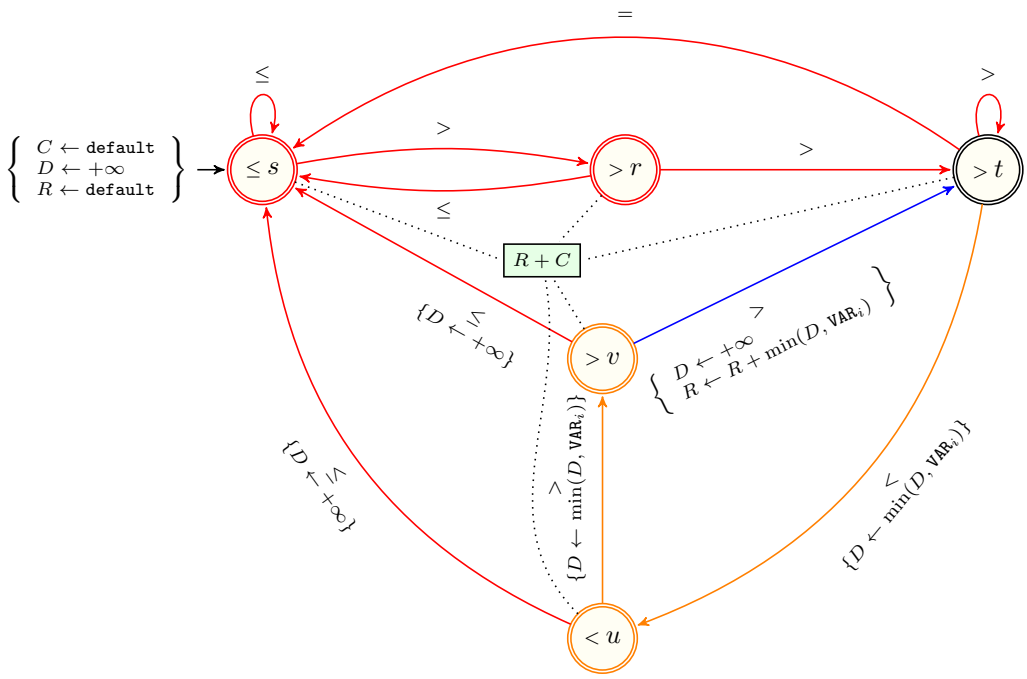


Figure 4.1291: Automaton for the SUM_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

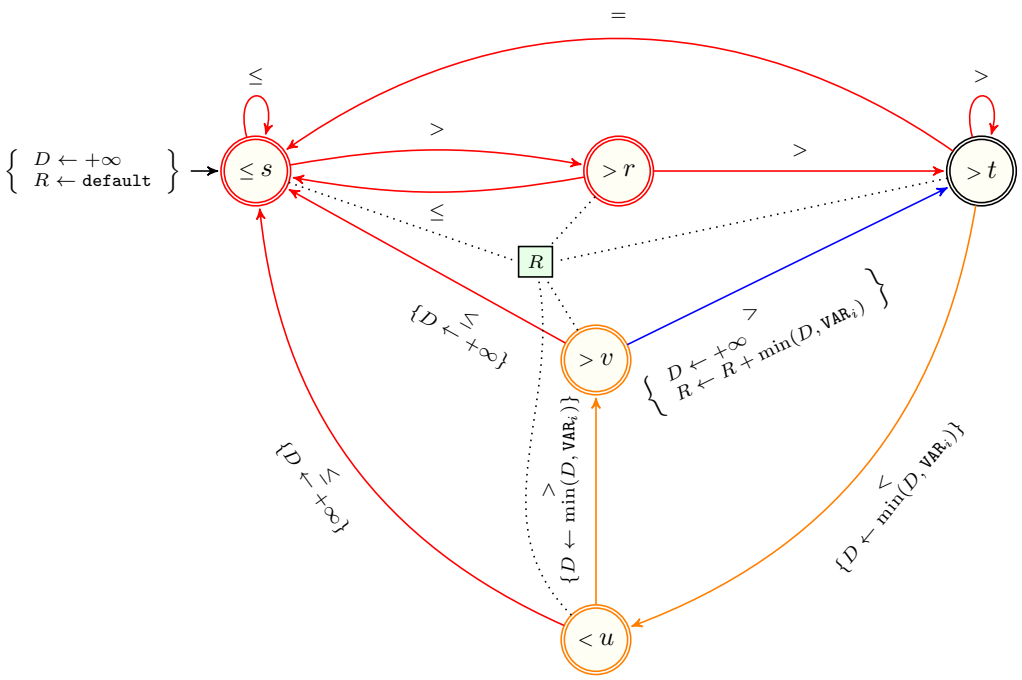


Figure 4.1292: Simplified automaton for the SUM_MIN_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_MIN_DECREASING

DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint SUM_MIN_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\sum \left(\begin{array}{l} (\texttt{np}-1) * \texttt{minv}, \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) / 2 \rfloor \end{array} \right) \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{minv} + \texttt{np} - 1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{minv}, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{maxv}-1 \end{array} \right) - \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) / 2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{maxv}-1-\texttt{np}+1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{maxv}-1, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right)$$

required(VARIABLES, var)

where

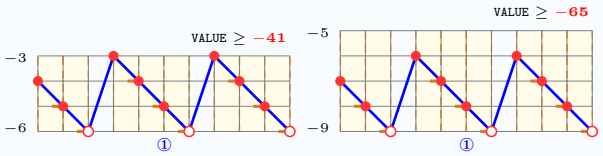
$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1)) \end{array} \right) \end{array} \right)$$

$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1)) \end{array} \right) \end{array} \right)$$

$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$

Purpose

VALUE is the sum of all minimum values in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.

Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$(14, \langle 3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4 \rangle)$

Figure 4.1293 provides an example where the SUM_MIN_DECREASING $(14, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

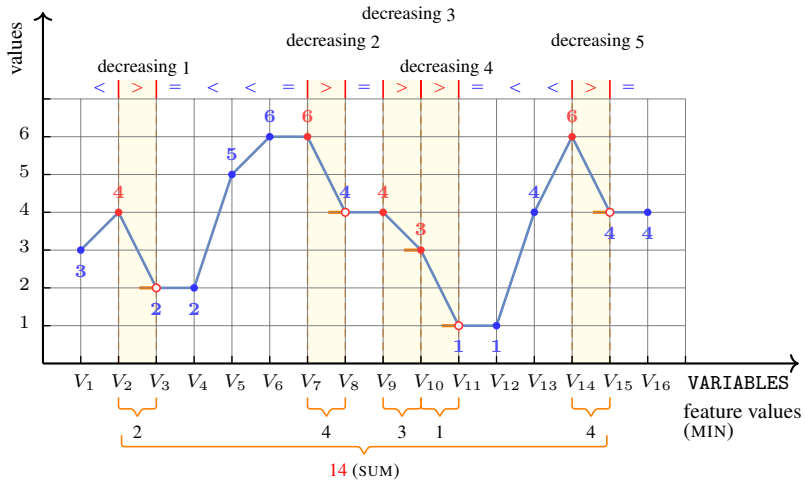


Figure 4.1293: Illustrating the SUM_MIN_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1294 and 4.1295 respectively depict the automaton associated with the constraint SUM_MIN_DECREASING and its simplified form.

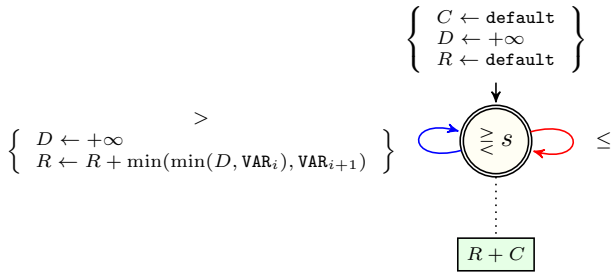


Figure 4.1294: Automaton for the SUM_MIN_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is 0

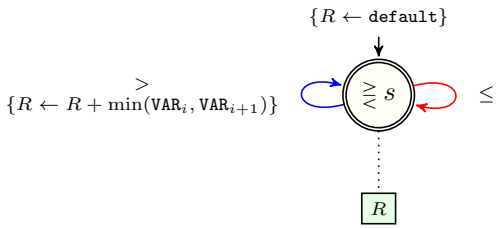


Figure 4.1295: Simplified automaton for the SUM_MIN_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

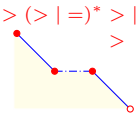
Table 4.299: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the SUM_MIN_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.300: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the SUM_MIN_DECREASING constraint defined as the composition of the DECREASING pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

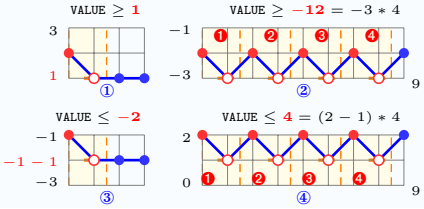
Constraint SUM_MIN_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '> (> | =)* > | >'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(7, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1296 provides an example where the SUM_MIN_DECREASING_SEQUENCE $(7, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

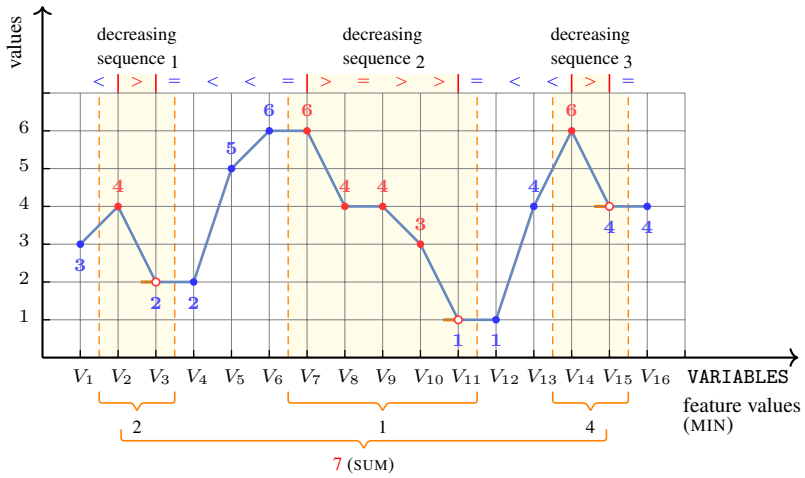


Figure 4.1296: Illustrating the SUM_MIN_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1297 and 4.1298 respectively depict the automaton associated with the constraint SUM_MIN_DECREASING_SEQUENCE and its simplified form.

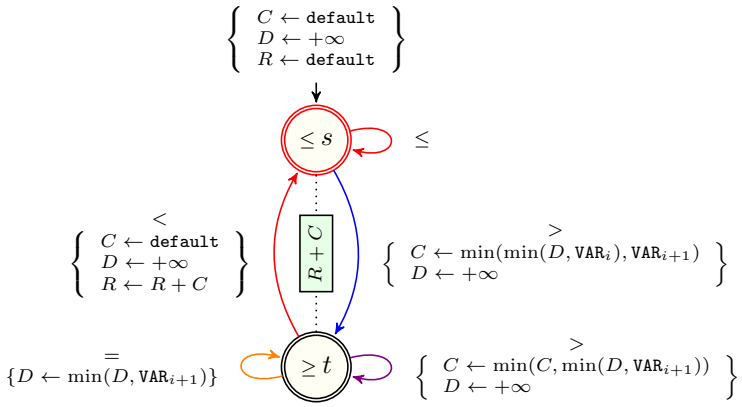


Figure 4.1297: Automaton for the SUM_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

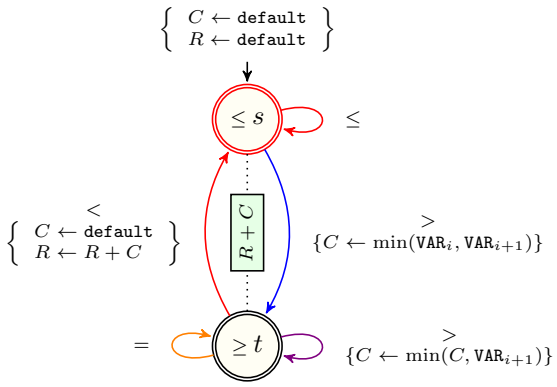


Figure 4.1298: Simplified automaton for the SUM_MIN_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

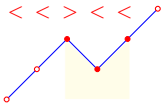
Table 4.301: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the SUM_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	0	0
t	$\overrightarrow{C}$	0 M

Table 4.302: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the SUM_MIN_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_DIP_ON_INCREASING_SEQUENCE



▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint SUM_MIN_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

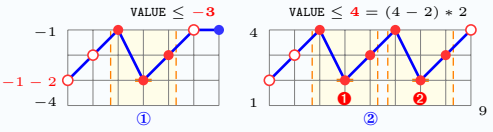
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}, \texttt{minv} * \texttt{np})$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 2①, (\texttt{maxv} - 2) * \texttt{np}②)$
required(VARIABLES, var)

where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 3)/3\rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all minimum values in each occurrence of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 2$ to index j.

Example $(3, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4\rangle)$

Figure 4.1299 provides an example where the SUM_MIN_DIP_ON_INCREASING_SEQUENCE $(3, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

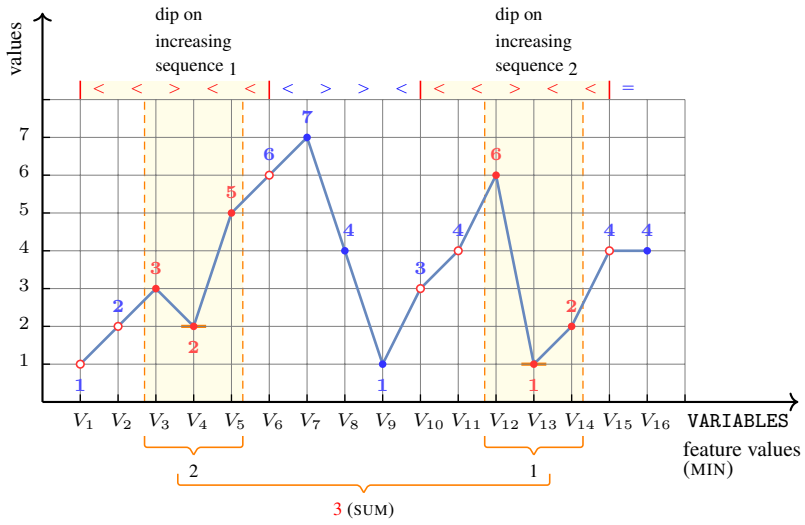


Figure 4.1299: Illustrating the SUM_MIN_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1300 and 4.1301 respectively depict the automaton associated with the constraint SUM_MIN_DIP_ON_INCREASING_SEQUENCE and its simplified form.

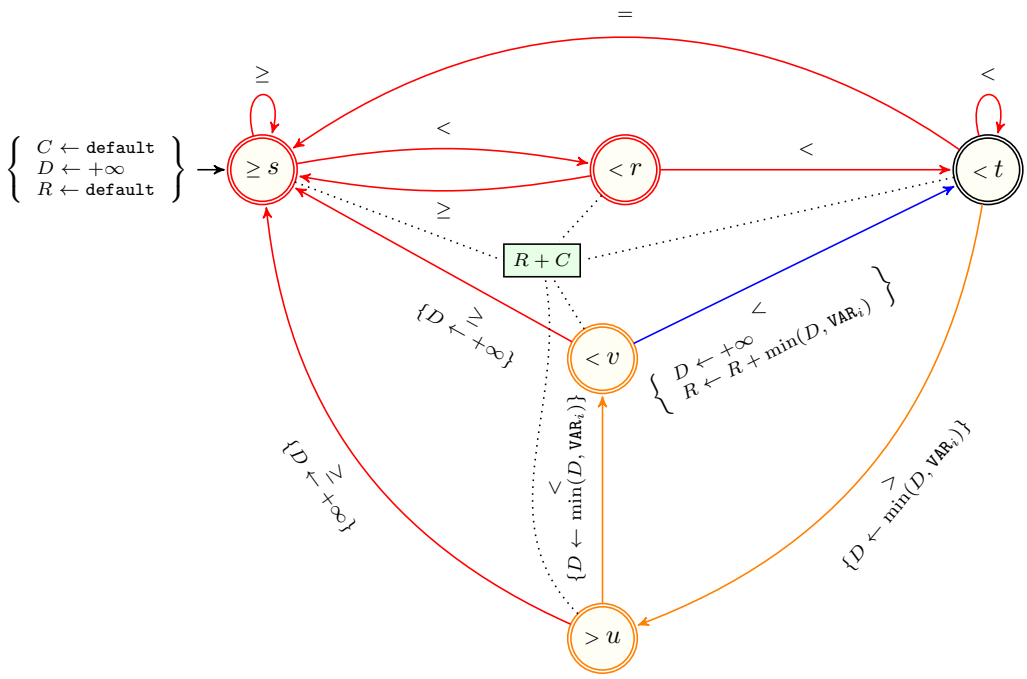


Figure 4.1300: Automaton for the SUM_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

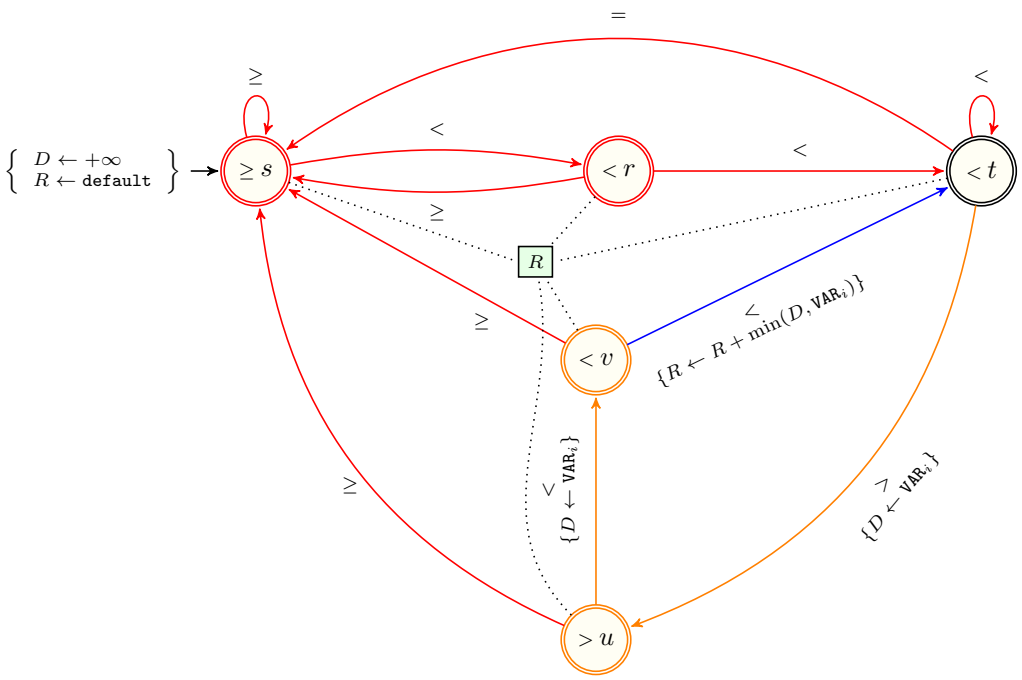
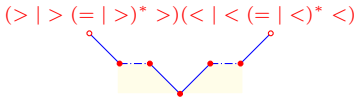


Figure 4.1301: Simplified automaton for the SUM_MIN_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.28 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_MIN_GORGE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

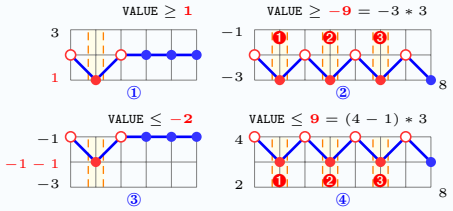
Constraint SUM_MIN_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 1$ to index j.

Example $(12, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7\rangle)$

Figure 4.1302 provides an example where the SUM_MIN_GORGE $(12, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

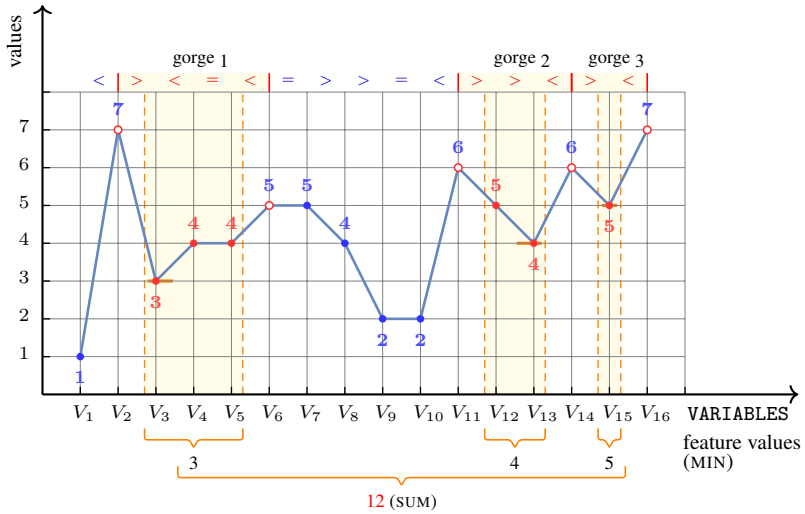


Figure 4.1302: Illustrating the SUM_MIN_GORGE constraint of the **Example** slot

Automaton Figures 4.1303 and 4.1304 respectively depict the automaton associated with the constraint SUM_MIN_GORGE and its simplified form.

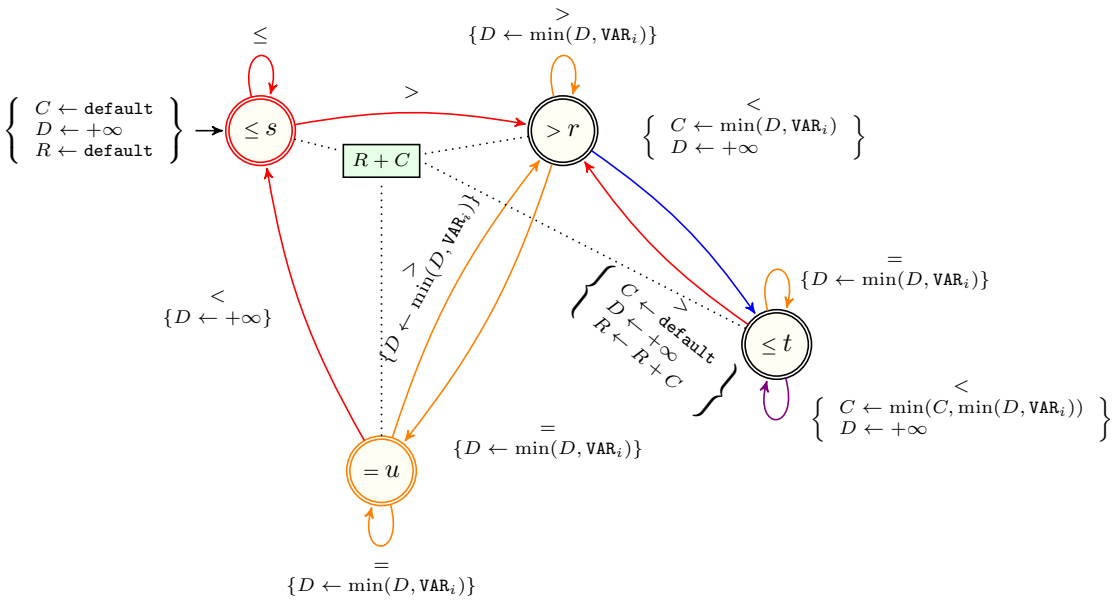


Figure 4.1303: Automaton for the SUM_MIN_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

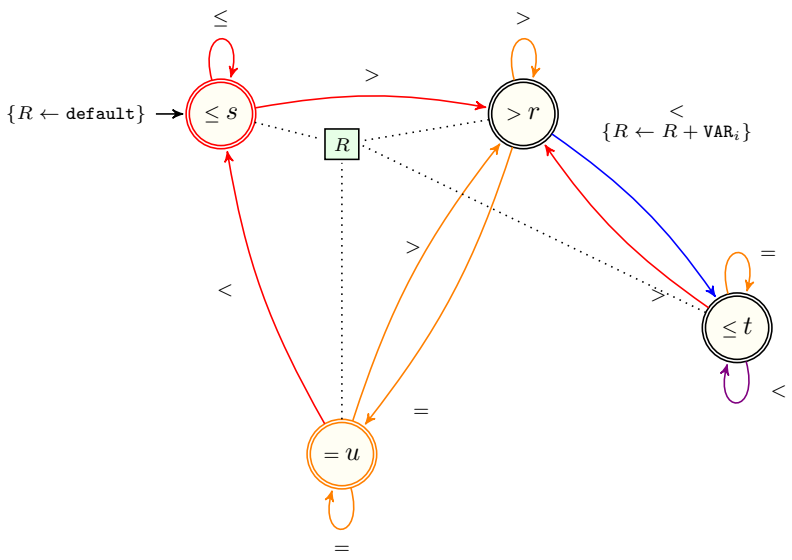


Figure 4.1304: Simplified automaton for the SUM_MIN_GORGE constraint obtained by applying decoration Table 3.39 to the seed transducer of the GORGE pattern where `default` is 0

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \mathtt{VAR}_{i+1})$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.303: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the SUM_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	$\texttt{VAR}_{i+1}$ C	0 R	0
t	0	0 L	0	0 L
u	0	0	0 R	0

Table 4.304: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the SUM_MIN_GORGE constraint defined as the composition of the GORGE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_INCREASING

▶ ▷ ◁ ◀

DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint SUM_MIN_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

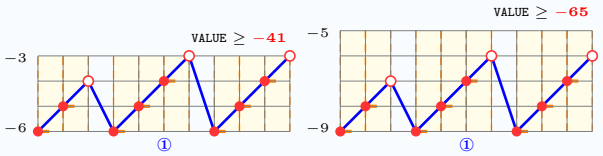
$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\sum \left(\begin{array}{l} (\texttt{np}-1) * \texttt{minv}, \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) / 2 \rfloor \end{array} \right) \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{minv} + \texttt{np} - 1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{minv}, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(\begin{array}{l} 0, \sum \left(\begin{array}{l} \texttt{q} * \left(\begin{array}{l} \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{maxv}-1 \end{array} \right) - \\ \lfloor \prod \left(\begin{array}{l} \texttt{np}-1, \\ \texttt{np}-2 \end{array} \right) / 2 \rfloor \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \texttt{maxv}-1-\texttt{np}+1 \end{array} \right), \\ \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np} \right) \right), \\ \texttt{maxv}-1, \\ \texttt{q}-1 \end{array} \right) \end{array} \right) \end{array} \right)$$

required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$
$\texttt{maxv} =$maxval(VARIABLES.var)
$\texttt{minv} =$minval(VARIABLES.var)
$\texttt{rv} =$range(VARIABLES.var)



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$$

Purpose

VALUE is the sum of all minimum values in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j+1$.

Example

$(12, \langle 4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3 \rangle)$

Figure 4.1305 provides an example where the SUM_MIN_INCREASING $(12, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

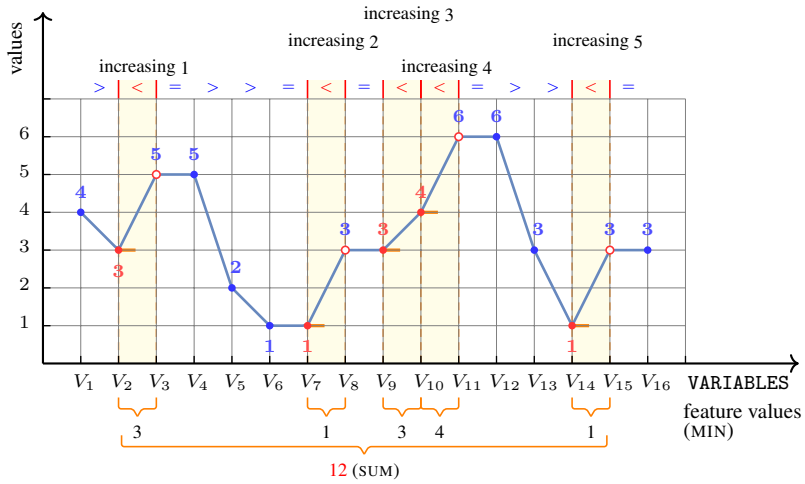


Figure 4.1305: Illustrating the SUM_MIN_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1306 and 4.1307 respectively depict the automaton associated with the constraint SUM_MIN_INCREASING and its simplified form.

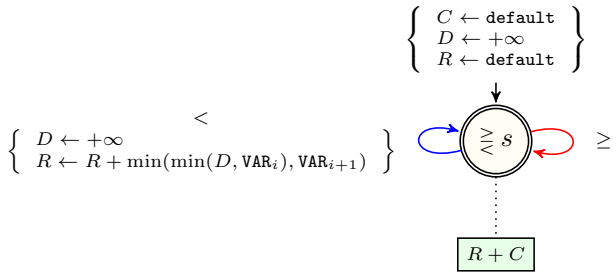


Figure 4.1306: Automaton for the SUM_MIN_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is 0

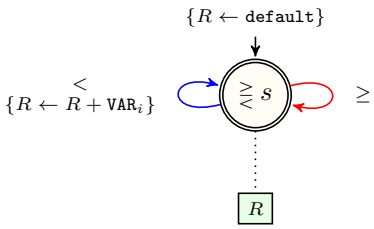


Figure 4.1307: Simplified automaton for the SUM_MIN_INCREASING constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

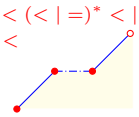
Table 4.305: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the SUM_MIN_INCREASING constraint defined as the composition of the INCREASING pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.306: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the SUM_MIN_INCREASING constraint defined as the composition of the INCREASING pattern , the feature MIN , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

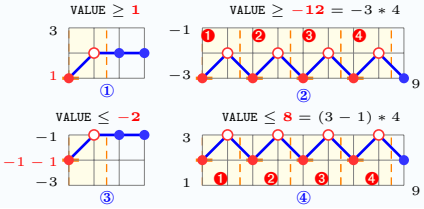
Constraint SUM_MIN_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1308 provides an example where the SUM_MIN_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

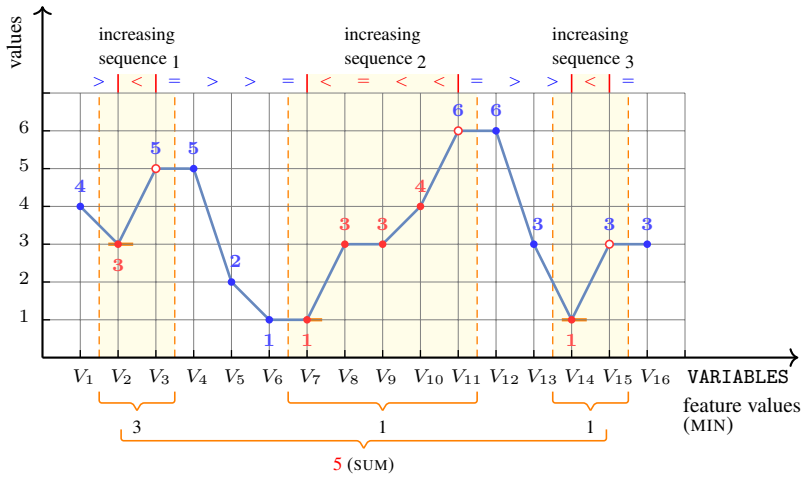


Figure 4.1308: Illustrating the SUM_MIN_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1309 and 4.1310 respectively depict the automaton associated with the constraint SUM_MIN_INCREASING_SEQUENCE and its simplified form.

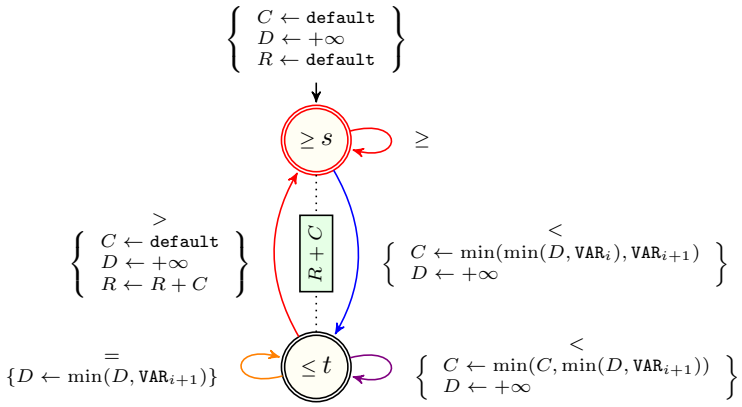


Figure 4.1309: Automaton for the SUM_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

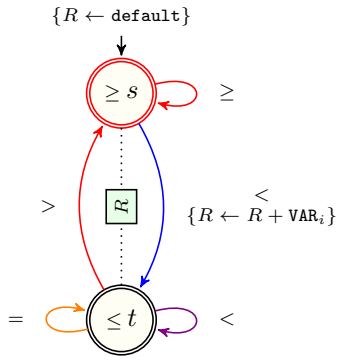


Figure 4.1310: Simplified automaton for the SUM_MIN_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.307: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the SUM_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

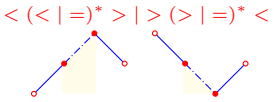
	s	t
s	0	$\overleftarrow{C}$
t	0	0 M

Table 4.308: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the SUM_MIN_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

Constraint SUM_MIN_INFLEXION(VALUE, VARIABLES)

Arguments

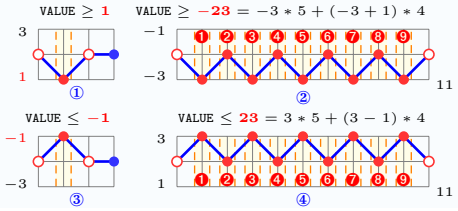
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\begin{array}{l} \texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0 \\ \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \lfloor (\texttt{np}+1)/2 \rfloor + (\texttt{minv}+1) * \lfloor \texttt{np}/2 \rfloor ②) \end{array} \right) \\ \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, \texttt{maxv} * \lfloor (\texttt{np}+1)/2 \rfloor + (\texttt{maxv}-1) * \lfloor \texttt{np}/2 \rfloor ④) \end{array} \right) \\ \texttt{required}(\texttt{VARIABLES}, \texttt{var}) \end{array}$$

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \texttt{sv} - 2)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of all minimum values in each occurrence of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(26, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.1311 provides an example where the SUM_MIN_INFLEXION $(26, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

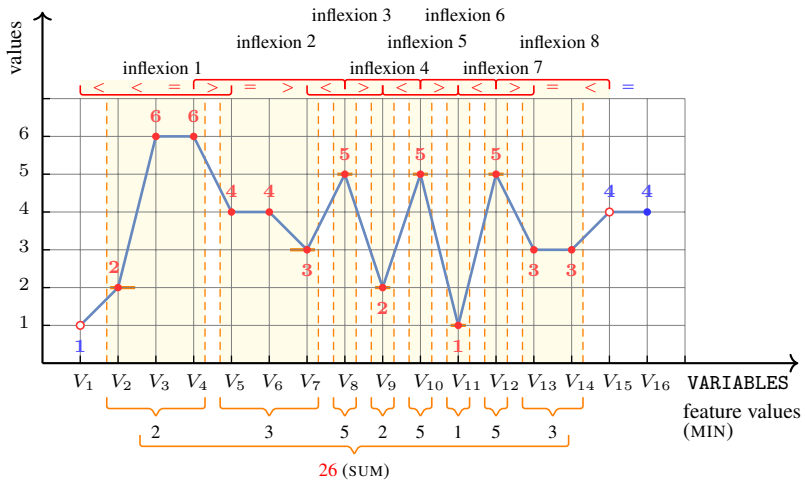


Figure 4.1311: Illustrating the SUM_MIN_INFLEXION constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1312 and 4.1313 respectively depict the automaton associated with the constraint SUM_MIN_INFLEXION and its simplified form.

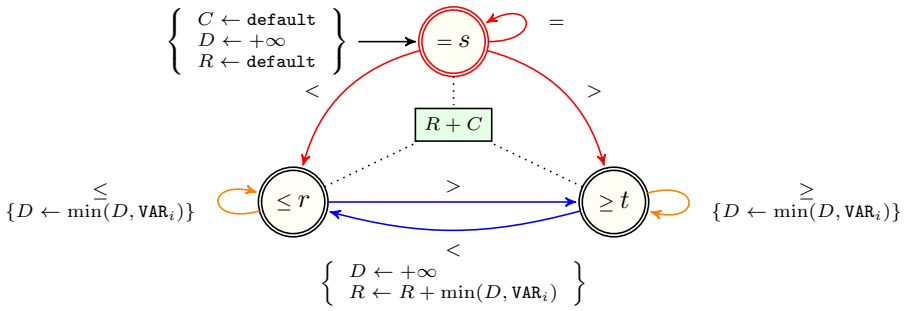


Figure 4.1312: Automaton for the SUM_MIN_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

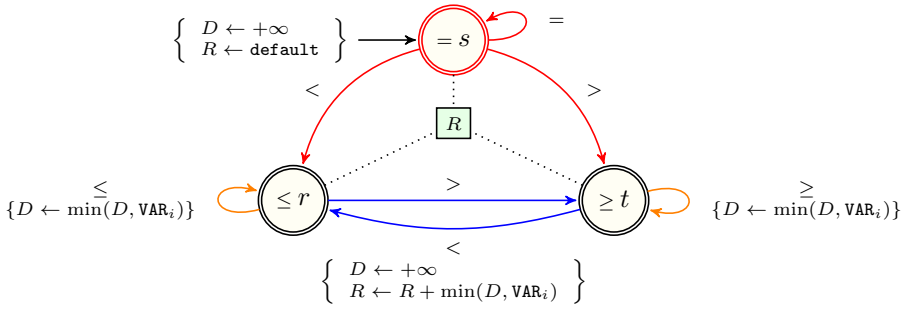


Figure 4.1313: Simplified automaton for the SUM_MIN_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

AGGREGATOR FEATURE PATTERN

SUM_MIN_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

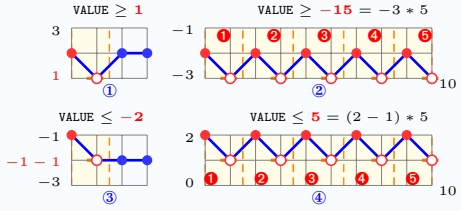
Constraint SUM_MIN_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(6, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1314 provides an example where the SUM_MIN_STRICTLY_DECREASING_SEQUENCE $(6, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

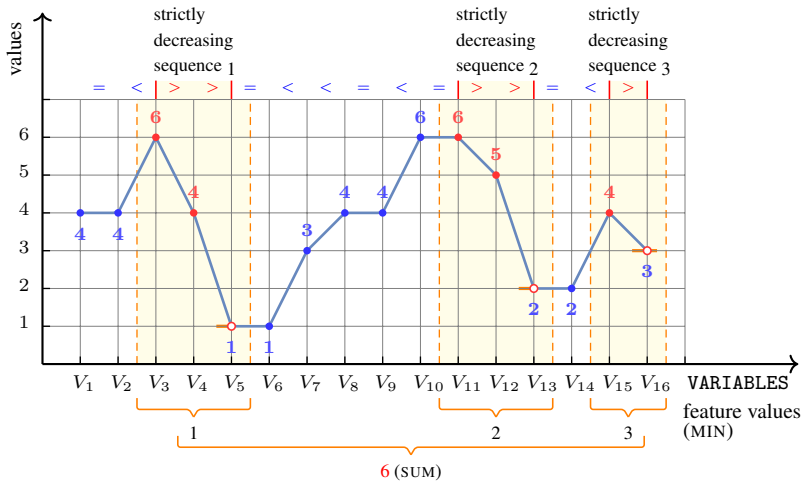


Figure 4.1314: Illustrating the SUM_MIN_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1315 and 4.1316 respectively depict the automaton associated with the constraint SUM_MIN_STRICTLY_DECREASING_SEQUENCE and its simplified form.

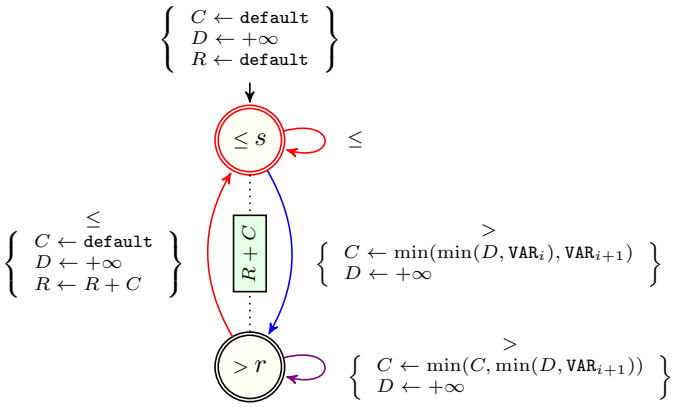


Figure 4.1315: Automaton for the SUM_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

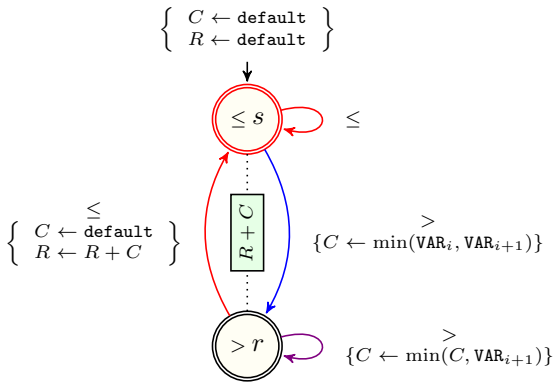


Figure 4.1316: Simplified automaton for the SUM_MIN_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.25 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.309: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the SUM_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	$\overrightarrow{C}$	0 M

Table 4.310: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the SUM_MIN_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

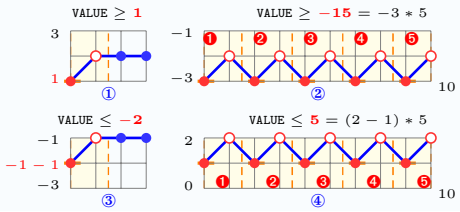
Constraint SUM_MIN_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature MIN computes the minimum of the values from index i to index $j + 1$.

Example $(5, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1317 provides an example where the SUM_MIN_STRICTLY_INCREASING_SEQUENCE $(5, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: VALUE determined by VARIABLES.

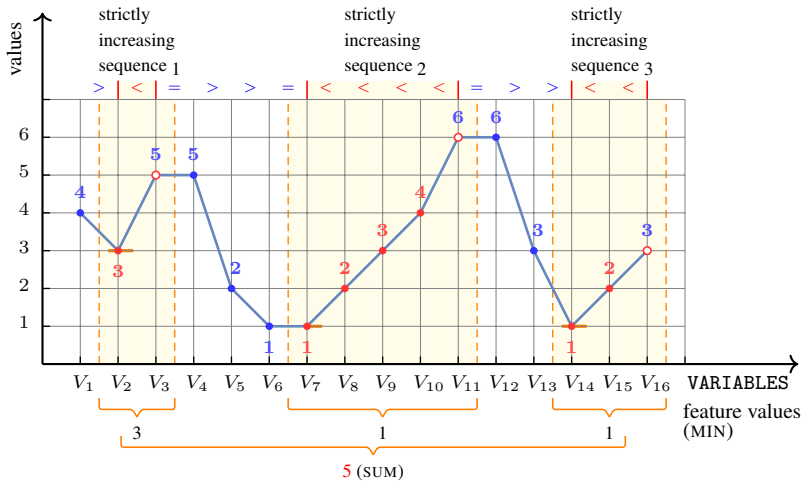


Figure 4.1317: Illustrating the SUM_MIN_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1318 and 4.1319 respectively depict the automaton associated with the constraint SUM_MIN_STRICTLY_INCREASING_SEQUENCE and its simplified form.

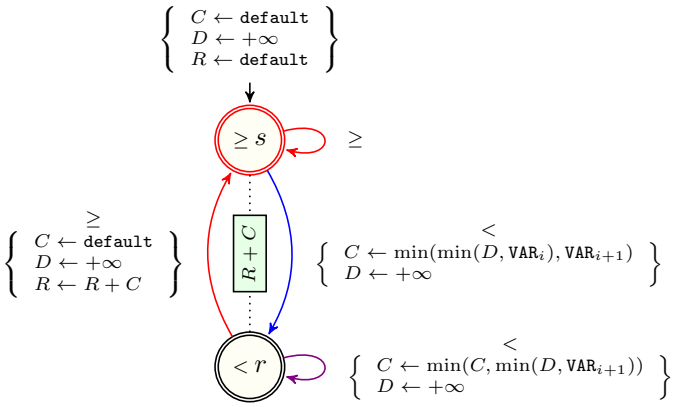


Figure 4.1318: Automaton for the SUM_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

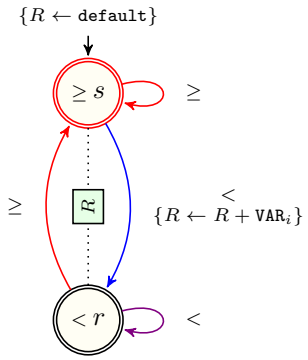


Figure 4.1319: Simplified automaton for the SUM_MIN_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.39 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D})$ M

Table 4.311: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the SUM_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

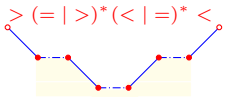
	s	r
s	0	$\overleftarrow{C}$
r	0	0 M

Table 4.312: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the SUM_MIN_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

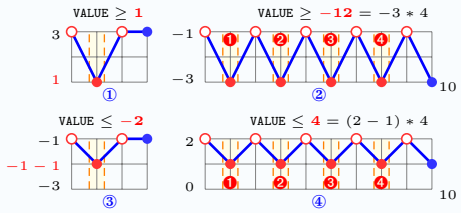
Constraint SUM_MIN_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * \texttt{np}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * \texttt{np}④)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$
minv =minval(VARIABLES.var)
maxv =maxval(VARIABLES.var)
rv =range(VARIABLES.var)



Purpose VALUE is the sum of all minimum values in each occurrence of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i + 1$ to index j.

Example $(10, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7\rangle)$

Figure 4.1320 provides an example where the SUM_MIN_VALLEY $(10, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

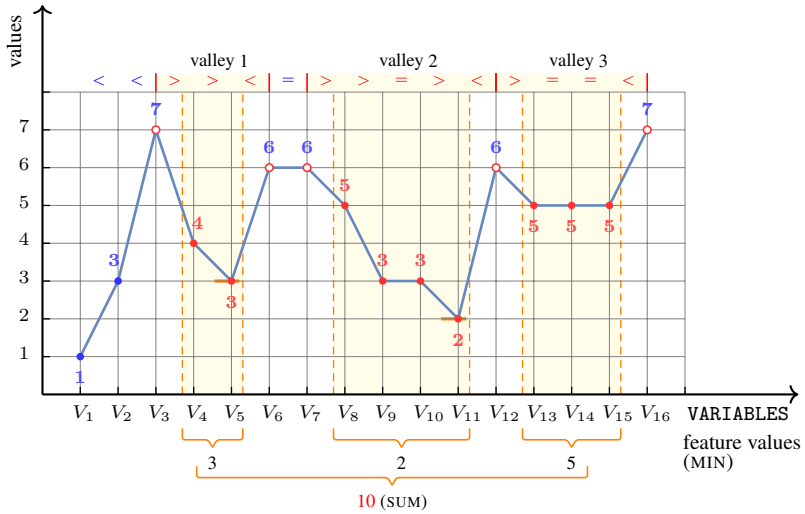


Figure 4.1320: Illustrating the SUM_MIN_VALLEY constraint of the **Example** slot

Automaton Figures 4.1321 and 4.1322 respectively depict the automaton associated with the constraint SUM_MIN_VALLEY and its simplified form.

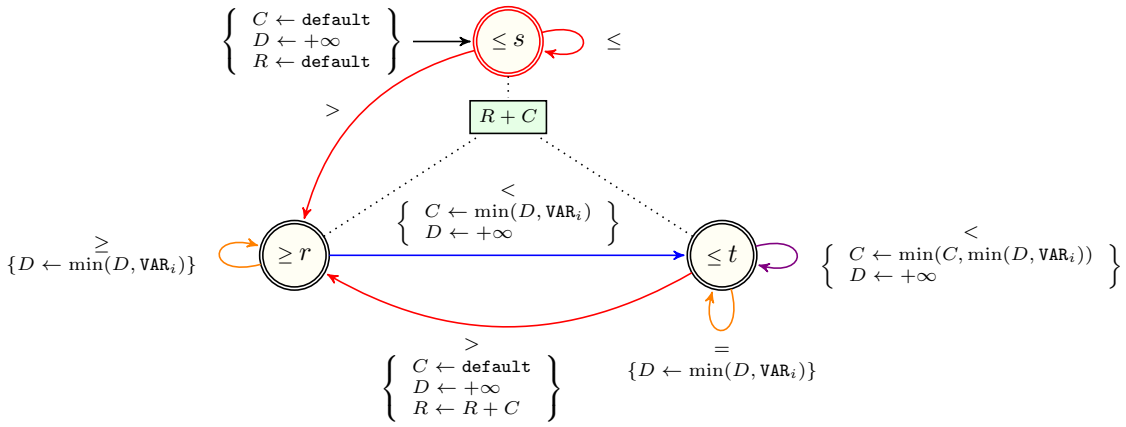


Figure 4.1321: Automaton for the SUM_MIN_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is 0

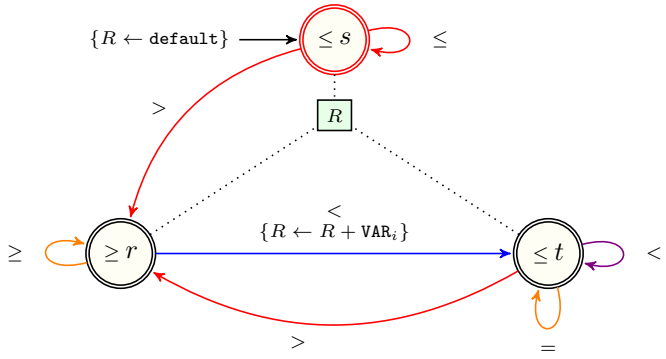


Figure 4.1322: Simplified automaton for the SUM_MIN_VALLEY constraint obtained by applying decoration Table 3.39 to the seed transducer of the VALLEY pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ C	$\min(\overleftarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\min(\overrightarrow{C}, \overrightarrow{D}, \overleftarrow{D}, \texttt{VAR}_{i+1})$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.313: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the SUM_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

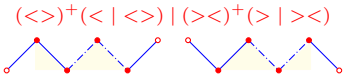
	s	r	t
s	0	0	0
r	0	$\texttt{VAR}_{i+1}$ C	0 R
t	0	0 L	0

Table 4.314: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the SUM_MIN_VALLEY constraint defined as the composition of the VALLEY pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_MIN_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint SUM_MIN_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$

$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv}, \texttt{minv} * \texttt{np1})$

$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv}, \min(\texttt{minv} * \texttt{np1}, \texttt{minv} * \texttt{np2}))$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [0, \texttt{qub}]} \texttt{max} \left(0, \sum \left(\begin{array}{l} \prod \left(\begin{array}{l} \lfloor (\texttt{sv} - 3 * \texttt{q})/4 \rfloor, \\ \texttt{maxv} - 1 \end{array} \right), \\ \texttt{q} * (\texttt{maxv} - 2), \\ \prod \left(\begin{array}{l} \min \left(1, \max \left(\texttt{maxv} - \texttt{minv} - 1, 0 \right) \right), \\ \min \left(1, \max \left(0, \texttt{q} * (\texttt{maxv} - 2) \right) \right), \\ \min \left(\begin{array}{l} \texttt{q}, \\ \left(\begin{array}{l} (\texttt{sv} - \lfloor (\texttt{sv} - 3 * \texttt{q})/4 \rfloor) * 4 - \\ \texttt{q} * 3 \end{array} \right) \bmod 3 \end{array} \right) \end{array} \right) \end{array} \right) \right)$$

required(VARIABLES, var)

where

$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$

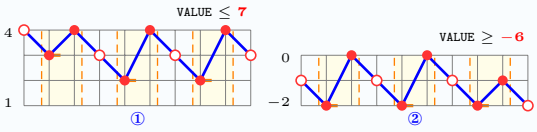
$\texttt{np1} = \lfloor \texttt{sv}/4 \rfloor$

$\texttt{np2} = \texttt{max}(0, \lfloor (\texttt{sv} - 1)/3 \rfloor)$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{qub} = \min(1, \max(\texttt{maxv} - \texttt{minv} - 1, 0)) * \texttt{np2}$



Purpose VALUE is the sum of all minimum values in each occurrence of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature MIN computes the minimum of the values from index $i+1$ to index j.

Example $(3, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.1323 provides an example where the SUM_MIN_ZIGZAG $(3, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

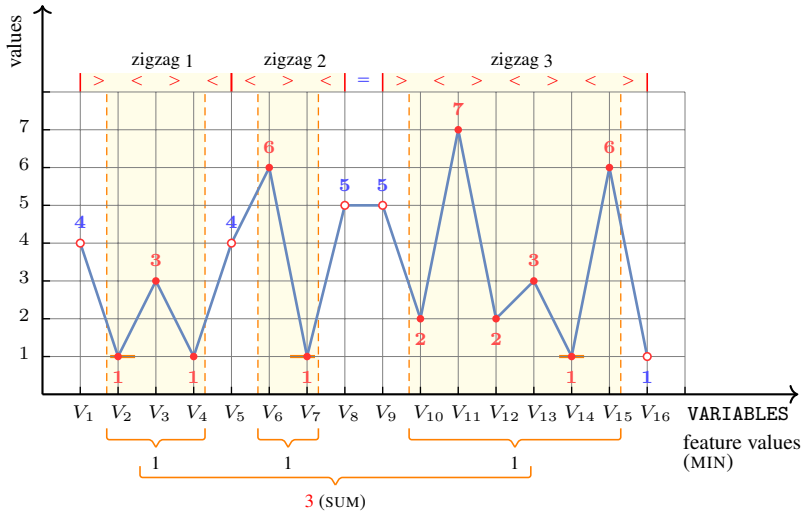


Figure 4.1323: Illustrating the SUM_MIN_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1324 and 4.1325 respectively depict the automaton associated with the constraint SUM_MIN_ZIGZAG and its simplified form.

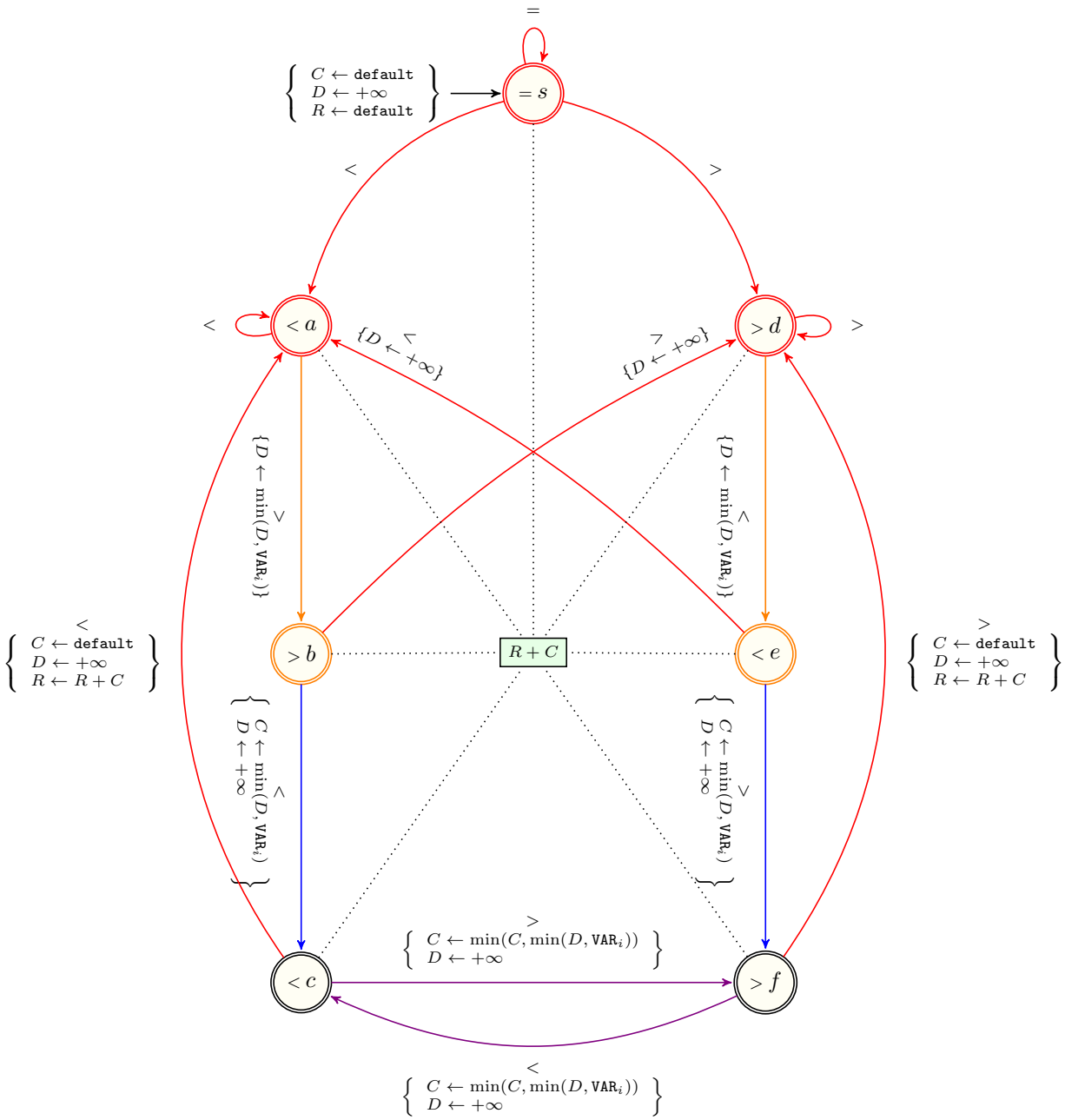


Figure 4.1324: Automaton for the SUM_MIN_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

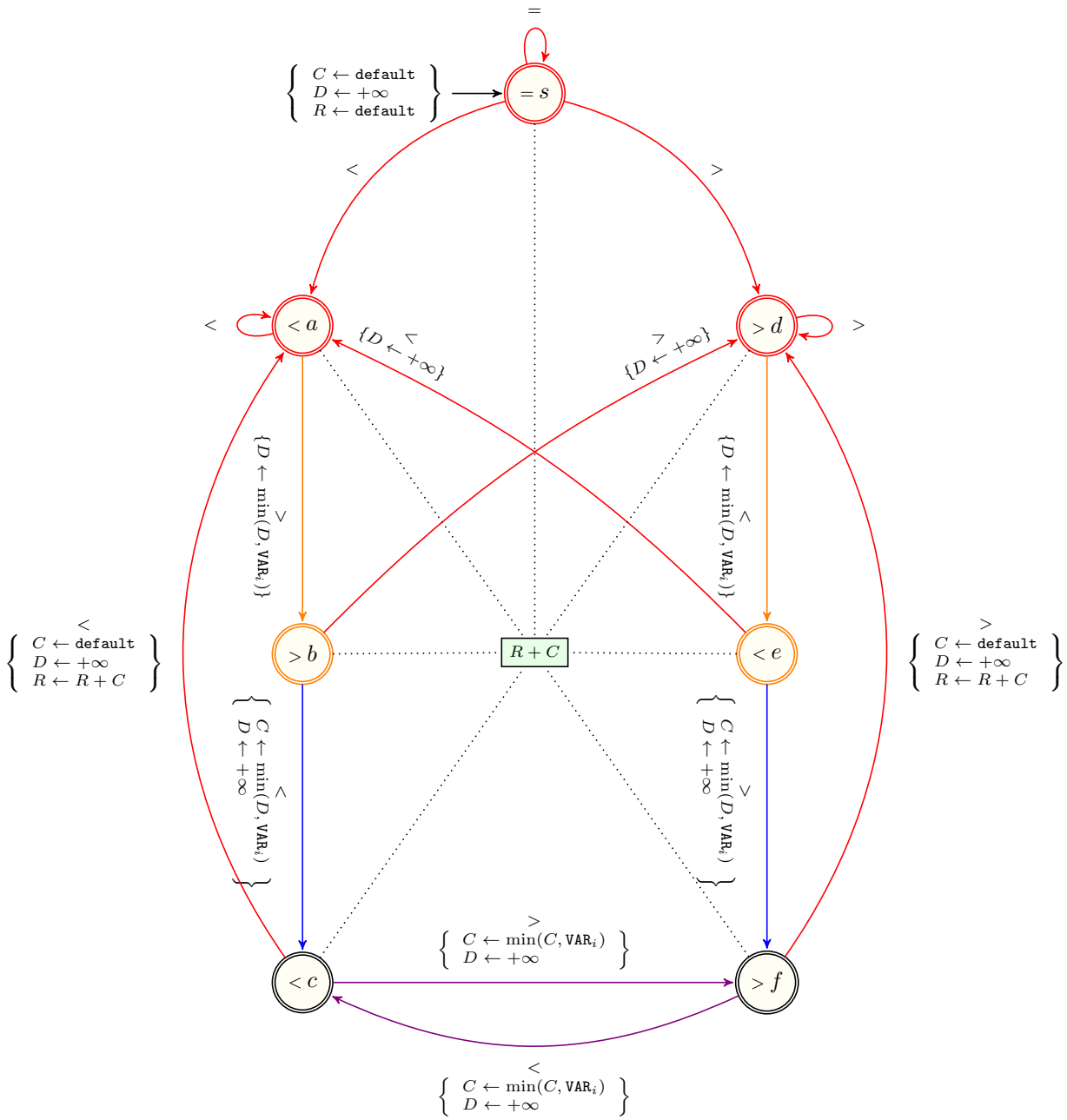


Figure 4.1325: Simplified automaton for the SUM_MIN_ZIGZAG constraint obtained by applying decoration Table 3.24 to the seed transducer of the ZIGZAG pattern where `default` is 0; missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value.

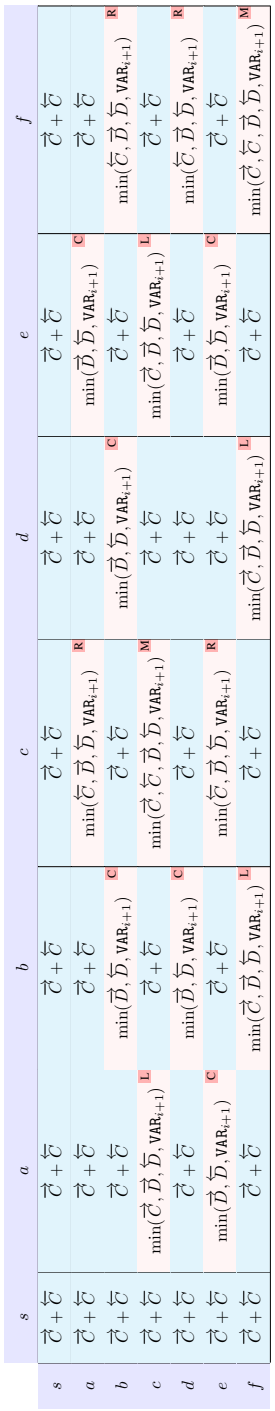


Table 4.315: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the SUM_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature MIN, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

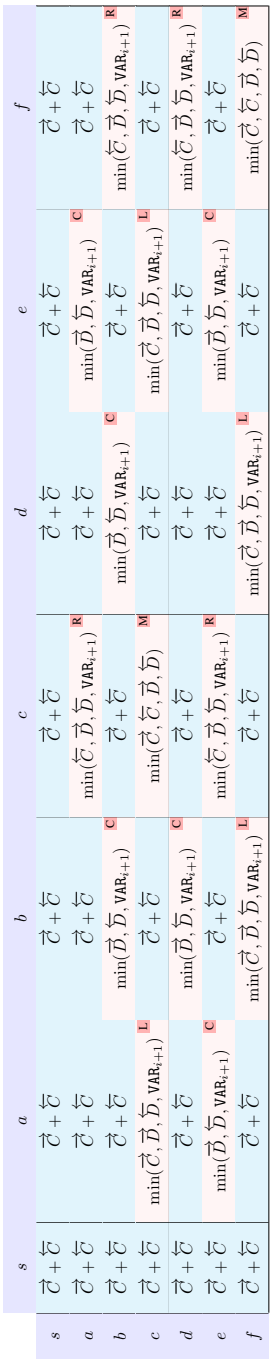


Table 4.316: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the SUM_MIN_ZIGZAG constraint defined as the composition of the ZIGZAG pattern , the feature MIN , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_RANGE_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

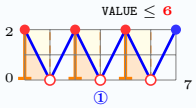
Constraint SUM_RANGE_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

VALUE is the sum of the differences between the largest and smallest value in each occurrence of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(9, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1326 provides an example where the SUM_RANGE_DECREASING $(9, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

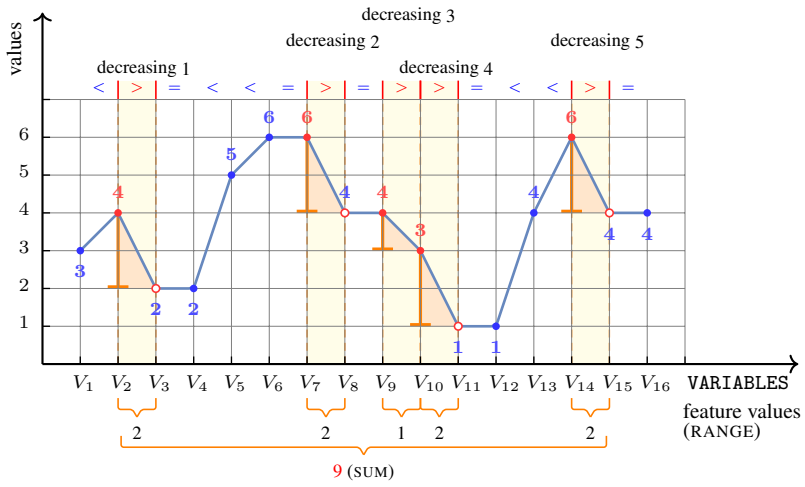


Figure 4.1326: Illustrating the SUM_RANGE_DECREASING constraint of the **Example** slot

Automaton

Figures 4.1327 and 4.1328 respectively depict the automaton associated with the constraint SUM_RANGE_DECREASING and its simplified form.

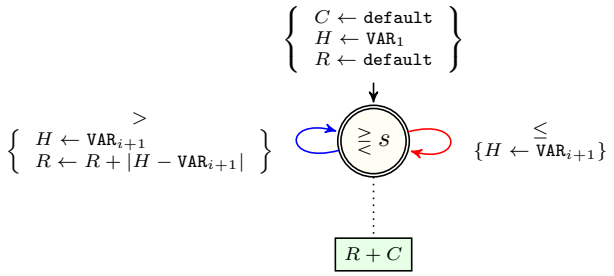


Figure 4.1327: Automaton for the SUM_RANGE_DECREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING pattern where `default` is 0

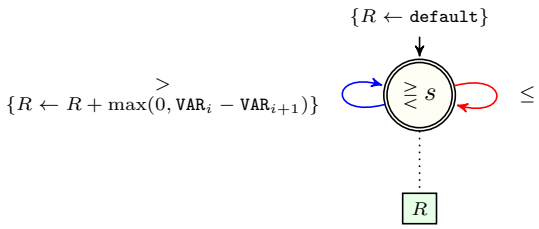
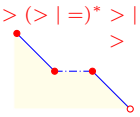


Figure 4.1328: Simplified automaton for the SUM_RANGE_DECREASING constraint obtained by applying decoration Table 3.46 to the seed transducer of the DECREASING pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_RANGE_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

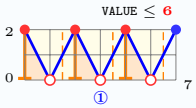
Constraint SUM_RANGE_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the sum of the differences between the largest and smallest value in each occurrence of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(9, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1329 provides an example where the SUM_RANGE_DECREASING_SEQUENCE $(9, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

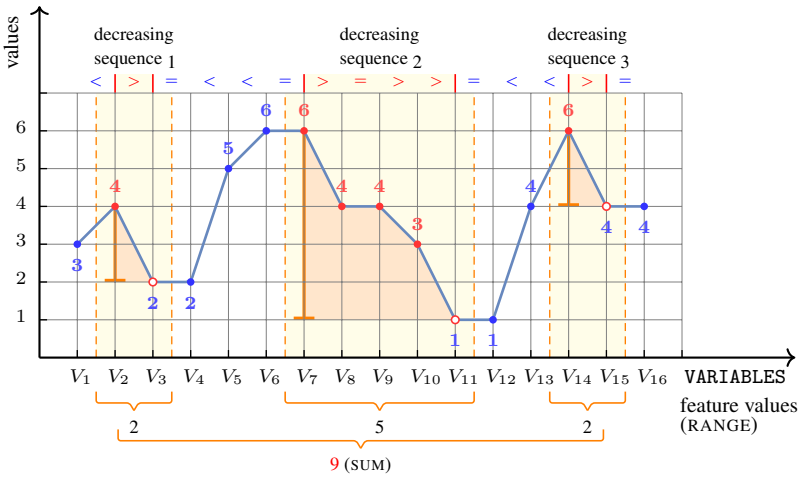


Figure 4.1329: Illustrating the SUM_RANGE_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Figures 4.1330 and 4.1331 respectively depict the automaton associated with the constraint SUM_RANGE_DECREASING_SEQUENCE and its simplified form.

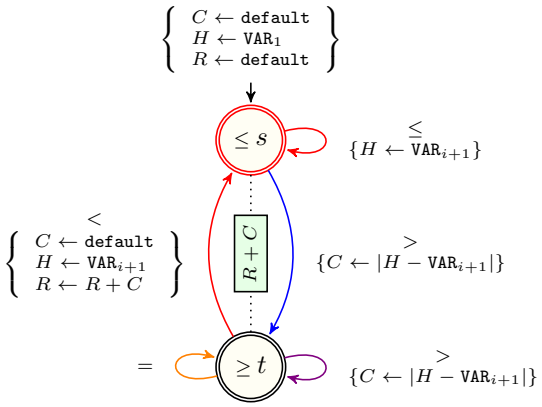


Figure 4.1330: Automaton for the SUM_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

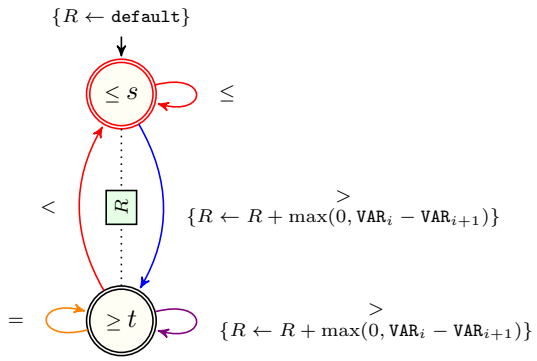


Figure 4.1331: Simplified automaton for the SUM_RANGE_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.46 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_RANGE_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

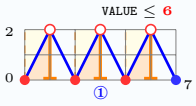
Constraint SUM_RANGE_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose

VALUE is the sum of the differences between the largest and smallest value in each occurrence of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(9, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1332 provides an example where the SUM_RANGE_INCREASING $(9, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

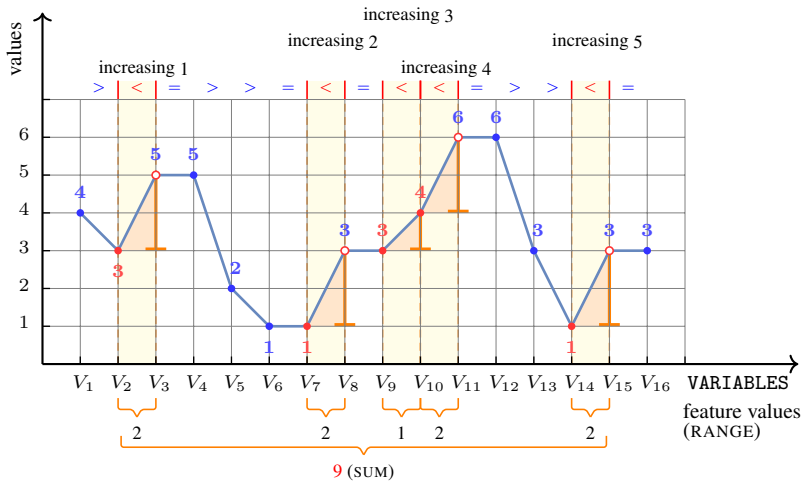


Figure 4.1332: Illustrating the SUM_RANGE_INCREASING constraint of the **Example** slot

Automaton

Figures 4.1333 and 4.1334 respectively depict the automaton associated with the constraint SUM_RANGE_INCREASING and its simplified form.

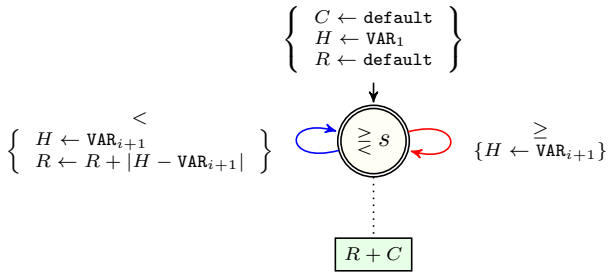


Figure 4.1333: Automaton for the SUM_RANGE_INCREASING constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING pattern where `default` is 0

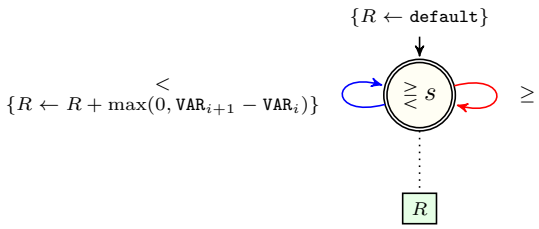
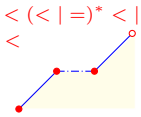


Figure 4.1334: Simplified automaton for the SUM_RANGE_INCREASING constraint obtained by applying decoration Table 3.47 to the seed transducer of the INCREASING pattern where `default` is 0; $R_i - R_{i-2} - X_i + X_{i-2} \geq 0$ and $R_i - R_{i-1} - X_i + X_{i-1} \geq 0$ are linear invariants.

AGGREGATOR FEATURE PATTERN

SUM_RANGE_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

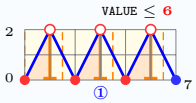
Constraint SUM_RANGE_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the sum of the differences between the largest and smallest value in each occurrence of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(9, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1335 provides an example where the SUM_RANGE_INCREASING_SEQUENCE $(9, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

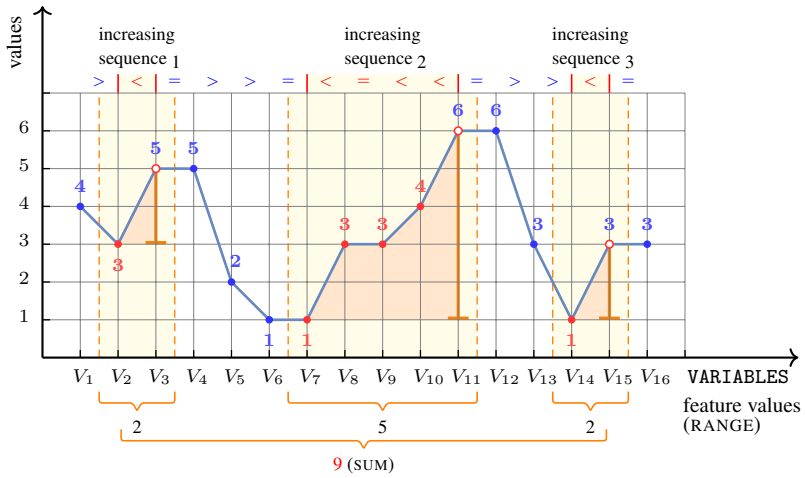


Figure 4.1335: Illustrating the SUM_RANGE_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1336 and 4.1337 respectively depict the automaton associated with the constraint SUM_RANGE_INCREASING_SEQUENCE and its simplified form.

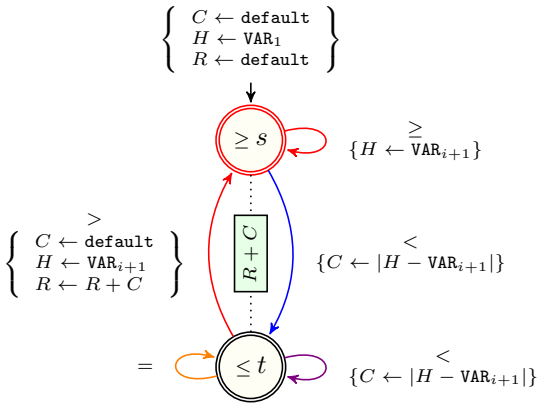


Figure 4.1336: Automaton for the SUM_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

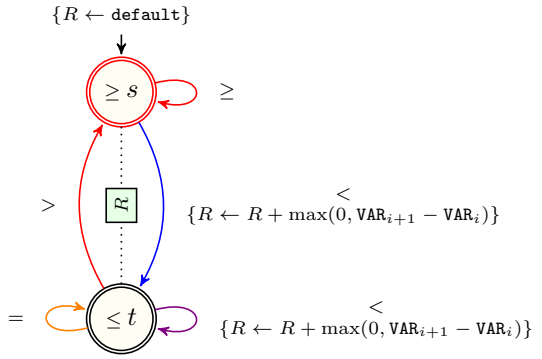


Figure 4.1337: Simplified automaton for the SUM_RANGE_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.47 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_RANGE_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

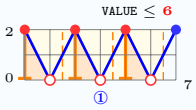
Constraint SUM_RANGE_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the sum of the differences between the largest and smallest value in each occurrence of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(10, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1338 provides an example where the SUM_RANGE_STRICTLY_DECREASING_SEQUENCE $(10, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

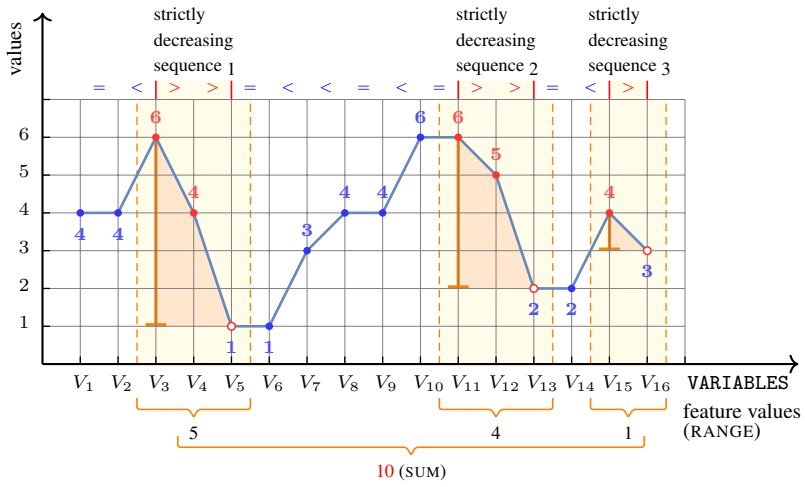


Figure 4.1338: Illustrating the SUM_RANGE_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1339 and 4.1340 respectively depict the automaton associated with the constraint SUM_RANGE_STRICTLY_DECREASING_SEQUENCE and its simplified form.

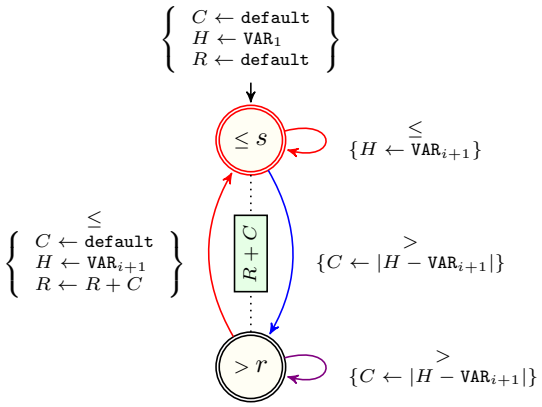


Figure 4.1339: Automaton for the SUM_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

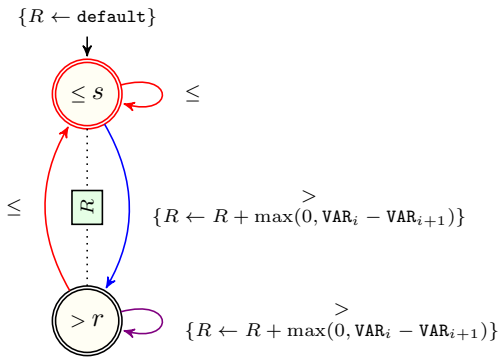


Figure 4.1340: Simplified automaton for the SUM_RANGE_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.46 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_RANGE_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

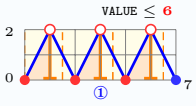
Constraint SUM_RANGE_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} \geq 0$
$\texttt{VALUE} \leq \lfloor \texttt{sv}/2 \rfloor * (\texttt{rv} - 1)$①
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$



Purpose VALUE is the sum of the differences between the largest and smallest value in each occurrence of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature RANGE computes the range of the values from index i to index $j + 1$.

Example $(9, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1341 provides an example where the SUM_RANGE_STRICTLY_INCREASING_SEQUENCE $(9, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

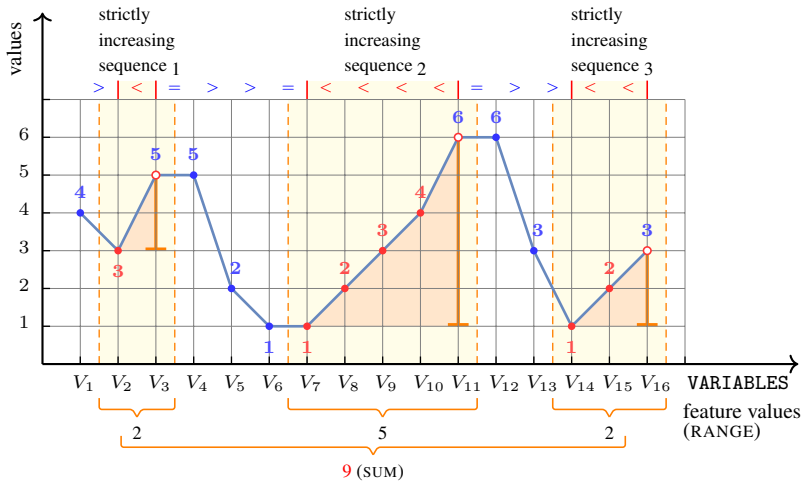


Figure 4.1341: Illustrating the SUM_RANGE_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1342 and 4.1343 respectively depict the automaton associated with the constraint SUM_RANGE_STRICTLY_INCREASING_SEQUENCE and its simplified form.

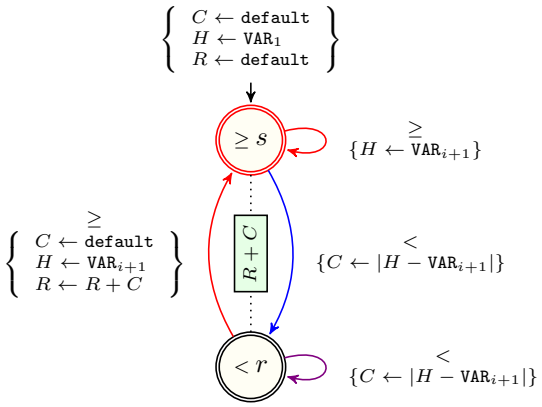


Figure 4.1342: Automaton for the SUM_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.48 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

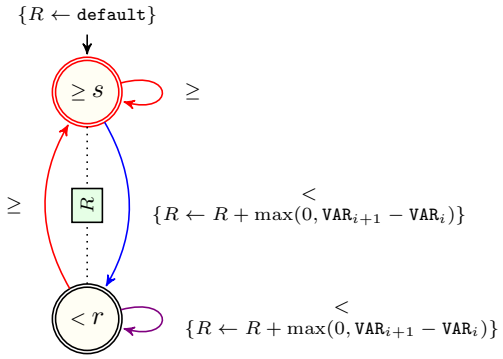
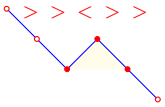


Figure 4.1343: Simplified automaton for the SUM_RANGE_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.47 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_SURF_BUMP_ON_DECREASING_SEQUENCE



▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint SUM_SURF_BUMP_ON_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

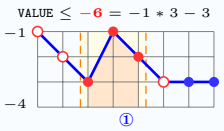
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

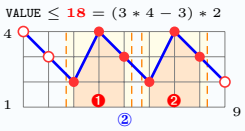
Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(3 * \texttt{minv} + 3, (3 * \texttt{minv} + 3) * \texttt{np})$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(3 * \texttt{maxv} - 3①, (3 * \texttt{maxv} - 3) * \texttt{np}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - 3 * \texttt{np} - \max((\texttt{sv} - 3) * (\texttt{maxv} - 3), 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{n2} \geq \min((\texttt{sv} - 3) * (\texttt{minv} + 3), 0) - 3 * \texttt{np} - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor (\texttt{sv} - 3)/3 \rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$





Purpose VALUE is the sum of the surface of occurrences of the BUMP_ON_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(27, \langle 7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3 \rangle)$

Figure 4.1344 provides an example where the SUM_SURF_BUMP_ON_DECREASING_SEQUENCE $(27, [7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3])$ constraint holds.

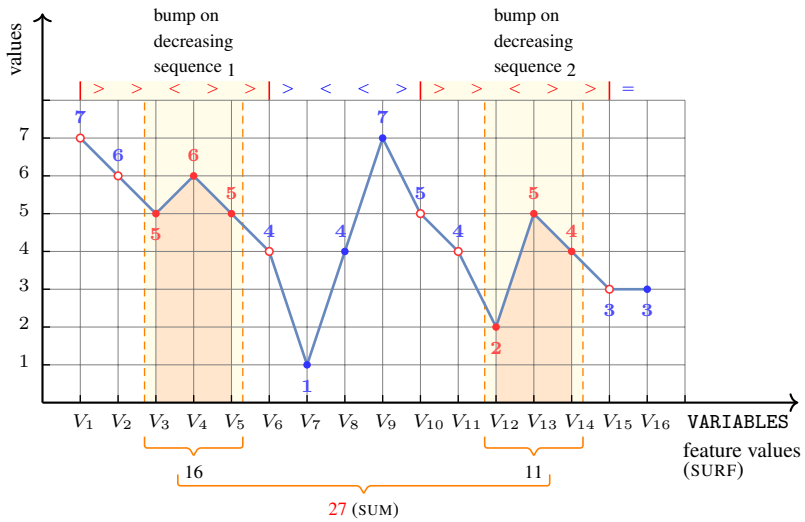


Figure 4.1344: Illustrating the SUM_SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1345 and 4.1346 respectively depict the automaton associated with the constraint SUM_SURF_BUMP_ON_DECREASING_SEQUENCE and its simplified form.

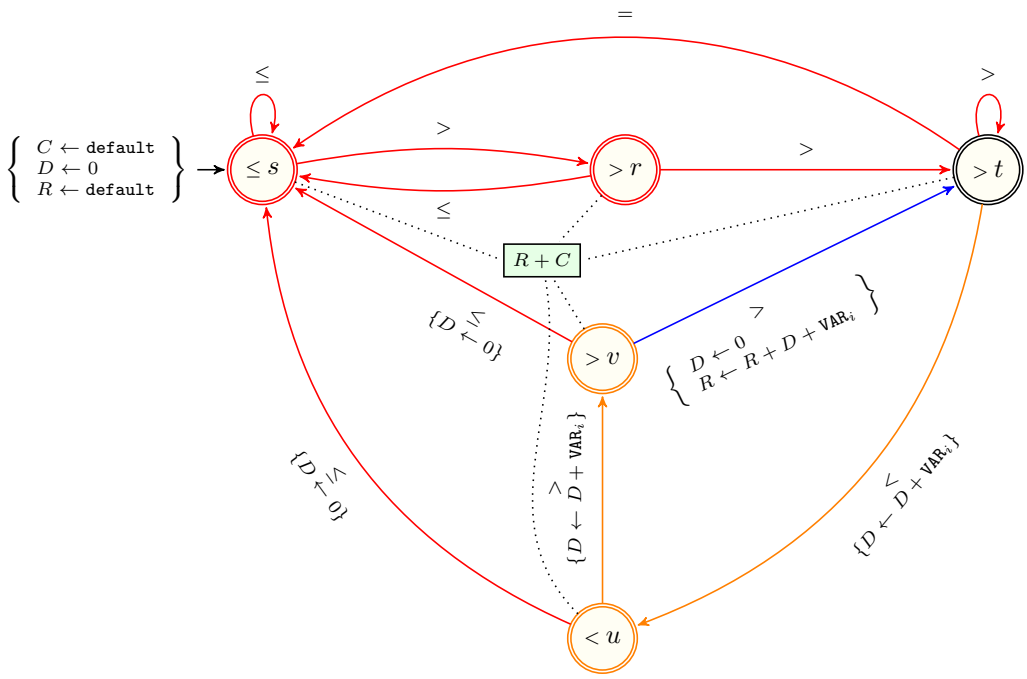


Figure 4.1345: Automaton for the SUM_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

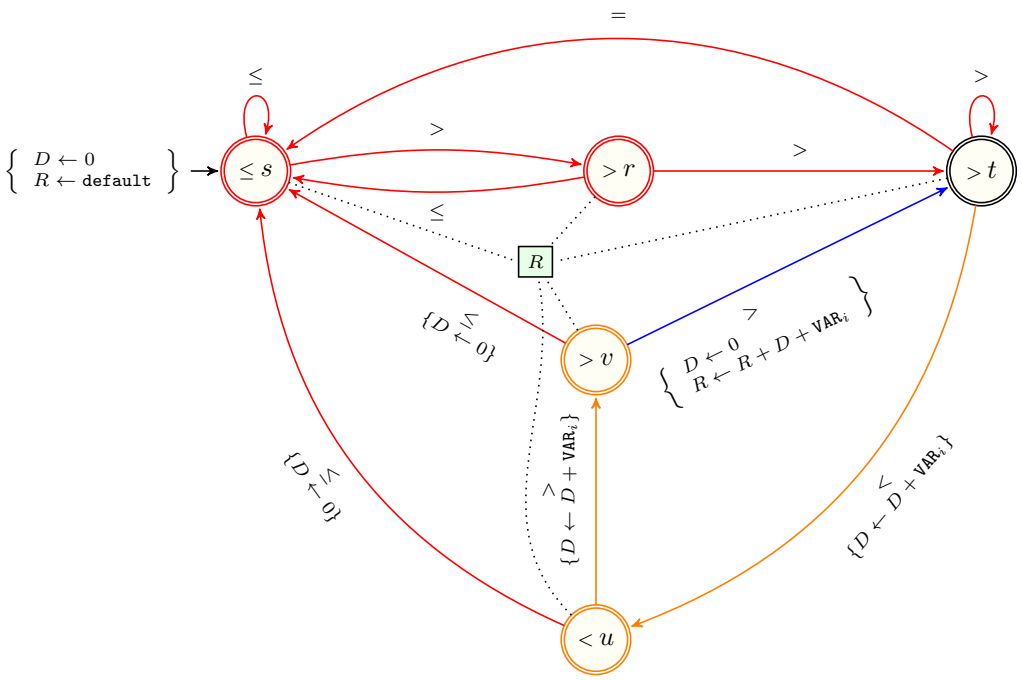


Figure 4.1346: Simplified automaton for the SUM_SURF_BUMP_ON_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the BUMP_ON_DECREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_SURF_DECREASING

DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint SUM_SURF_DECREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

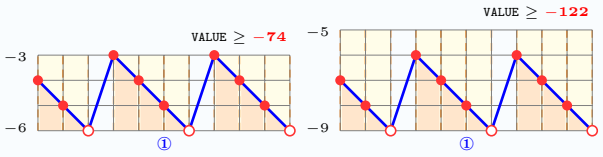
$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(0, \sum \left(\prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np} - 1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \sum \left(\begin{array}{l} 2 * \texttt{minv} - 1, \\ 2 * \texttt{np} \end{array} \right) \end{array} \right), \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2 - \texttt{np} \right) \right), \\ 2 * \texttt{minv} + 1, \\ \texttt{q} - 1 \end{array} \right) \right) \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(0, \sum \left(\prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np} - 1 \right) \right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ 2 * \texttt{maxv} + 1 - \\ 2 * \texttt{np} \end{array} \right), \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2 - \texttt{np} \right) \right), \\ 2 * \texttt{maxv} - 1, \\ \texttt{q} - 1 \end{array} \right) \right) \right)$$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}| + 1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{minv}| + 1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}| + 1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv}, \texttt{rv}), |\texttt{maxv}| + 1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$$

Purpose

VALUE is the sum of the surface of occurrences of the DECREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '>'.

Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$(37, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1347 provides an example where the SUM_SURF_DECREASING $(37, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

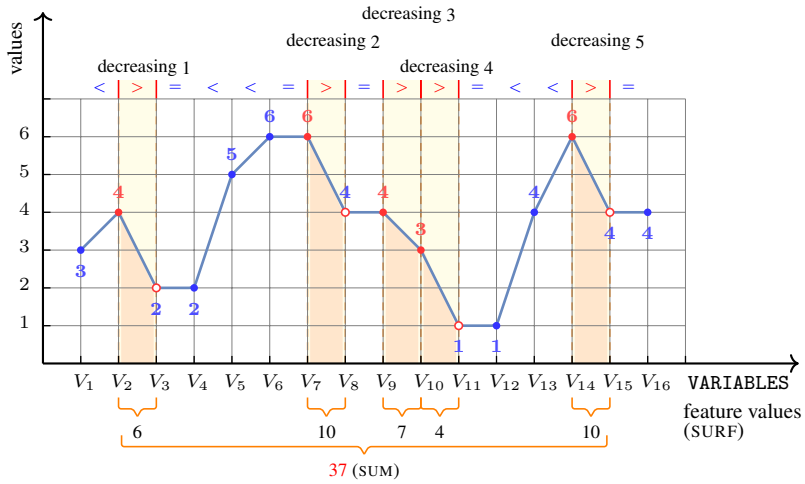


Figure 4.1347: Illustrating the SUM_SURF_DECREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1348 and 4.1349 respectively depict the automaton associated with the constraint SUM_SURF_DECREASING and its simplified form.

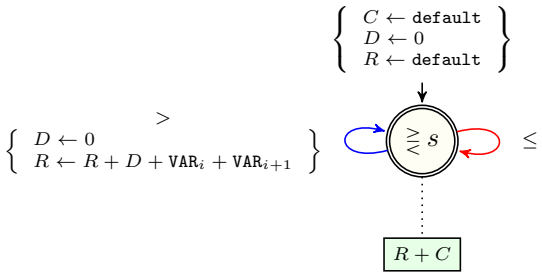


Figure 4.1348: Automaton for the SUM_SURF_DECREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING pattern where `default` is 0

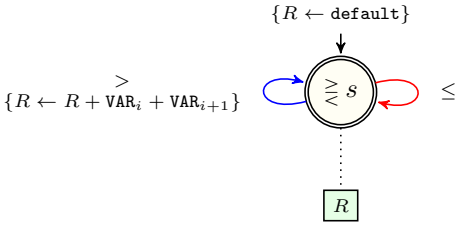


Figure 4.1349: Simplified automaton for the SUM_SURF_DECREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the DECREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

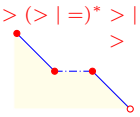
Table 4.317: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the SUM_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.318: Concrete glue matrix, derived from the parametrised glue matrix 3.4, for the simplified automaton of the SUM_SURF_DECREASING constraint defined as the composition of the DECREASING pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint SUM_SURF_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 1 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = 0$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = 0 \vee \mathtt{VALUE} \geq \min(2 * \mathtt{minv} + 1, (2 * \mathtt{minv} + 1) * \mathtt{np})$

$\mathtt{rv} \geq 3 \Rightarrow$

$$\bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \geq \min \left(\begin{array}{l} 2 * \mathtt{minv} + 1①, \\ (2 * \mathtt{minv} + 1) * \mathtt{np} + \min(0, \mathtt{sv} \bmod 2 * (\mathtt{minv} + 2))② \end{array} \right) \end{array} \right)$$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = 0 \vee \mathtt{VALUE} \leq \max(2 * \mathtt{maxv} - 1, (2 * \mathtt{maxv} - 1) * \mathtt{np})$

$\mathtt{rv} \geq 3 \Rightarrow$

$$\bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \leq \max \left(\begin{array}{l} 2 * \mathtt{maxv} - 1③, \\ (2 * \mathtt{maxv} - 1) * \mathtt{np} + \max(0, \mathtt{sv} \bmod 2 * (\mathtt{maxv} - 2))④ \end{array} \right) \end{array} \right)$$

$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{maxv} - 1, \mathtt{maxv} \rangle)$

$\mathtt{rv} = 2 \vee \mathtt{maxv} = 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \max(0, \mathtt{np} * (2 * \mathtt{maxv} - 3))$

$\mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{np} - \mathtt{sv} * (\mathtt{maxv} - 2)$

$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[1, \mathtt{sv}], \langle \mathtt{minv}, \mathtt{minv} + 1 \rangle)$

$\mathtt{rv} = 2 \vee \mathtt{minv} = -1 \Rightarrow \mathtt{n2} \geq \min(0, \mathtt{np} * (2 * \mathtt{minv} + 3)) - \mathtt{VALUE}$

$\mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq \mathtt{sv} * (\mathtt{minv} + 2) - \mathtt{np} - \mathtt{VALUE}$

required(VARIABLES, var)

where

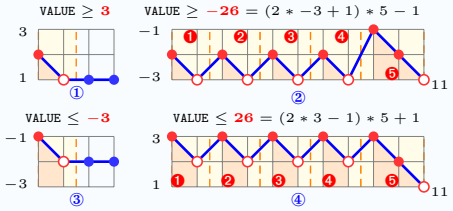
$\mathtt{sv} = |\mathtt{VARIABLES}|$

$\mathtt{np} = \lfloor \mathtt{sv}/2 \rfloor$

minv =minval(VARIABLES.var)

maxv =maxval(VARIABLES.var)

rv =range(VARIABLES.var)



Purpose

VALUE is the sum of the surface of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.

Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(34, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4\rangle)$

Figure 4.1350 provides an example where the SUM_SURF_DECREASING_SEQUENCE $(34, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4])$ constraint holds.

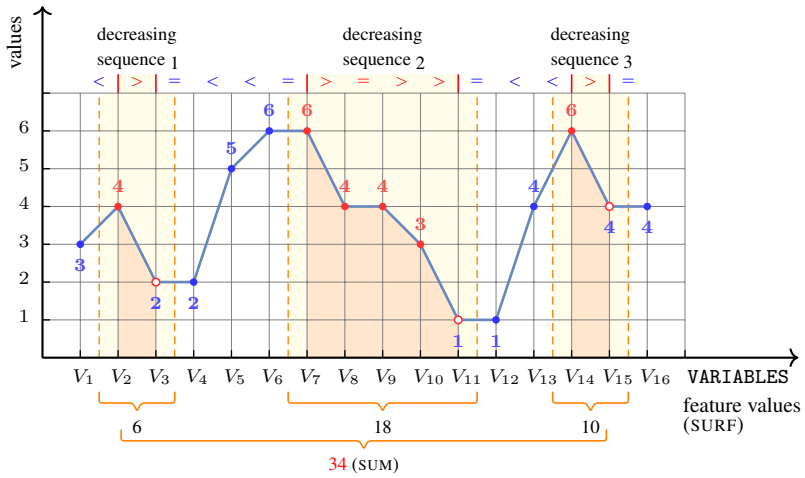


Figure 4.1350: Illustrating the SUM_SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1351 and 4.1352 respectively depict the automaton associated with the constraint SUM_SURF_DECREASING_SEQUENCE and its simplified form.

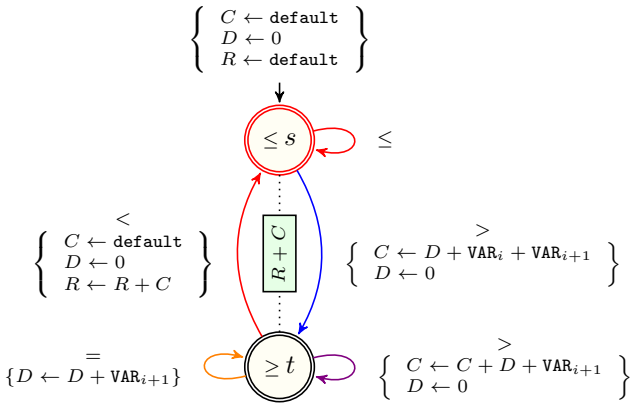


Figure 4.1351: Automaton for the SUM_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

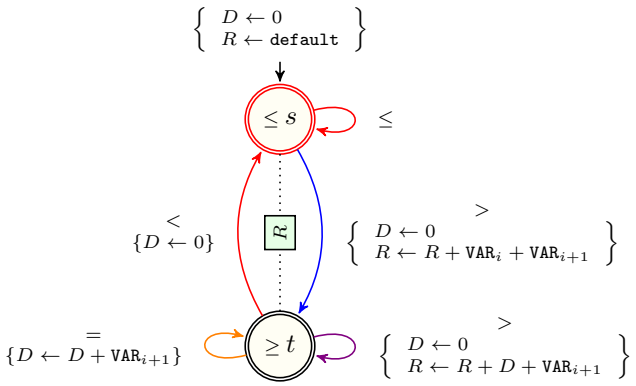


Figure 4.1352: Simplified automaton for the SUM_SURF_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.30 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

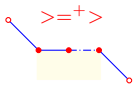
Table 4.319: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the SUM_SURF_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.320: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the SUM_SURF_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_DECREASING_TERRACE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_TERRACE pattern.

Constraint SUM_SURF_DECREASING_TERRACE(VALUE, VARIABLES)

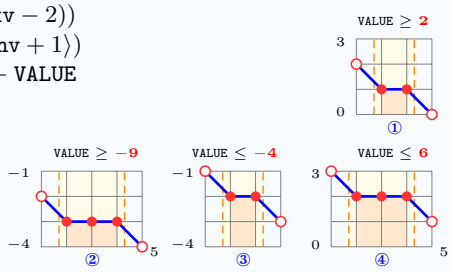
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1)③, (\texttt{sv} - 2) * (\texttt{maxv} - 1)④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of the surface of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(12, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3 \rangle)$

Figure 4.1353 provides an example where the SUM_SURF_DECREASING_TERRACE $(12, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

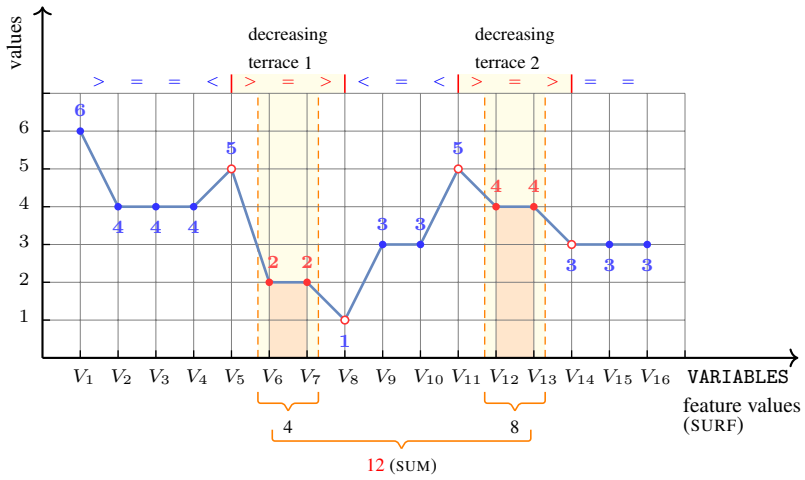


Figure 4.1353: Illustrating the SUM_SURF_DECREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1354 and 4.1355 respectively depict the automaton associated with the constraint SUM_SURF_DECREASING_TERRACE and its simplified form.

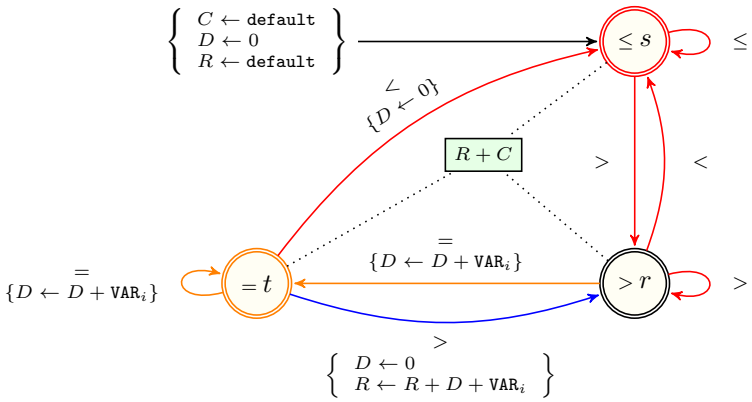


Figure 4.1354: Automaton for the SUM_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0

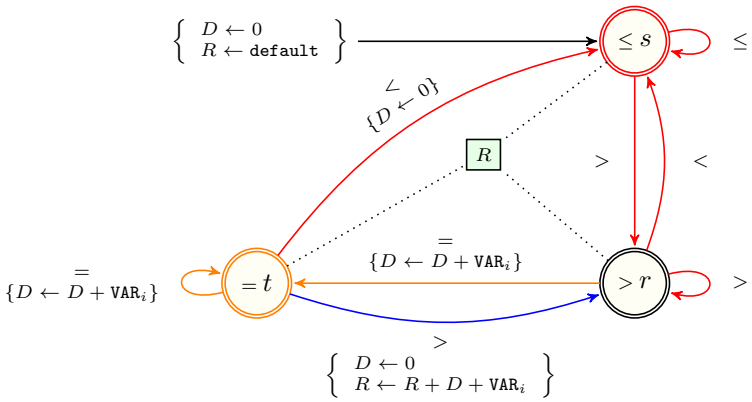


Figure 4.1355: Simplified automaton for the SUM_SURF_DECREASING_TERRACE constraint obtained by applying decoration Table 3.29 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C

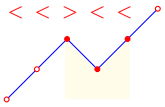
Table 4.321: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the SUM_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \mathtt{VAR}_{i+1}$ C

Table 4.322: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the SUM_SURF_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint SUM_SURF_DIP_ON_INCREASING_SEQUENCE(VALUE, VARIABLES)

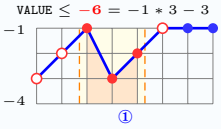
Arguments

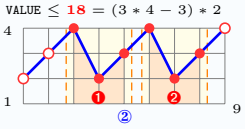
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 5 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(3 * \texttt{minv} + 3, (3 * \texttt{minv} + 3) * \texttt{np})$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(3 * \texttt{maxv} - 3①, (3 * \texttt{maxv} - 3) * \texttt{np}②)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{maxv} - 2, \texttt{maxv} - 1, \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - 3 * \texttt{np} - \max((\texttt{sv} - 3) * (\texttt{maxv} - 3), 0)$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[3, \texttt{sv} - 1], \langle \texttt{minv}, \texttt{minv} + 1, \texttt{minv} + 2 \rangle)$
$\texttt{n2} \geq \min((\texttt{sv} - 3) * (\texttt{minv} + 3), 0) - 3 * \texttt{np} - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor (\texttt{sv} - 3)/3 \rfloor)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$





Purpose VALUE is the sum of the surface of occurrences of the DIP_ON_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<>><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example $(19, \langle 1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4 \rangle)$

Figure 4.1356 provides an example where the SUM_SURF_DIP_ON_INCREASING_SEQUENCE $(19, [1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 5$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

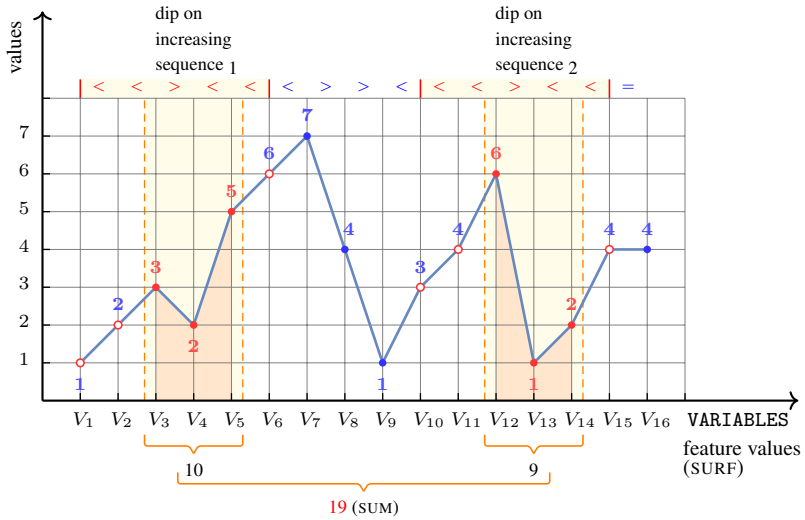


Figure 4.1356: Illustrating the SUM_SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1357 and 4.1358 respectively depict the automaton associated with the constraint SUM_SURF_DIP_ON_INCREASING_SEQUENCE and its simplified form.

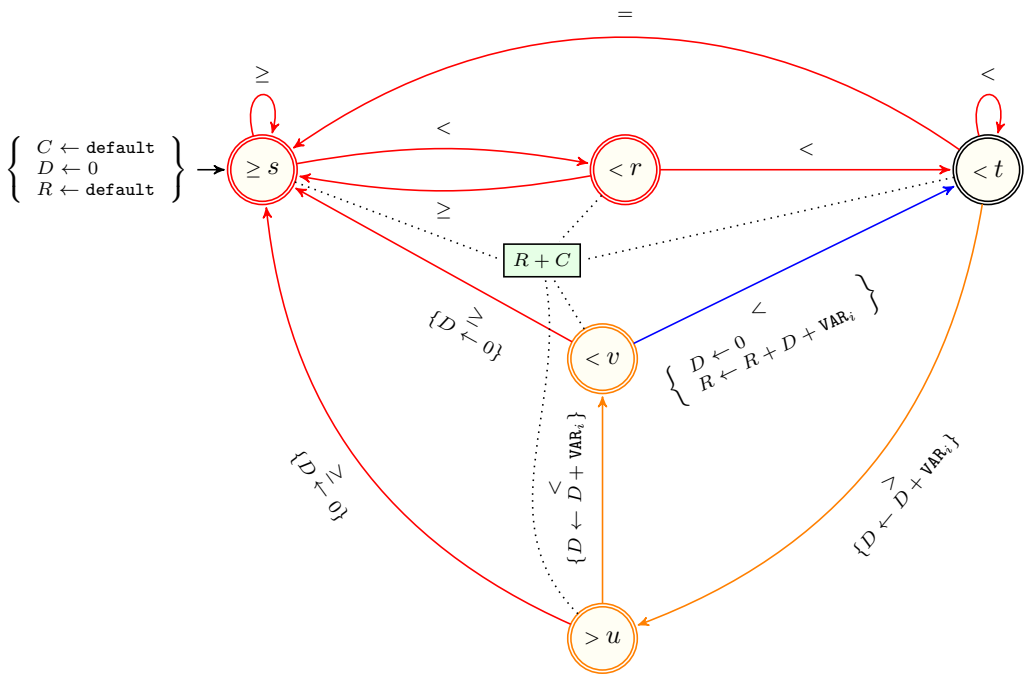


Figure 4.1357: Automaton for the SUM_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

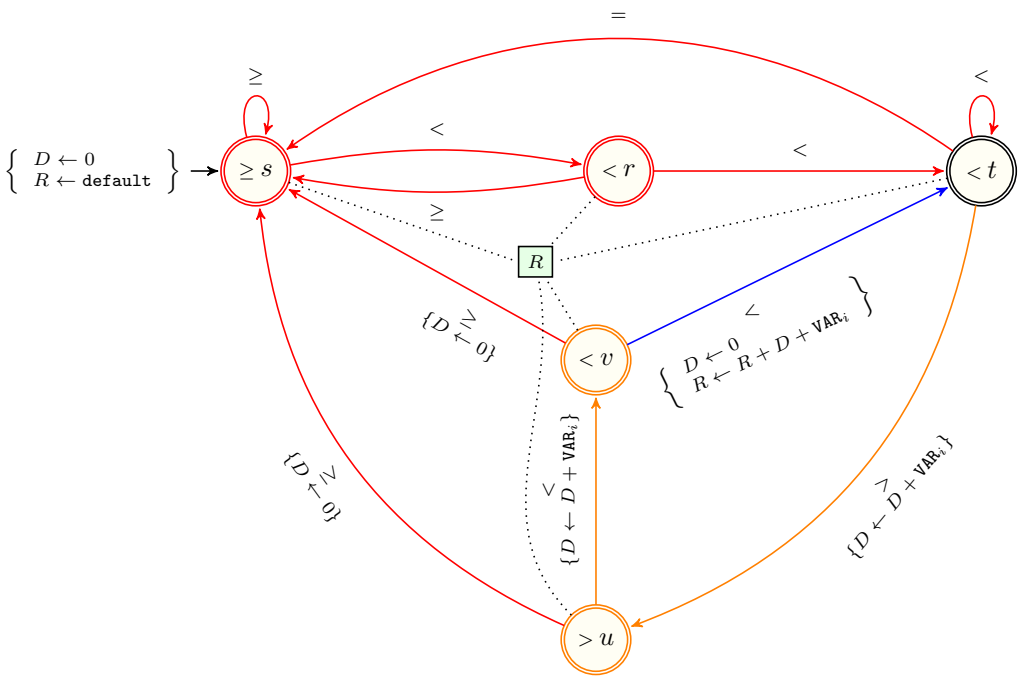
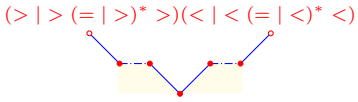


Figure 4.1358: Simplified automaton for the SUM_SURF_DIP_ON_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.26 to the seed transducer of the DIP_ON_INCREASING_SEQUENCE pattern where `default` is 0

AGGREGATOR FEATURE PATTERN

SUM_SURF_GORGE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the GORGE pattern.

Constraint SUM_SURF_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\mathtt{sv} \leq 2 \vee \mathtt{rv} \leq 1 \Rightarrow \mathtt{VALUE} = 0$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = 0 \vee \mathtt{VALUE} \geq \min(\mathtt{minv}①, \mathtt{minv} * \mathtt{np}②)$

$\mathtt{rv} \geq 3 \Rightarrow$
$\bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \geq \min(\mathtt{minv}③, \min((\mathtt{sv}-2) * (\mathtt{minv}+1) - 1④, \mathtt{minv} * \mathtt{np}⑤)) \end{array} \right)$

$\mathtt{rv} = 2 \Rightarrow \mathtt{VALUE} = 0 \vee \mathtt{VALUE} \leq \max(\mathtt{maxv}-1, (\mathtt{maxv}-1) * \mathtt{np})$

$\mathtt{rv} \geq 3 \Rightarrow$
$\bigvee \left(\begin{array}{l} \mathtt{VALUE} = 0, \\ \mathtt{VALUE} \leq \max(\mathtt{maxv}-1, \max((\mathtt{sv}-2) * (\mathtt{maxv}-1) - 1, (\mathtt{maxv}-1) * \mathtt{np})) \end{array} \right)$

$\mathtt{among}(\mathtt{n1}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{maxv}-1 \rangle)$

$\mathtt{rv} = 2 \vee \mathtt{maxv} = 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - \mathtt{np} * \max(0, \mathtt{maxv}-2)$

$\mathtt{rv} > 2 \wedge \mathtt{maxv} > 1 \Rightarrow \mathtt{n1} \geq \mathtt{VALUE} - (\mathtt{sv}-2) * (\mathtt{maxv}-2)$

$\mathtt{among}(\mathtt{n2}, \mathtt{VARIABLES}[2, \mathtt{sv}-1], \langle \mathtt{minv}, \mathtt{minv}+1 \rangle)$

$\mathtt{rv} = 2 \vee \mathtt{minv} = -1 \Rightarrow \mathtt{n2} \geq \mathtt{np} * \min(0, \mathtt{minv}+1) - \mathtt{VALUE}$

$\mathtt{rv} > 2 \wedge \mathtt{minv} < -1 \Rightarrow \mathtt{n2} \geq (\mathtt{sv}-2) * (\mathtt{minv}+2) - 1 - \mathtt{VALUE}$

$\mathtt{required}(\mathtt{VARIABLES}, \mathtt{var})$

where

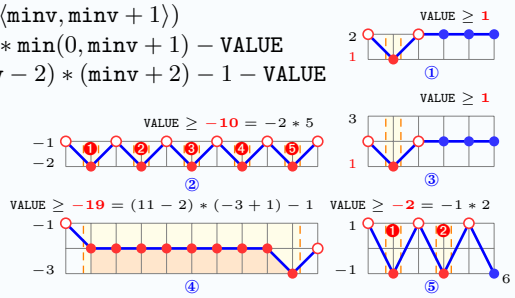
$\mathtt{sv} = |\mathtt{VARIABLES}|$

$\mathtt{np} = \max(0, \lfloor (\mathtt{sv}-1)/2 \rfloor)$

$\mathtt{minv} = \mathtt{minval}(\mathtt{VARIABLES.var})$

$\mathtt{maxv} = \mathtt{maxval}(\mathtt{VARIABLES.var})$

$\mathtt{rv} = \mathtt{range}(\mathtt{VARIABLES.var})$



Purpose

VALUE is the sum of the surface of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.

Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(25, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7 \rangle)$

Figure 4.1359 provides an example where the SUM_SURF_GORGE $(25, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

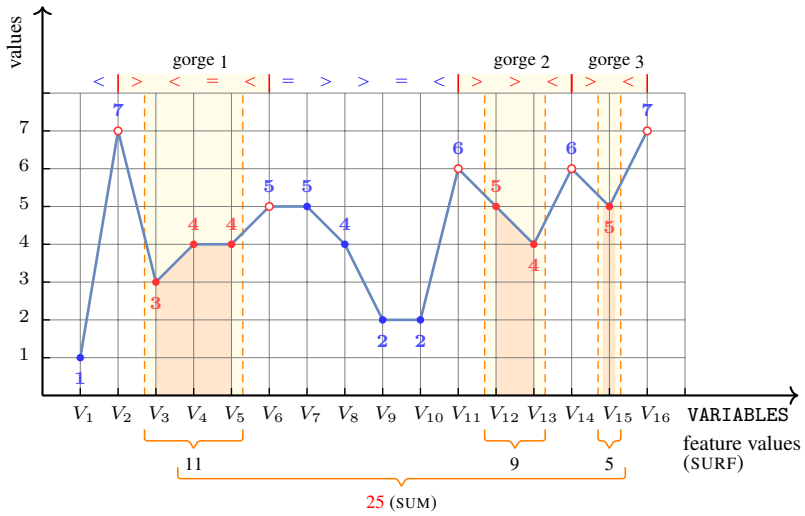


Figure 4.1359: Illustrating the SUM_SURF_GORGE constraint of the **Example** slot

Typical $|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1360 and 4.1361 respectively depict the automaton associated with the constraint SUM_SURF_GORGE and its simplified form.

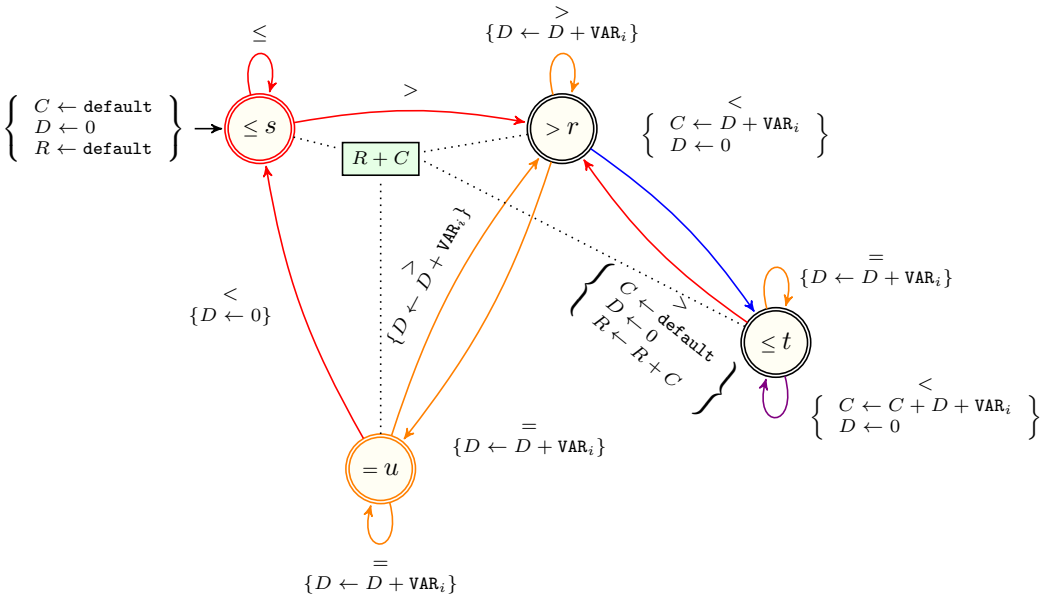


Figure 4.1360: Automaton for the SUM_SURF_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

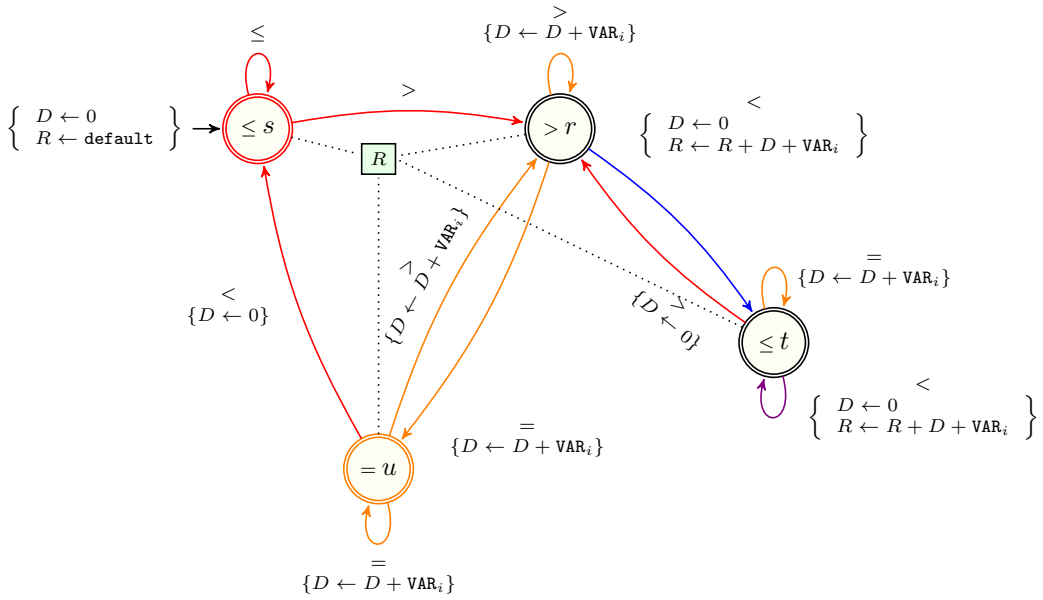


Figure 4.1361: Simplified automaton for the SUM_SURF_GORGE constraint obtained by applying decoration Table 3.26 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.323: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the SUM_SURF_GORGE constraint defined as the composition of the GORGE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	0

Table 4.324: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the SUM_SURF_GORGE constraint defined as the composition of the GORGE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_INCREASING

DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint SUM_SURF_INCREASING(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\texttt{VALUE} \geq \min_{\texttt{q} \in [\texttt{lb1},\texttt{ub1}]} \texttt{min} \left(0, \sum \left(\prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1\right)\right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ \sum \left(\begin{array}{l} 2*\texttt{minv}-1, \\ 2*\texttt{np} \end{array} \right) \end{array} \right), \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np}\right)\right), \\ 2*\texttt{minv}+1, \\ \texttt{q}-1 \end{array} \right) \right) \right) \quad ①$$

$$\texttt{VALUE} \leq \max_{\texttt{q} \in [\texttt{lb2},\texttt{ub2}]} \texttt{max} \left(0, \sum \left(\prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, \texttt{np}-1\right)\right), \\ \texttt{sv} \bmod \texttt{np} * \texttt{q}, \\ 2*\texttt{maxv}+1- \\ 2*\texttt{np} \end{array} \right), \prod \left(\begin{array}{l} \texttt{max}\left(0, \texttt{min}\left(1, 2-\texttt{np}\right)\right), \\ 2*\texttt{maxv}-1, \\ \texttt{q}-1 \end{array} \right) \right) \right)$$

$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where

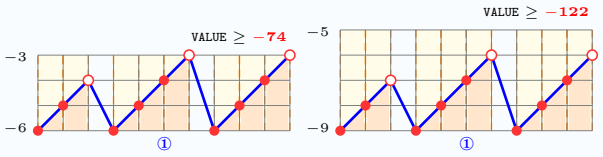
$\texttt{sv} = |\texttt{VARIABLES}|$

$\texttt{np} = \lfloor \texttt{sv}/\texttt{q} \rfloor$

$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$

$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$

$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



$$\texttt{lb1} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{minv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub1} = \lfloor \texttt{sv}/2 \rfloor + 1$$

$$\texttt{lb2} = \texttt{min} \left(\begin{array}{l} \lfloor \texttt{sv}/2 \rfloor + 1, \\ \sum \left(\begin{array}{l} \lfloor \texttt{sv}/\texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1) \rfloor, \\ \texttt{min}(1, \texttt{sv} \bmod \texttt{min}(\texttt{min}(\texttt{sv},\texttt{rv}), |\texttt{maxv}|+1)) \end{array} \right) \end{array} \right)$$

$$\texttt{ub2} = \lfloor \texttt{sv}/2 \rfloor + 1$$

Purpose

VALUE is the sum of the surface of occurrences of the INCREASING pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '<'.

Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example

$(33, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1362 provides an example where the SUM_SURF_INCREASING $(33, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

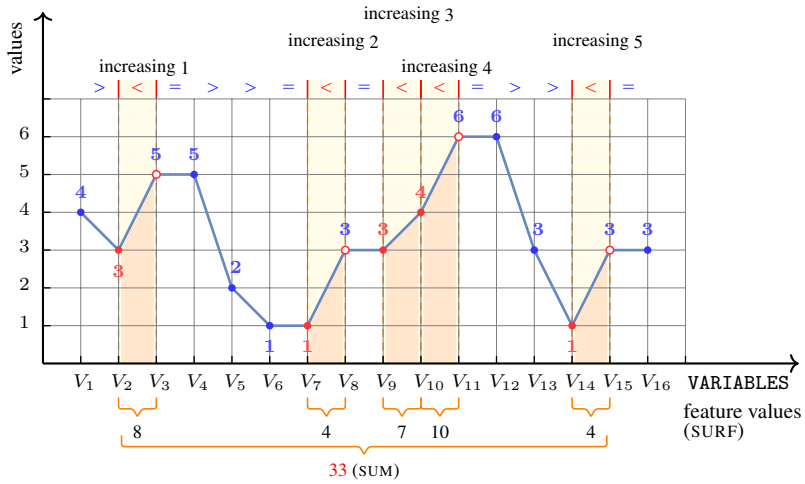


Figure 4.1362: Illustrating the SUM_SURF_INCREASING constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1363 and 4.1364 respectively depict the automaton associated with the constraint SUM_SURF_INCREASING and its simplified form.

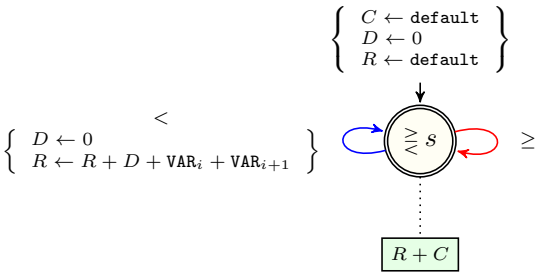


Figure 4.1363: Automaton for the SUM_SURF_INCREASING constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING pattern where `default` is 0

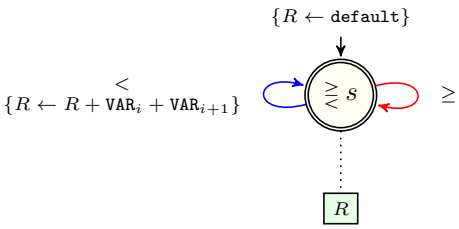


Figure 4.1364: Simplified automaton for the SUM_SURF_INCREASING constraint obtained by applying decoration Table 3.40 to the seed transducer of the INCREASING pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

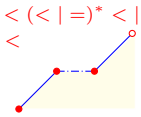
Table 4.325: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the SUM_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s
s	0

Table 4.326: Concrete glue matrix, derived from the parametrised glue matrix 3.8, for the simplified automaton of the SUM_SURF_INCREASING constraint defined as the composition of the INCREASING pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint SUM_SURF_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

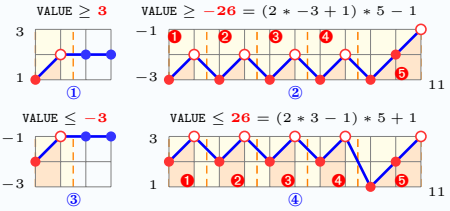
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(2 * \texttt{minv} + 1, (2 * \texttt{minv} + 1) * \texttt{np})$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min \left(\begin{array}{l} 2 * \texttt{minv} + 1①, \\ (2 * \texttt{minv} + 1) * \texttt{np} + \min(0, \texttt{sv} \bmod 2 * (\texttt{minv} + 2))② \end{array} \right) \end{array} \right)$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv} - 1, (2 * \texttt{maxv} - 1) * \texttt{np})$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max \left(\begin{array}{l} 2 * \texttt{maxv} - 1③, \\ (2 * \texttt{maxv} - 1) * \texttt{np} + \max(0, \texttt{sv} \bmod 2 * (\texttt{maxv} - 2))④ \end{array} \right) \end{array} \right)$$
$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$$
$$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{np} * (2 * \texttt{maxv} - 3))$$
$$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{np} - \texttt{sv} * (\texttt{maxv} - 2)$$
$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$$
$$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \min(0, \texttt{np} * (2 * \texttt{minv} + 3)) - \texttt{VALUE}$$
$$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq \texttt{sv} * (\texttt{minv} + 2) - \texttt{np} - \texttt{VALUE}$$
$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$$\texttt{sv} = |\texttt{VARIABLES}|$$
$$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$$
$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$
$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$
$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose

VALUE is the sum of the surface of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.

Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(29, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1365 provides an example where the SUM_SURF_INCREASING_SEQUENCE $(29, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

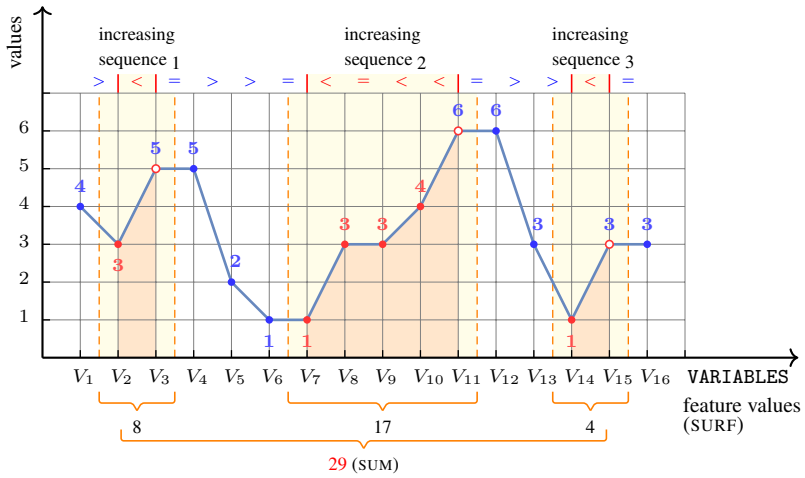


Figure 4.1365: Illustrating the SUM_SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1366 and 4.1367 respectively depict the automaton associated with the constraint SUM_SURF_INCREASING_SEQUENCE and its simplified form.

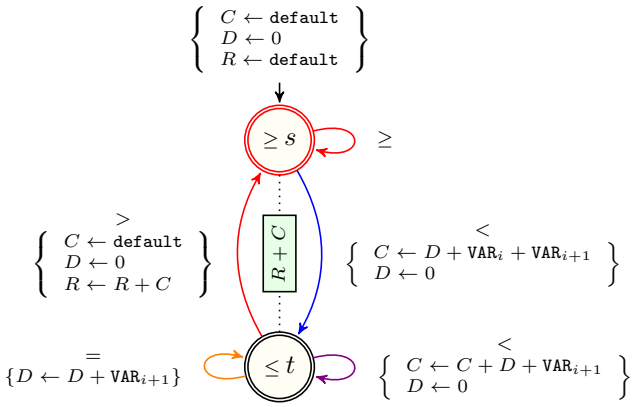


Figure 4.1366: Automaton for the SUM_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

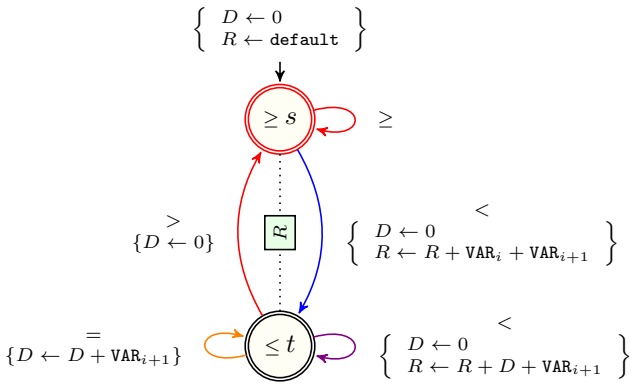


Figure 4.1367: Simplified automaton for the SUM_SURF_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.30 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.327: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the SUM_SURF_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

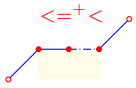
	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.328: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the SUM_SURF_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

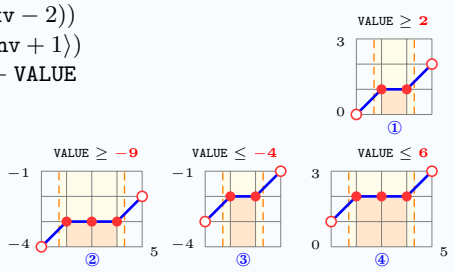
Constraint SUM_SURF_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(2 * (\texttt{maxv} - 1)③, (\texttt{sv} - 2) * (\texttt{maxv} - 1)④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \max(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \min(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the sum of the surface of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(19, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.1368 provides an example where the SUM_SURF_INCREASING_TERRACE $(19, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 2$

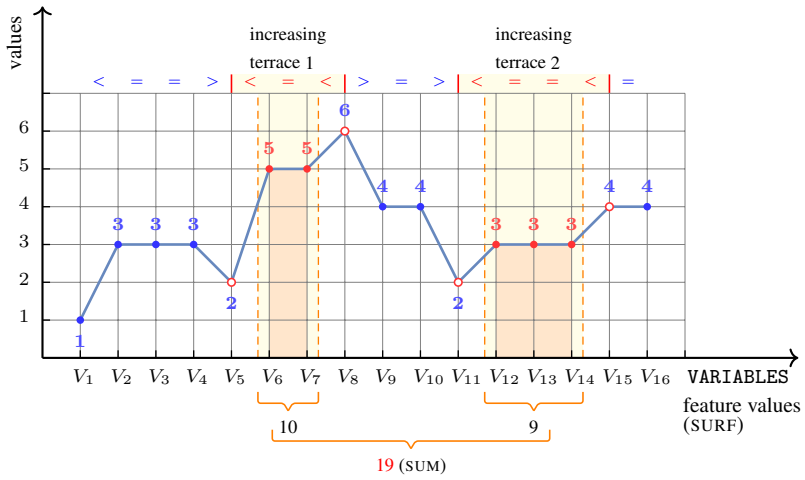


Figure 4.1368: Illustrating the SUM_SURF_INCREASING_TERRACE constraint of the **Example** slot

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1369 and 4.1370 respectively depict the automaton associated with the constraint SUM_SURF_INCREASING_TERRACE and its simplified form.

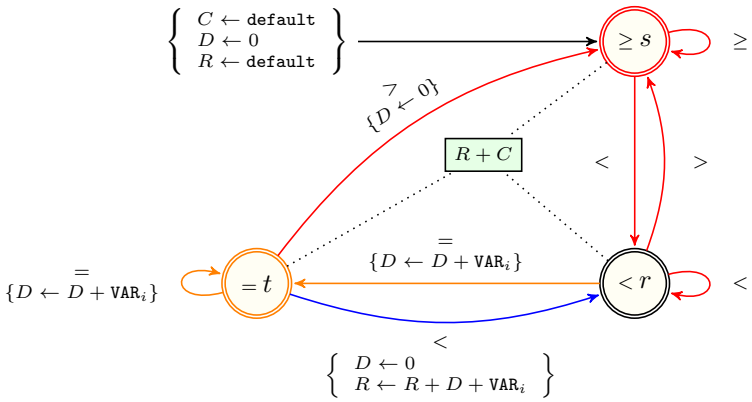


Figure 4.1369: Automaton for the SUM_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0

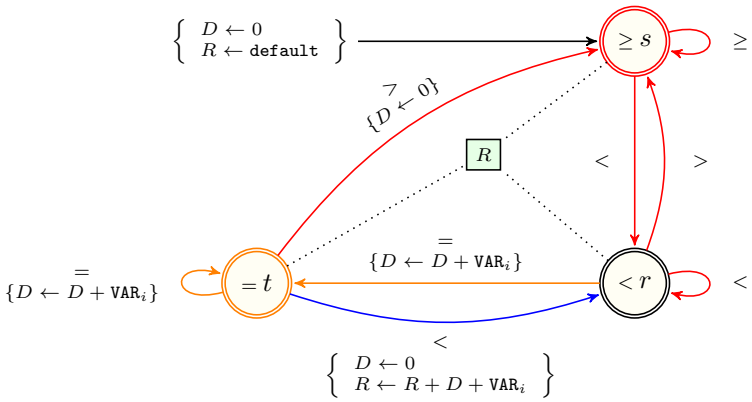


Figure 4.1370: Simplified automaton for the SUM_SURF_INCREASING_TERRACE constraint obtained by applying decoration Table 3.29 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.329: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the SUM_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

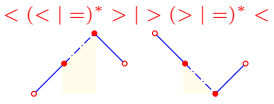
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.330: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the SUM_SURF_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_INFLEXION

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the INFLEXION pattern.

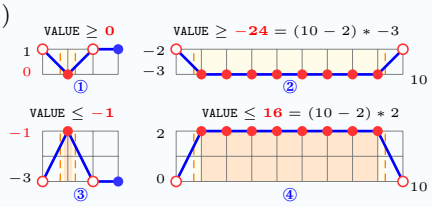
Constraint SUM_SURF_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, (\texttt{sv} - 2) * \texttt{minv}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, (\texttt{sv} - 2) * \texttt{maxv}④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the sum of the surface of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(49, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.1371 provides an example where the SUM_SURF_INFLEXION $(49, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

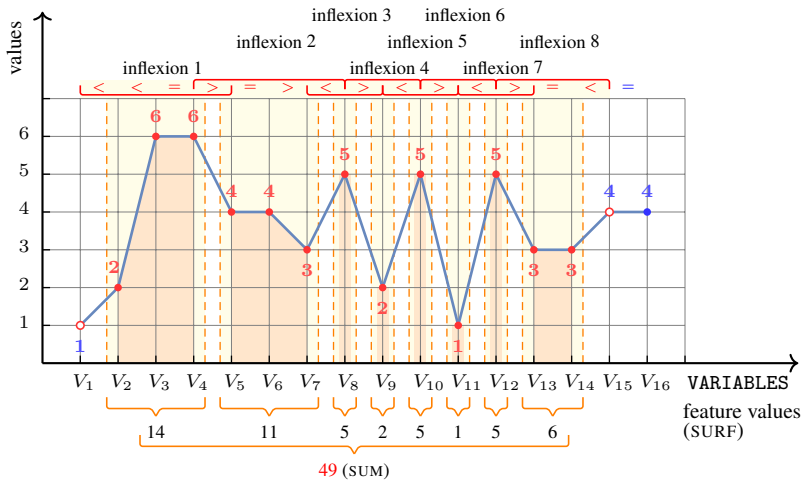


Figure 4.1371: Illustrating the SUM_SURF_INFLEXION constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1372 and 4.1373 respectively depict the automaton associated with the constraint SUM_SURF_INFLEXION and its simplified form.

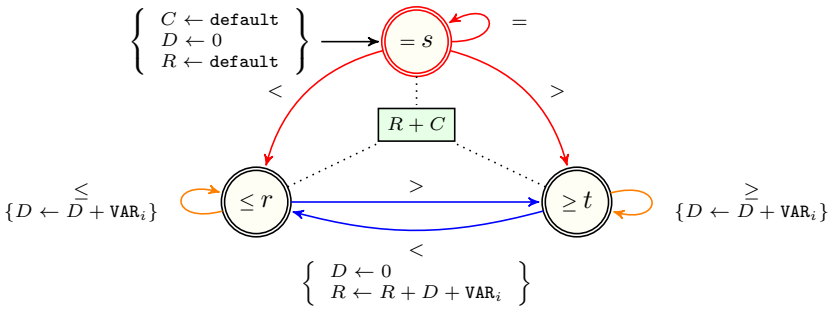


Figure 4.1372: Automaton for the SUM_SURF_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

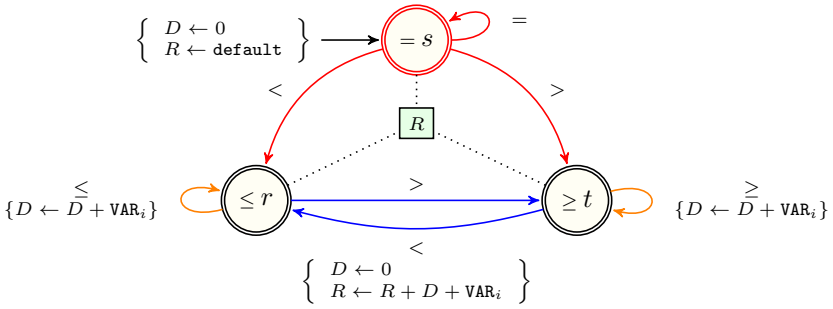
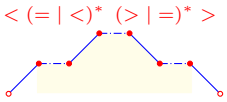


Figure 4.1373: Simplified automaton for the SUM_SURF_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

AGGREGATOR FEATURE PATTERN

SUM_SURF_PEAK

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

Constraint SUM_SURF_PEAK(VALUE, VARIABLES)

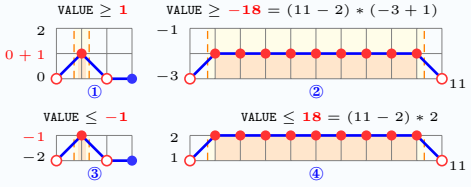
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, (\texttt{sv} - 2) * \texttt{maxv}④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the sum of the surface of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(32, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1 \rangle)$

Figure 4.1374 provides an example where the SUM_SURF_PEAK $(32, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

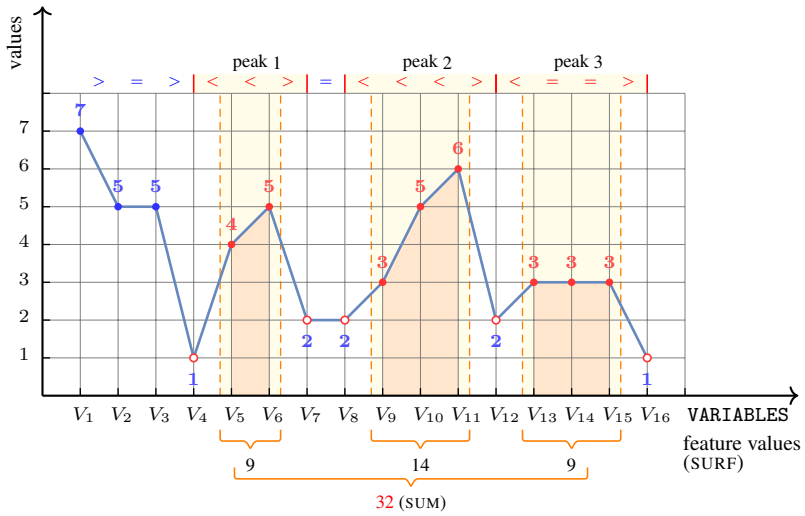


Figure 4.1374: Illustrating the SUM_SURF_PEAK constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1375 and 4.1376 respectively depict the automaton associated with the constraint SUM_SURF_PEAK and its simplified form.

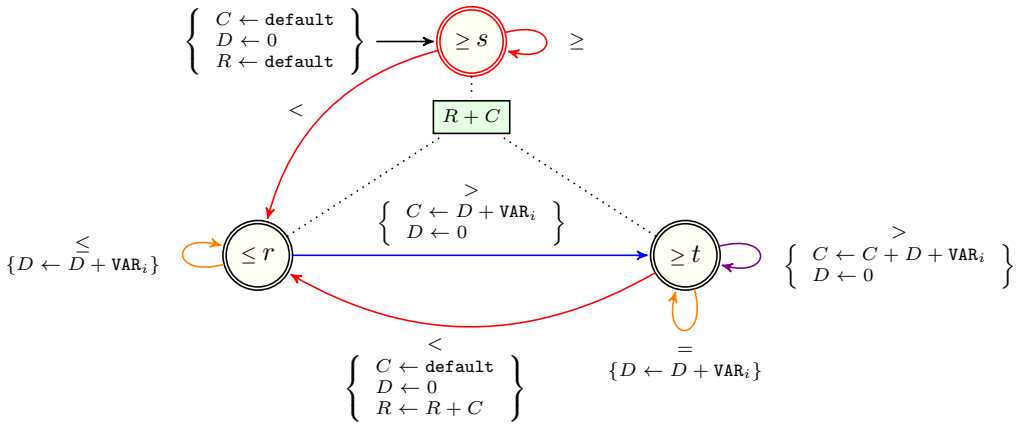


Figure 4.1375: Automaton for the SUM_SURF_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is 0

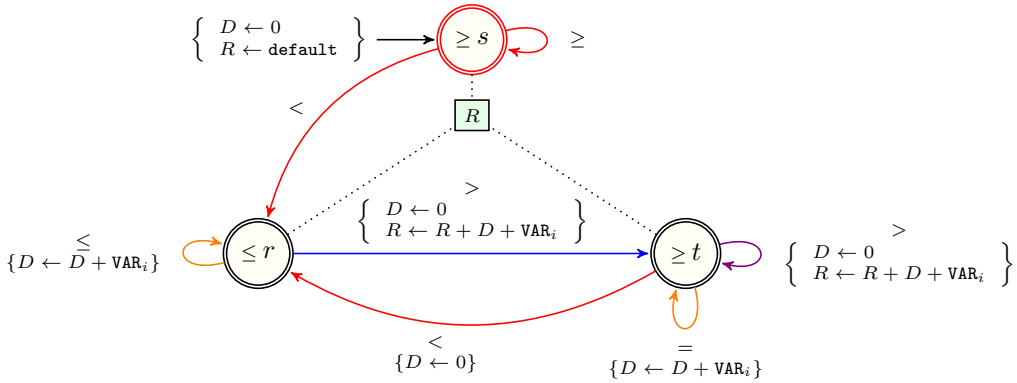


Figure 4.1376: Simplified automaton for the SUM_SURF_PEAK constraint obtained by applying decoration Table 3.26 to the seed transducer of the PEAK pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.331: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the SUM_SURF_PEAK constraint defined as the composition of the PEAK pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	0

Table 4.332: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the SUM_SURF_PEAK constraint defined as the composition of the PEAK pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint SUM_SURF_PLAIN(VALUE, VARIABLES)

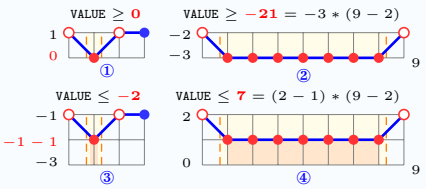
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * (\texttt{sv} - 2)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * (\texttt{sv} - 2)④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{c} = \texttt{min}(1, \texttt{VALUE} \bmod (\texttt{sv} - 2))$



Purpose VALUE is the sum of the surface of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^* <$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(15, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3 \rangle)$

Figure 4.1377 provides an example where the SUM_SURF_PLAIN $(15, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

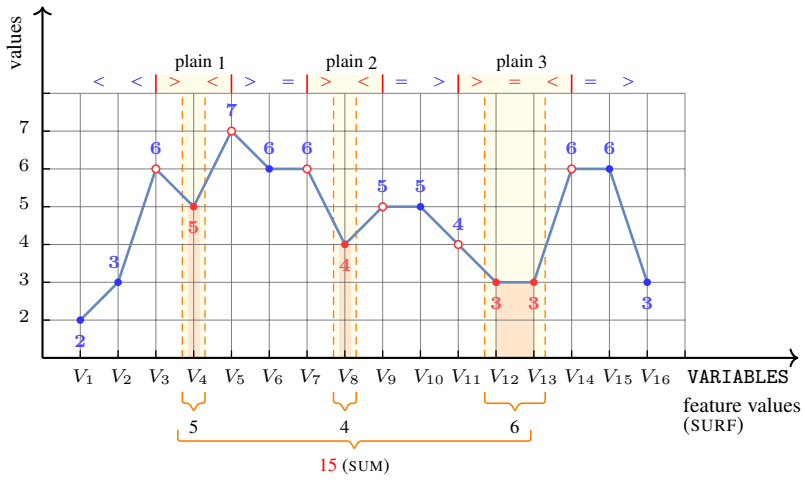


Figure 4.1377: Illustrating the SUM_SURF_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1378 and 4.1379 respectively depict the automaton associated with the constraint SUM_SURF_PLAIN and its simplified form.

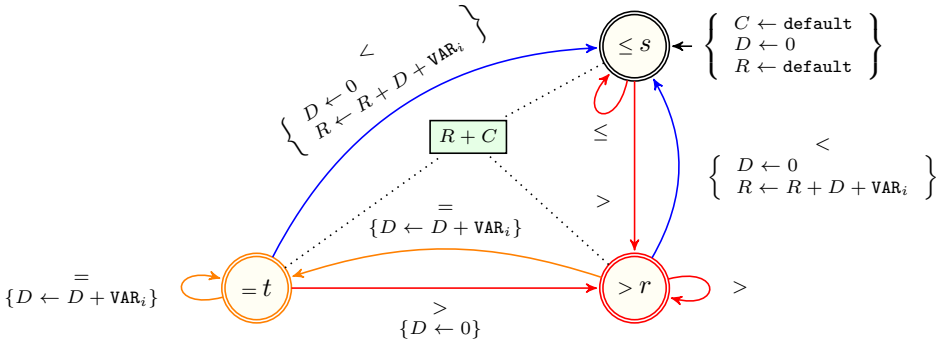


Figure 4.1378: Automaton for the SUM_SURF_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is 0

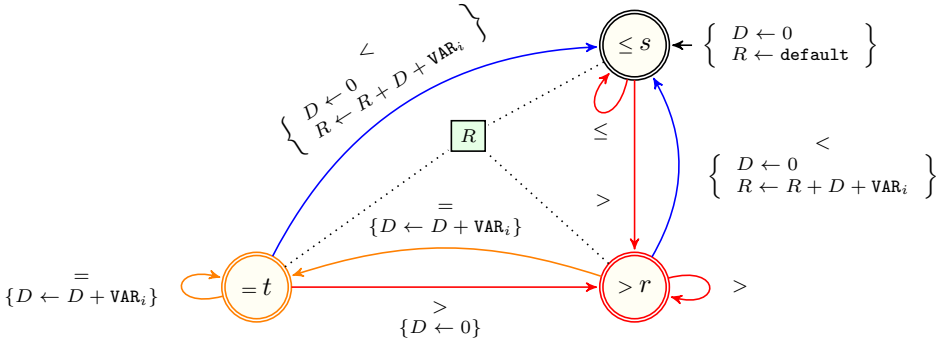


Figure 4.1379: Simplified automaton for the SUM_SURF_PLAIN constraint obtained by applying decoration Table 3.29 to the seed transducer of the PLAIN pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.333: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the SUM_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.334: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the SUM_SURF_PLAIN constraint defined as the composition of the PLAIN pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_PLATEAU

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

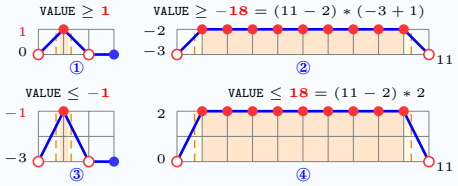
Constraint SUM_SURF_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv} + 1①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv}③, (\texttt{sv} - 2) * \texttt{maxv}④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{c} = \texttt{min}(1, \texttt{VALUE} \bmod (\texttt{sv} - 2))$



Purpose VALUE is the sum of the surface of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(17, \langle 7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5 \rangle)$

Figure 4.1380 provides an example where the SUM_SURF_PLATEAU $(17, [7, 5, 2, 3, 1, 2, 2, 4, 3, 3, 4, 5, 5, 2, 2, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

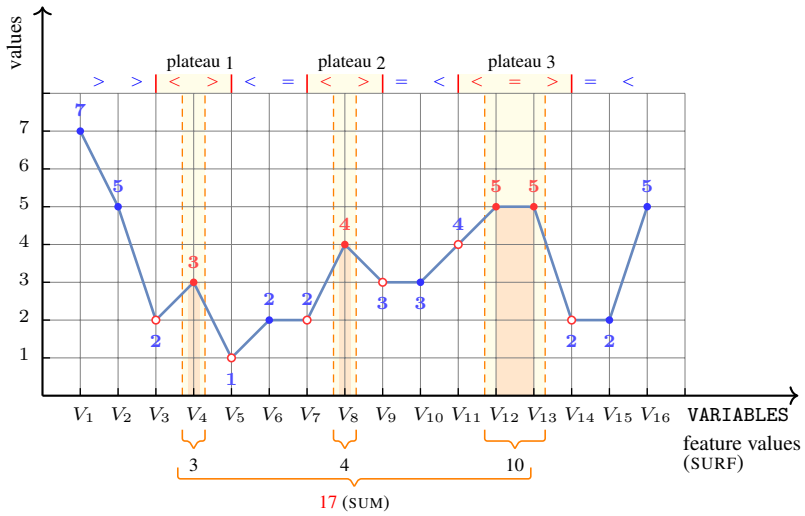


Figure 4.1380: Illustrating the SUM_SURF_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1381 and 4.1382 respectively depict the automaton associated with the constraint SUM_SURF_PLATEAU and its simplified form.

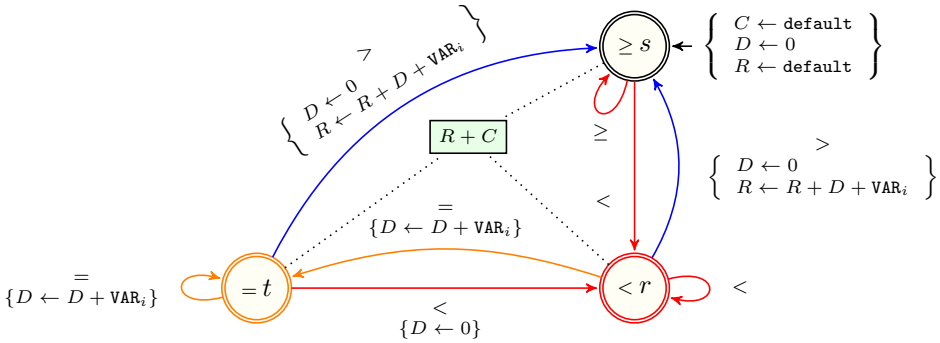


Figure 4.1381: Automaton for the SUM_SURF_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is 0

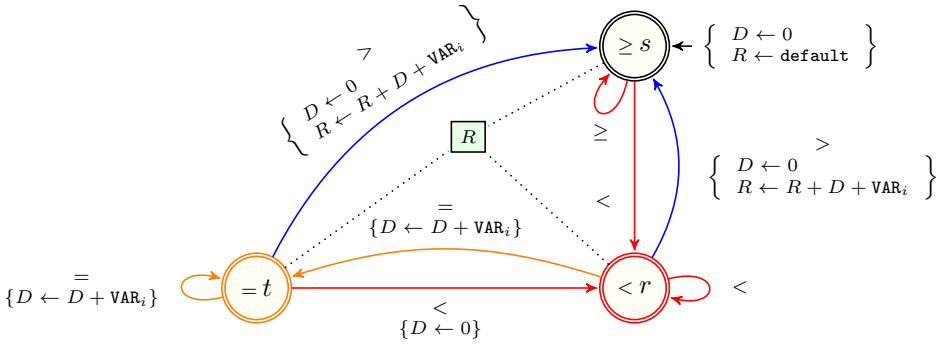


Figure 4.1382: Simplified automaton for the SUM_SURF_PLATEAU constraint obtained by applying decoration Table 3.29 to the seed transducer of the PLATEAU pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.335: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the SUM_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

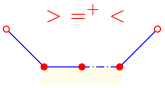
	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.336: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the SUM_SURF_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

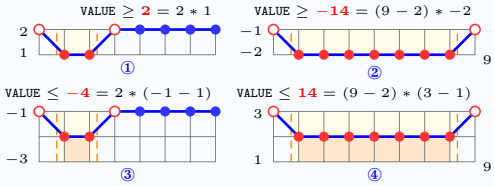
Constraint SUM_SURF_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}①, (\texttt{sv} - 2) * \texttt{minv}②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(2 * (\texttt{maxv} - 1)③, (\texttt{sv} - 2) * (\texttt{maxv} - 1)④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$
where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{c} = \texttt{min}(1, \texttt{VALUE} \bmod (\texttt{sv} - 2))$



Purpose

VALUE is the sum of the surface of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$> =^{+} <$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(27, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5 \rangle)$

Figure 4.1383 provides an example where the SUM_SURF_PROPER_PLAIN $(27, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

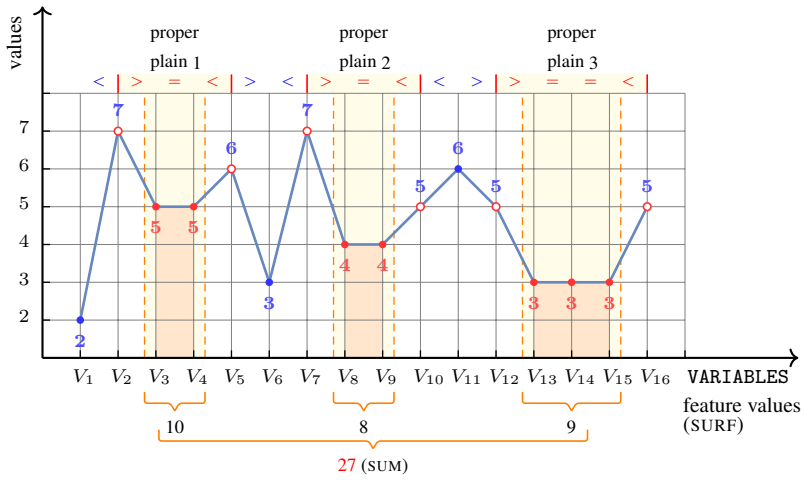


Figure 4.1383: Illustrating the SUM_SURF_PROPER_PLAIN constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1384 and 4.1385 respectively depict the automaton associated with the constraint SUM_SURF_PROPER_PLAIN and its simplified form.

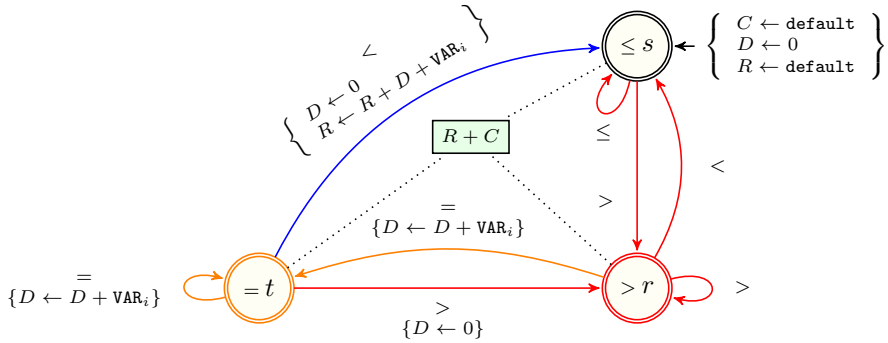


Figure 4.1384: Automaton for the SUM_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

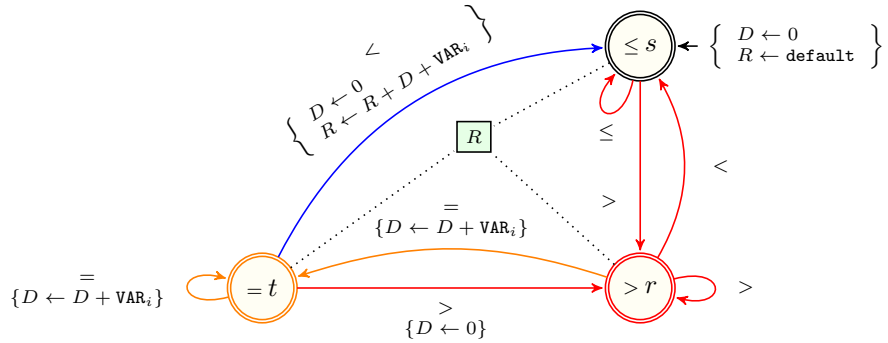


Figure 4.1385: Simplified automaton for the SUM_SURF_PROPER_PLAIN constraint obtained by applying decoration Table 3.29 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.337: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the SUM_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

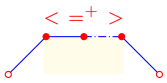
	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.338: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the SUM_SURF_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint SUM_SURF_PROPER_PLATEAU(VALUE, VARIABLES)

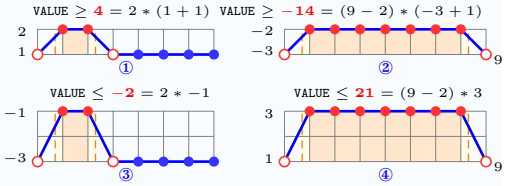
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(2 * (\texttt{minv} + 1)①, (\texttt{sv} - 2) * (\texttt{minv} + 1)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}③, (\texttt{sv} - 2) * \texttt{maxv}④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2 - \texttt{c}) * (\texttt{minv} + 2)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{c} = \texttt{min}(1, \texttt{VALUE} \bmod (\texttt{sv} - 2))$



Purpose

VALUE is the sum of the surface of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(29, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3 \rangle)$

Figure 4.1386 provides an example where the SUM_SURF_PROPER_PLATEAU $(29, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

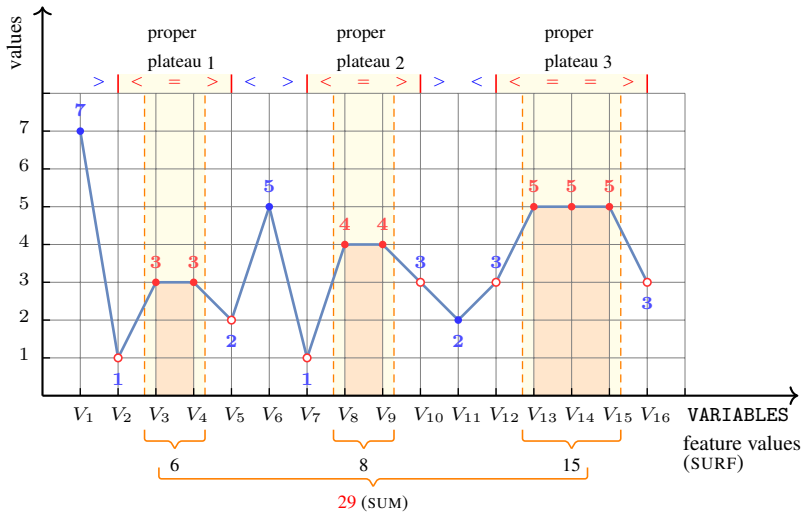


Figure 4.1386: Illustrating the SUM_SURF_PROPER_PLATEAU constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1387 and 4.1388 respectively depict the automaton associated with the constraint SUM_SURF_PROPER_PLATEAU and its simplified form.

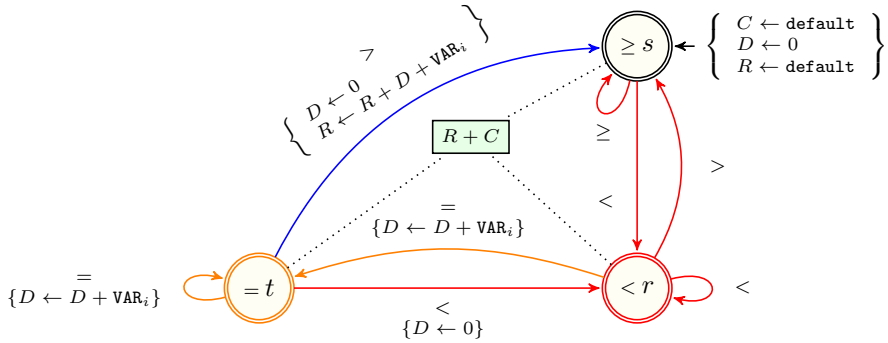


Figure 4.1387: Automaton for the SUM_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

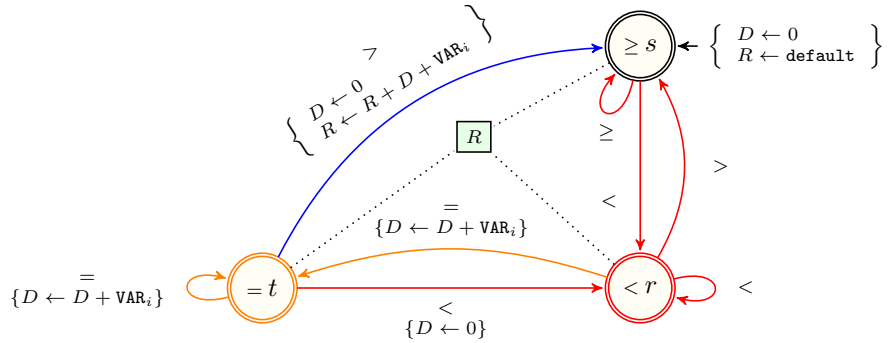


Figure 4.1388: Simplified automaton for the SUM_SURF_PROPER_PLATEAU constraint obtained by applying decoration Table 3.29 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.339: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the SUM_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C

Table 4.340: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the SUM_SURF_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_STEADY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STEADY pattern.

Constraint SUM_SURF_STEADY(VALUE, VARIABLES)

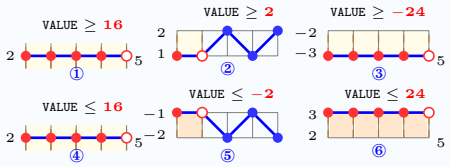
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq 2 * \texttt{minv} * \texttt{np}$①
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}$②$, 2 * \texttt{minv} * \texttt{np}$③$)$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{maxv} * \texttt{np}$④
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}$⑤$, 2 * \texttt{maxv} * \texttt{np}$⑥$)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{np} - \texttt{max}(0, (2 * \texttt{sv} - 1) * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (2 * \texttt{sv} - 1) * (\texttt{minv} + 1)) - \texttt{np} - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \texttt{max}(0, \texttt{sv} - 1)$
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of the surface of occurrences of the STEADY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(60, \langle 1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6 \rangle)$

Figure 4.1389 provides an example where the SUM_SURF_STEADY $(60, [1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

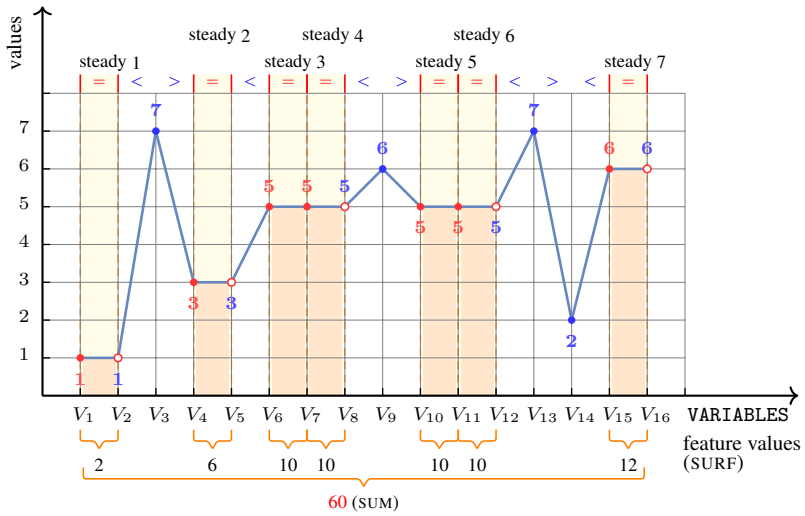


Figure 4.1389: Illustrating the SUM_SURF_STEADY constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1390 and 4.1391 respectively depict the automaton associated with the constraint SUM_SURF_STEADY and its simplified form.

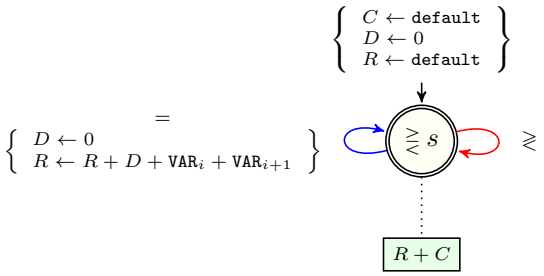


Figure 4.1390: Automaton for the SUM_SURF_STEADY constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY pattern where `default` is 0

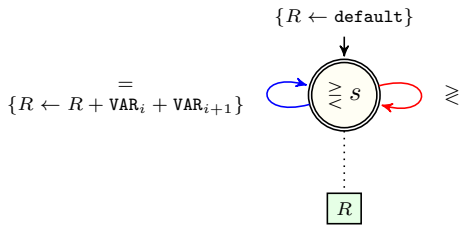


Figure 4.1391: Simplified automaton for the SUM_SURF_STEADY constraint obtained by applying decoration Table 3.40 to the seed transducer of the STEADY pattern where `default` is 0

	s
s	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.341: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the SUM_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

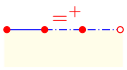
	s
s	0

Table 4.342: Concrete glue matrix, derived from the parametrised glue matrix 3.16, for the simplified automaton of the SUM_SURF_STEADY constraint defined as the composition of the STEADY pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint SUM_SURF_STEADY_SEQUENCE(VALUE, VARIABLES)

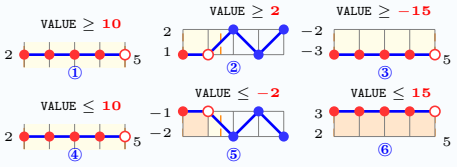
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \geq \texttt{minv} * \texttt{sv}①$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv}②, \texttt{minv} * \texttt{sv}③)$
$\texttt{rv} = 1 \wedge \texttt{sv} \geq 2 \Rightarrow \texttt{VALUE} \leq \texttt{maxv} * \texttt{sv}④$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv}⑤, \texttt{maxv} * \texttt{sv}⑥)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{sv} * (\texttt{maxv} - 1))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, \texttt{sv} * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose VALUE is the sum of the surface of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(31, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.1392 provides an example where the SUM_SURF_STEADY_SEQUENCE $(31, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

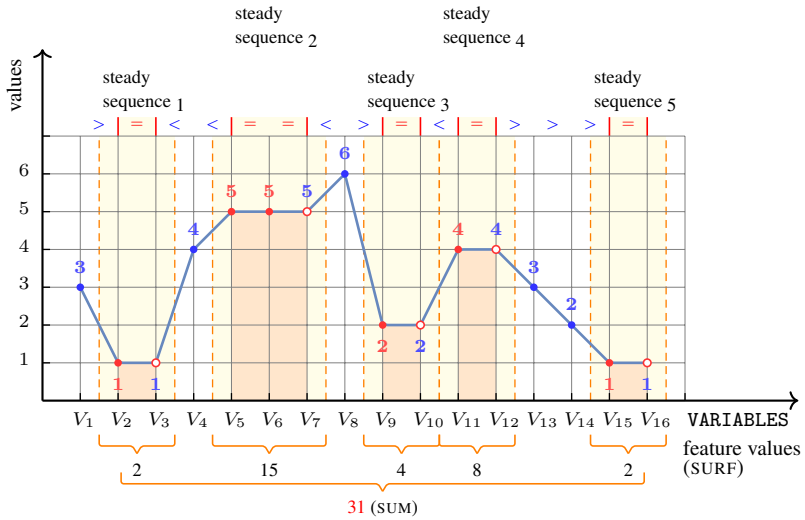


Figure 4.1392: Illustrating the SUM_SURF_STEADY_SEQUENCE constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1393 and 4.1394 respectively depict the automaton associated with the constraint SUM_SURF_STEADY_SEQUENCE and its simplified form.

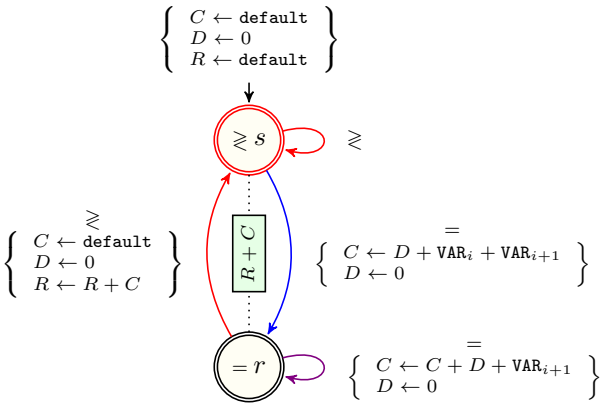


Figure 4.1393: Automaton for the SUM_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

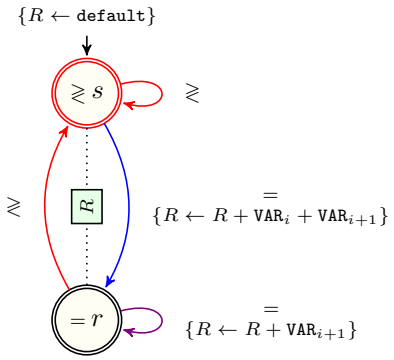


Figure 4.1394: Simplified automaton for the SUM_SURF_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STEADY_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.343: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the SUM_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	$-\texttt{VAR}_{i+1}$ M

Table 4.344: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the SUM_SURF_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint SUM_SURF_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

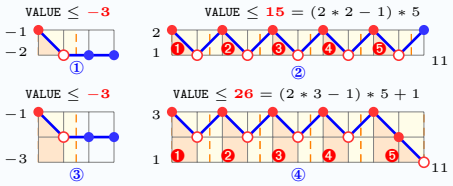
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(2 * \texttt{minv} + 1, (2 * \texttt{minv} + 1) * \texttt{np})$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \texttt{min} \left(\begin{array}{l} 2 * \texttt{minv} + 1, \\ (2 * \texttt{minv} + 1) * \texttt{np} + \texttt{min}(0, \texttt{sv} \bmod 2 * (\texttt{minv} + 2)) \end{array} \right) \end{array} \right)$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(2 * \texttt{maxv} - 1①, (2 * \texttt{maxv} - 1) * \texttt{np}②)$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \texttt{max} \left(\begin{array}{l} 2 * \texttt{maxv} - 1③, \\ (2 * \texttt{maxv} - 1) * \texttt{np} + \texttt{max}(0, \texttt{sv} \bmod 2 * (\texttt{maxv} - 2))④ \end{array} \right) \end{array} \right)$$
$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$$
$$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, \texttt{np} * (\texttt{maxv} - 2)) - \texttt{np}$$
$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$$
$$\texttt{n2} \geq \texttt{min}(0, \texttt{np} * (\texttt{minv} + 2)) - \texttt{np} - \texttt{VALUE}$$
$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$$\texttt{sv} = |\texttt{VARIABLES}|$$
$$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$$
$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$
$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$
$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose

VALUE is the sum of the surface of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.

An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.

Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(31, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1395 provides an example where the SUM_SURF_STRICTLY_DECREASING_SEQUENCE $(31, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

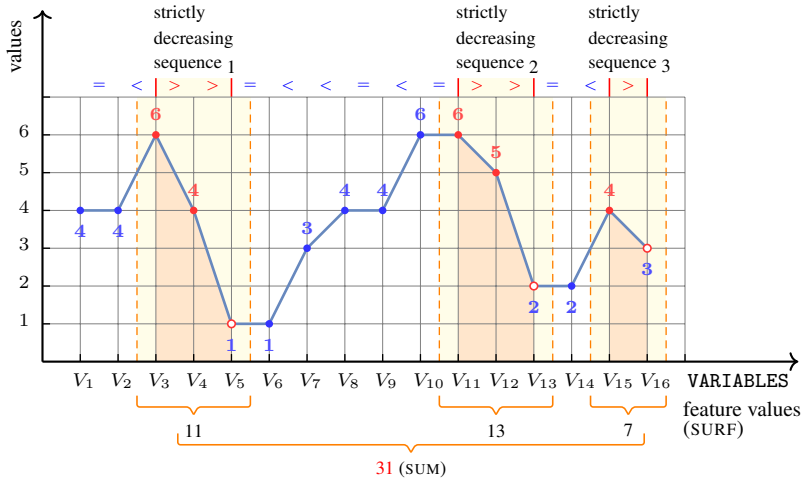


Figure 4.1395: Illustrating the SUM_SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Typical $|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1396 and 4.1397 respectively depict the automaton associated with the constraint SUM_SURF_STRICTLY_DECREASING_SEQUENCE and its simplified form.

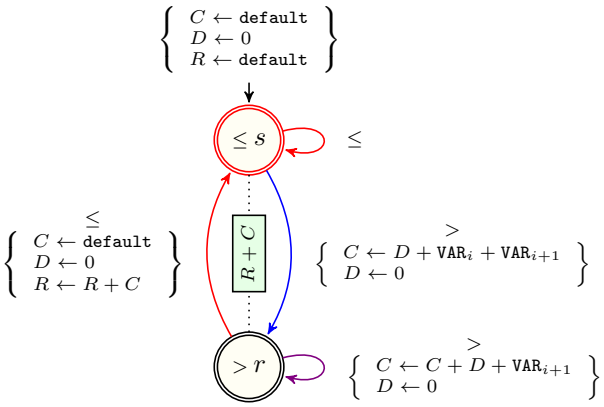


Figure 4.1396: Automaton for the SUM_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

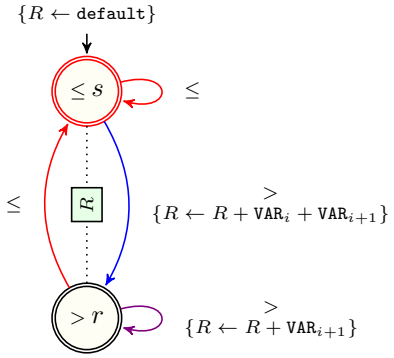


Figure 4.1397: Simplified automaton for the SUM_SURF_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.345: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the SUM_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	$-\texttt{VAR}_{i+1}$ M

Table 4.346: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the SUM_SURF_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_STRICTLY_INCREASING_SEQUENCE

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint SUM_SURF_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

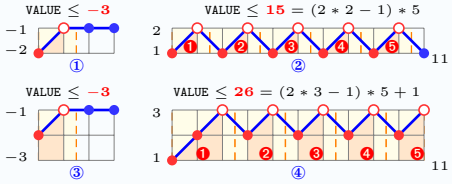
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(2 * \texttt{minv} + 1, (2 * \texttt{minv} + 1) * \texttt{np})$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min \left(\begin{array}{l} 2 * \texttt{minv} + 1, \\ (2 * \texttt{minv} + 1) * \texttt{np} + \min(0, \texttt{sv} \bmod 2 * (\texttt{minv} + 2)) \end{array} \right) \end{array} \right)$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(2 * \texttt{maxv} - 1①, (2 * \texttt{maxv} - 1) * \texttt{np}②)$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max \left(\begin{array}{l} 2 * \texttt{maxv} - 1③, \\ (2 * \texttt{maxv} - 1) * \texttt{np} + \max(0, \texttt{sv} \bmod 2 * (\texttt{maxv} - 2))④ \end{array} \right) \end{array} \right)$$
$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$$
$$\texttt{n1} \geq \texttt{VALUE} - \max(0, \texttt{np} * (\texttt{maxv} - 2)) - \texttt{np}$$
$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[1, \texttt{sv}], \langle \texttt{minv}, \texttt{minv} + 1 \rangle)$$
$$\texttt{n2} \geq \min(0, \texttt{np} * (\texttt{minv} + 2)) - \texttt{np} - \texttt{VALUE}$$
$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$$\texttt{sv} = |\texttt{VARIABLES}|$$
$$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$$
$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$
$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$
$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose

VALUE is the sum of the surface of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example $(30, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1398 provides an example where the SUM_SURF_STRICTLY_INCREASING_SEQUENCE $(30, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

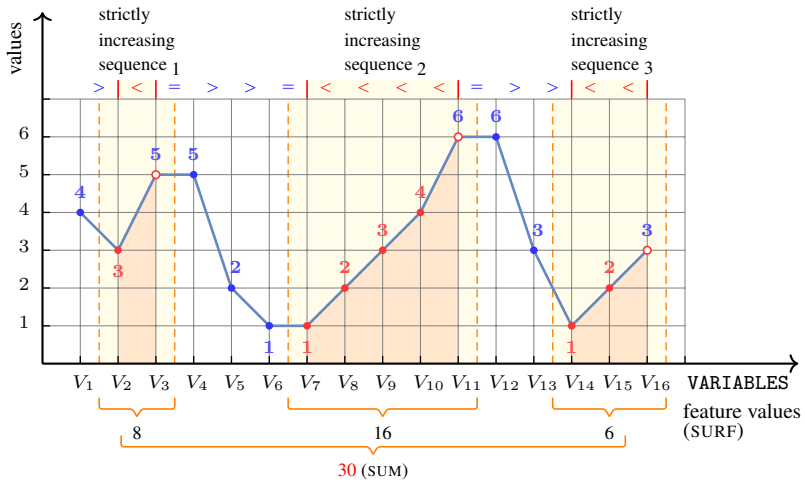


Figure 4.1398: Illustrating the SUM_SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Typical $|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Arg. properties Functional dependency: `VALUE` determined by `VARIABLES`.

Automaton

Figures 4.1399 and 4.1400 respectively depict the automaton associated with the constraint SUM_SURF_STRICTLY_INCREASING_SEQUENCE and its simplified form.

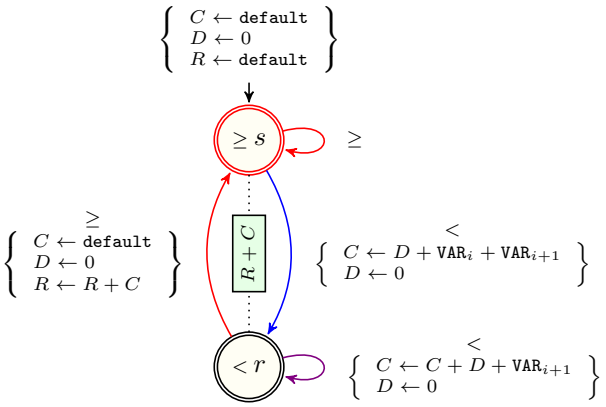


Figure 4.1399: Automaton for the SUM_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

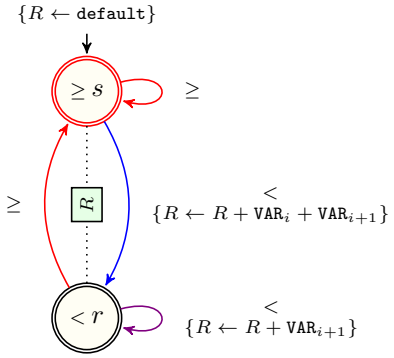


Figure 4.1400: Simplified automaton for the SUM_SURF_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - \texttt{VAR}_{i+1}$ M

Table 4.347: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the SUM_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	$-\texttt{VAR}_{i+1}$ M

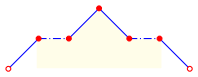
Table 4.348: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the SUM_SURF_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the SUMMIT pattern.

Constraint SUM_SURF_SUMMIT(VALUE, VARIABLES)

Arguments

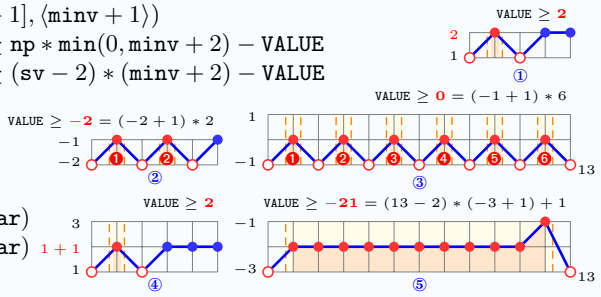
VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \min(\texttt{minv} + 1①, (\texttt{minv} + 1) * \texttt{np}②)$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min((\texttt{minv} + 1) * \texttt{np}③, \min(\texttt{minv} + 1④, (\texttt{sv} - 2) * (\texttt{minv} + 1) + 1⑤)) \end{array} \right)$$
$$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \max(\texttt{maxv}, \texttt{maxv} * \texttt{np})$$
$$\texttt{rv} \geq 3 \Rightarrow \bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max(\texttt{maxv} * \texttt{np}, \max(\texttt{maxv}, (\texttt{sv} - 2) * (\texttt{maxv} - 1) + 1)) \end{array} \right)$$
$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1, \texttt{maxv} \rangle)$$
$$\texttt{rv} = 2 \vee \texttt{maxv} = 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - \texttt{np} * \max(0, \texttt{maxv} - 1)$$
$$\texttt{rv} > 2 \wedge \texttt{maxv} > 1 \Rightarrow \texttt{n1} \geq \texttt{VALUE} - (\texttt{sv} - 2) * (\texttt{maxv} - 2) - 1$$
$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} + 1 \rangle)$$
$$\texttt{rv} = 2 \vee \texttt{minv} = -1 \Rightarrow \texttt{n2} \geq \texttt{np} * \min(0, \texttt{minv} + 2) - \texttt{VALUE}$$
$$\texttt{rv} > 2 \wedge \texttt{minv} < -1 \Rightarrow \texttt{n2} \geq (\texttt{sv} - 2) * (\texttt{minv} + 2) - \texttt{VALUE}$$
$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

where

$$\texttt{sv} = |\texttt{VARIABLES}|$$
$$\texttt{np} = \max(0, \lfloor (\texttt{sv} - 1)/2 \rfloor)$$
$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$
$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$
$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose

VALUE is the sum of the surface of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '(< | < (= | <)* <)(> | > (= | >)* >)'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(23, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1 \rangle)$

Figure 4.1401 provides an example where the SUM_SURF_SUMMIT $(23, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

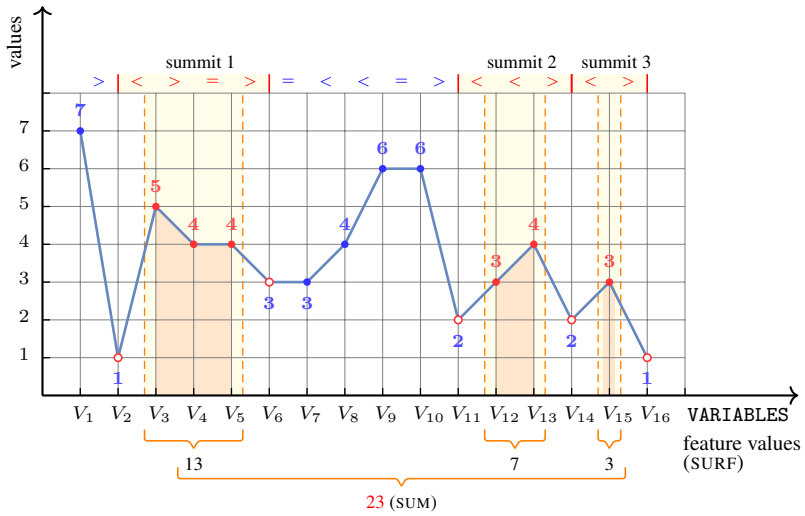


Figure 4.1401: Illustrating the SUM_SURF_SUMMIT constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1402 and 4.1403 respectively depict the automaton associated with the constraint SUM_SURF_SUMMIT and its simplified form.

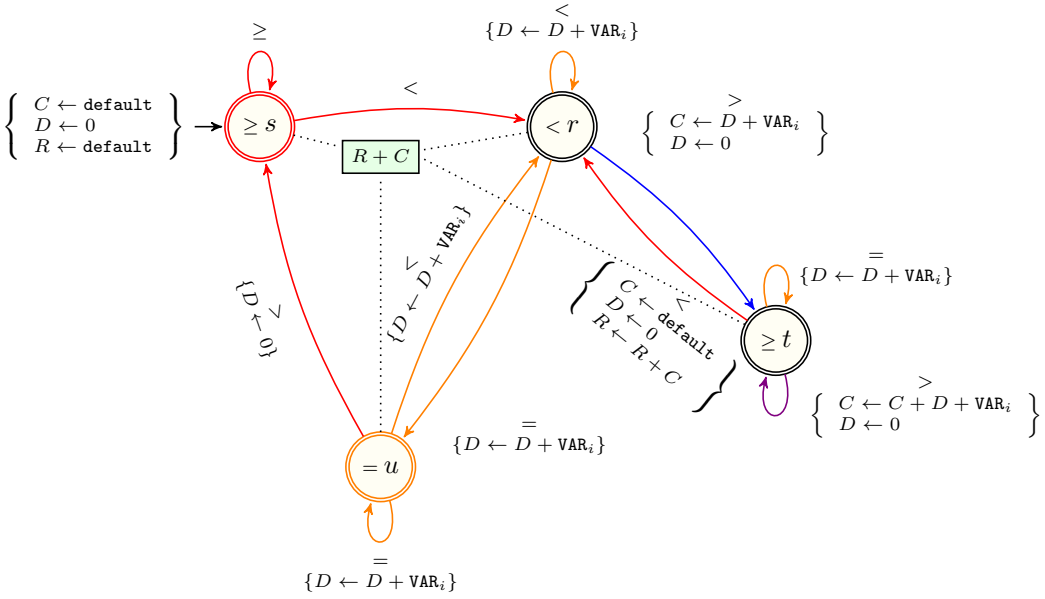


Figure 4.1402: Automaton for the SUM_SURF_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

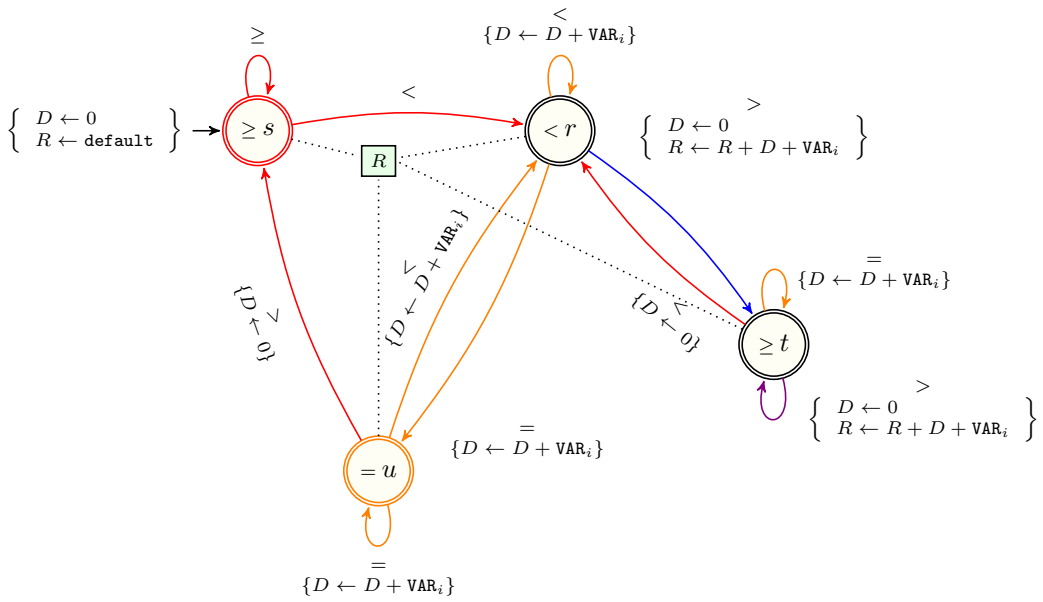


Figure 4.1403: Simplified automaton for the SUM_SURF_SUMMIT constraint obtained by applying decoration Table 3.26 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.349: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the SUM_SURF_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

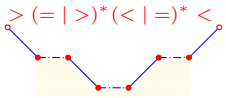
	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R	0

Table 4.350: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the SUM_SURF_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_VALLEY

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint SUM_SURF_VALLEY(VALUE, VARIABLES)

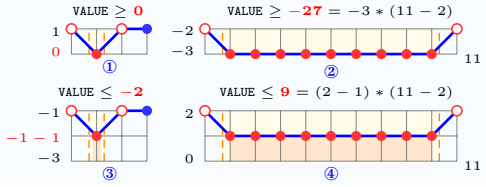
Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq \texttt{min}(\texttt{minv}①, \texttt{minv} * (\texttt{sv} - 2)②)$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \leq \texttt{max}(\texttt{maxv} - 1③, (\texttt{maxv} - 1) * (\texttt{sv} - 2)④)$
$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{maxv} - 1 \rangle)$
$\texttt{n1} \geq \texttt{VALUE} - \texttt{max}(0, (\texttt{sv} - 2) * (\texttt{maxv} - 2))$
$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv} - 1], \langle \texttt{minv} \rangle)$
$\texttt{n2} \geq \texttt{min}(0, (\texttt{sv} - 2) * (\texttt{minv} + 1)) - \texttt{VALUE}$
$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$

where
$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$
$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$



Purpose

VALUE is the sum of the surface of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example $(35, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.1404 provides an example where the SUM_SURF_VALLEY $(35, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical

$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

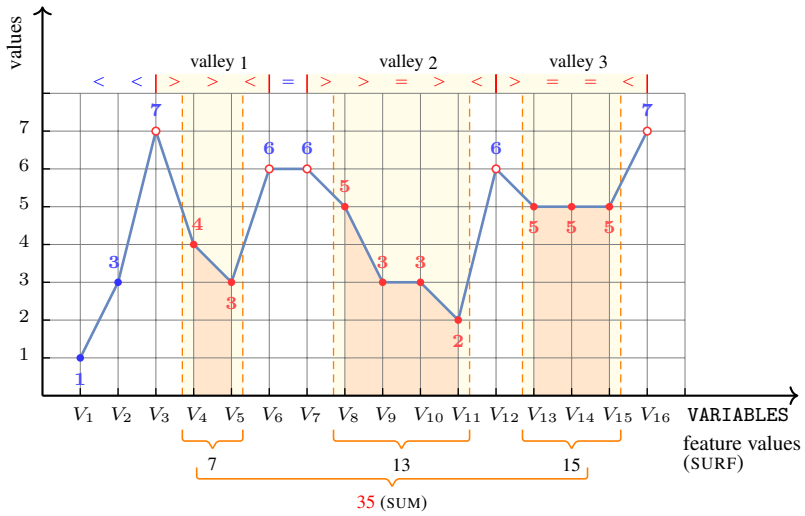


Figure 4.1404: Illustrating the SUM_SURF_VALLEY constraint of the **Example** slot

Symmetry Items of VARIABLES can be reversed.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

Automaton

Figures 4.1405 and 4.1406 respectively depict the automaton associated with the constraint SUM_SURF_VALLEY and its simplified form.

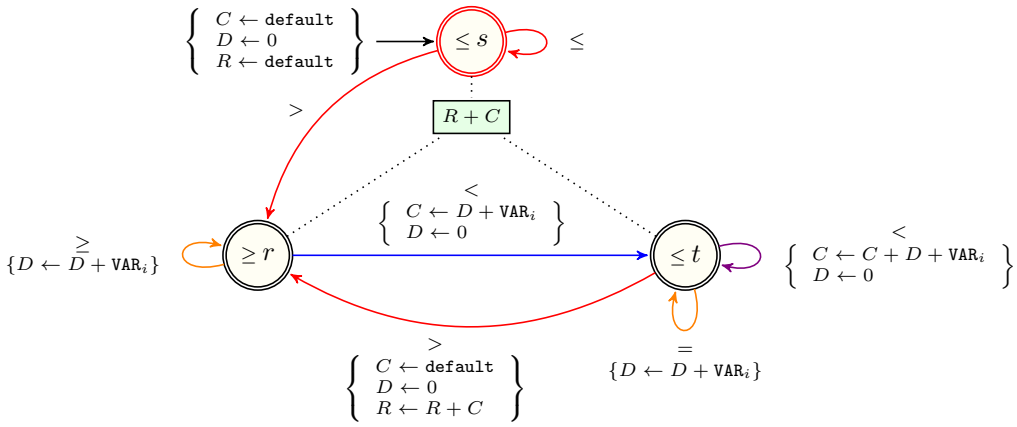


Figure 4.1405: Automaton for the SUM_SURF_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is 0

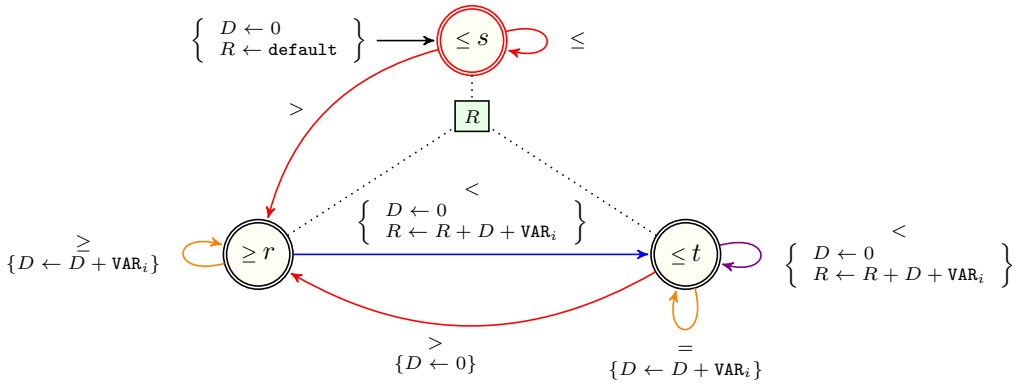


Figure 4.1406: Simplified automaton for the SUM_SURF_VALLEY constraint obtained by applying decoration Table 3.26 to the seed transducer of the VALLEY pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.351: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the SUM_SURF_VALLEY constraint defined as the composition of the VALLEY pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

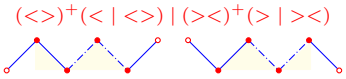
	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ C	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + \texttt{VAR}_{i+1}$ L	0

Table 4.352: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the SUM_SURF_VALLEY constraint defined as the composition of the VALLEY pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_SURF_ZIGZAG

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint SUM_SURF_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$$

$$\bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \geq \min \left(\begin{array}{l} 2 * \texttt{minv} + 1 ①, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{minv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{minv}+1) ② \end{array} \right) \end{array} \right)$$

$$\bigvee \left(\begin{array}{l} \texttt{VALUE} = 0, \\ \texttt{VALUE} \leq \max \left(\begin{array}{l} 2 * \texttt{maxv} - 1 ③, \\ \lfloor (\texttt{sv}-1)/2 \rfloor * \texttt{maxv} + \lfloor (\texttt{sv}-2)/2 \rfloor * (\texttt{maxv}-1) ④ \end{array} \right) \end{array} \right)$$

$$\texttt{among}(\texttt{n1}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{maxv}-1, \texttt{maxv} \rangle)$$

$$\texttt{n1} \geq \texttt{VALUE} - \lfloor (\texttt{sv}-1)/2 \rfloor - \max(0, (\texttt{sv}-2) * (\texttt{maxv}-2))$$

$$\texttt{n2} \geq \min(0, (\texttt{sv}-2) * (\texttt{minv}+2)) - \lfloor (\texttt{sv}-1)/2 \rfloor - \texttt{VALUE}$$

$$\texttt{among}(\texttt{n2}, \texttt{VARIABLES}[2, \texttt{sv}-1], \langle \texttt{minv}, \texttt{minv}+1 \rangle)$$

$$\texttt{required}(\texttt{VARIABLES}, \texttt{var})$$

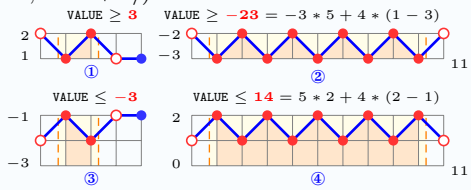
where

$$\texttt{minv} = \texttt{minval}(\texttt{VARIABLES.var})$$

$$\texttt{maxv} = \texttt{maxval}(\texttt{VARIABLES.var})$$

$$\texttt{sv} = |\texttt{VARIABLES}|$$

$$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$$



Purpose VALUE is the sum of the surface of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example $(33, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.1407 provides an example where the SUM_SURF_ZIGZAG $(33, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

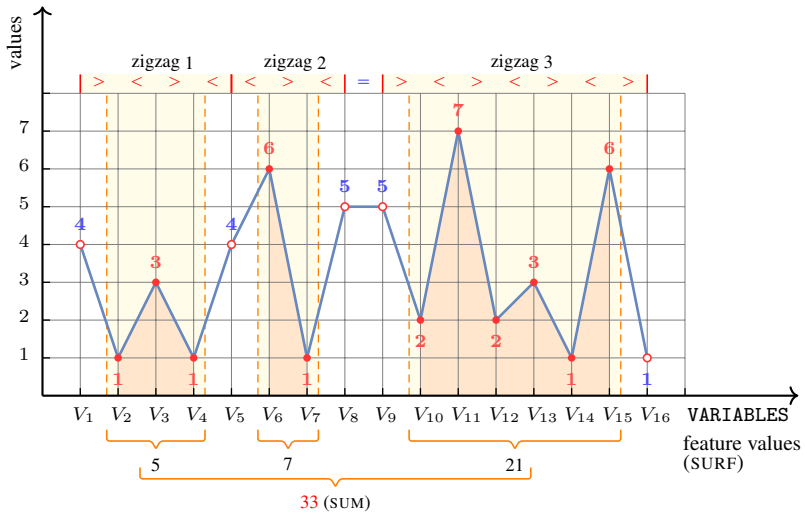


Figure 4.1407: Illustrating the SUM_SURF_ZIGZAG constraint of the **Example** slot

Typical

$|\texttt{VARIABLES}| > 3$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry

Items of VARIABLES can be reversed.

Arg. properties

Functional dependency: VALUE determined by VARIABLES.

Automaton Figures 4.1408 and 4.1409 respectively depict the automaton associated with the constraint SUM_SURF_ZIGZAG and its simplified form.

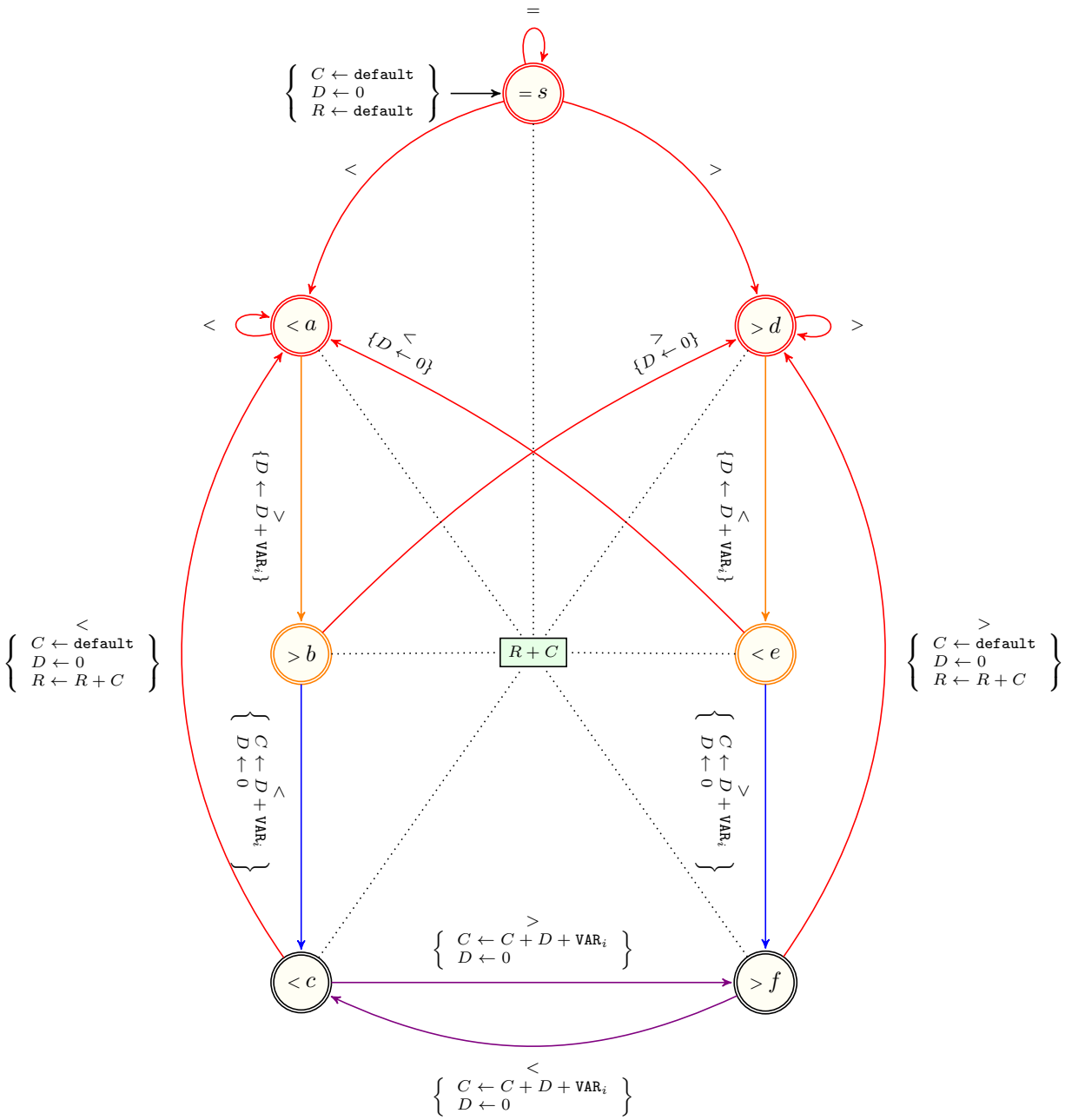


Figure 4.1408: Automaton for the SUM_SURF_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

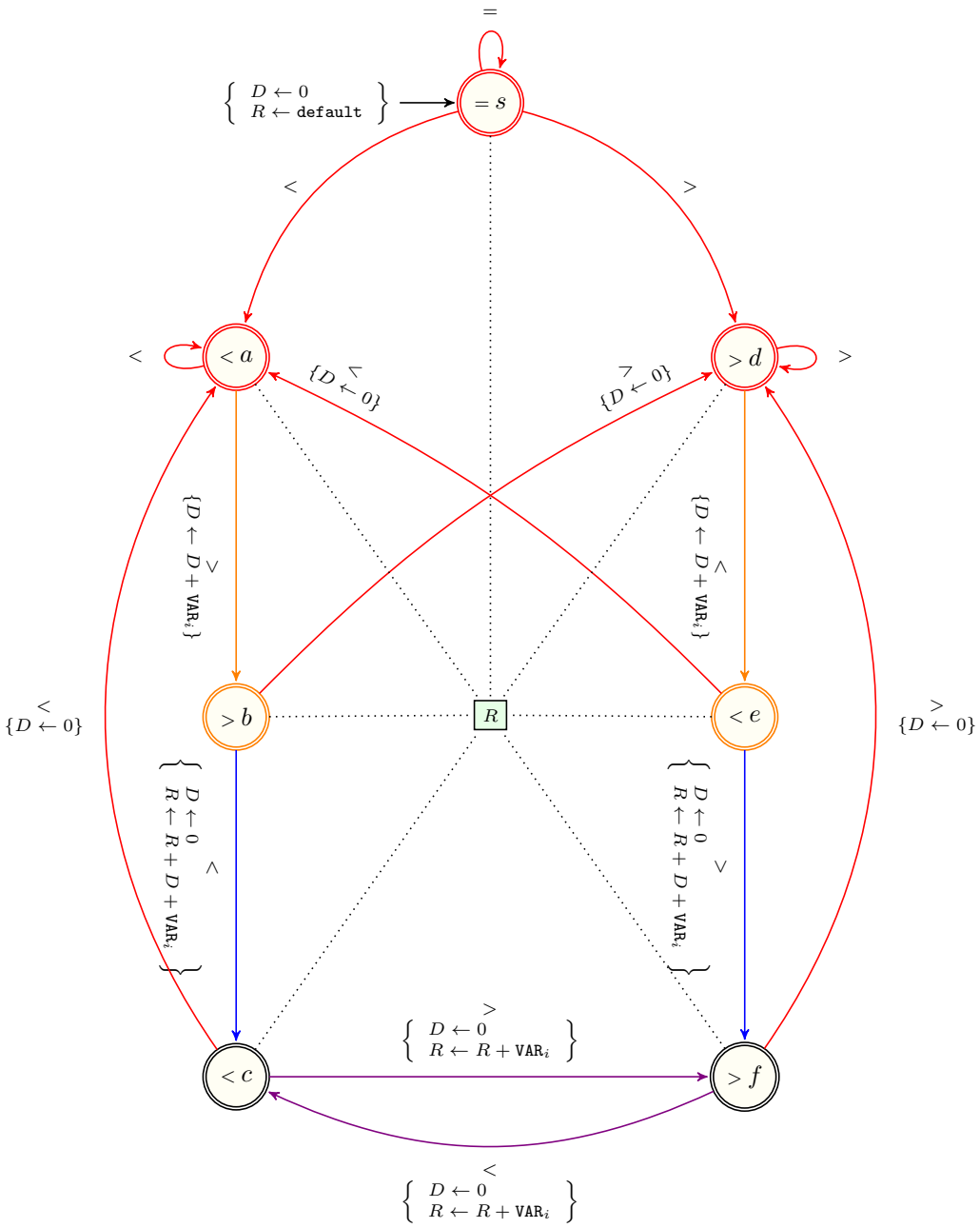


Figure 4.1409: Simplified automaton for the SUM_SURF_ZIGZAG constraint obtained by applying decoration Table 3.29 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value

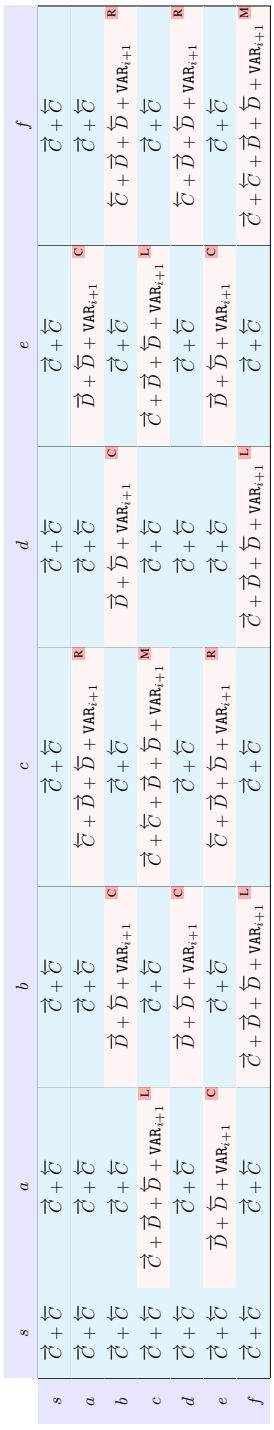


Table 4.353: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the SUM_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature SURF, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

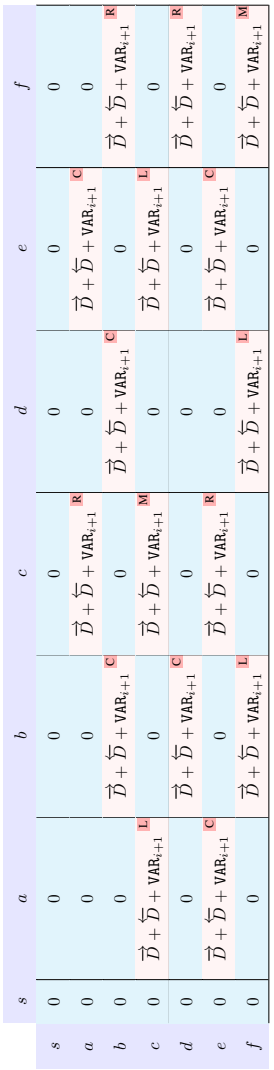
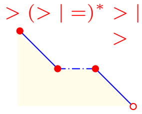


Table 4.354: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the simplified automaton of the SUM_SURF_ZIGZAG constraint defined as the composition of the ZIGZAG pattern , the feature SURF , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_DECREASING_SEQUENCE



► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

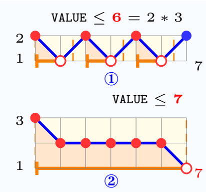
Constraint SUM_WIDTH_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{np}$ ①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$ ②
required(VARIABLES, var)
where
$\texttt{rv} = \texttt{range}(\texttt{VARIABLES.var})$
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$



(see also fig. 4.1413)

Purpose

VALUE is the sum of the width of occurrences of the DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(9, \langle 3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4 \rangle)$

Figure 4.1410 provides an example where the SUM_WIDTH_DECREASING_SEQUENCE (9, [3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4]) constraint holds.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

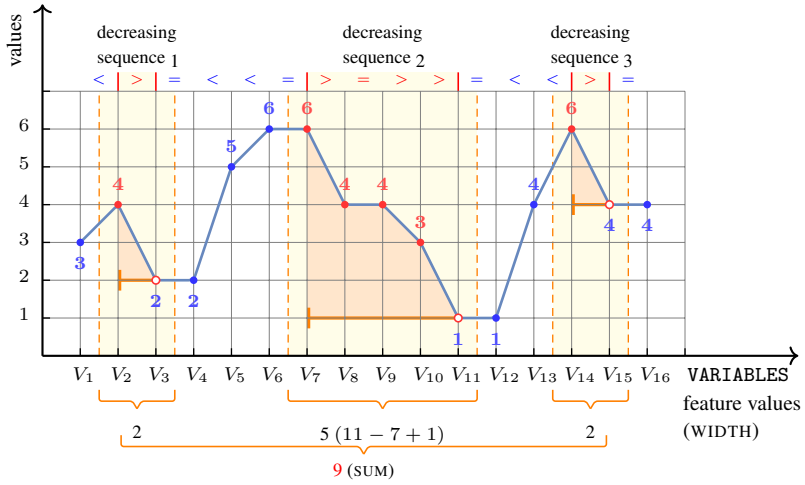


Figure 4.1410: Illustrating the SUM_WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1411 and 4.1412 respectively depict the automaton associated with the constraint SUM_WIDTH_DECREASING_SEQUENCE and its simplified form.

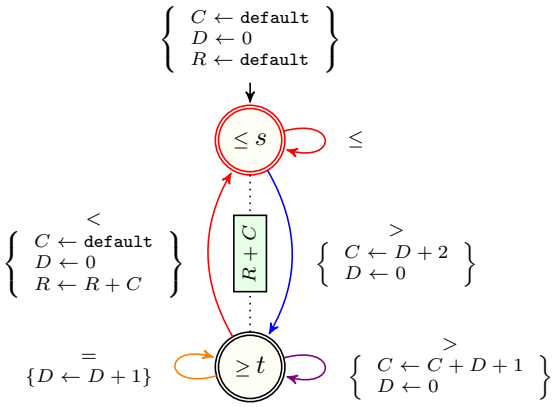


Figure 4.1411: Automaton for the SUM_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.355: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the SUM_WIDTH_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

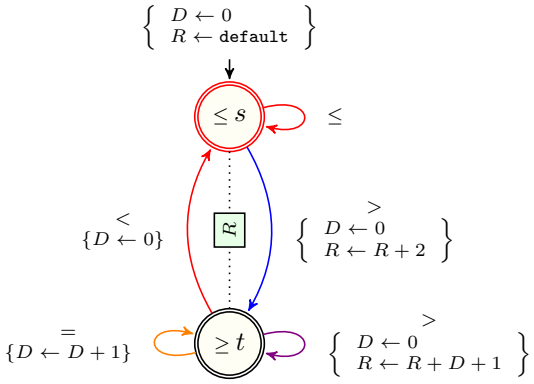


Figure 4.1412: Simplified automaton for the SUM_WIDTH_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.30 to the seed transducer of the DECREASING_SEQUENCE pattern where default is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.356: Concrete glue matrix, derived from the parametrised glue matrix 3.5, for the simplified automaton of the SUM_WIDTH_DECREASING_SEQUENCE constraint defined as the composition of the DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

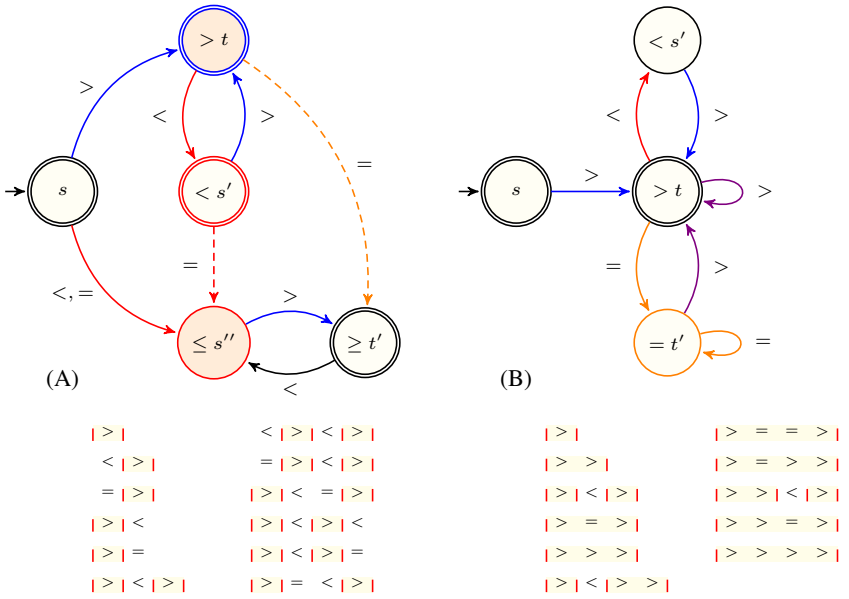


Figure 4.1413: (**top**) Automata without registers for the SUM_WIDTH_DECREASING_SEQUENCE_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_DECREASING_SEQUENCE_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the DECREASING_SEQUENCE pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2, i.e. $2 \cdot \left\lfloor \frac{\mathtt{sv}}{2} \right\rfloor$ of the **Restrictions** slot (see ①); (B) strictly greater than 2, i.e. sv (see ②). Within (A) states s, s' and t' are accepting when $\mathtt{sv} \bmod 2 = 1$, while state t is accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

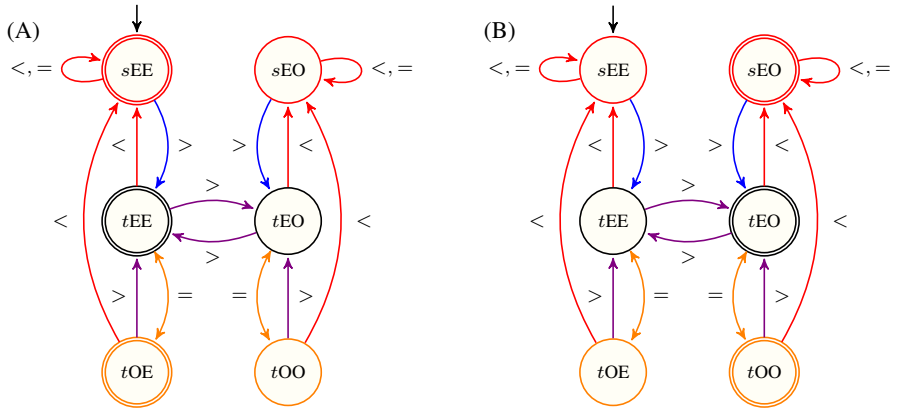
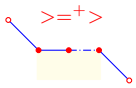


Figure 4.1414: Automata without registers for the (A) SUM_WIDTH_DECREASING_SEQUENCE_IS_EVEN and the (B) SUM_WIDTH_DECREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the DECREASING_SEQUENCE pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

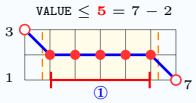
Constraint SUM_WIDTH_DECREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1418)

Purpose VALUE is the sum of the width of occurrences of the DECREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(4, \langle 6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3 \rangle)$

Figure 4.1415 provides an example where the SUM_WIDTH_DECREASING_TERRACE $(4, [6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

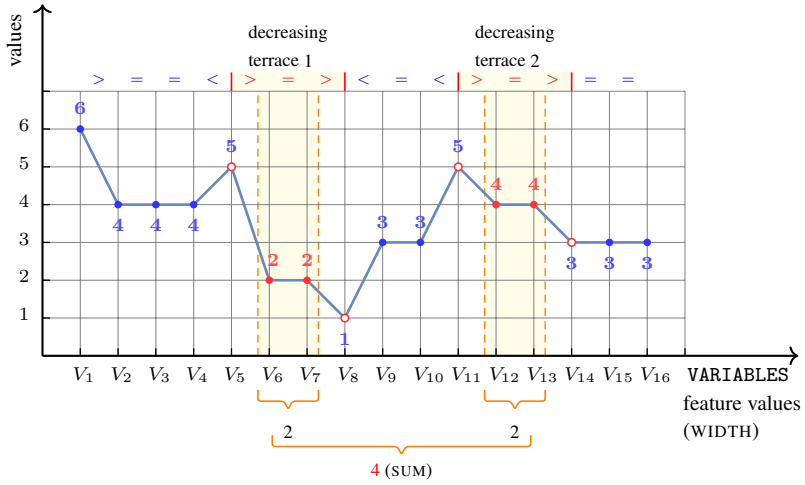


Figure 4.1415: Illustrating the SUM_WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.1416 and 4.1417 respectively depict the automaton associated with the constraint SUM_WIDTH_DECREASING_TERRACE and its simplified form.

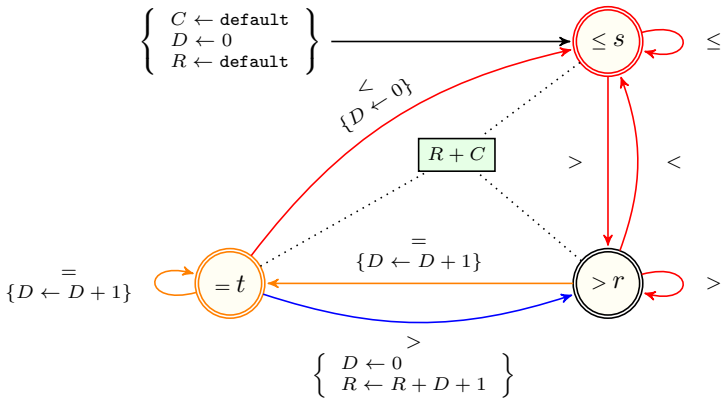


Figure 4.1416: Automaton for the SUM_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.357: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the SUM_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

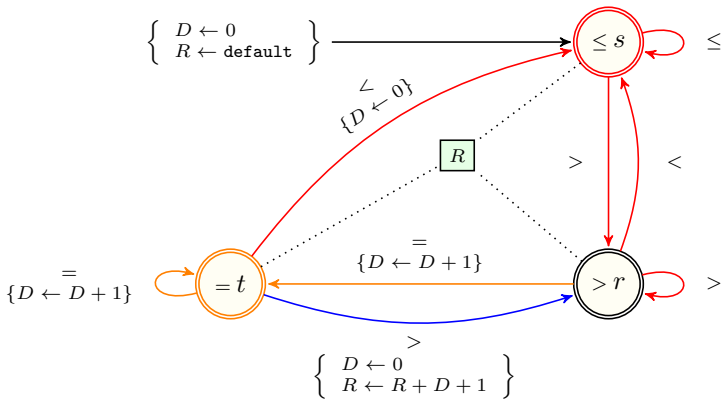


Figure 4.1417: Simplified automaton for the SUM_WIDTH_DECREASING_TERRACE constraint obtained by applying decoration Table 3.29 to the seed transducer of the DECREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.358: Concrete glue matrix, derived from the parametrised glue matrix 3.6, for the simplified automaton of the SUM_WIDTH_DECREASING_TERRACE constraint defined as the composition of the DECREASING_TERRACE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

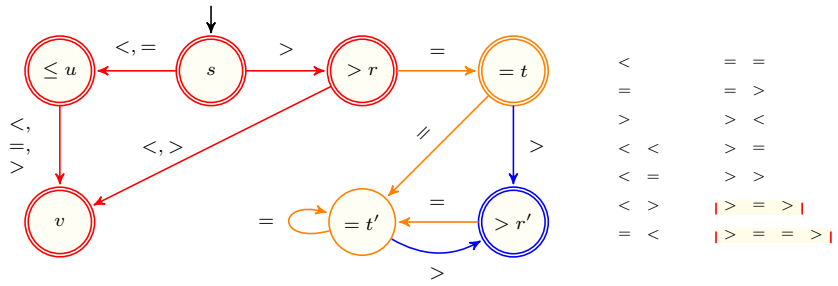


Figure 4.1418: (**left**) Automaton without registers for the SUM_WIDTH_DECREASING_TERRACE_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the DECREASING_TERRACE pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

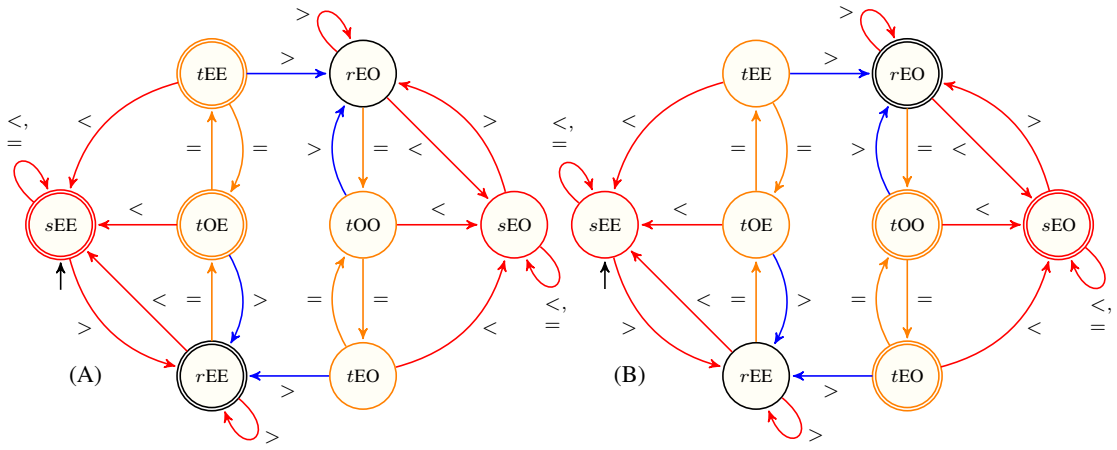
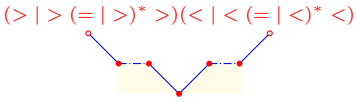


Figure 4.1419: Automata without registers for the (A) SUM_WIDTH_DECREASING_TERRACE_IS_EVEN and the (B) SUM_WIDTH_DECREASING_TERRACE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the DECREASING_TERRACE pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_GORGE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the GORGE pattern.

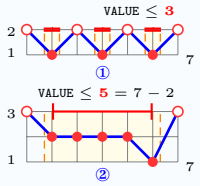
Constraint SUM_WIDTH_GORGE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{np}$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \max(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \max(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$



(see also fig. 4.1423)

Purpose VALUE is the sum of the width of occurrences of the GORGE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(6, \langle 1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7\rangle)$

Figure 4.1420 provides an example where the SUM_WIDTH_GORGE $(6, [1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

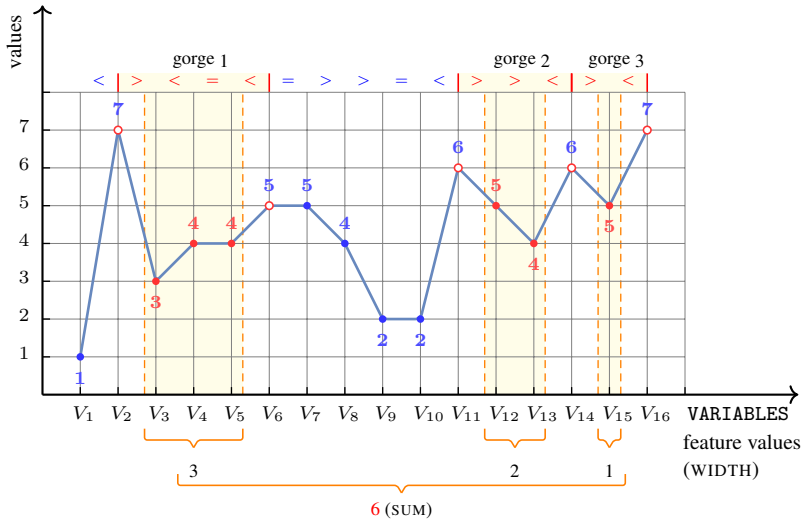


Figure 4.1420: Illustrating the SUM_WIDTH_GORGE constraint of the **Example** slot

Automaton Figures 4.1421 and 4.1422 respectively depict the automaton associated with the constraint SUM_WIDTH_GORGE and its simplified form.

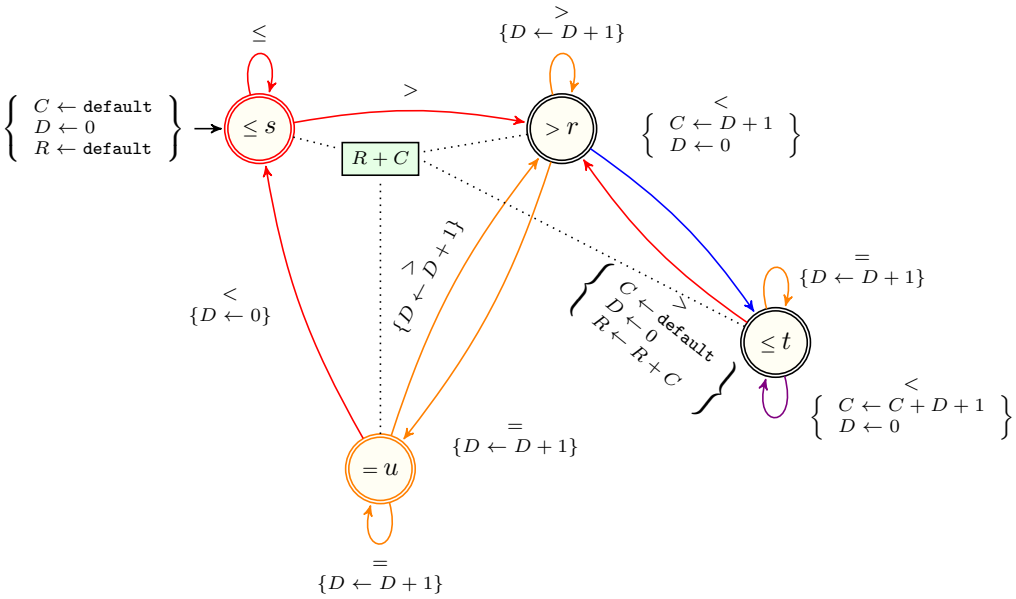


Figure 4.1421: Automaton for the SUM_WIDTH_GORGE constraint obtained by applying decoration Table 3.37 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

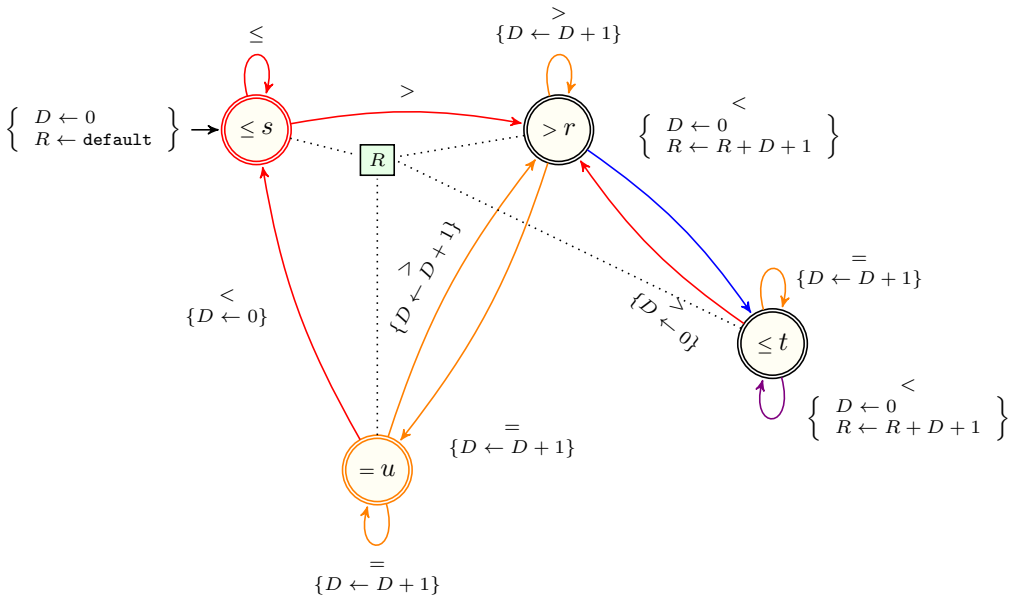


Figure 4.1422: Simplified automaton for the SUM_WIDTH_GORGE constraint obtained by applying decoration Table 3.26 to the seed transducer of the GORGE pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.359: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the SUM_WIDTH_GORGE constraint defined as the composition of the GORGE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0

Table 4.360: Concrete glue matrix, derived from the parametrised glue matrix 3.7, for the simplified automaton of the SUM_WIDTH_GORGE constraint defined as the composition of the GORGE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

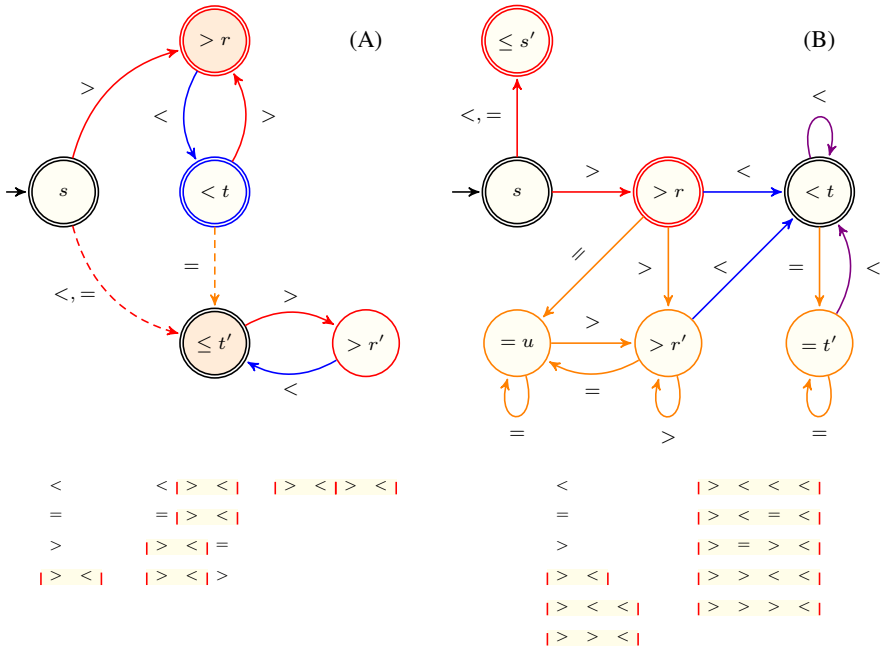


Figure 4.1423: (**top**) Automata without registers for the SUM_WIDTH_GORGE_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_GORGE_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the GORGE pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2, i.e. $\max\left(0, \left\lfloor \frac{\mathtt{sv}-1}{2} \right\rfloor\right)$ of the **Restrictions** slot (see ①), (B) strictly greater than 2, i.e. $\max(0, \mathtt{sv} - 2)$ (see ②). Within (A) states s and t are accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

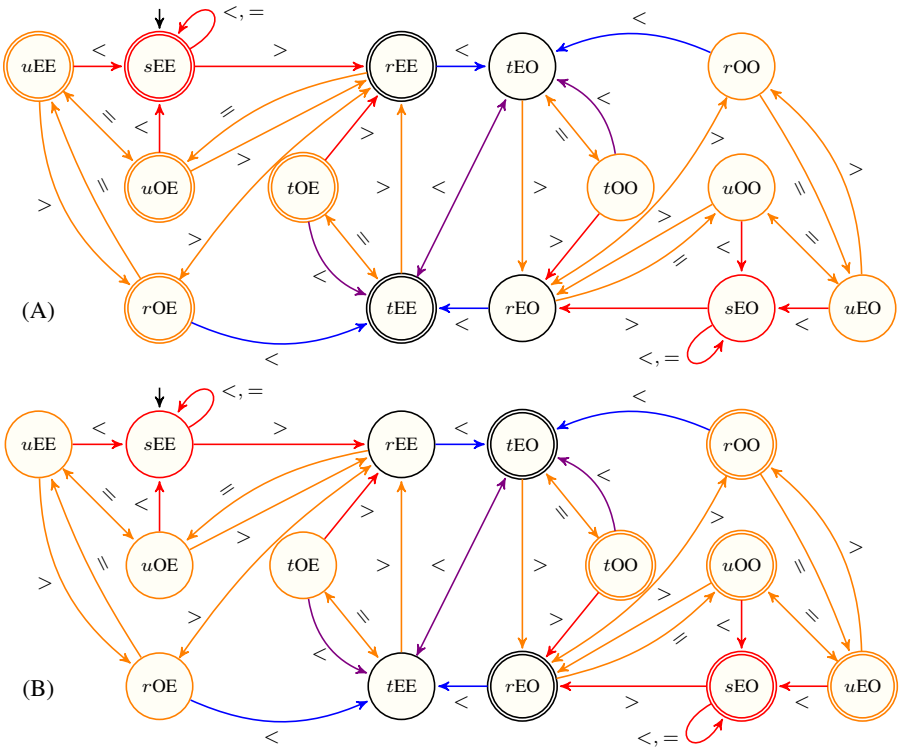
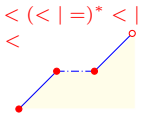


Figure 4.1424: Automata without registers for the (A) SUM_WIDTH_GORGE_IS_EVEN and the (B) SUM_WIDTH_GORGE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the GORGE pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

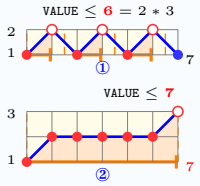
Constraint SUM_WIDTH_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq 2 * \texttt{np}$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
rv =range(VARIABLES.var)
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{np} = \lfloor \texttt{sv}/2 \rfloor$



(see also fig. 4.1428)

Purpose VALUE is the sum of the width of occurrences of the INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(9, \langle 4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3 \rangle)$

Figure 4.1425 provides an example where the SUM_WIDTH_INCREASING_SEQUENCE $(9, [4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

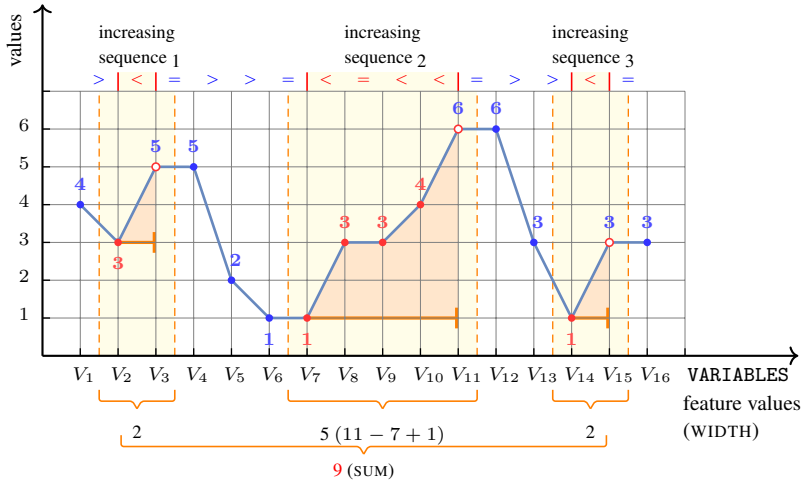


Figure 4.1425: Illustrating the SUM_WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1426 and 4.1427 respectively depict the automaton associated with the constraint SUM_WIDTH_INCREASING_SEQUENCE and its simplified form.

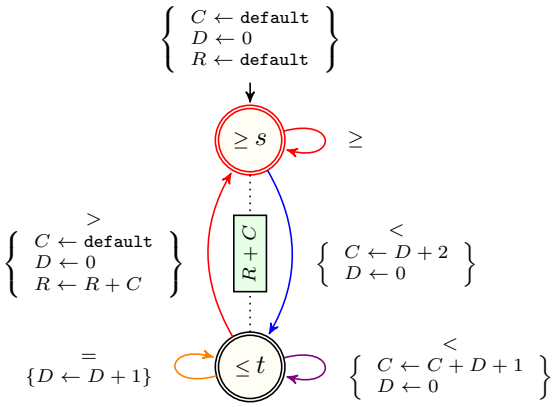


Figure 4.1426: Automaton for the SUM_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0

	s	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.361: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the SUM_WIDTH_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern , the feature WIDTH , and the aggregator sum ; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

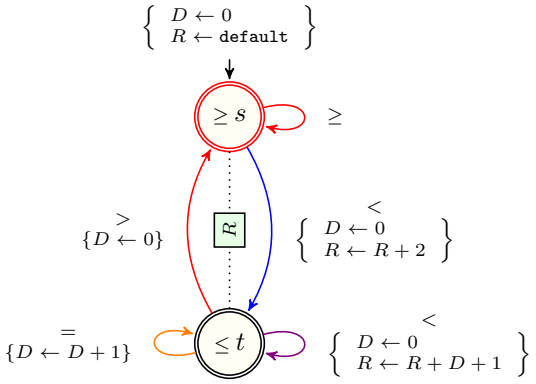


Figure 4.1427: Simplified automaton for the SUM_WIDTH_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.30 to the seed transducer of the INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	t
s	0	0
t	0	$\overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.362: Concrete glue matrix, derived from the parametrised glue matrix 3.9, for the simplified automaton of the SUM_WIDTH_INCREASING_SEQUENCE constraint defined as the composition of the INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

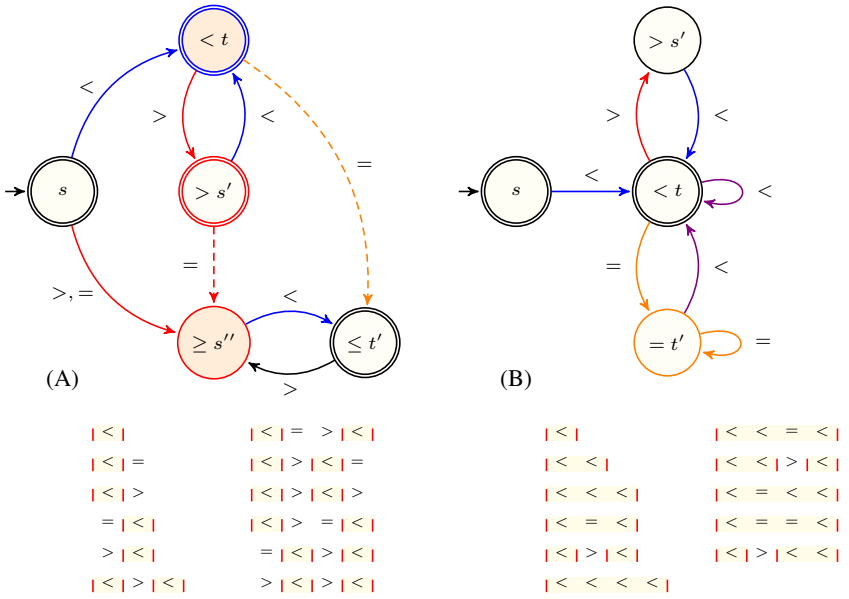


Figure 4.1428: (**top**) Automata without registers for the SUM_WIDTH_INCREASING_SEQUENCE_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_INCREASING_SEQUENCE_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the INCREASING_SEQUENCE pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2, i.e. $2 \cdot \left\lfloor \frac{\mathtt{sv}}{2} \right\rfloor$ of the **Restrictions** slot (see ①), (B) strictly greater than 2, i.e. sv (see ②). Within (A) states s, s' and t' are accepting when $\mathtt{sv} \bmod 2 = 1$, while state t is accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

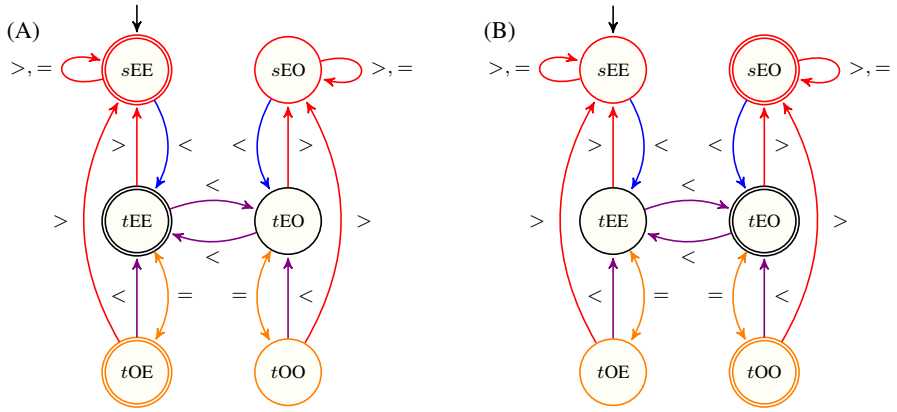
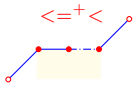


Figure 4.1429: Automata without registers for the (A) SUM_WIDTH_INCREASING_SEQUENCE_IS_EVEN and the (B) SUM_WIDTH_INCREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the INCREASING_SEQUENCE pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

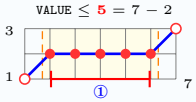
Constraint SUM_WIDTH_INCREASING_TERRACE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 2 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1433)

Purpose VALUE is the sum of the width of occurrences of the INCREASING_TERRACE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(5, \langle 1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4 \rangle)$

Figure 4.1430 provides an example where the SUM_WIDTH_INCREASING_TERRACE $(5, [1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 2

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

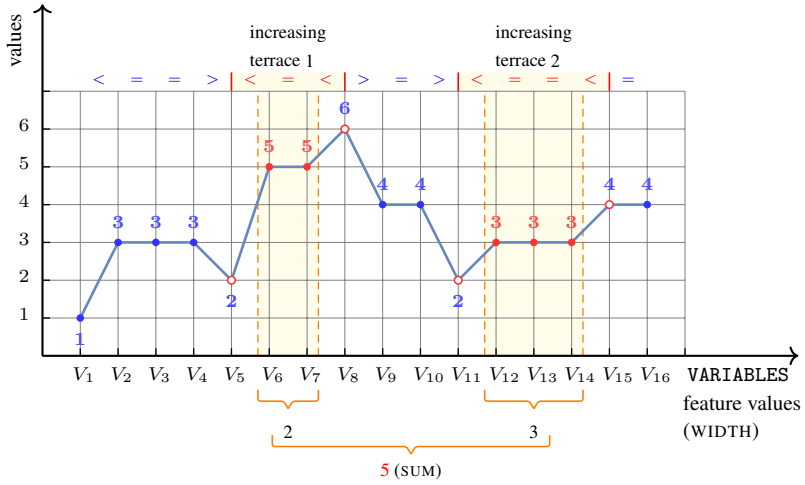


Figure 4.1430: Illustrating the SUM_WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton

Figures 4.1431 and 4.1432 respectively depict the automaton associated with the constraint SUM_WIDTH_INCREASING_TERRACE and its simplified form.

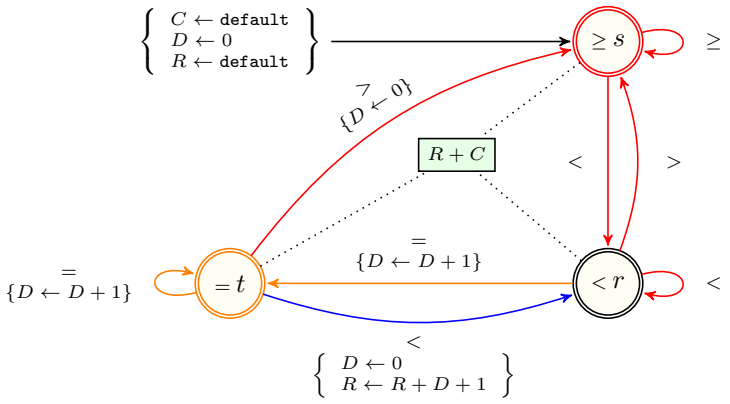


Figure 4.1431: Automaton for the SUM_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.37 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.363: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the SUM_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

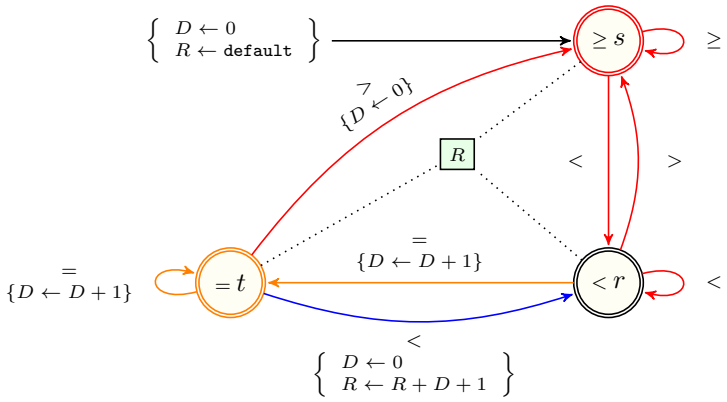


Figure 4.1432: Simplified automaton for the SUM_WIDTH_INCREASING_TERRACE constraint obtained by applying decoration Table 3.29 to the seed transducer of the INCREASING_TERRACE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.364: Concrete glue matrix, derived from the parametrised glue matrix 3.10, for the simplified automaton of the SUM_WIDTH_INCREASING_TERRACE constraint defined as the composition of the INCREASING_TERRACE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

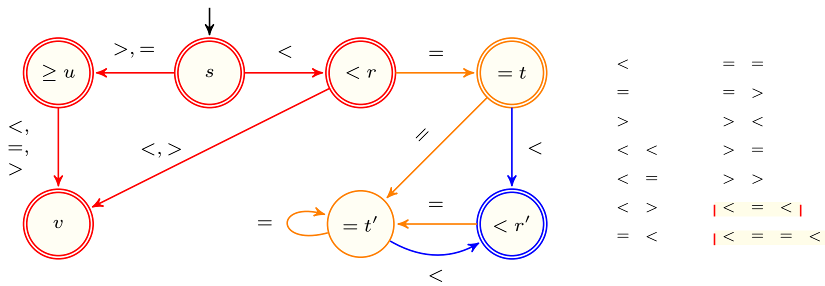


Figure 4.1433: (**left**) Automaton without registers for the SUM_WIDTH_INCREASING_TERRACE_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the INCREASING_TERRACE pattern of a sequence of sv variables, i.e. $\max(0, \texttt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\texttt{sv} - 1 \in \{1, 2, 3, 4\}$.

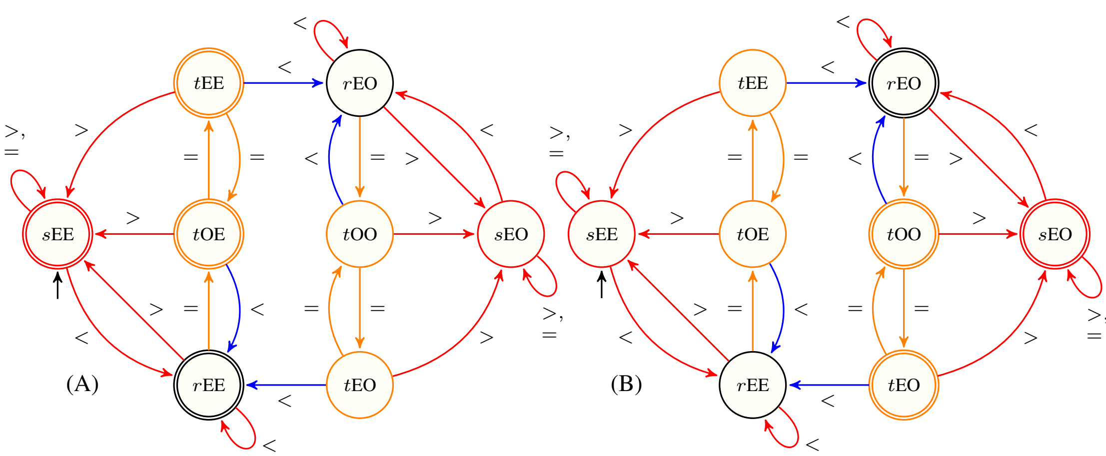
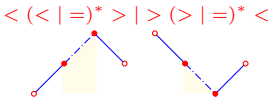


Figure 4.1434: Automata without registers for the SUM_WIDTH_INCREASING_TERRACE_IS_EVEN and the (B) SUM_WIDTH_INCREASING_TERRACE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the INCREASING_TERRACE pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

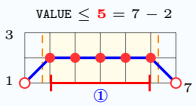
Constraint SUM_WIDTH_INFLEXION(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1438)

Purpose

VALUE is the sum of the width of occurrences of the INFLEXION pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(13, \langle 1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4 \rangle)$

Figure 4.1435 provides an example where the SUM_WIDTH_INFLEXION $(13, [1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

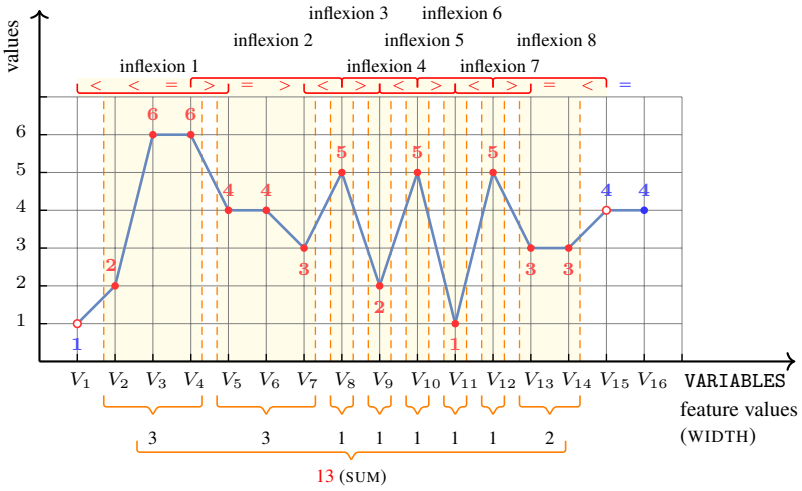


Figure 4.1435: Illustrating the SUM_WIDTH_INFLEXION constraint of the **Example** slot

Automaton

Figures 4.1436 and 4.1437 respectively depict the automaton associated with the constraint SUM_WIDTH_INFLEXION and its simplified form.

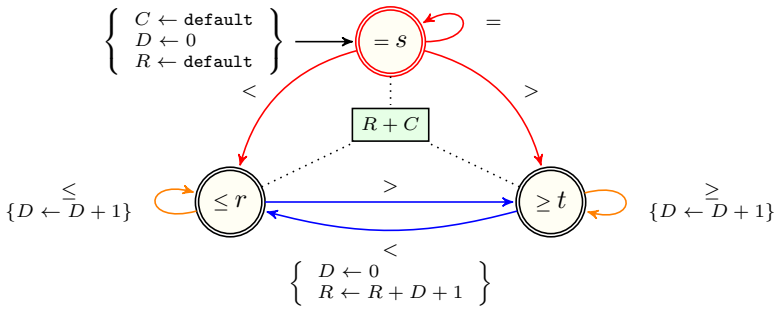


Figure 4.1436: Automaton for the SUM_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.37 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$)

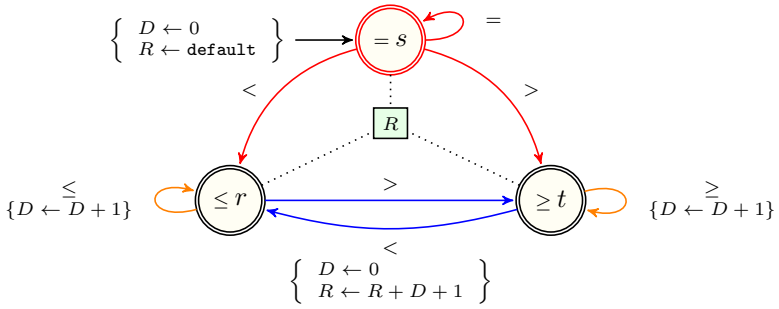


Figure 4.1437: Simplified automaton for the SUM_WIDTH_INFLEXION constraint obtained by applying decoration Table 3.26 to the seed transducer of the INFLEXION pattern where `default` is 0 (transition $r \rightarrow t$ has the same registers updates as transition $t \rightarrow r$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

Specialisation

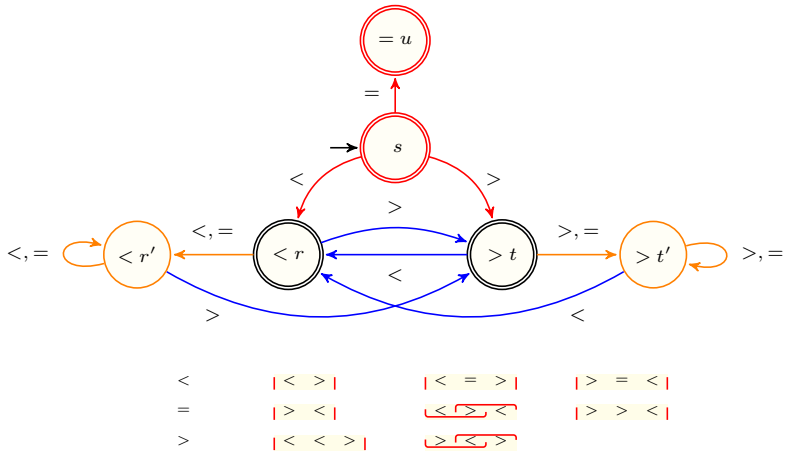


Figure 4.1438: (**top**) Automaton without registers for the SUM_WIDTH_INFLEXION_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the INFLEXION pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3\}$.

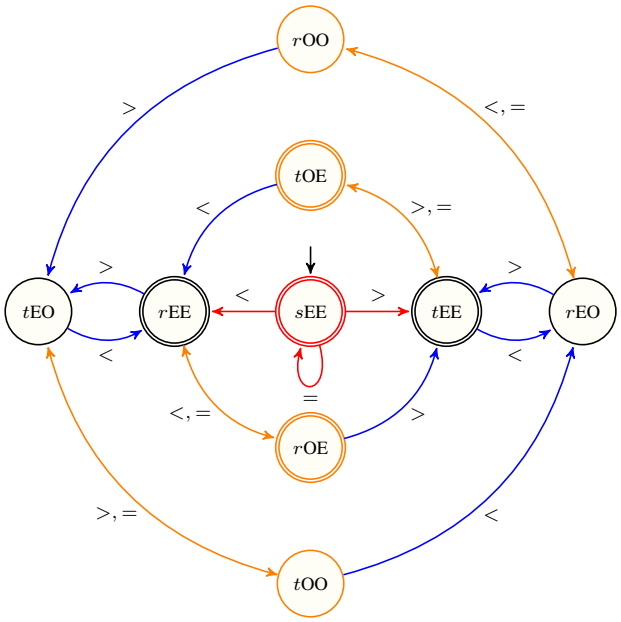
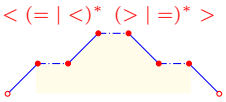


Figure 4.1439: Automaton without registers for the SUM_WIDTH_INFLEXION_IS_EVEN constraint; it achieves an even sum of width of the INFLEXION pattern on a sequence of n variables; within the name of each state the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd. The automaton for the SUM_WIDTH_INFLEXION_IS_ODD constraint achieves an odd sum of width of the INFLEXION pattern on a sequence of n variables; it is obtained by switching the accepting and the non-accepting states.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_PEAK



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PEAK pattern.

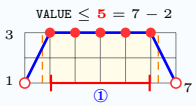
Constraint SUM_WIDTH_PEAK(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1443)

Purpose VALUE is the sum of the width of occurrences of the PEAK pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(8, \langle 7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1\rangle)$

Figure 4.1440 provides an example where the SUM_WIDTH_PEAK $(8, [7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

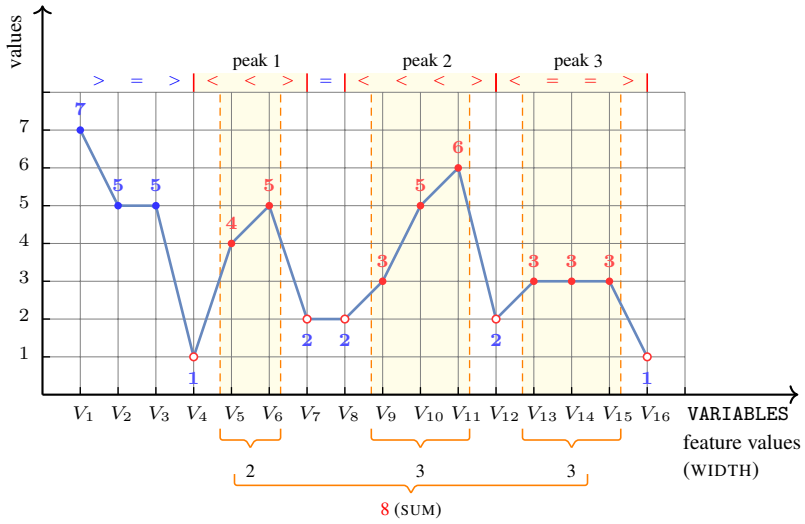


Figure 4.1440: Illustrating the SUM_WIDTH_PEAK constraint of the **Example** slot

Automaton

Figures 4.1441 and 4.1442 respectively depict the automaton associated with the constraint SUM_WIDTH_PEAK and its simplified form.

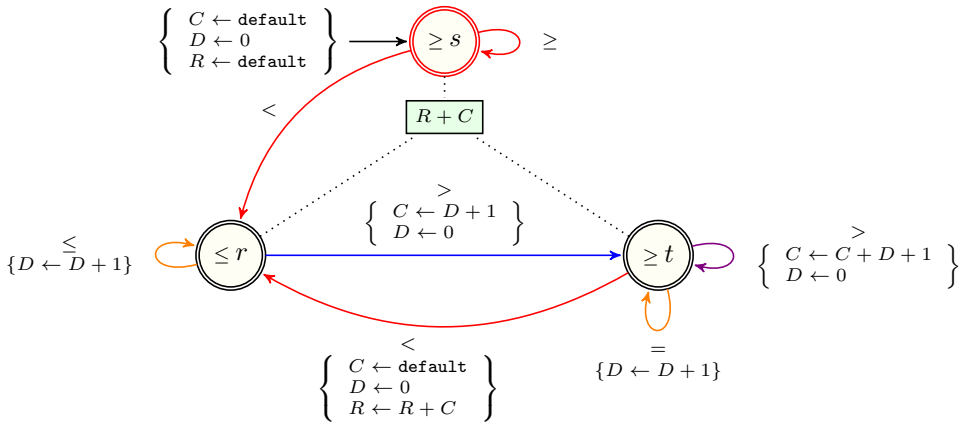


Figure 4.1441: Automaton for the SUM_WIDTH_PEAK constraint obtained by applying decoration Table 3.37 to the seed transducer of the PEAK pattern where `default` is 0

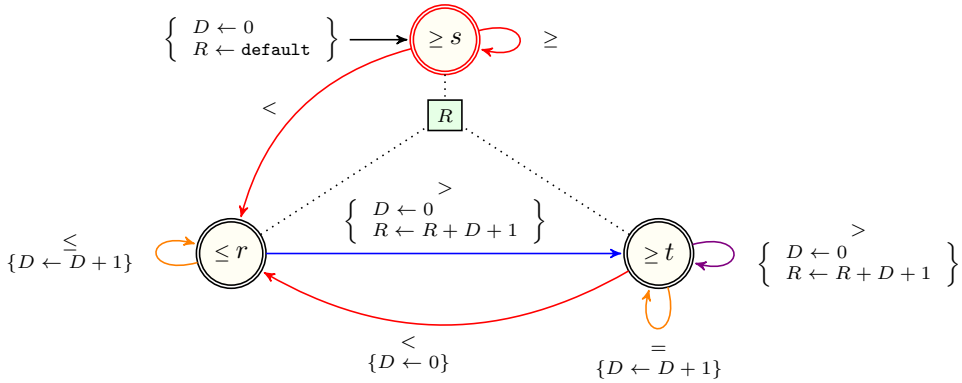


Figure 4.1442: Simplified automaton for the SUM_WIDTH_PEAK constraint obtained by applying decoration Table 3.26 to the seed transducer of the PEAK pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.365: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the SUM_WIDTH_PEAK constraint defined as the composition of the PEAK pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0

Table 4.366: Concrete glue matrix, derived from the parametrised glue matrix 3.11, for the simplified automaton of the SUM_WIDTH_PEAK constraint defined as the composition of the PEAK pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

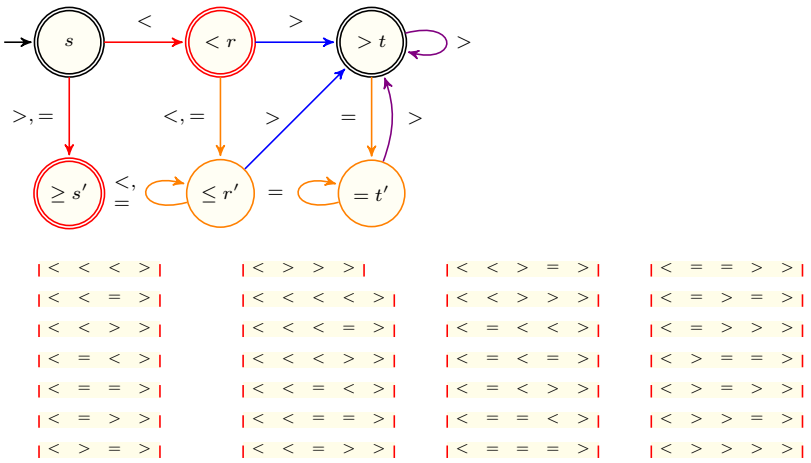


Figure 4.1443: (**top**) Automaton without registers for the SUM_WIDTH_PEAK_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the PEAK pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

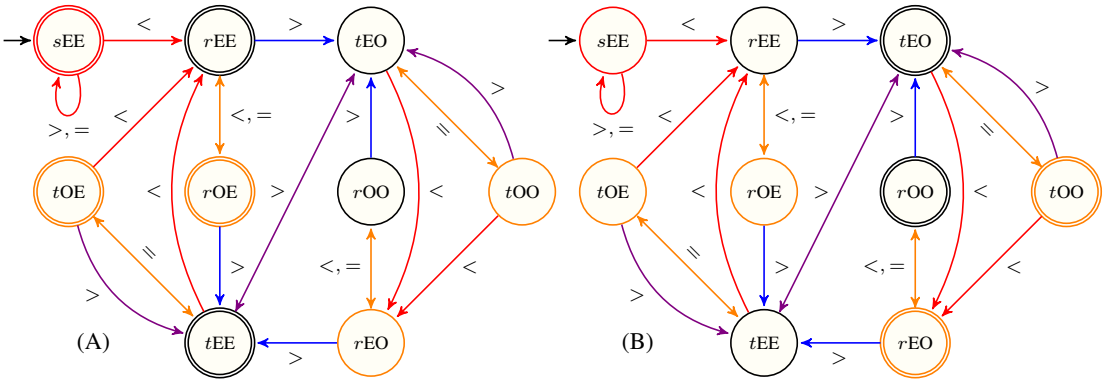


Figure 4.1444: Automata without registers for the (A) SUM_WIDTH_PEAK_IS_EVEN and the (B) SUM_WIDTH_PEAK_IS_ODD constraints; they respectively achieve an even/odd sum of width of the PEAK pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

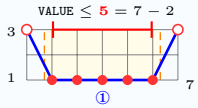
Constraint SUM_WIDTH_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1448)

Purpose VALUE is the sum of the width of occurrences of the PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(4, \langle 2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3\rangle)$

Figure 4.1445 provides an example where the SUM_WIDTH_PLAIN $(4, [2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

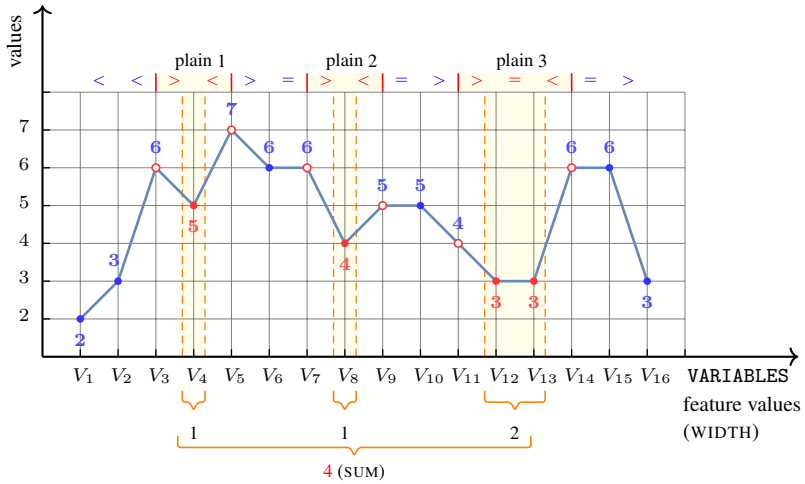


Figure 4.1445: Illustrating the SUM_WIDTH_PLAIN constraint of the **Example** slot

Automaton

Figures 4.1446 and 4.1447 respectively depict the automaton associated with the constraint SUM_WIDTH_PLAIN and its simplified form.

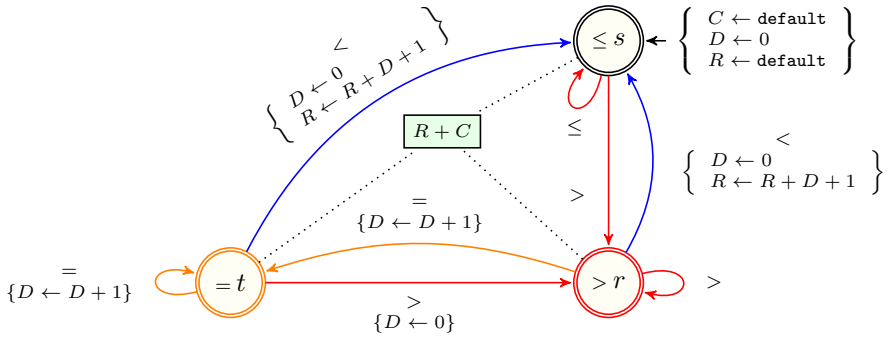


Figure 4.1446: Automaton for the SUM_WIDTH_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLAIN pattern where `default` is 0

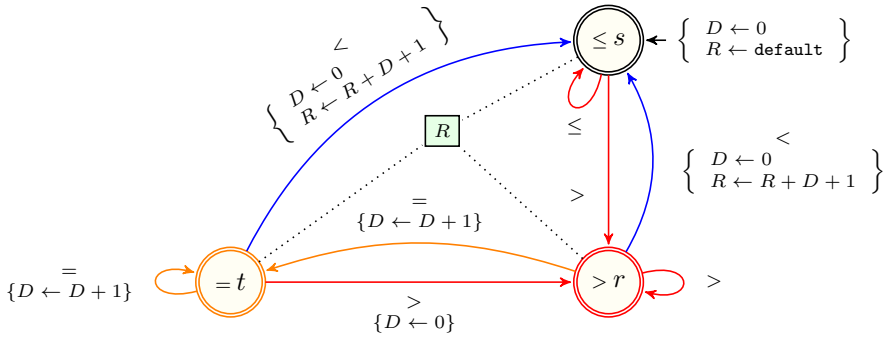


Figure 4.1447: Simplified automaton for the SUM_WIDTH_PLAIN constraint obtained by applying decoration Table 3.29 to the seed transducer of the PLAIN pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.367: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the SUM_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.368: Concrete glue matrix, derived from the parametrised glue matrix 3.12, for the simplified automaton of the SUM_WIDTH_PLAIN constraint defined as the composition of the PLAIN pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

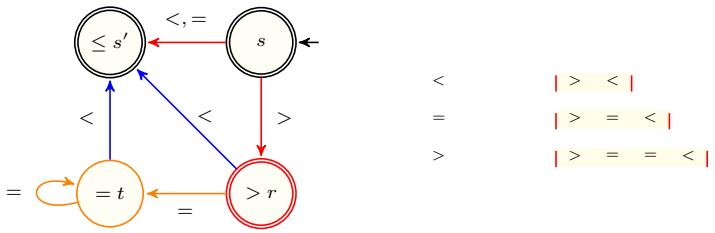


Figure 4.1448: (**left**) Automaton without registers for the SUM_WIDTH_PLAIN_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the PLAIN pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

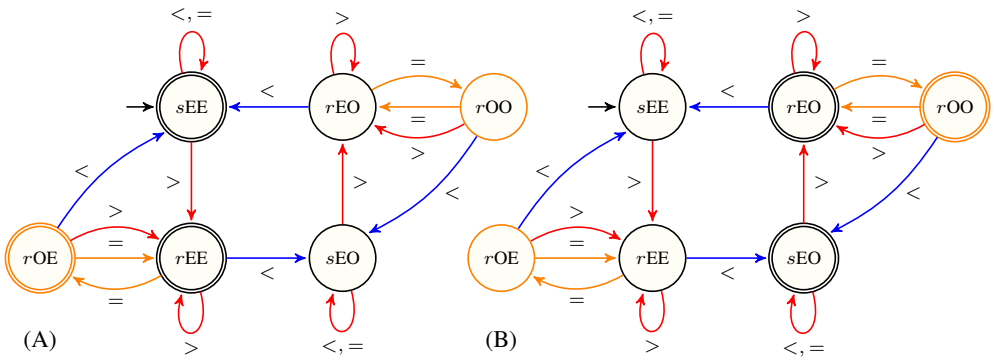


Figure 4.1449: Automata without registers for the (A) SUM_WIDTH_PLAIN_IS_EVEN and the (B) SUM_WIDTH_PLAIN_IS_ODD constraints; they respectively achieve an even/odd sum of width of the PLAIN pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

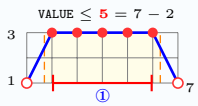
Constraint SUM_WIDTH_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1453)

Purpose VALUE is the sum of the width of occurrences of the PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(7, \langle 1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5 \rangle)$

Figure 4.1450 provides an example where the SUM_WIDTH_PLATEAU $(7, [1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

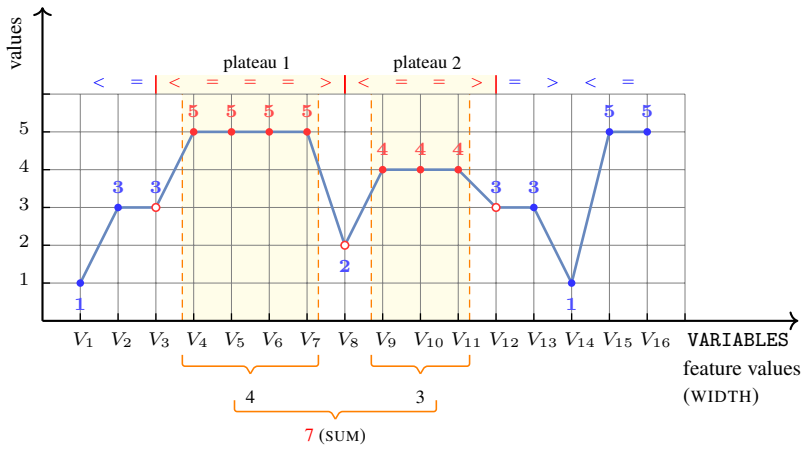


Figure 4.1450: Illustrating the SUM_WIDTH_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.1451 and 4.1452 respectively depict the automaton associated with the constraint SUM_WIDTH_PLATEAU and its simplified form.

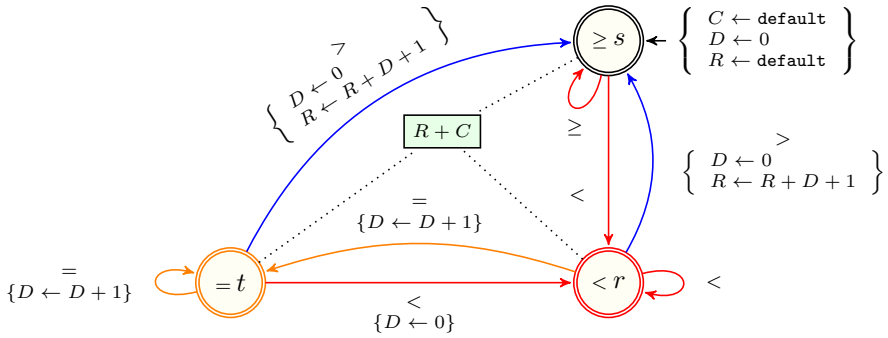


Figure 4.1451: Automaton for the SUM_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PLATEAU pattern where `default` is 0

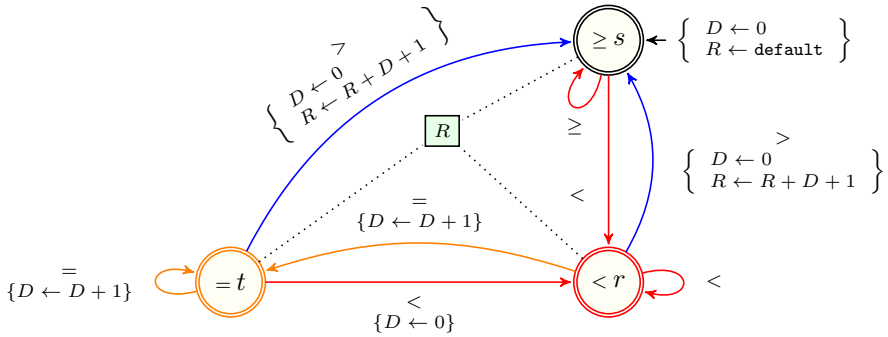


Figure 4.1452: Simplified automaton for the SUM_WIDTH_PLATEAU constraint obtained by applying decoration Table 3.29 to the seed transducer of the PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.369: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the SUM_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.370: Concrete glue matrix, derived from the parametrised glue matrix 3.13, for the simplified automaton of the SUM_WIDTH_PLATEAU constraint defined as the composition of the PLATEAU pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

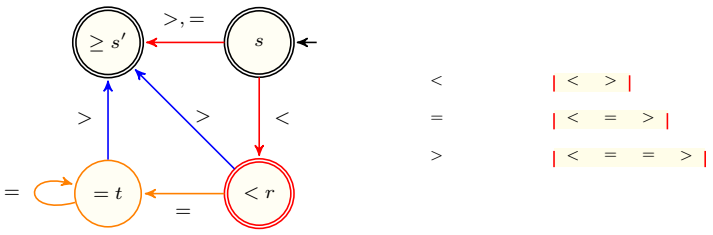


Figure 4.1453: (**left**) Automaton without registers for the SUM_WIDTH_PLATEAU_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the PLATEAU pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

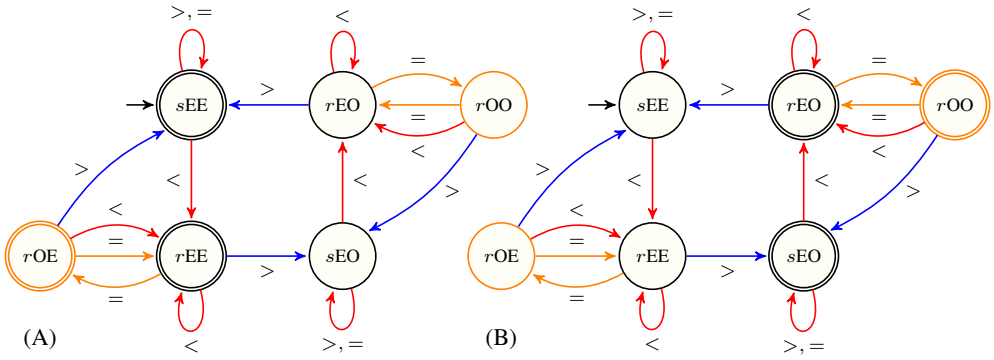
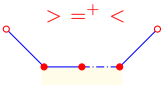


Figure 4.1454: Automata without registers for the (A) SUM_WIDTH_PLATEAU_IS_EVEN and the (B) SUM_WIDTH_PLATEAU_IS_ODD constraints; they respectively achieve an even/odd sum of width of the PLATEAU pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

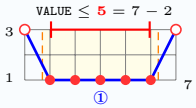
Constraint SUM_WIDTH_PROPER_PLAIN(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1458)

Purpose VALUE is the sum of the width of occurrences of the PROPER_PLAIN pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(7, \langle 2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5\rangle)$

Figure 4.1455 provides an example where the SUM_WIDTH_PROPER_PLAIN $(7, [2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

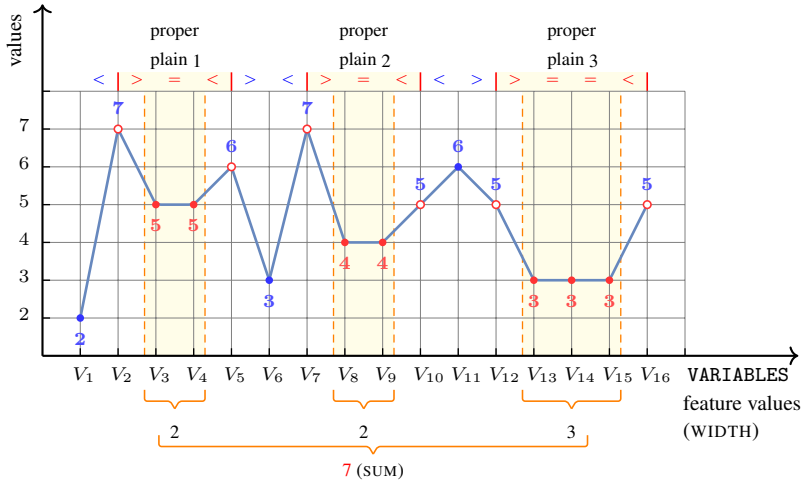


Figure 4.1455: Illustrating the SUM_WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton

Figures 4.1456 and 4.1457 respectively depict the automaton associated with the constraint SUM_WIDTH_PROPER_PLAIN and its simplified form.

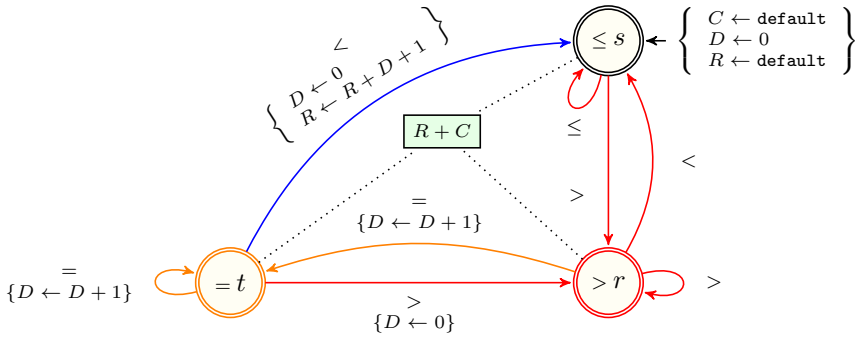


Figure 4.1456: Automaton for the SUM_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0

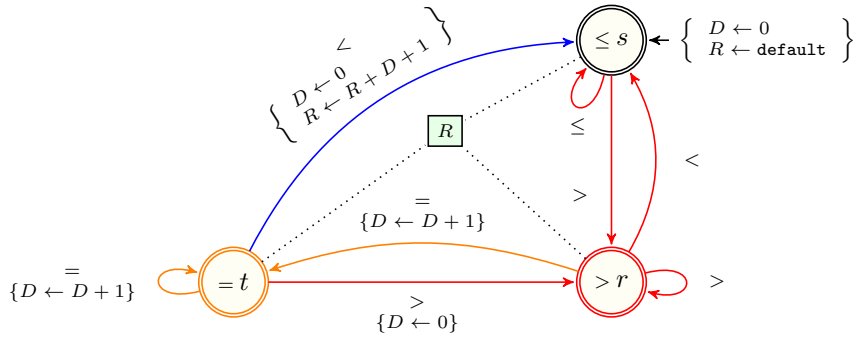


Figure 4.1457: Simplified automaton for the SUM_WIDTH_PROPER_PLAIN constraint obtained by applying decoration Table 3.29 to the seed transducer of the PROPER_PLAIN pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.371: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the SUM_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.372: Concrete glue matrix, derived from the parametrised glue matrix 3.14, for the simplified automaton of the SUM_WIDTH_PROPER_PLAIN constraint defined as the composition of the PROPER_PLAIN pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

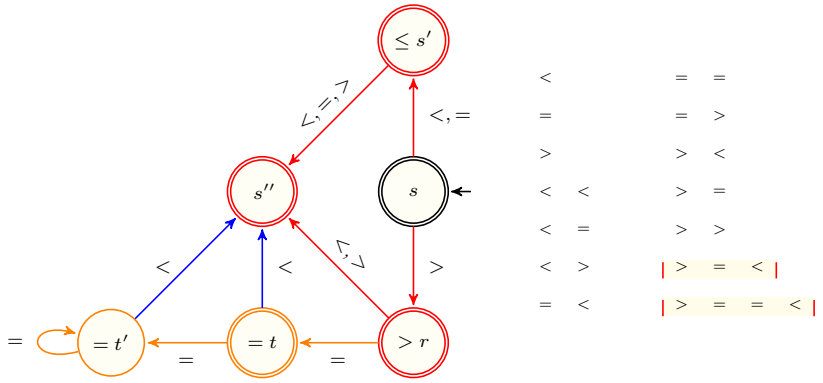


Figure 4.1458: (**left**) Automaton without registers for the SUM_WIDTH_PROPER_PLAIN_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the PROPER_PLAIN pattern of a sequence of sv variables, i.e. $\max(0, \texttt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\texttt{sv} - 1 \in \{1, 2, 3, 4\}$.

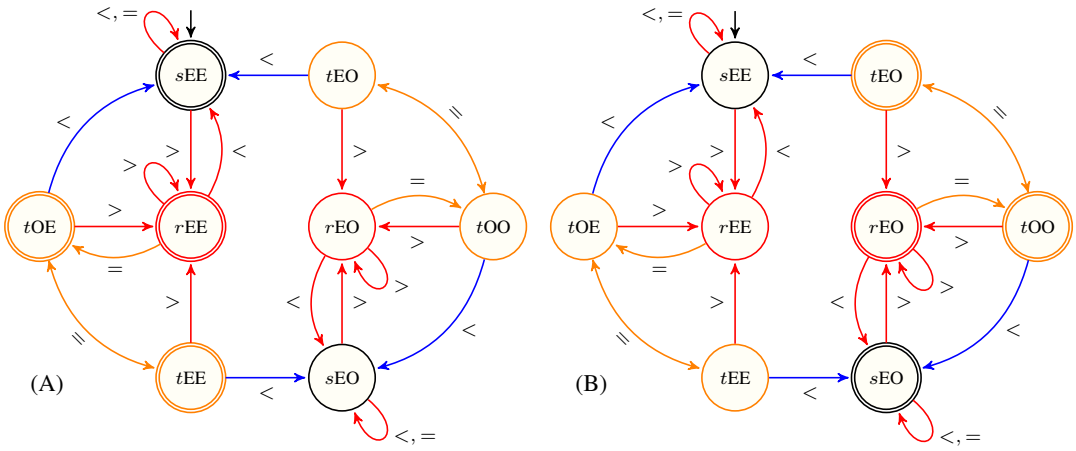
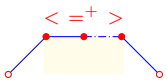


Figure 4.1459: Automata without registers for the (A) SUM_WIDTH_PROPER_PLAIN_IS_EVEN and the (B) SUM_WIDTH_PROPER_PLAIN_IS_ODD constraints; they respectively achieve an even/odd sum of width of the PROPER_PLAIN pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

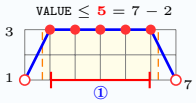
Constraint SUM_WIDTH_PROPER_PLATEAU(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2\textcircled{1})$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1463)

Purpose VALUE is the sum of the width of occurrences of the PROPER_PLATEAU pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(7, \langle 7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3 \rangle)$

Figure 4.1460 provides an example where the SUM_WIDTH_PROPER_PLATEAU $(7, [7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

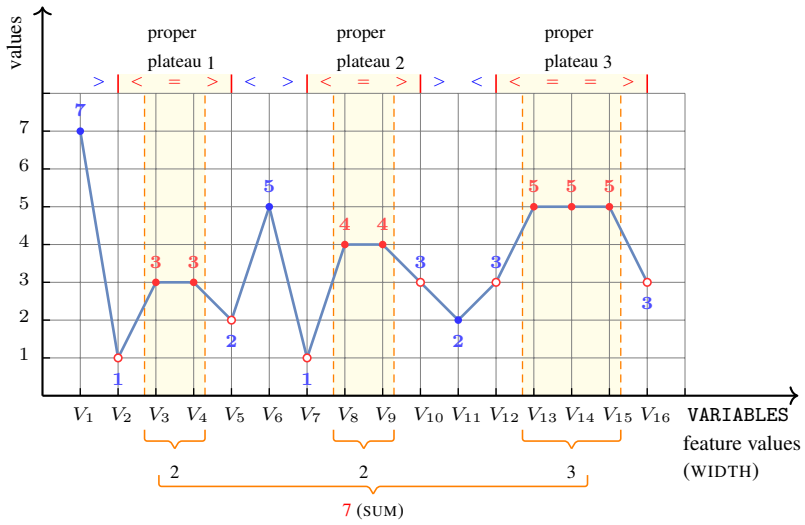


Figure 4.1460: Illustrating the SUM_WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton

Figures 4.1461 and 4.1462 respectively depict the automaton associated with the constraint SUM_WIDTH_PROPER_PLATEAU and its simplified form.

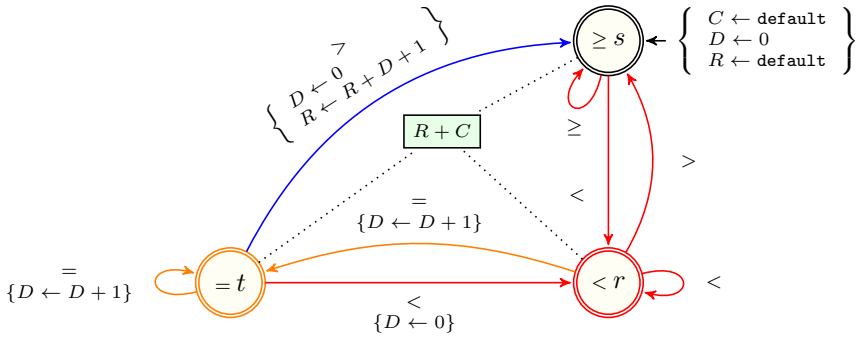


Figure 4.1461: Automaton for the SUM_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.37 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0

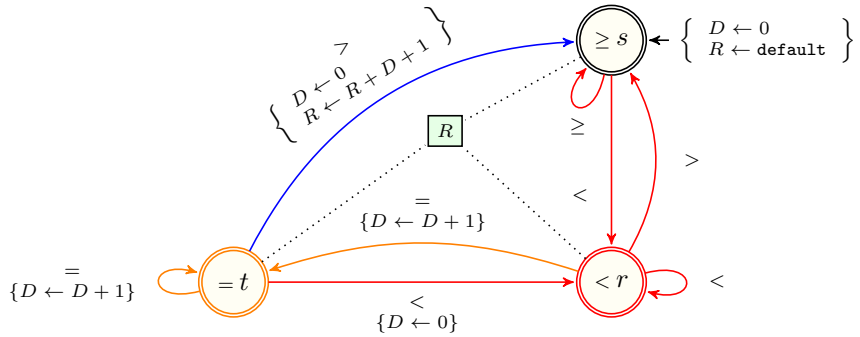


Figure 4.1462: Simplified automaton for the SUM_WIDTH_PROPER_PLATEAU constraint obtained by applying decoration Table 3.29 to the seed transducer of the PROPER_PLATEAU pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.373: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the SUM_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t
s	0	0	0
r	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ C

Table 4.374: Concrete glue matrix, derived from the parametrised glue matrix 3.15, for the simplified automaton of the SUM_WIDTH_PROPER_PLATEAU constraint defined as the composition of the PROPER_PLATEAU pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

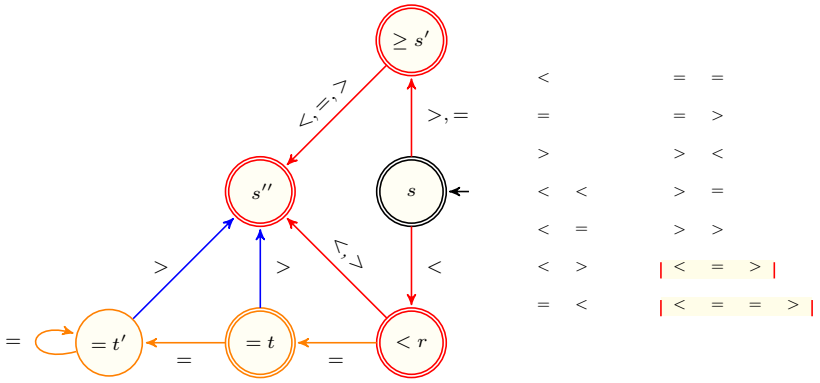


Figure 4.1463: (**left**) Automaton without registers for the SUM_WIDTH_PROPER_PLATEAU_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the PROPER_PLATEAU pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

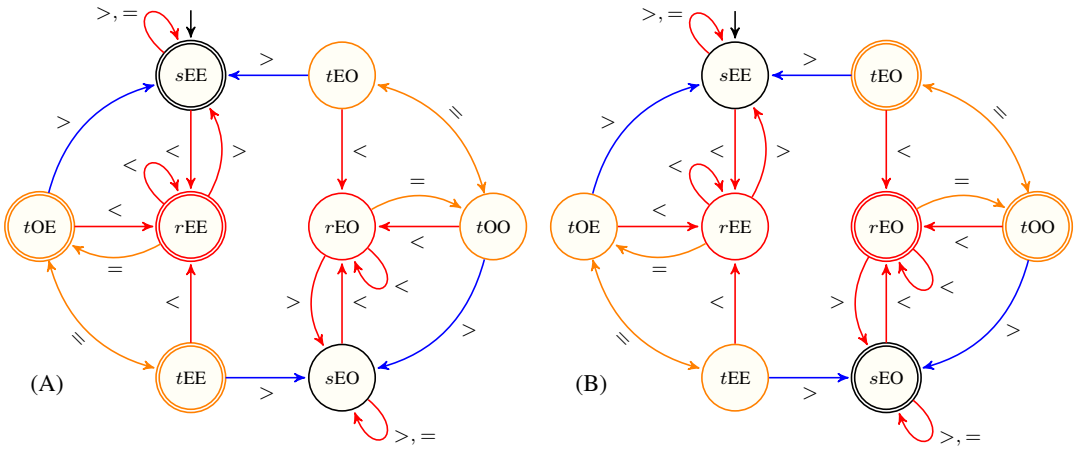
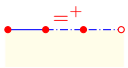


Figure 4.1464: Automata without registers for the (A) SUM_WIDTH_PROPER_PLATEAU_IS_EVEN and the (B) SUM_WIDTH_PROPER_PLATEAU_IS_ODD constraints; they respectively achieve an even/odd sum of width of the PROPER_PLATEAU pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_STEADY_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

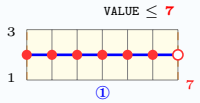
Constraint SUM_WIDTH_STEADY_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{rv} = 1 \Rightarrow \texttt{VALUE} \geq \texttt{sv}$
$\texttt{rv} \geq 2 \Rightarrow \texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \texttt{sv}$①
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



(see also fig. 4.1468)

Purpose VALUE is the sum of the width of occurrences of the STEADY_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(11, \langle 3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1 \rangle)$

Figure 4.1465 provides an example where the SUM_WIDTH_STEADY_SEQUENCE $(11, [3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$

Symmetries

- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

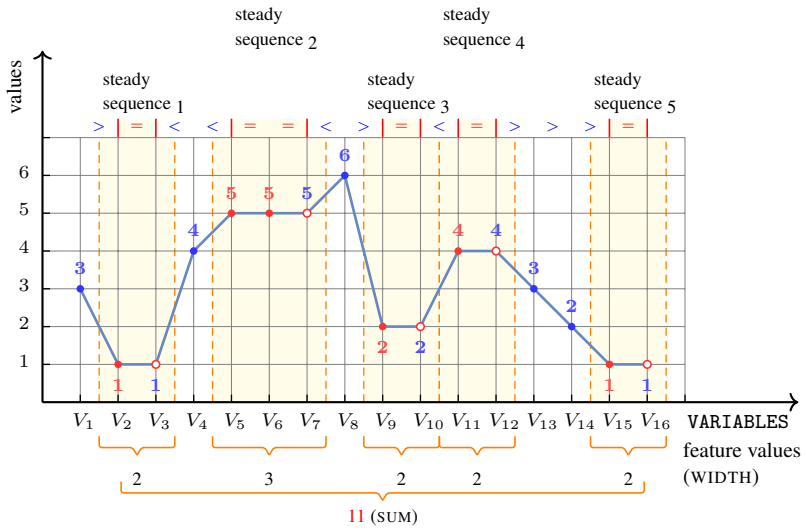


Figure 4.1465: Illustrating the SUM_WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1466 and 4.1467 respectively depict the automaton associated with the constraint SUM_WIDTH_STEADY_SEQUENCE and its simplified form.

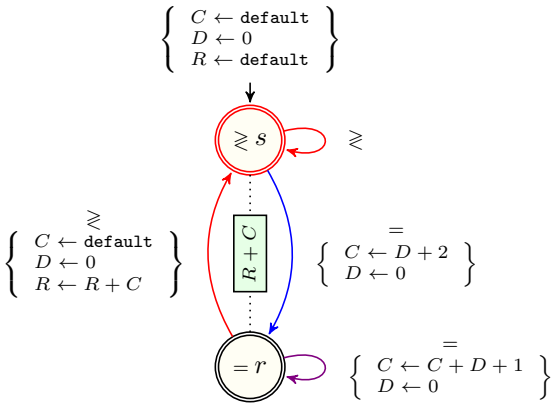


Figure 4.1466: Automaton for the SUM_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STEADY_SEQUENCE pattern where default is 0

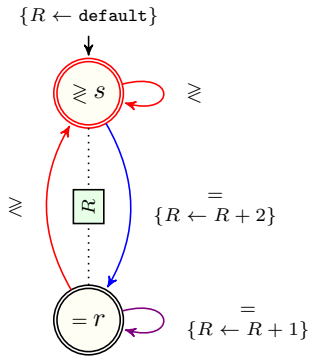


Figure 4.1467: Simplified automaton for the SUM_WIDTH_STEADY_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STEADY_SEQUENCE pattern where default is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.375: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the SUM_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r
s	0	0
r	0	-1 M

Table 4.376: Concrete glue matrix, derived from the parametrised glue matrix 3.17, for the simplified automaton of the SUM_WIDTH_STEADY_SEQUENCE constraint defined as the composition of the STEADY_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

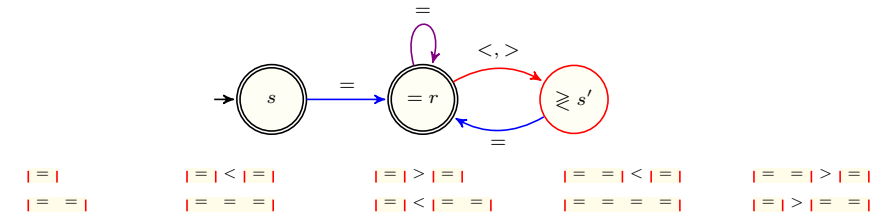


Figure 4.1468: (**top**) Automaton without registers for the SUM_WIDTH_STEADY_SEQUENCE_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the STEADY_SEQUENCE pattern of a sequence of sv variables, i.e. sv of the **Restrictions** slot (see ①). (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

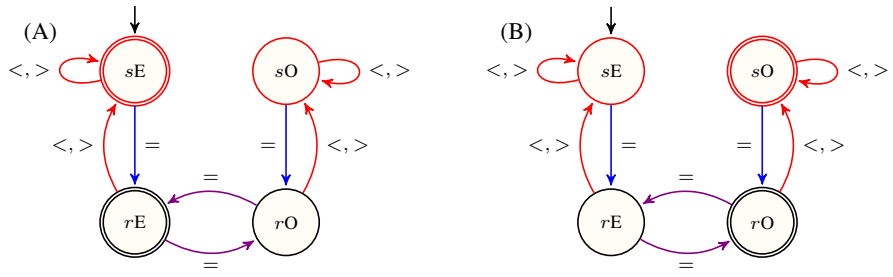


Figure 4.1469: Automaton without registers for the (A) SUM_WIDTH_STEADY_SEQUENCE_IS_EVEN and the (B) SUM_WIDTH_STEADY_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the STEADY_SEQUENCE pattern on a sequence of n variables; within the name of each state the second symbol represents whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

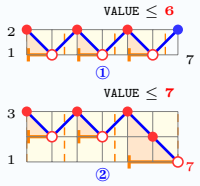
Constraint SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{sv} - \texttt{sv} \bmod 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



(see also fig. 4.1473)

Purpose VALUE is the sum of the width of occurrences of the STRICTLY_DECREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(8, \langle 4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3 \rangle)$

Figure 4.1470 provides an example where the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE $(8, [4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3])$ constraint holds.

Typical $|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

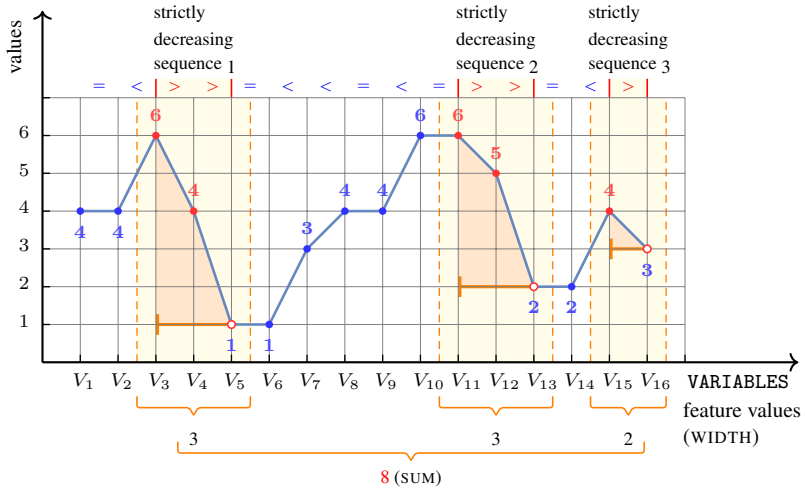


Figure 4.1470: Illustrating the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1471 and 4.1472 respectively depict the automaton associated with the constraint SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE and its simplified form.

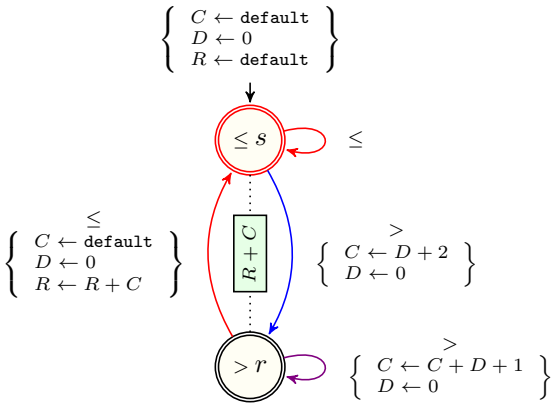


Figure 4.1471: Automaton for the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.377: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

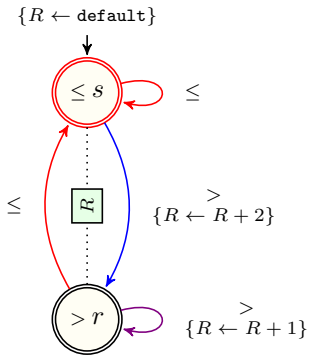


Figure 4.1472: Simplified automaton for the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_DECREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	-1 M

Table 4.378: Concrete glue matrix, derived from the parametrised glue matrix 3.18, for the simplified automaton of the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE constraint defined as the composition of the STRICTLY_DECREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

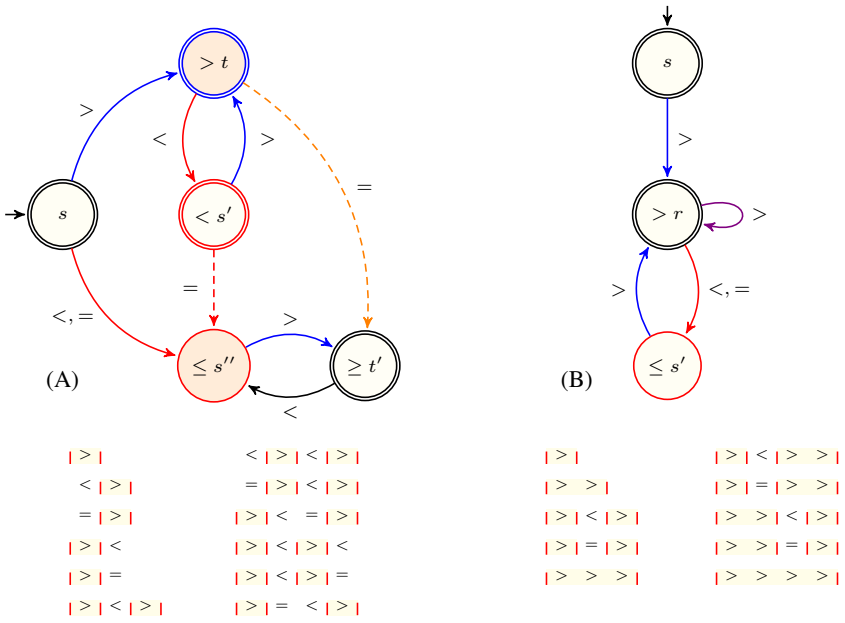


Figure 4.1473: (**top**) Automata without registers for the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the STRICTLY_DECREASING_SEQUENCE pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2, i.e. $\mathtt{sv} - \mathtt{sv} \bmod 2$ of the **Restrictions** slot (see ①), (B) strictly greater than 2, i.e. sv (see ②). Within (A) states s, s' and t' are accepting when $\mathtt{sv} \bmod 2 = 1$, while state t is accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

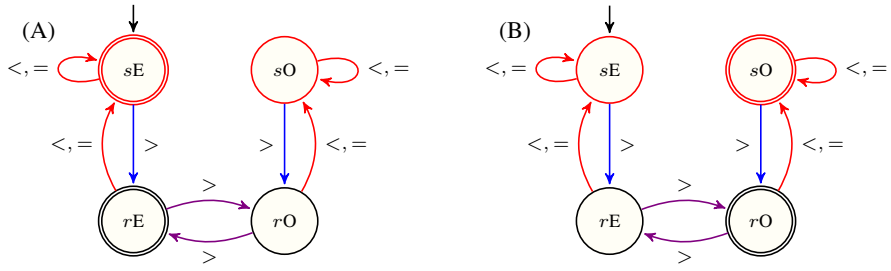


Figure 4.1474: Automata without registers for the (A) SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE_IS_EVEN and the (B) SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the STRICTLY_DECREASING_SEQUENCE pattern on a sequence of n variables; the second symbol represents whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

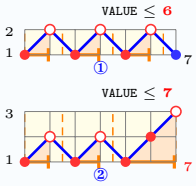
Constraint SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 1 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{sv} - \texttt{sv} \bmod 2$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{sv}$②
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)



(see also fig. 4.1478)

Purpose VALUE is the sum of the width of occurrences of the STRICTLY_INCREASING_SEQUENCE pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example $(10, \langle 4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3 \rangle)$

Figure 4.1475 provides an example where the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE $(10, [4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 1$
range(VARIABLES.var) > 1

Symmetry One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

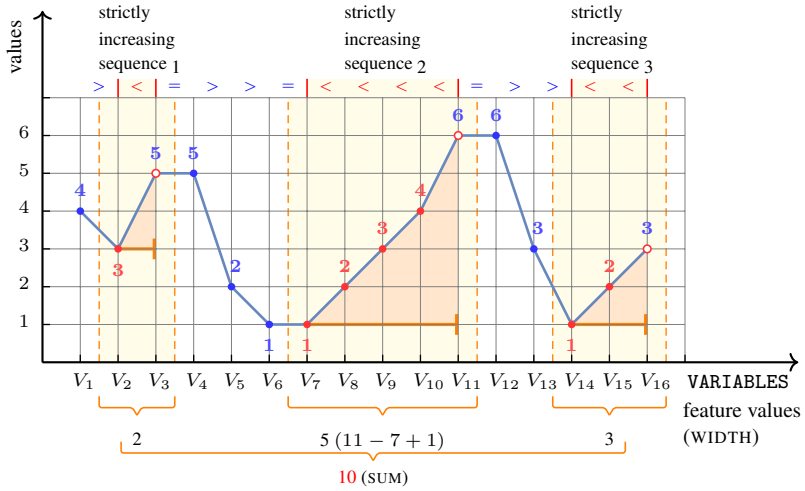


Figure 4.1475: Illustrating the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton

Figures 4.1476 and 4.1477 respectively depict the automaton associated with the constraint SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE and its simplified form.

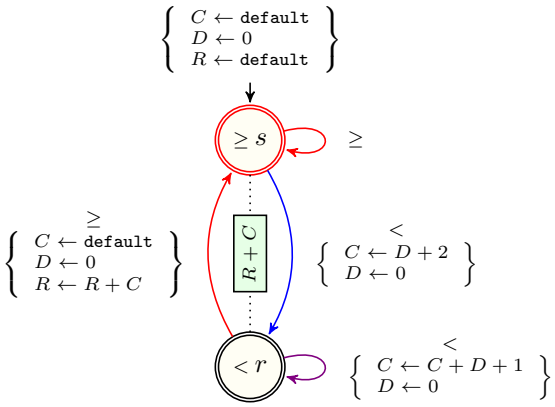


Figure 4.1476: Automaton for the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.37 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0

	s	r
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} - 1$ M

Table 4.379: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

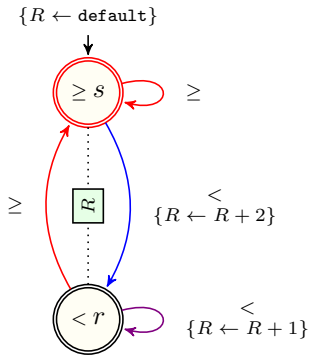


Figure 4.1477: Simplified automaton for the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint obtained by applying decoration Table 3.40 to the seed transducer of the STRICTLY_INCREASING_SEQUENCE pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

	s	r
s	0	0
r	0	-1 M

Table 4.380: Concrete glue matrix, derived from the parametrised glue matrix 3.19, for the simplified automaton of the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE constraint defined as the composition of the STRICTLY_INCREASING_SEQUENCE pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

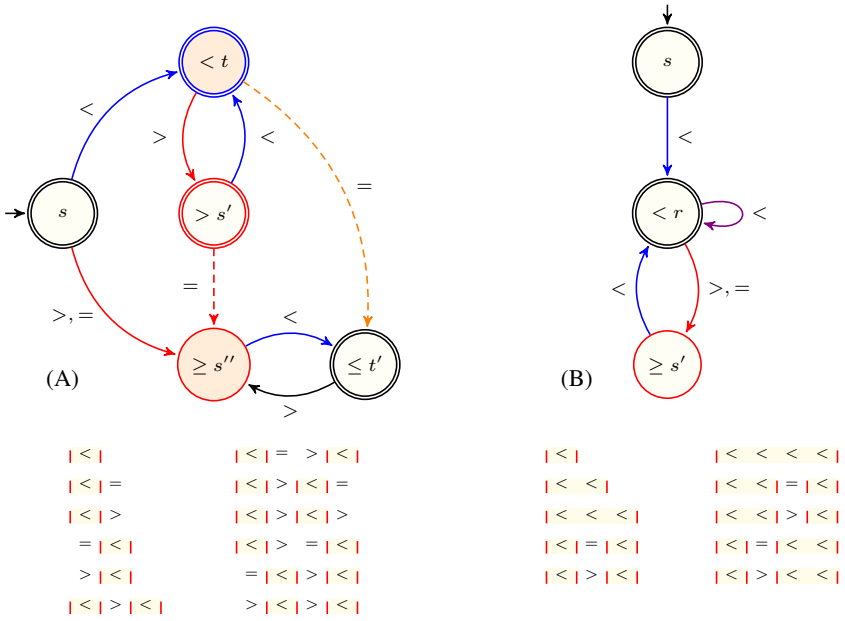


Figure 4.1478: (**top**) Automata without registers for the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the STRICTLY_INCREASING_SEQUENCE pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2 of the **Restrictions** slot (see ①), i.e. $\mathtt{sv} - \mathtt{sv} \bmod 2$, (B) strictly greater than 2, i.e. sv (see ②). Within (A) states s, s' and t' are accepting when $\mathtt{sv} \bmod 2 = 1$, while state t is accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

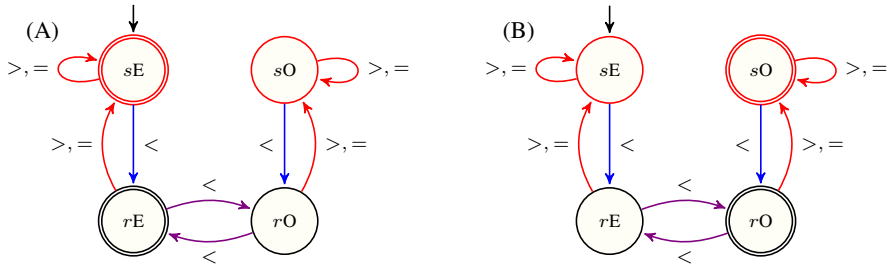


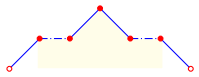
Figure 4.1479: Automata without registers for the (A) SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE_IS_EVEN and the (B) SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE_IS_ODD constraints; they respectively achieve an even/odd sum of width of the STRICTLY_INCREASING_SEQUENCE pattern on a sequence of n variables; the second symbol represents whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the SUMMIT pattern.

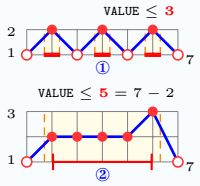
Constraint SUM_WIDTH_SUMMIT(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{rv} = 2 \Rightarrow \texttt{VALUE} \leq \texttt{np}$①
$\texttt{rv} \geq 3 \Rightarrow \texttt{VALUE} \leq \texttt{max}(0, \texttt{sv} - 2$②$)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} =$range(VARIABLES.var)
$\texttt{np} = \texttt{max}(0, \lfloor(\texttt{sv} - 1)/2\rfloor)$



(see also fig. 4.1483)

Purpose VALUE is the sum of the width of occurrences of the SUMMIT pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(6, \langle 7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1\rangle)$

Figure 4.1480 provides an example where the SUM_WIDTH_SUMMIT $(6, [7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

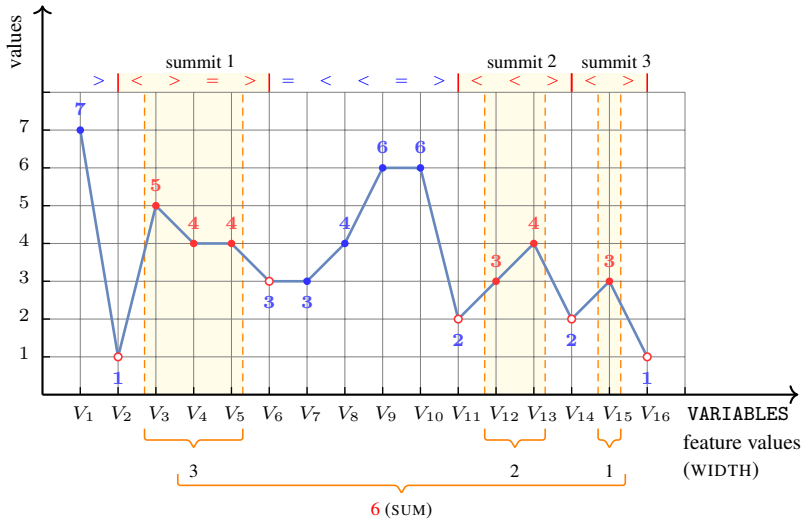


Figure 4.1480: Illustrating the SUM_WIDTH_SUMMIT constraint of the **Example** slot

Automaton Figures 4.1481 and 4.1482 respectively depict the automaton associated with the constraint SUM_WIDTH_SUMMIT and its simplified form.

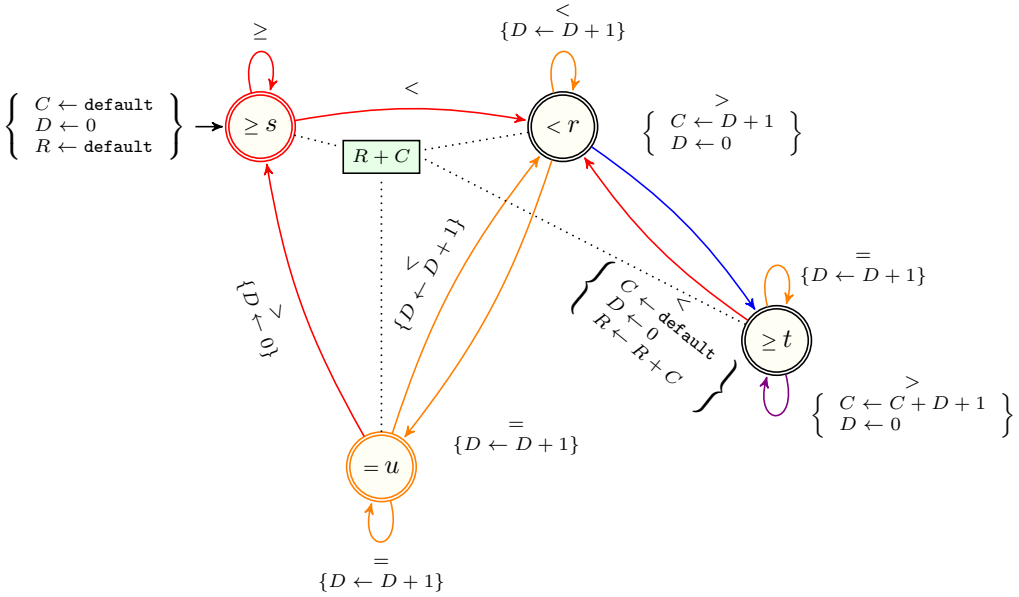


Figure 4.1481: Automaton for the SUM_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.37 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$)

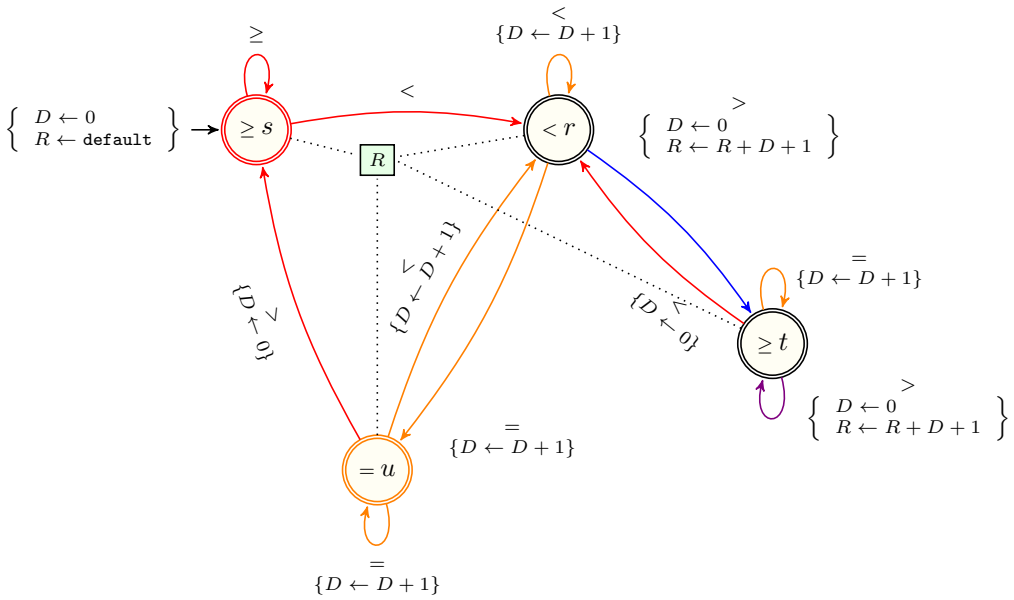


Figure 4.1482: Simplified automaton for the SUM_WIDTH_SUMMIT constraint obtained by applying decoration Table 3.26 to the seed transducer of the SUMMIT pattern where `default` is 0 (transition $u \rightarrow r$ has the same register update as transition $r \rightarrow u$); $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t	u
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\overrightarrow{C} + \overleftarrow{C}$
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L
u	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.381: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the SUM_WIDTH_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	r	t	u
s	0	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L
u	0	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ R	0

Table 4.382: Concrete glue matrix, derived from the parametrised glue matrix 3.20, for the simplified automaton of the SUM_WIDTH_SUMMIT constraint defined as the composition of the SUMMIT pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

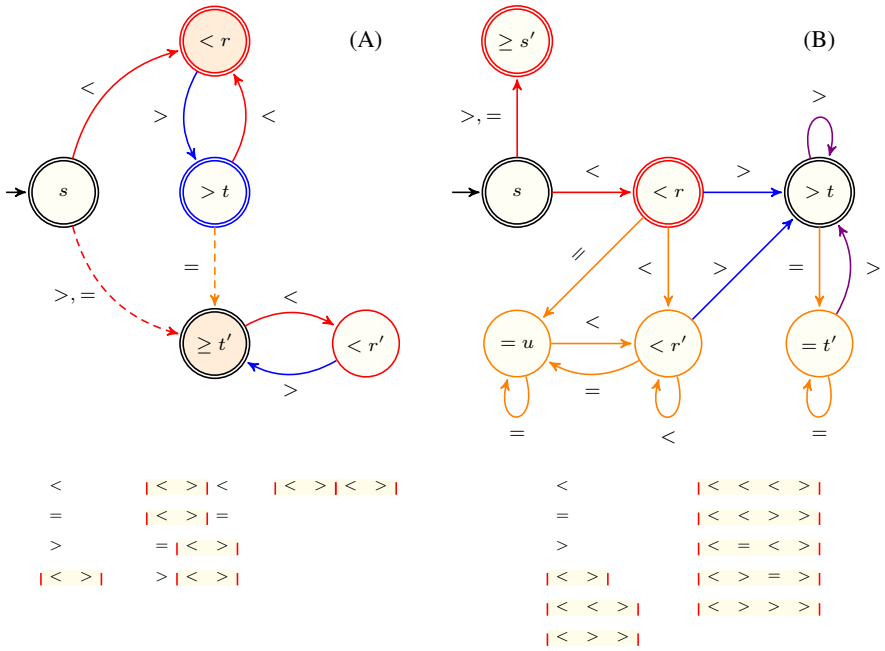


Figure 4.1483: (**top**) Automata without registers for the SUM_WIDTH_SUMMIT_EQ_UP_WHEN_RANGE_EQ_2 and the SUM_WIDTH_SUMMIT_EQ_UP constraints; they describe all sequences maximising the sum of the widths of all the occurrences of the SUMMIT pattern of a sequence of sv variables when the difference between the maximum and the minimum of the variables plus one of the sequence of variables is (A) equal to 2, i.e. $\max\left(0, \left\lfloor \frac{\mathtt{sv}-1}{2} \right\rfloor\right)$ of the **Restrictions** slot (see ①), (B) strictly greater than 2, i.e. $\max(0, \mathtt{sv} - 2)$ (see ②). Within (A) states s and t are accepting when $\mathtt{sv} \bmod 2 = 1$, while states r and t' are accepting when $\mathtt{sv} \bmod 2 = 0$. (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{1, 2, 3, 4\}$.

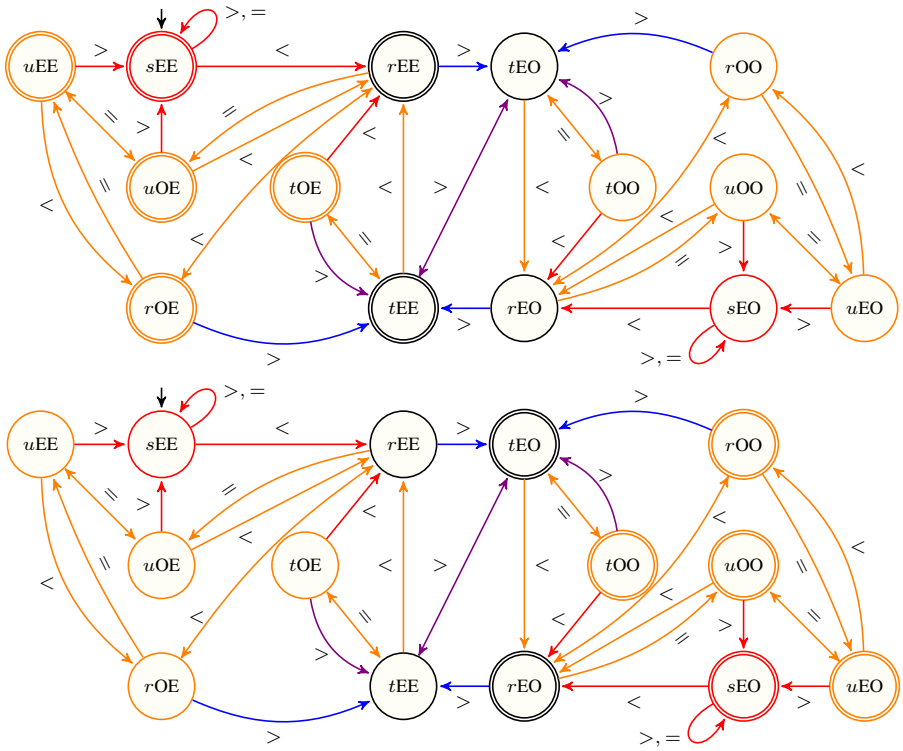


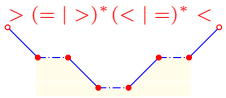
Figure 4.1484: Automata without registers for the (A) SUM_WIDTH_SUMMIT_IS_EVEN and the (B) SUM_WIDTH_SUMMIT_IS_ODD constraints; they respectively achieve an even/odd sum of width of the SUMMIT pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

SUM_WIDTH_SUMMIT 3029

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

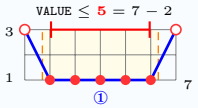
Constraint SUM_WIDTH_VALLEY(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 2 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 1$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
$\texttt{rv} = \text{range}(\texttt{VARIABLES.var})$



(see also fig. 4.1488)

Purpose

VALUE is the sum of the width of occurrences of the VALLEY pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(9, \langle 1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7 \rangle)$

Figure 4.1485 provides an example where the SUM_WIDTH_VALLEY $(9, [1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 2$
$\text{range}(\texttt{VARIABLES.var}) > 1$

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

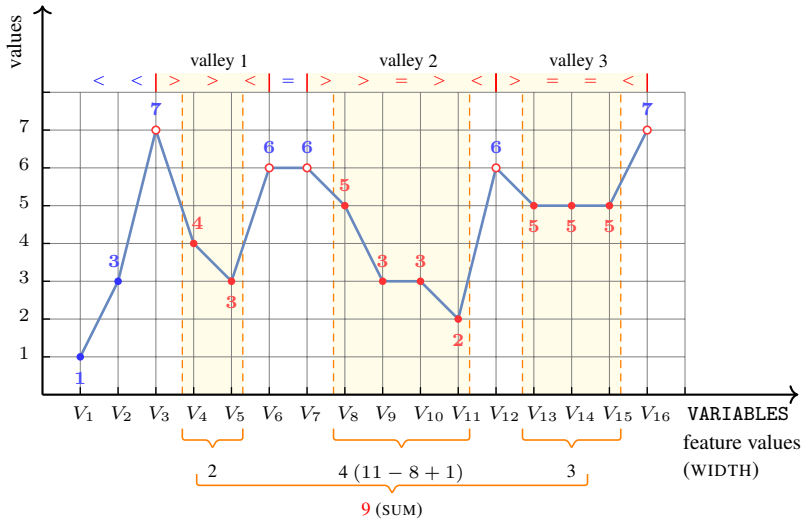


Figure 4.1485: Illustrating the SUM_WIDTH_VALLEY constraint of the **Example** slot

Automaton

Figures 4.1486 and 4.1487 respectively depict the automaton associated with the constraint SUM_WIDTH_VALLEY and its simplified form.

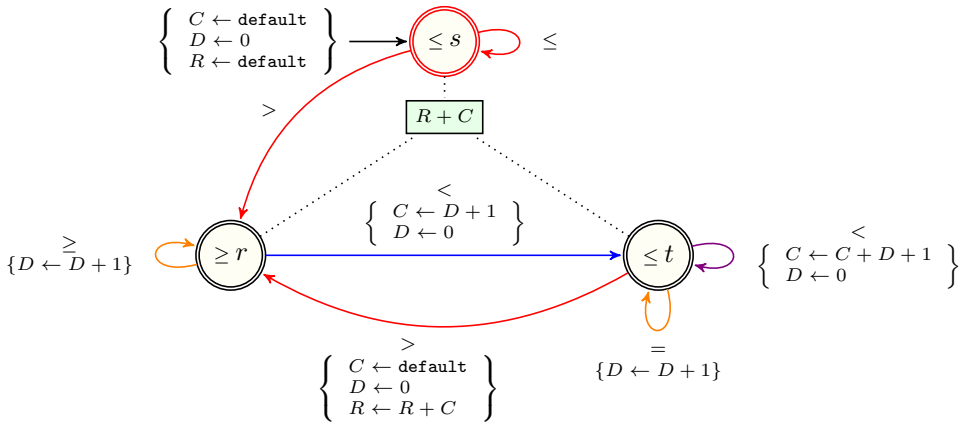


Figure 4.1486: Automaton for the SUM_WIDTH_VALLEY constraint obtained by applying decoration Table 3.37 to the seed transducer of the VALLEY pattern where `default` is 0

	s	r	t
s	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overleftarrow{C}$
r	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overleftarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ R
t	$\overrightarrow{C} + \overleftarrow{C}$	$\overrightarrow{C} + \overrightarrow{D} + \overleftarrow{D} + 1$ L	$\overrightarrow{C} + \overleftarrow{C}$

Table 4.383: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the SUM_WIDTH_VALLEY constraint defined as the composition of the VALLEY pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

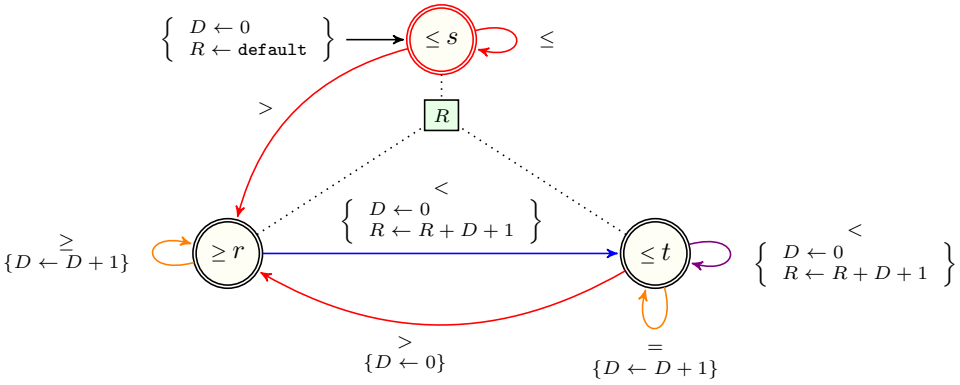


Figure 4.1487: Simplified automaton for the SUM_WIDTH_VALLEY constraint obtained by applying decoration Table 3.26 to the seed transducer of the VALLEY pattern where `default` is 0; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + D_{i-1} + 1 \geq 0$ are linear invariants.

	s	r	t
s	0	0	0
r	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ C	$\overrightarrow{D} + \overleftarrow{D} + 1$ R
t	0	$\overrightarrow{D} + \overleftarrow{D} + 1$ L	0

Table 4.384: Concrete glue matrix, derived from the parametrised glue matrix 3.21, for the simplified automaton of the SUM_WIDTH_VALLEY constraint defined as the composition of the VALLEY pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

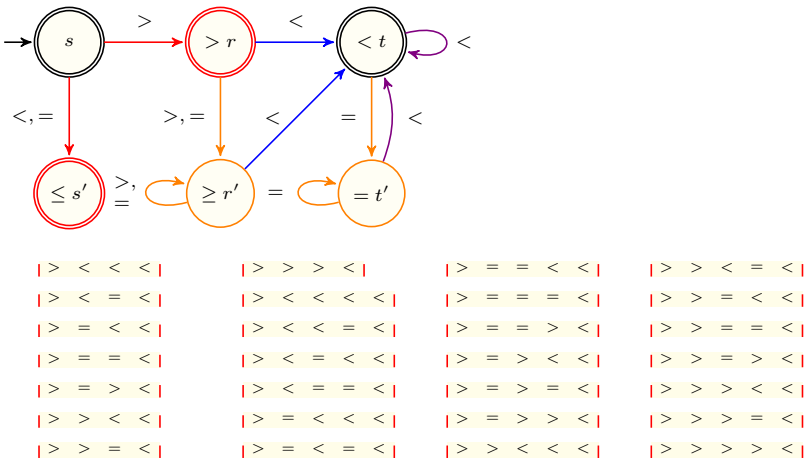


Figure 4.1488: (**top**) Automaton without registers for the SUM_WIDTH_VALLEY_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the VALLEY pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**bottom**) All corresponding solutions for $\mathtt{sv} - 1 \in \{4, 5\}$.

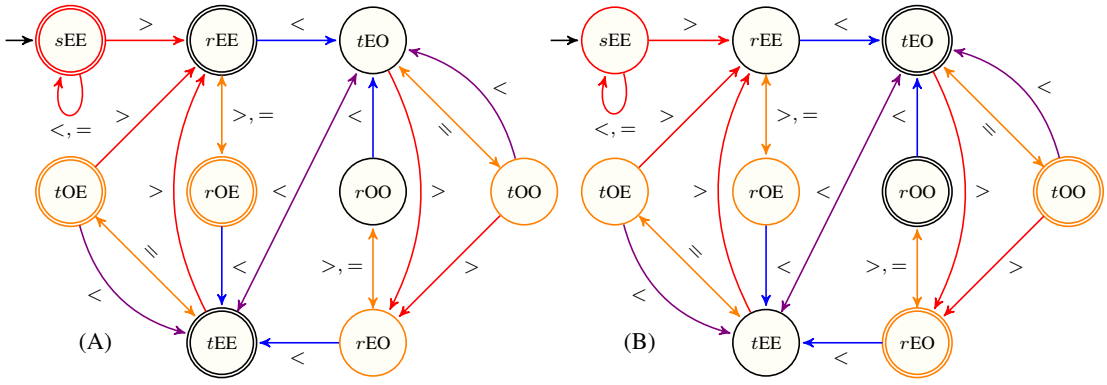
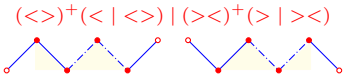


Figure 4.1489: Automata without registers for the SUM_WIDTH_VALLEY_IS_EVEN and the (B) SUM_WIDTH_VALLEY_IS_ODD constraints; they respectively achieve an even/odd sum of width of the VALLEY pattern on a sequence of n variables; the second symbol represents whether the register D is even or odd, while the third symbol denotes whether the register R is even or odd.

AGGREGATOR FEATURE PATTERN

SUM_WIDTH_ZIGZAG

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

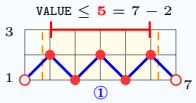
Constraint SUM_WIDTH_ZIGZAG(VALUE, VARIABLES)

Arguments

VALUE	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

$\texttt{sv} \leq 3 \vee \texttt{rv} \leq 1 \Rightarrow \texttt{VALUE} = 0$
$\texttt{VALUE} = 0 \vee \texttt{VALUE} \geq 2$
$\texttt{VALUE} \leq \max(0, \texttt{sv} - 2①)$
required(VARIABLES, var)
where
$\texttt{sv} = |\texttt{VARIABLES}|$
rv =range(VARIABLES.var)



(see also fig. 4.1493)

Purpose

VALUE is the sum of the width of occurrences of the ZIGZAG pattern in the time-series given by the VARIABLES collection. If the pattern does not occur, VALUE takes the default value 0.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example $(11, \langle 4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1 \rangle)$

Figure 4.1490 provides an example where the SUM_WIDTH_ZIGZAG $(11, [4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1])$ constraint holds.

Typical
$|\texttt{VARIABLES}| > 3$
range(VARIABLES.var) > 1

Symmetries
- Items of VARIABLES can be reversed.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Arg. properties Functional dependency: VALUE determined by VARIABLES.

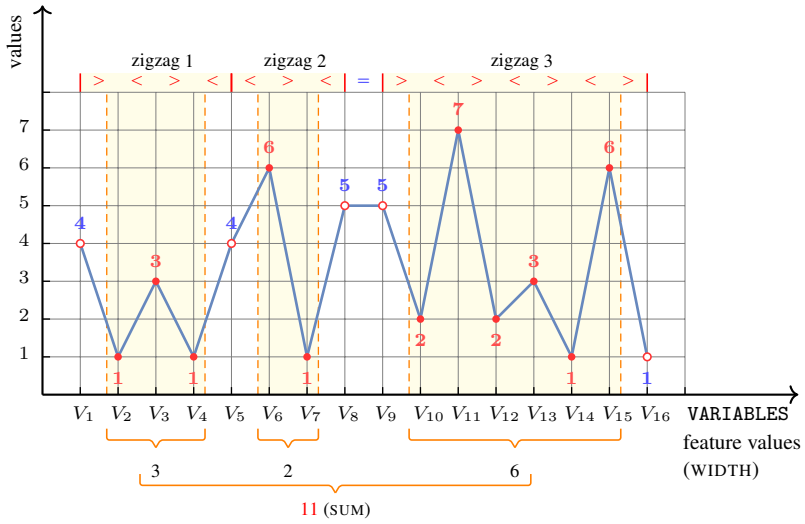


Figure 4.1490: Illustrating the SUM_WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Figures 4.1491 and 4.1492 respectively depict the automaton associated with the constraint SUM_WIDTH_ZIGZAG and its simplified form.

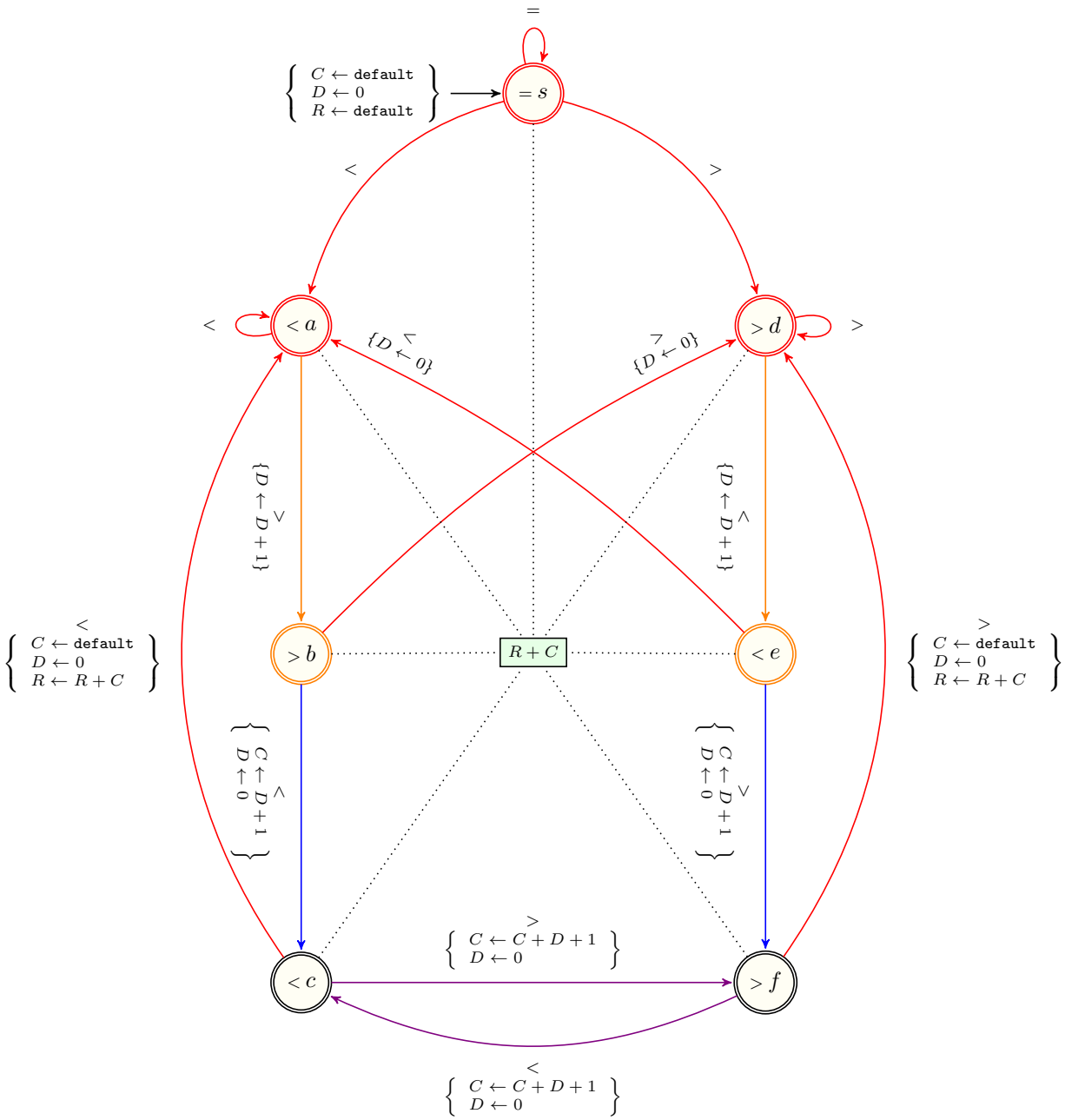


Figure 4.1491: Automaton for the SUM_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.37 to the seed transducer of the ZIGZAG pattern where `default` is 0; (1) missing transitions from a, b, c, d, e, f to s are labelled by $=$; (2) on transitions from b, c, e, f to s the register D is reset to its initial value; (3) on transitions from c, f to s the register R is updated wrt C and the register C is reset to its initial value

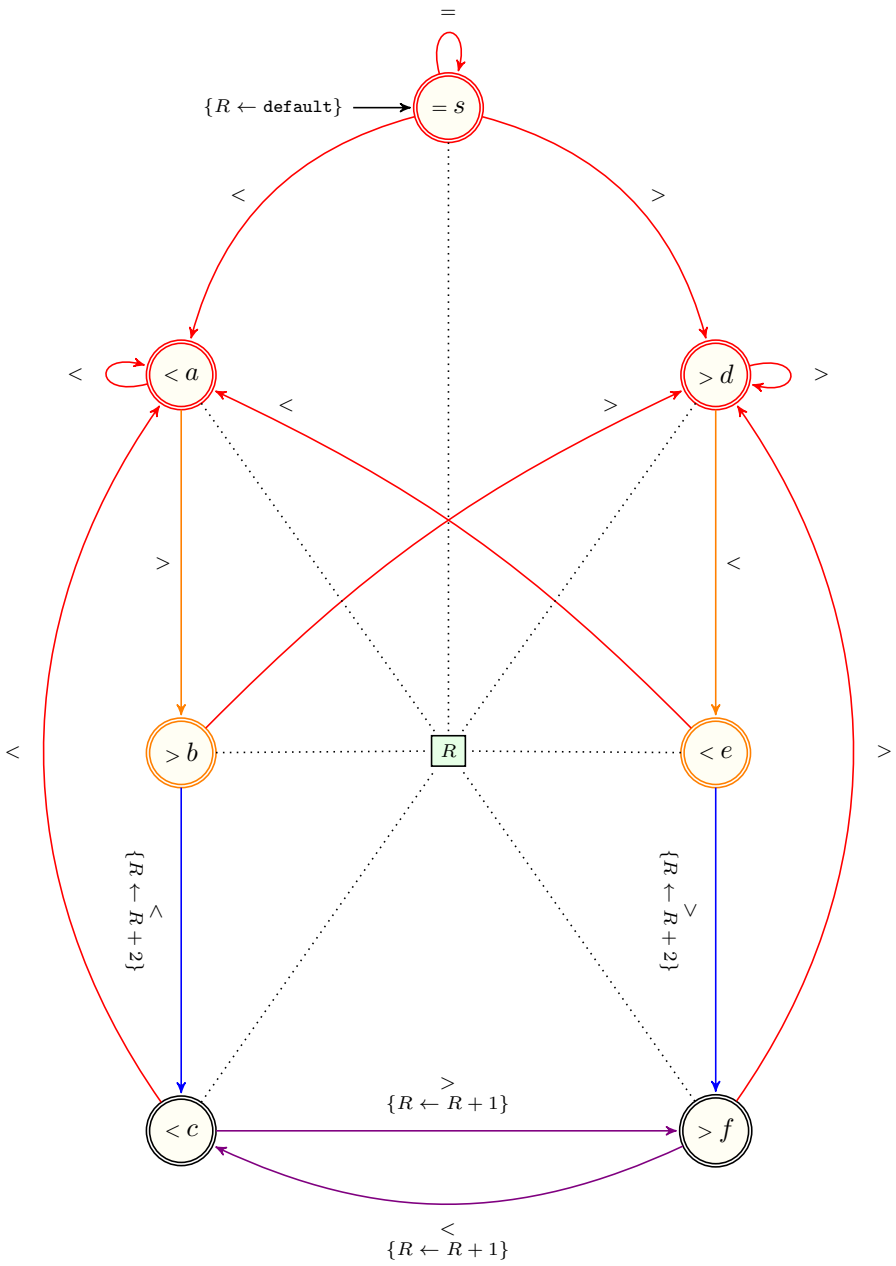


Figure 4.1492: Simplified automaton for the SUM_WIDTH_ZIGZAG constraint obtained by applying decoration Table 3.41 to the seed transducer of the ZIGZAG pattern where `default` is 0; missing transitions from a, b, c, d, e, f to s are labelled by $=$; $R_i - R_{i-1} \geq 0$ and $-R_i + R_{i-1} + 2 \geq 0$ are linear invariants.

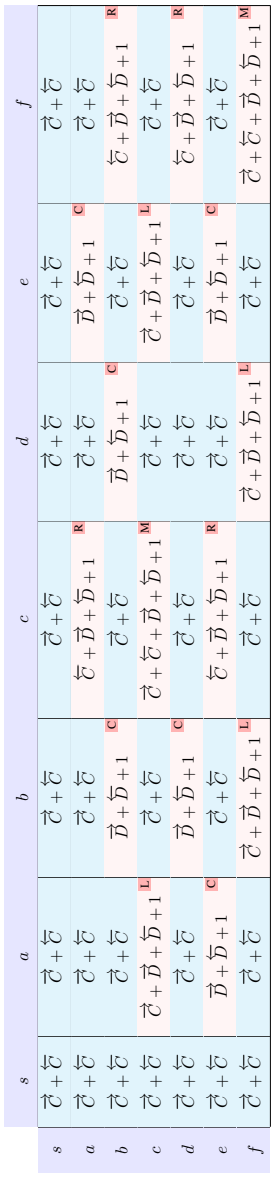


Table 4.385: Concrete glue matrix, derived from the parametrised glue matrix 3.22, for the SUM_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

	s	a	b	c	d	e	f
s	0	0	0	0	0	0	0
a	0	0	0	1 R	0	2 C	0
b	0	0	3 C	0	2 C	0	2 R
c	0	1 L	0	1 M	0	2 L	0
d	0	0	2 C	0	0	0	1 R
e	0	2 C	0	2 R	0	3 C	0
f	0	0	2 L	0	1 L	0	1 M

Table 4.386: Concrete glue matrix, not directly derived from the parametrised glue matrix 3.22, for the simplified automaton of the SUM_WIDTH_ZIGZAG constraint defined as the composition of the ZIGZAG pattern, the feature WIDTH, and the aggregator sum; cells of the glue matrix are coloured with the colour of the constituent to which they are related.

Specialisation

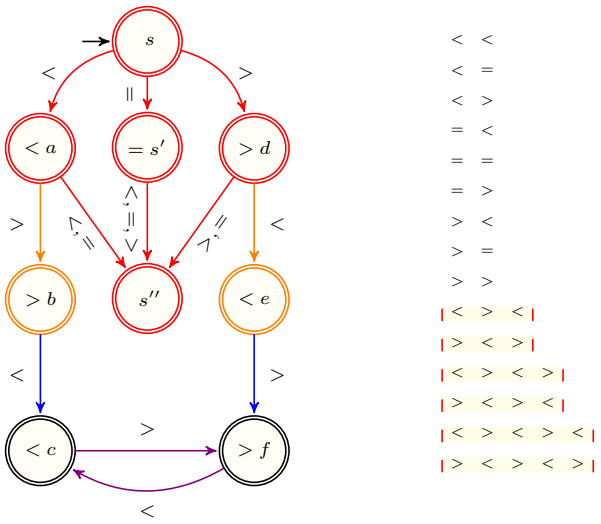


Figure 4.1493: (**left**) Automaton without registers for the SUM_WIDTH_ZIGZAG_EQ_UP constraint; it describes all sequences maximising the sum of the widths of all the occurrences of the ZIGZAG pattern of a sequence of sv variables, i.e. $\max(0, \mathtt{sv} - 2)$ of the **Restrictions** slot (see ①). (**right**) All corresponding solutions for $\mathtt{sv} - 1 \in \{2, 3, 4, 5\}$.

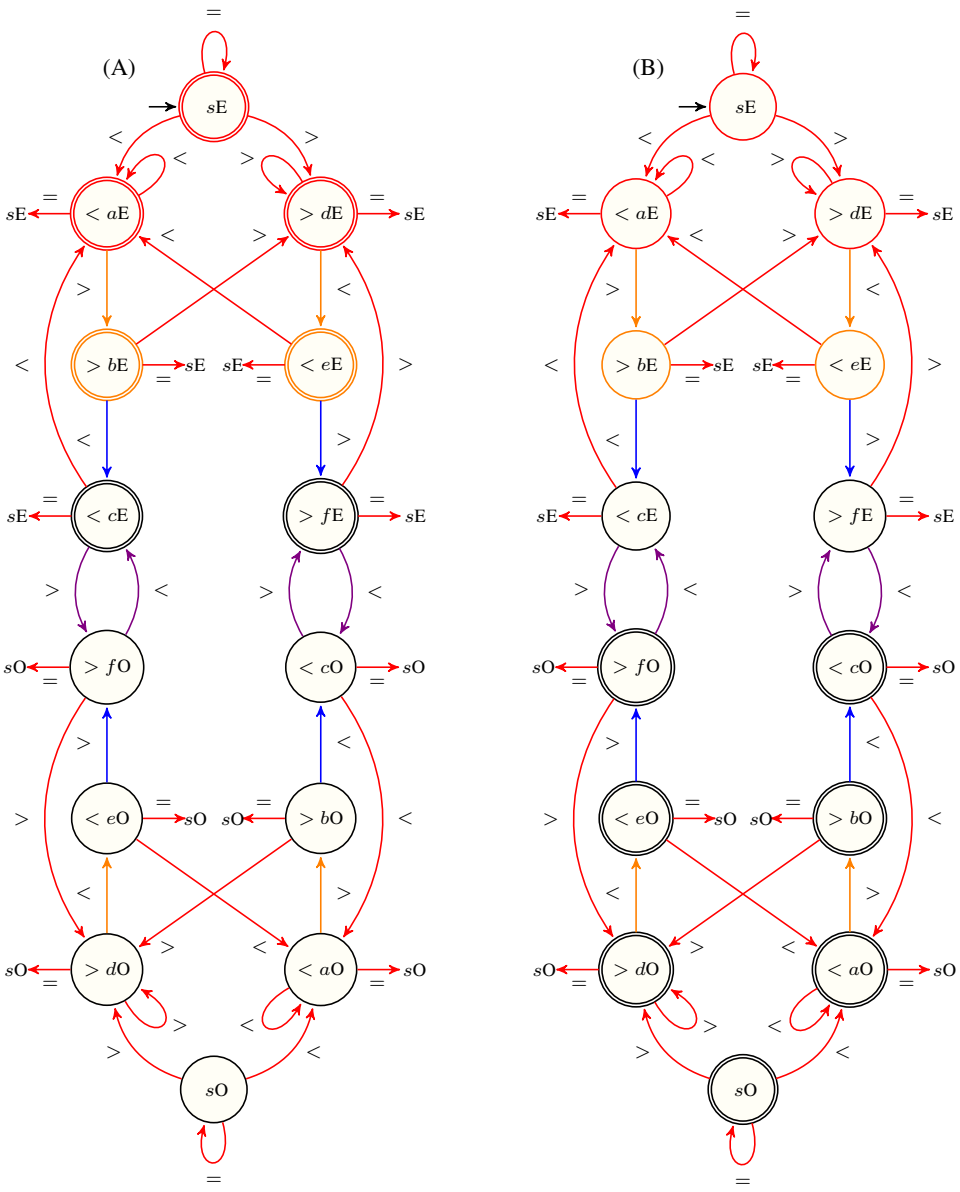
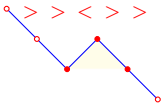


Figure 4.1494: Automata without registers for the (A) SUM_WIDTH_ZIGZAG_IS_EVEN and the (B) SUM_WIDTH_ZIGZAG_IS_ODD constraints; they respectively achieve an even/odd sum of width of the ZIGZAG pattern on a sequence of n variables; the second symbol represents whether the register R is even or odd.

FEATURE PATTERN

SURF_BUMP_ON_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the BUMP_ON_DECREASING_SEQUENCE pattern.

Constraint SURF_BUMP_ON_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of BUMP_ON_DECREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of BUMP_ON_DECREASING_SEQUENCE.
An occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE is the subsequence which matches the regular expression '>><>>'.
Assume that the occurrence of the pattern BUMP_ON_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example Figure 4.1495 provides an example where the SURF_BUMP_ON_DECREASING_SEQUENCE $([7, 6, 5, 6, 5, 4, 1, 4, 7, 5, 4, 2, 5, 4, 3, 3], [0, 0, 0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 11, 0, 0], 0)$ constraint holds.

Typical

$|\text{VARIABLES}| > 5$
$\text{range}(\text{VARIABLES.var}) > 2$

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

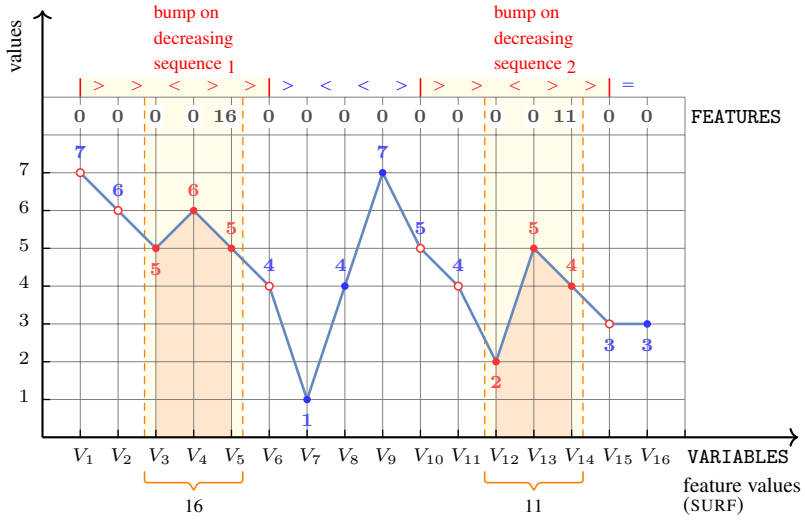


Figure 4.1495: Illustrating the SURF_BUMP_ON_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_DECREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING pattern.

Constraint SURF_DECREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING.
An occurrence of the pattern DECREASING is the subsequence which matches the regular expression '$>$'.
Assume that the occurrence of the pattern DECREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example Figure 4.1496 provides an example where the SURF_DECREASING $([3,4,2,2,5,6,6,4,4,3,1,1,4,6,4,4],[0,6,0,0,0,0,0,10,0,7,4,0,0,0,10,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

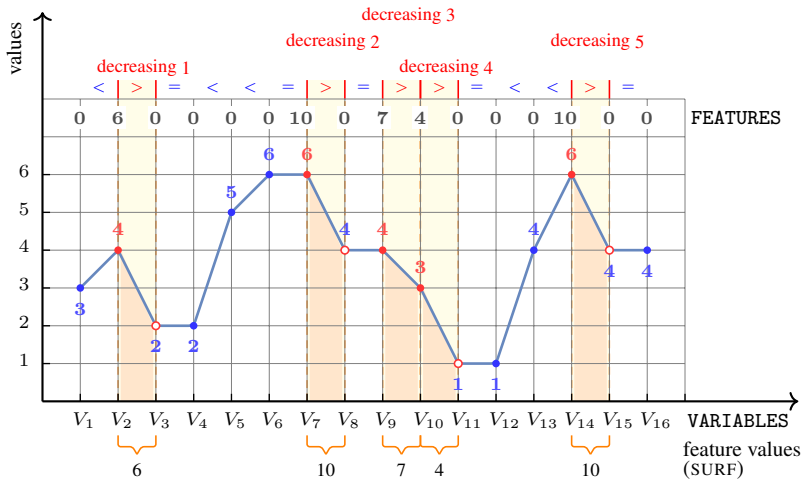
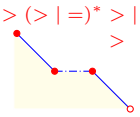


Figure 4.1496: Illustrating the SURF_DECREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_DECREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint SURF_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_SEQUENCE.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example Figure 4.1497 provides an example where the SURF_DECREASING_SEQUENCE $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 6, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 0, 10, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

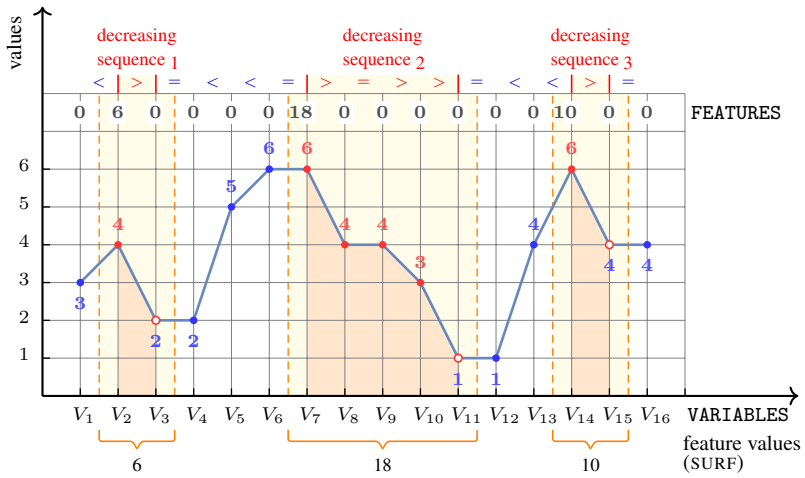


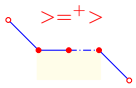
Figure 4.1497: Illustrating the SURF_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_DECREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint SURF_DECREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_TERRACE is identified then FEATURES[i] is the default value DEFAULT; otherwise FEATURES[i] gives the feature value of the corresponding occurrence of DECREASING_TERRACE.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1498 provides an example where the SURF_DECREASING_TERRACE $([6,4,4,4,5,2,2,1,3,3,5,4,4,3,3,3],[0,0,0,0,0,0,0,4,0,0,0,0,0,8,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

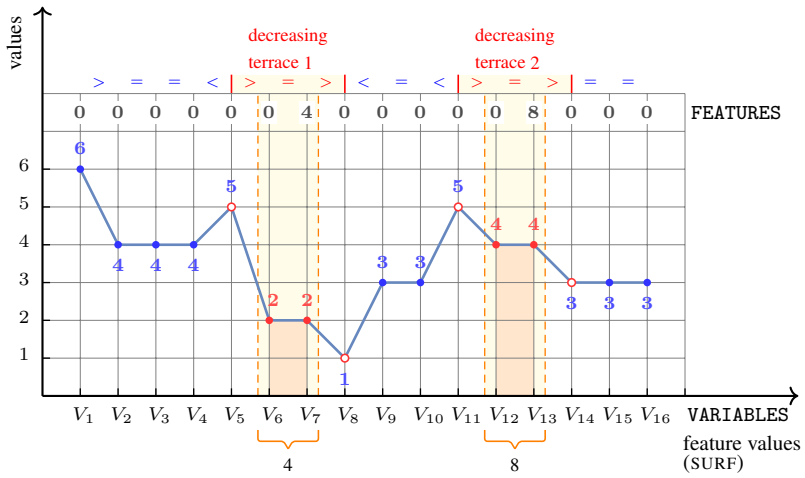
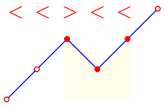


Figure 4.1498: Illustrating the SURF_DECREASING_TERRACE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_DIP_ON_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the DIP_ON_INCREASING_SEQUENCE pattern.

Constraint SURF_DIP_ON_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DIP_ON_INCREASING_SEQUENCE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DIP_ON_INCREASING_SEQUENCE.
An occurrence of the pattern DIP_ON_INCREASING_SEQUENCE is the subsequence which matches the regular expression '<<><<'.
Assume that the occurrence of the pattern DIP_ON_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 2$ to index j.

Example Figure 4.1499 provides an example where the SURF_DIP_ON_INCREASING_SEQUENCE $([1, 2, 3, 2, 5, 6, 7, 4, 1, 3, 4, 6, 1, 2, 4, 4], [0, 0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 5
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

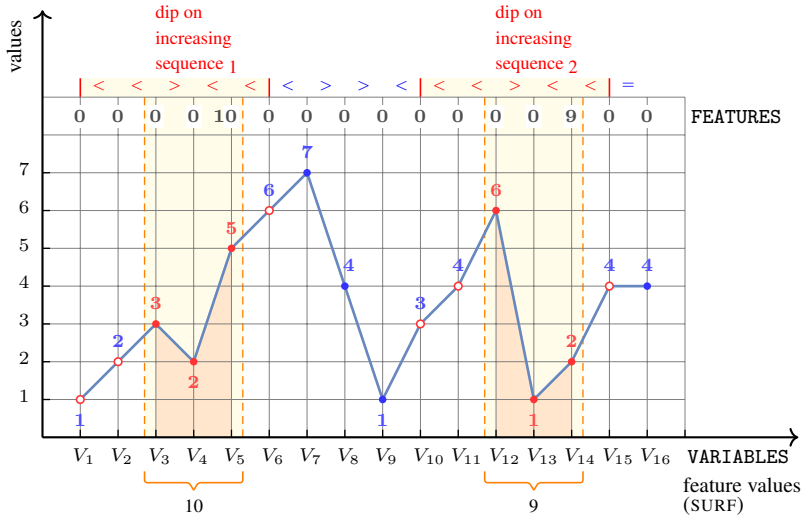


Figure 4.1499: Illustrating the SURF_DIP_ON_INCREASING_SEQUENCE constraint of the **Example** slot

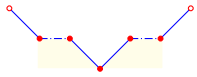
Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_GORGE

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$



Origin Based on the GORGE pattern.

Constraint SURF_GORGE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of GORGE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of GORGE.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1500 provides an example where the SURF_GORGE $([1,7,3,4,4,5,5,4,2,2,6,5,4,6,5,7],[0,0,11,0,0,0,0,0,0,0,0,0,9,0,5,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

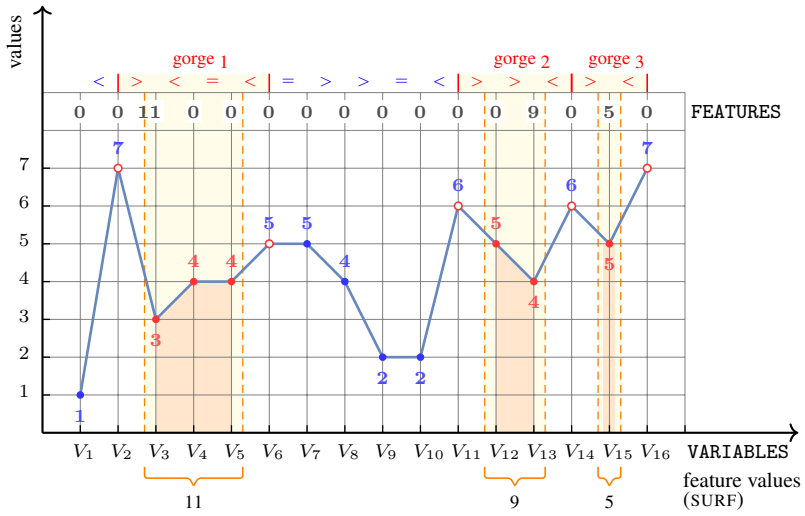


Figure 4.1500: Illustrating the SURF_GORGE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_INCREASING

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING pattern.

Constraint SURF_INCREASING(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING.
An occurrence of the pattern INCREASING is the subsequence which matches the regular expression '$<$'.
Assume that the occurrence of the pattern INCREASING starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example Figure 4.1501 provides an example where the SURF_INCREASING $([4,3,5,5,2,1,1,3,3,4,6,6,3,1,3,3],[0,8,0,0,0,0,4,0,7,10,0,0,0,4,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

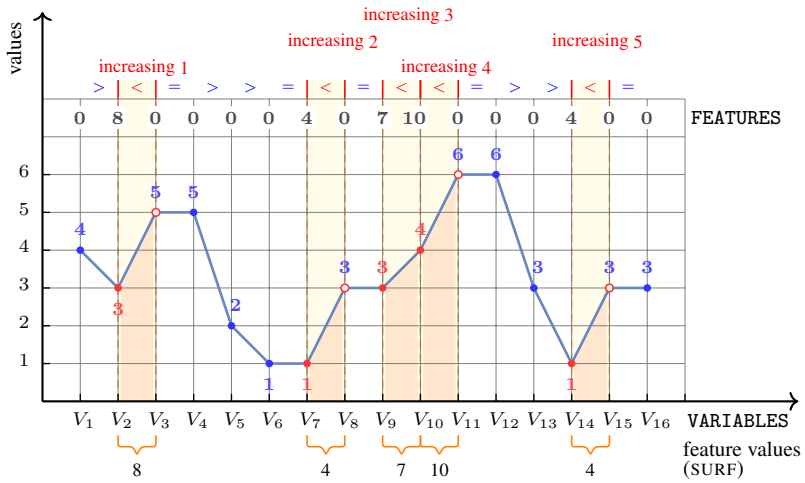
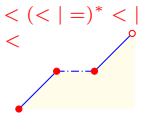


Figure 4.1501: Illustrating the SURF_INCREASING constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_INCREASING_SEQUENCE



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint SURF_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_SEQUENCE.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example Figure 4.1502 provides an example where the SURF_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 8, 0, 0, 0, 0, 17, 0, 0, 0, 0, 0, 0, 4, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

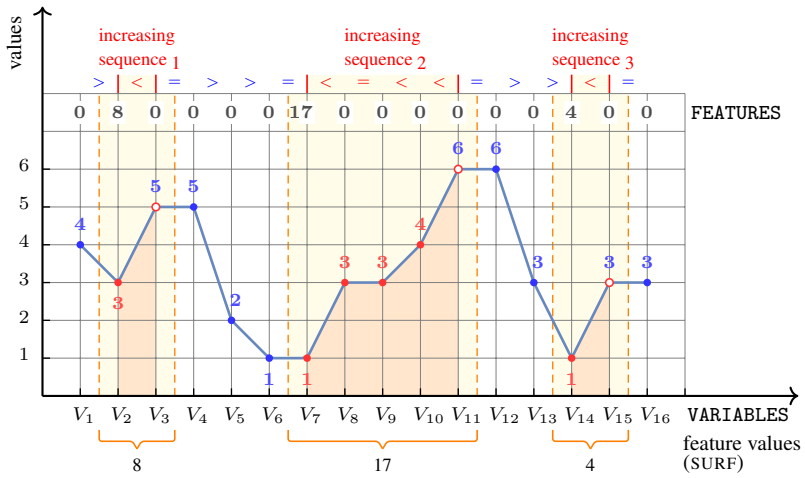


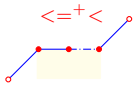
Figure 4.1502: Illustrating the SURF_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_INCREASING_TERRACE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint SURF_INCREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_TERRACE.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1503 provides an example where the SURF_INCREASING_TERRACE $([1,3,3,3,2,5,5,6,4,4,2,3,3,3,4,4],[0,0,0,0,0,0,0,10,0,0,0,0,0,0,0,9,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

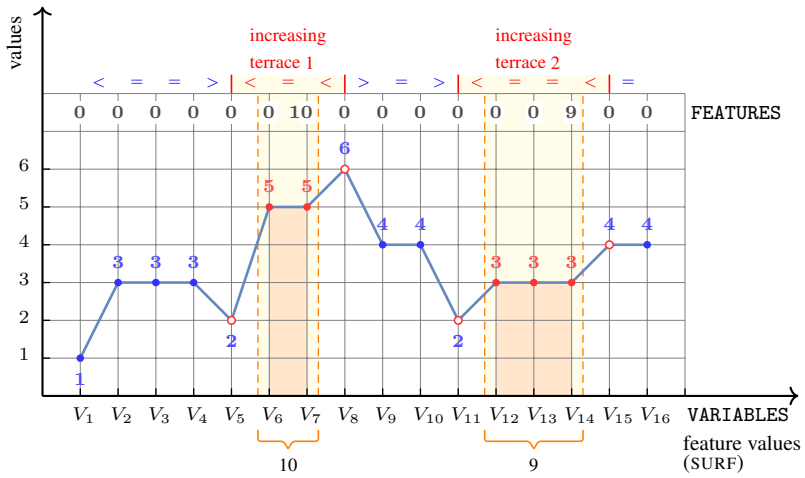


Figure 4.1503: Illustrating the SURF_INCREASING_TERRACE constraint of the **Example** slot

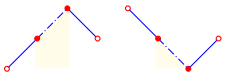
Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_INFLEXION

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

< (< | =)* > | > (> | =)* <



Origin Based on the INFLEXION pattern.

Constraint SURF_INFLEXION(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INFLEXION is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INFLEXION.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example Figure 4.1504 provides an example where the SURF_INFLEXION $([1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4], [0, 0, 0, 14, 0, 0, 11, 5, 2, 5, 1, 5, 0, 6, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

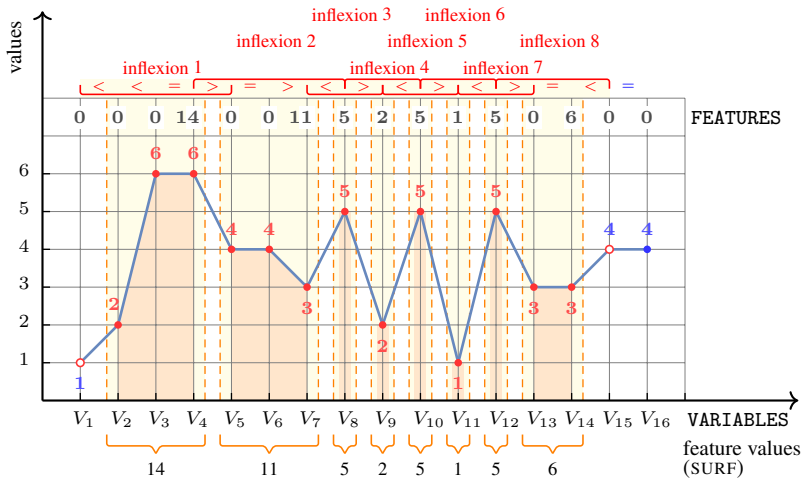


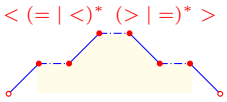
Figure 4.1504: Illustrating the SURF_INFLEXION constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_PEAK

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the PEAK pattern.

Constraint SURF_PEAK(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PEAK is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PEAK.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1505 provides an example where the SURF_PEAK $([7,5,5,1,4,5,2,2,3,5,6,2,3,3,3,1],[0,0,0,0,0,0,9,0,0,0,0,14,0,0,0,9,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

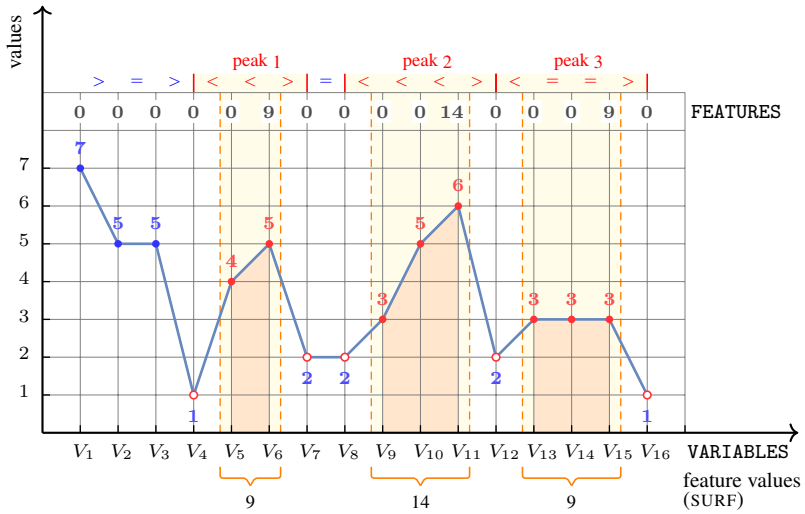


Figure 4.1505: Illustrating the SURF_PEAK constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_PLAIN

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the PLAIN pattern.

Constraint SURF_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLAIN.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^*<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1506 provides an example where the SURF_PLAIN $([2,3,6,5,7,6,6,4,5,5,4,3,3,6,6,3],[0,0,0,5,0,0,0,4,0,0,0,0,6,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

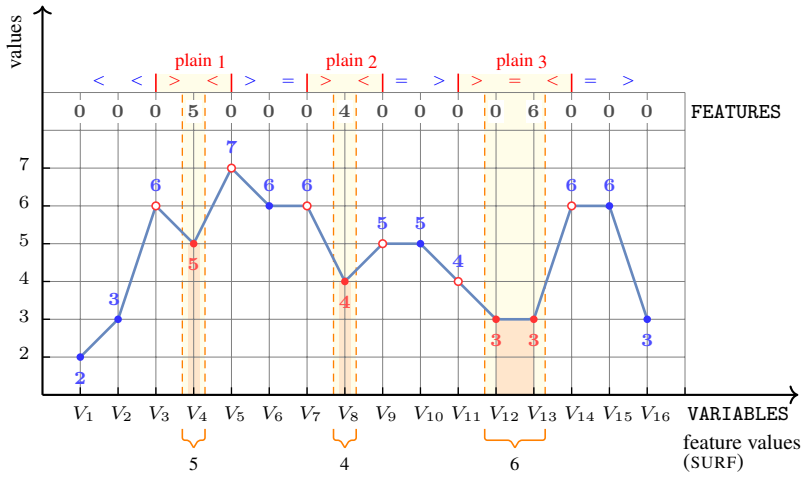


Figure 4.1506: Illustrating the SURF_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint SURF_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLATEAU.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^*>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1507 provides an example where the SURF_PLATEAU $([7,5,2,3,1,2,2,4,3,3,4,5,5,2,2,5],[0,0,0,3,0,0,0,4,0,0,0,0,10,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

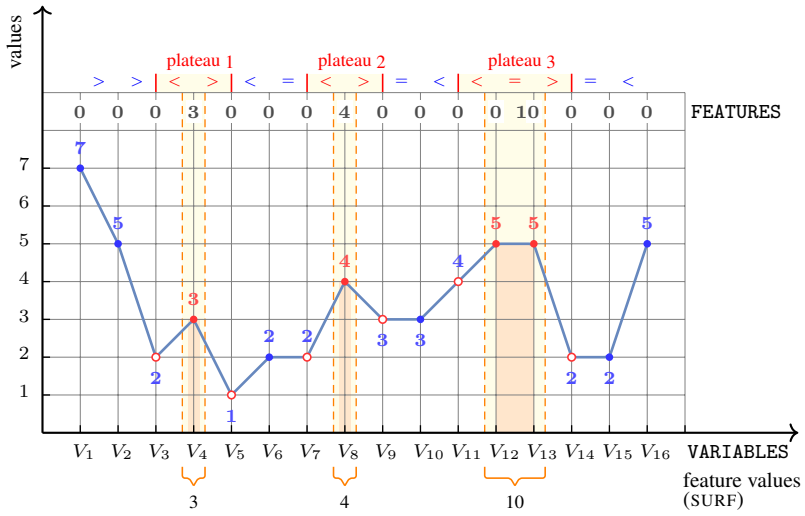


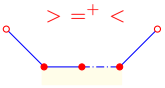
Figure 4.1507: Illustrating the SURF_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_PROPER_PLAIN

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint SURF_PROPER_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLAIN.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1508 provides an example where the SURF_PROPER_PLAIN $([2,7,5,5,6,3,7,4,4,5,6,5,3,3,3,5],[0,0,0,10,0,0,0,0,0,8,0,0,0,0,0,0,9,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

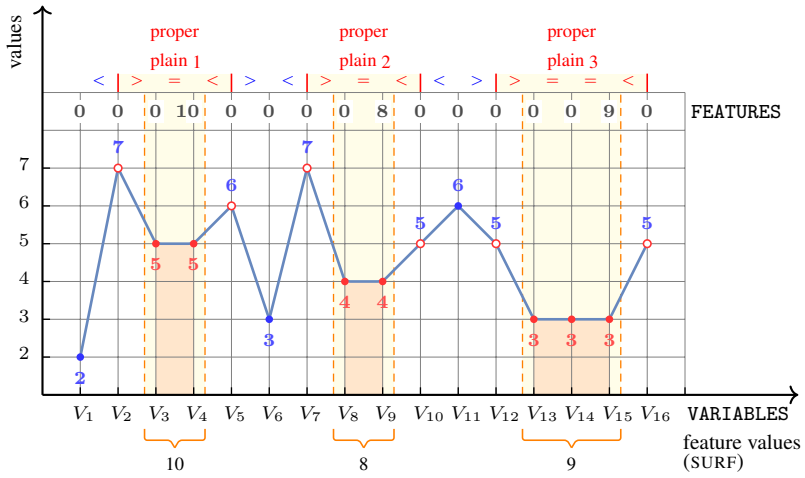


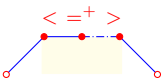
Figure 4.1508: Illustrating the SURF_PROPER_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_PROPER_PLATEAU

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint SURF_PROPER_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLATEAU.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1509 provides an example where the SURF_PROPER_PLATEAU $([7,1,3,3,2,5,1,4,4,3,2,3,5,5,5,3],[0,0,0,6,0,0,0,0,0,8,0,0,0,0,0,15,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

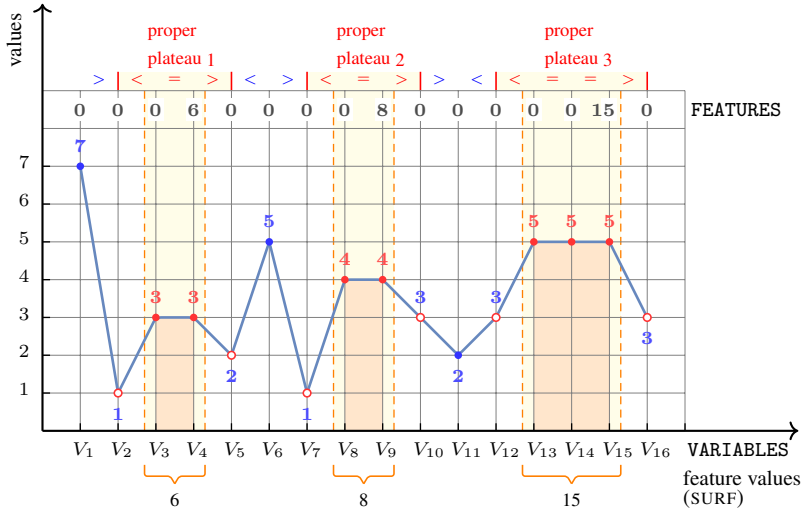


Figure 4.1509: Illustrating the SURF_PROPER_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_STEADY

▶ ▷ ◁ ◀ **DESCRIPTION** **AUTOMATON**

Origin Based on the STEADY pattern.

Constraint SURF_STEADY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY.
An occurrence of the pattern STEADY is the subsequence which matches the regular expression '='.
Assume that the occurrence of the pattern STEADY starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example Figure 4.1510 provides an example where the SURF_STEADY $([1, 1, 7, 3, 3, 5, 5, 5, 6, 5, 5, 5, 7, 2, 6, 6], [2, 0, 0, 6, 0, 10, 10, 0, 0, 10, 10, 0, 0, 0, 12, 0], 0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

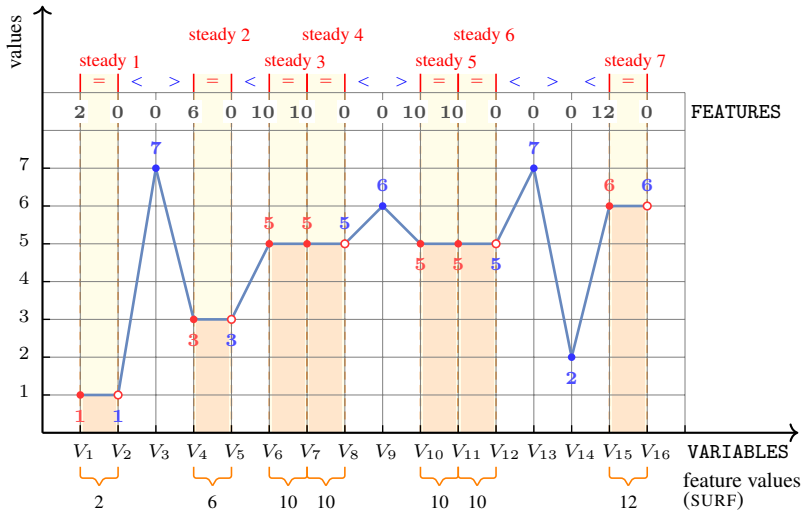


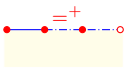
Figure 4.1510: Illustrating the SURF_STEADY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_STEADY_SEQUENCE

▶ ▷ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint SURF_STEADY_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY_SEQUENCE.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example Figure 4.1511 provides an example where the SURF_STEADY_SEQUENCE $([3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1], [0, 2, 0, 0, 15, 0, 0, 0, 4, 0, 8, 0, 0, 0, 2, 0], 0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

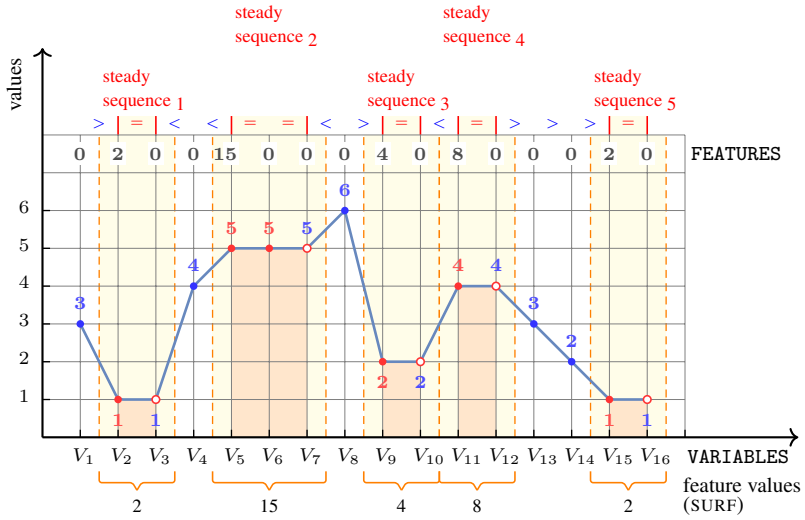


Figure 4.1511: Illustrating the SURF_STEADY_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_STRICTLY_DECREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint SURF_STRICTLY_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_DECREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j + 1$.

Example Figure 4.1512 provides an example where the SURF_STRICTLY_DECREASING_SEQUENCE $([4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3], [0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 13, 0, 0, 0, 7, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

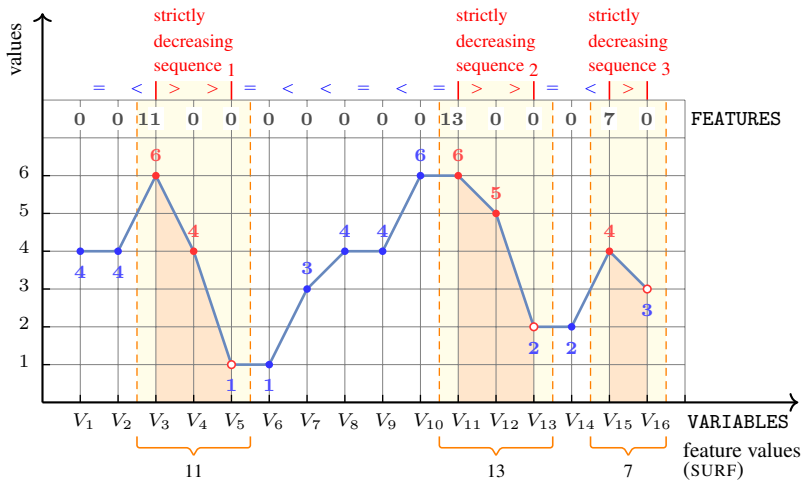


Figure 4.1512: Illustrating the SURF_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_STRICTLY_INCREASING_SEQUENCE

► ▷ ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint SURF_STRICTLY_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_INCREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature SURF computes the sum of the values from index i to index $j+1$.

Example Figure 4.1513 provides an example where the SURF_STRICTLY_INCREASING_SEQUENCE $([4,3,5,5,2,1,1,2,3,4,6,6,3,1,2,3],[0,8,0,0,0,0,16,0,0,0,0,0,0,6,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

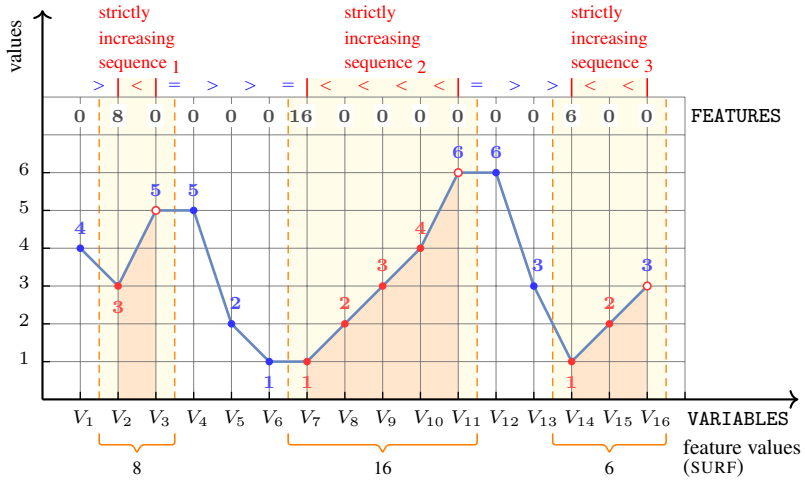


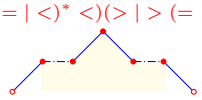
Figure 4.1513: Illustrating the SURF_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_SUMMIT

$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$



► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**

Origin Based on the SUMMIT pattern.

Constraint SURF_SUMMIT(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of SUMMIT is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of SUMMIT.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i + 1$ to index j.

Example Figure 4.1514 provides an example where the SURF_SUMMIT $([7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1], [0, 0, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

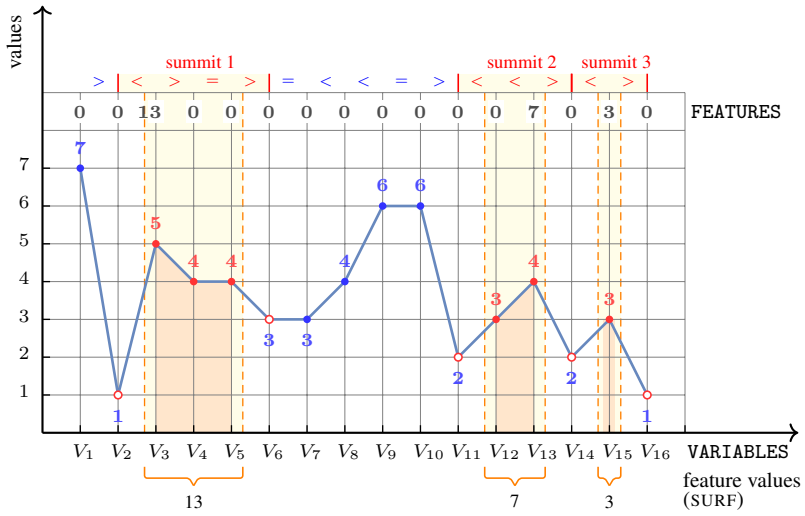


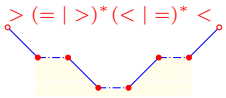
Figure 4.1514: Illustrating the SURF_SUMMIT constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_VALLEY

► ▷ ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint SURF_VALLEY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of VALLEY is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of VALLEY.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1515 provides an example where the SURF_VALLEY $([1,3,7,4,3,6,6,5,3,3,2,6,5,5,5,7],[0,0,0,0,7,0,0,0,0,0,0,13,0,0,0,15,0],0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

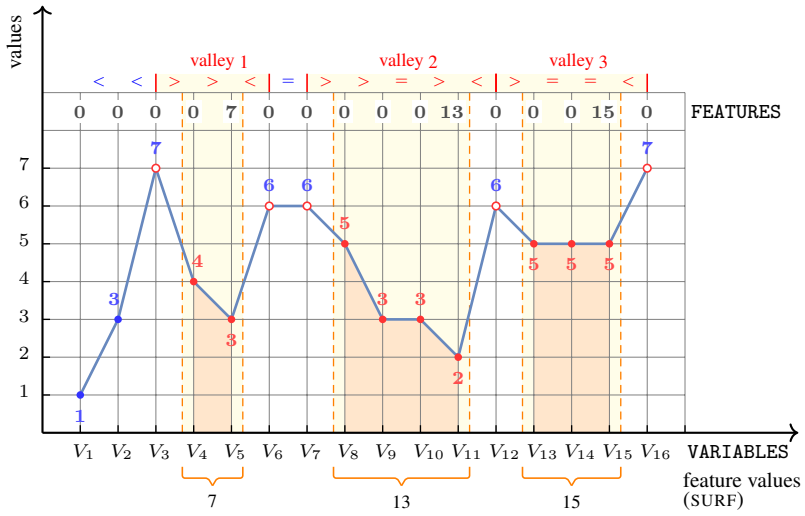


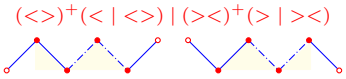
Figure 4.1515: Illustrating the SURF_VALLEY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

SURF_ZIGZAG

► ▷ ◁ ◄ **DESCRIPTION** **AUTOMATON**



Origin Based on the ZIGZAG pattern.

Constraint SURF_ZIGZAG(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
|VARIABLES| = |FEATURES|

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of ZIGZAG is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of ZIGZAG.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature SURF computes the sum of the values from index $i+1$ to index j.

Example Figure 4.1516 provides an example where the SURF_ZIGZAG $([4,1,3,1,4,6,1,5,5,2,7,2,3,1,6,1],[0,0,5,0,0,0,7,0,0,0,21,0,0,0,0,0],0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

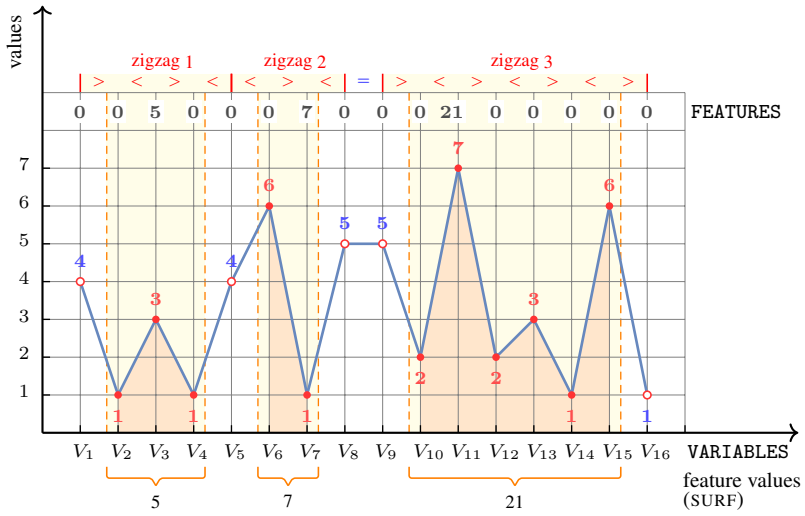
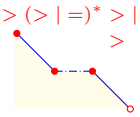


Figure 4.1516: Illustrating the SURF_ZIGZAG constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_DECREASING_SEQUENCE



► ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the DECREASING_SEQUENCE pattern.

Constraint WIDTH_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
rv = 2 ⇒ FEATURES.var ≤ 2
rv ≥ 3 ⇒ FEATURES.var ≤ sv
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_SEQUENCE.
An occurrence of the pattern DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$> (> \mid =)^* > \mid >$'.
Assume that the occurrence of the pattern DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example Figure 4.1517 provides an example where the WIDTH_DECREASING_SEQUENCE $([3, 4, 2, 2, 5, 6, 6, 4, 4, 3, 1, 1, 4, 6, 4, 4], [0, 2, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 2, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

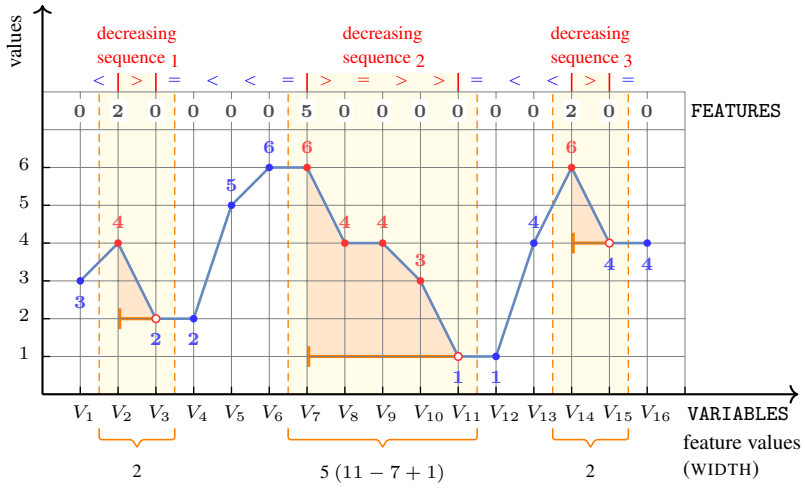


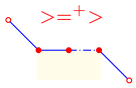
Figure 4.1517: Illustrating the WIDTH_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_DECREASING_TERRACE

► ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the DECREASING_TERRACE pattern.

Constraint WIDTH_DECREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 2 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of DECREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of DECREASING_TERRACE.
An occurrence of the pattern DECREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$>=^{+}>$'.
Assume that the occurrence of the pattern DECREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1518 provides an example where the WIDTH_DECREASING_TERRACE $([6, 4, 4, 4, 5, 2, 2, 1, 3, 3, 5, 4, 4, 3, 3, 3], [0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

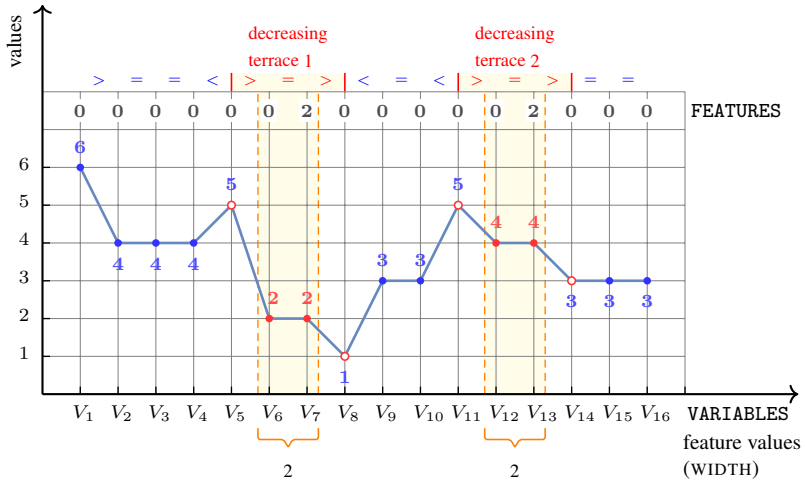


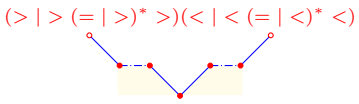
Figure 4.1518: Illustrating the WIDTH_DECREASING_TERRACE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_GORGE

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the GORGE pattern.

Constraint WIDTH_GORGE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
FEATURES.var ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
rv = 2 ⇒ FEATURES.var ≤ 1
rv ≥ 3 ⇒ FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of GORGE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of GORGE.
An occurrence of the pattern GORGE is the *maximal* subsequence which matches the regular expression '$(> \mid > (= \mid >)^* >)(< \mid < (= \mid <)^* <)$'.
Assume that the occurrence of the pattern GORGE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1519 provides an example where the WIDTH_GORGE $([1, 7, 3, 4, 4, 5, 5, 4, 2, 2, 6, 5, 4, 6, 5, 7], [0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

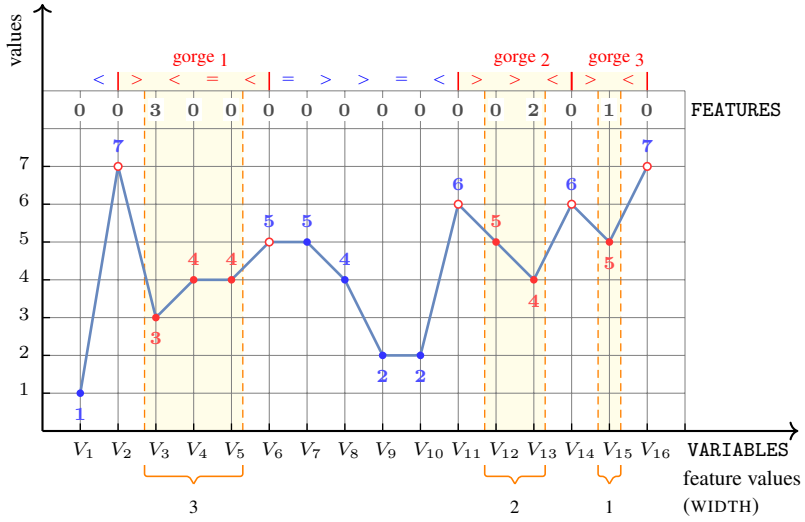
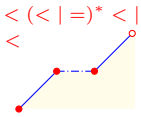


Figure 4.1519: Illustrating the WIDTH_GORGE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_INCREASING_SEQUENCE



► ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the INCREASING_SEQUENCE pattern.

Constraint WIDTH_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
rv = 2 ⇒ FEATURES.var ≤ 2
rv ≥ 3 ⇒ FEATURES.var ≤ sv
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_SEQUENCE.
An occurrence of the pattern INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* < \mid <$'.
Assume that the occurrence of the pattern INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example Figure 4.1520 provides an example where the WIDTH_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 3, 3, 4, 6, 6, 3, 1, 3, 3], [0, 2, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 2, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

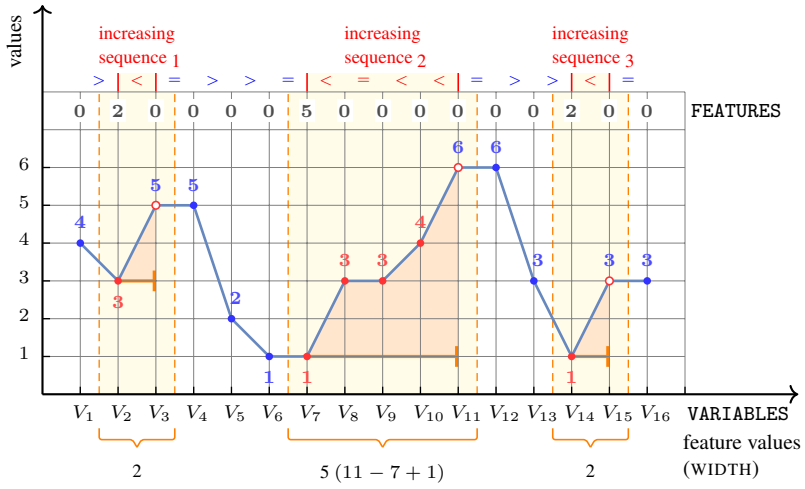


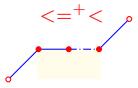
Figure 4.1520: Illustrating the WIDTH_INCREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_INCREASING_TERRACE

► ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the INCREASING_TERRACE pattern.

Constraint WIDTH_INCREASING_TERRACE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 2 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INCREASING_TERRACE is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INCREASING_TERRACE.
An occurrence of the pattern INCREASING_TERRACE is the *maximal* subsequence which matches the regular expression '$<=^{+}<$'.
Assume that the occurrence of the pattern INCREASING_TERRACE starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1521 provides an example where the WIDTH_INCREASING_TERRACE $([1, 3, 3, 3, 2, 5, 5, 6, 4, 4, 2, 3, 3, 3, 4, 4], [0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 2

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

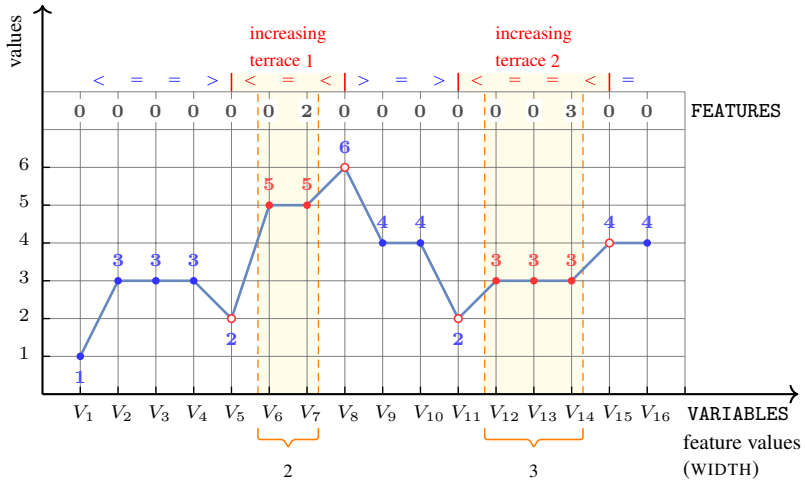


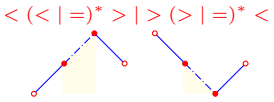
Figure 4.1521: Illustrating the WIDTH_INCREASING_TERRACE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_INFLEXION

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the INFLEXION pattern.

Constraint WIDTH_INFLEXION(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of INFLEXION is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of INFLEXION.
An occurrence of the pattern INFLEXION is the *maximal* subsequence which matches the regular expression '$< (< \mid =)^* > \mid > (> \mid =)^* <$'.
Assume that the occurrence of the pattern INFLEXION starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1522 provides an example where the WIDTH_INFLEXION $([1, 2, 6, 6, 4, 4, 3, 5, 2, 5, 1, 5, 3, 3, 4, 4], [0, 0, 0, 3, 0, 0, 3, 1, 1, 1, 1, 1, 0, 2, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

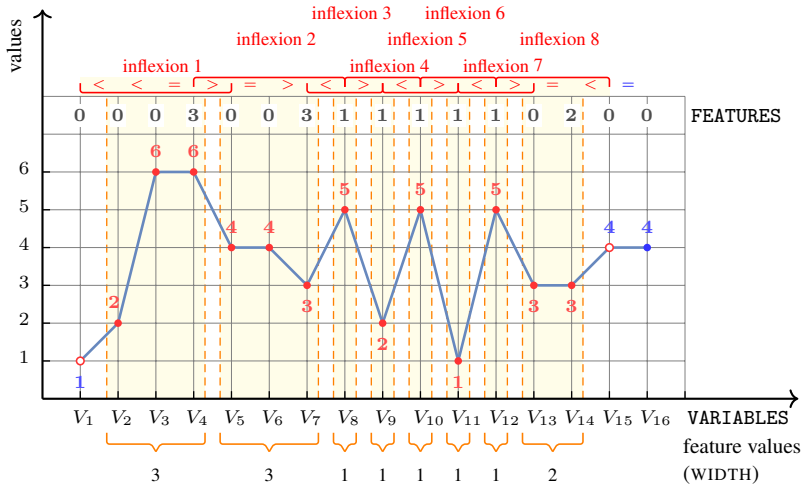


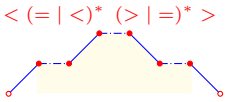
Figure 4.1522: Illustrating the WIDTH_INFLEXION constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_PEAK

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the PEAK pattern.

Constraint WIDTH_PEAK(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PEAK is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PEAK.
An occurrence of the pattern PEAK is the *maximal* subsequence which matches the regular expression '$< (= \mid <)^* (> \mid =)^* >$'.
Assume that the occurrence of the pattern PEAK starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1523 provides an example where the WIDTH_PEAK $([7, 5, 5, 1, 4, 5, 2, 2, 3, 5, 6, 2, 3, 3, 3, 1], [0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

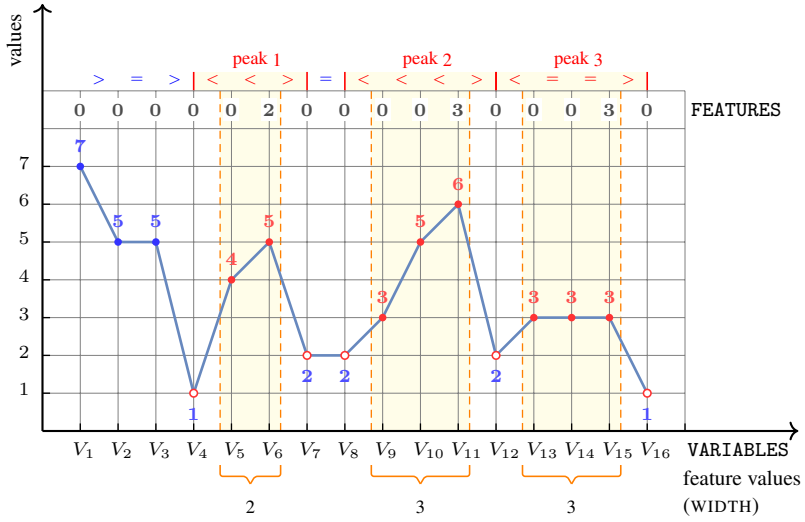


Figure 4.1523: Illustrating the WIDTH_PEAK constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_PLAIN

► ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the PLAIN pattern.

Constraint WIDTH_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLAIN.
An occurrence of the pattern PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{*}<$'.
Assume that the occurrence of the pattern PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1524 provides an example where the WIDTH_PLAIN $([2, 3, 6, 5, 7, 6, 6, 4, 5, 5, 4, 3, 3, 6, 6, 3], [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

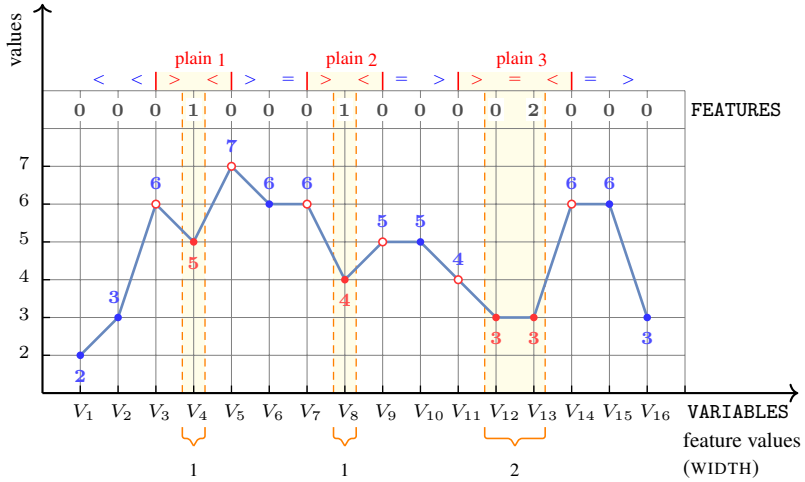


Figure 4.1524: Illustrating the WIDTH_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_PLATEAU

▶ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the PLATEAU pattern.

Constraint WIDTH_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PLATEAU.
An occurrence of the pattern PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{*}>$'.
Assume that the occurrence of the pattern PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1525 provides an example where the WIDTH_PLATEAU $([1, 3, 3, 5, 5, 5, 5, 2, 4, 4, 4, 3, 3, 1, 5, 5], [0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

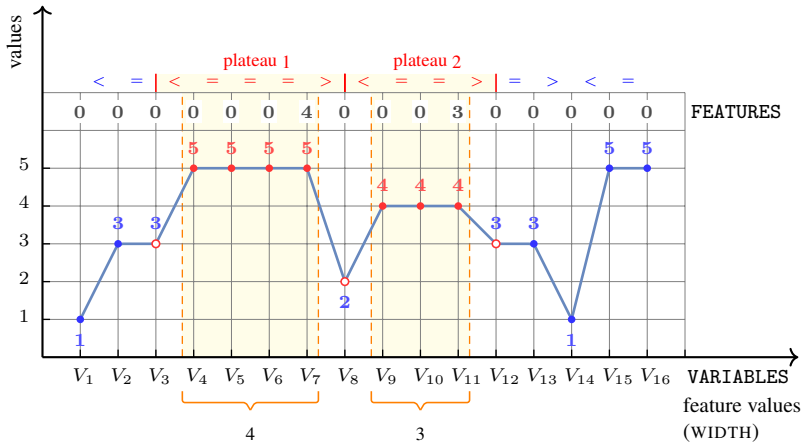


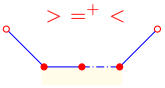
Figure 4.1525: Illustrating the WIDTH_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_PROPER_PLAIN

► ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLAIN pattern.

Constraint WIDTH_PROPER_PLAIN(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLAIN is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLAIN.
An occurrence of the pattern PROPER_PLAIN is the *maximal* subsequence which matches the regular expression '$>=^{+}<$'.
Assume that the occurrence of the pattern PROPER_PLAIN starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1526 provides an example where the WIDTH_PROPER_PLAIN $([2, 7, 5, 5, 6, 3, 7, 4, 4, 5, 6, 5, 3, 3, 3, 5], [0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

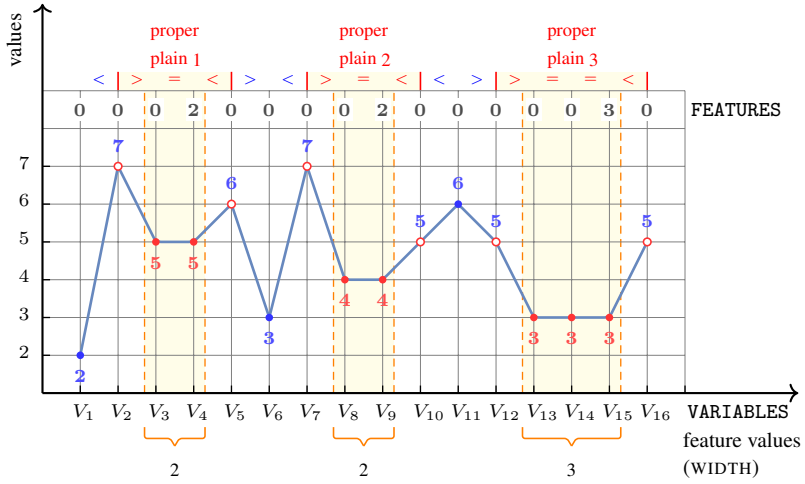


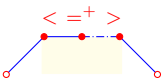
Figure 4.1526: Illustrating the WIDTH_PROPER_PLAIN constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_PROPER_PLATEAU

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the PROPER_PLATEAU pattern.

Constraint WIDTH_PROPER_PLATEAU(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0

where

sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of PROPER_PLATEAU is identified then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of PROPER_PLATEAU.
An occurrence of the pattern PROPER_PLATEAU is the *maximal* subsequence which matches the regular expression '$<=^{+}>$'.
Assume that the occurrence of the pattern PROPER_PLATEAU starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1527 provides an example where the WIDTH_PROPER_PLATEAU $([7, 1, 3, 3, 2, 5, 1, 4, 4, 3, 2, 3, 5, 5, 5, 3], [0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

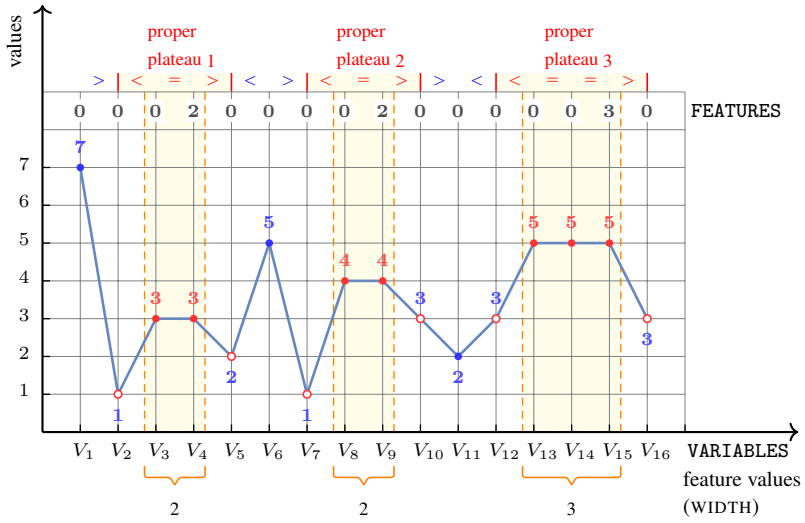


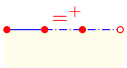
Figure 4.1527: Illustrating the WIDTH_PROPER_PLATEAU constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_STEADY_SEQUENCE

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the STEADY_SEQUENCE pattern.

Constraint WIDTH_STEADY_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
FEATURES.var ≥ 0
FEATURES.var ≤ sv
sv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
rv = 1 ∧ sv ≥ 2 ⇒maxval(FEATURES.var) = sv
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STEADY_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STEADY_SEQUENCE.
An occurrence of the pattern STEADY_SEQUENCE is the *maximal* subsequence which matches the regular expression '$=^+$'.
Assume that the occurrence of the pattern STEADY_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example Figure 4.1528 provides an example where the WIDTH_STEADY_SEQUENCE $([3, 1, 1, 4, 5, 5, 5, 6, 2, 2, 4, 4, 3, 2, 1, 1], [0, 2, 0, 0, 3, 0, 0, 0, 2, 0, 2, 0, 0, 0, 2, 0], 0)$ constraint holds.

Typical |VARIABLES| > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

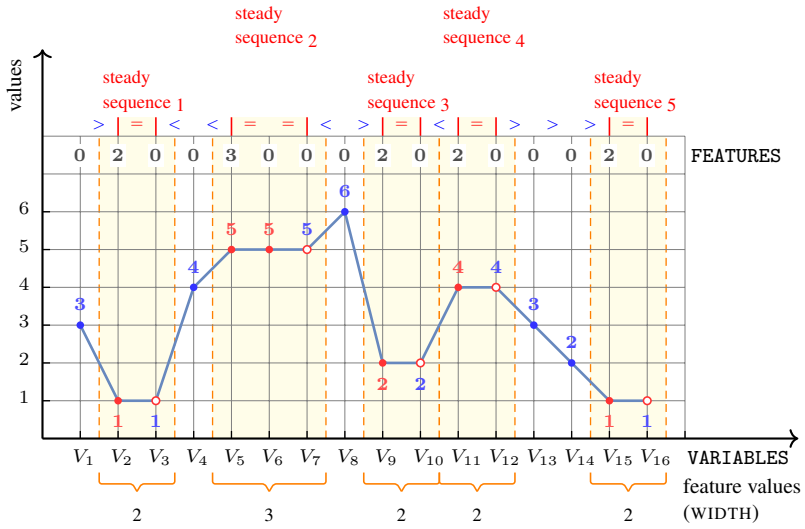


Figure 4.1528: Illustrating the WIDTH_STEADY_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_STRICTLY_DECREASING_SEQUENCE

▶ ◁ ◀ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_DECREASING_SEQUENCE pattern.

Constraint WIDTH_STRICTLY_DECREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ min(sv, rv)
DEFAULT = 0

where

sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_DECREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_DECREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_DECREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$>^+$'.
Assume that the occurrence of the pattern STRICTLY_DECREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example Figure 4.1529 provides an example where the WIDTH_STRICTLY_DECREASING_SEQUENCE $([4, 4, 6, 4, 1, 1, 3, 4, 4, 6, 6, 5, 2, 2, 4, 3], [0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

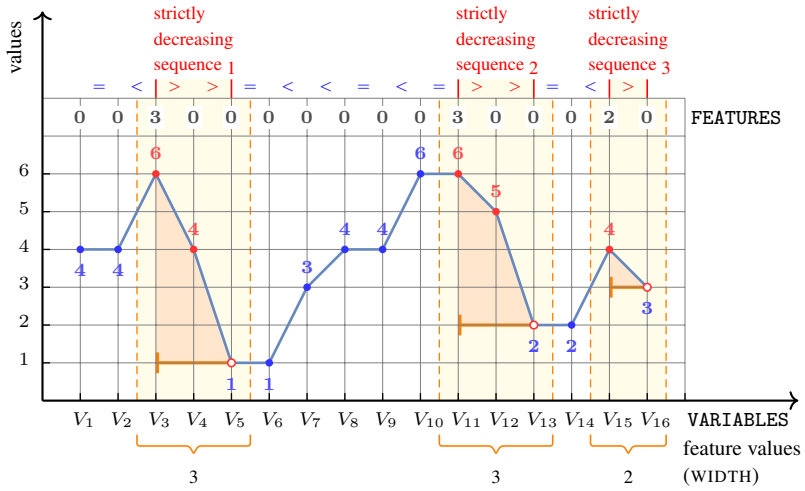


Figure 4.1529: Illustrating the WIDTH_STRICTLY_DECREASING_SEQUENCE constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_STRICTLY_INCREASING_SEQUENCE

► ◁ ◄ DESCRIPTION AUTOMATON

Origin Based on the STRICTLY_INCREASING_SEQUENCE pattern.

Constraint WIDTH_STRICTLY_INCREASING_SEQUENCE(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 1 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ min(sv, rv)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of STRICTLY_INCREASING_SEQUENCE is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of STRICTLY_INCREASING_SEQUENCE.
An occurrence of the pattern STRICTLY_INCREASING_SEQUENCE is the *maximal* subsequence which matches the regular expression '$<^+$'.
Assume that the occurrence of the pattern STRICTLY_INCREASING_SEQUENCE starts at position i and ends at position j. The feature WIDTH computes the value $j - i + 2$.

Example Figure 4.1530 provides an example where the WIDTH_STRICTLY_INCREASING_SEQUENCE $([4, 3, 5, 5, 2, 1, 1, 2, 3, 4, 6, 6, 3, 1, 2, 3], [0, 2, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 3, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 1
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

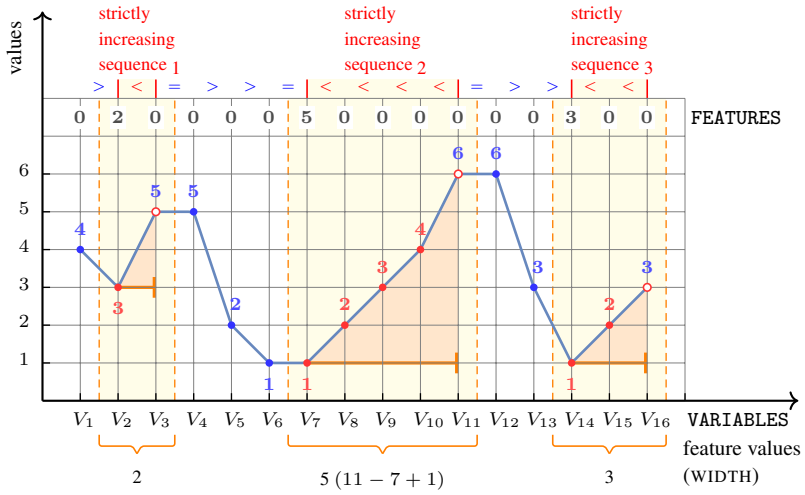


Figure 4.1530: Illustrating the WIDTH_STRICTLY_INCREASING_SEQUENCE constraint of the **Example** slot

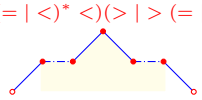
Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_SUMMIT

(< | < (= | <)* <)(> | > (= | >)* >)

▶ ◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the SUMMIT pattern.

Constraint WIDTH_SUMMIT(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
rv = 2 ⇒ FEATURES.var ≤ 1
rv ≥ 3 ⇒ FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of SUMMIT is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of SUMMIT.
An occurrence of the pattern SUMMIT is the *maximal* subsequence which matches the regular expression '$(< \mid < (= \mid <)^* <)(> \mid > (= \mid >)^* >)$'.
Assume that the occurrence of the pattern SUMMIT starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1531 provides an example where the WIDTH_SUMMIT $([7, 1, 5, 4, 4, 3, 3, 4, 6, 6, 2, 3, 4, 2, 3, 1], [0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

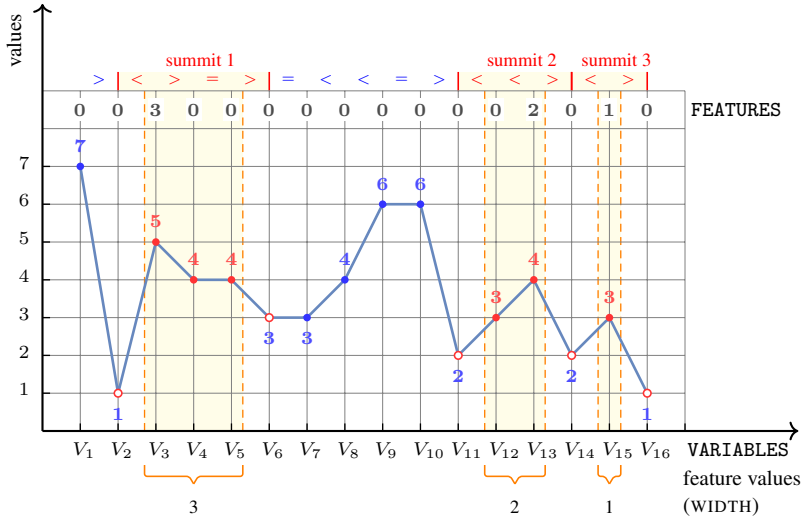


Figure 4.1531: Illustrating the WIDTH_SUMMIT constraint of the **Example** slot

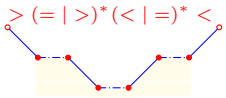
Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

WIDTH_SUMMIT 3193

FEATURE PATTERN

WIDTH_VALLEY

► ◁ ◄ DESCRIPTION AUTOMATON



Origin Based on the VALLEY pattern.

Constraint WIDTH_VALLEY(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 2 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var ≥ 0
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of VALLEY is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of VALLEY.
An occurrence of the pattern VALLEY is the *maximal* subsequence which matches the regular expression '$> (= \mid >)^* (< \mid =)^* <$'.
Assume that the occurrence of the pattern VALLEY starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1532 provides an example where the WIDTH_VALLEY $([1, 3, 7, 4, 3, 6, 6, 5, 3, 3, 2, 6, 5, 5, 5, 7], [0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 2
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

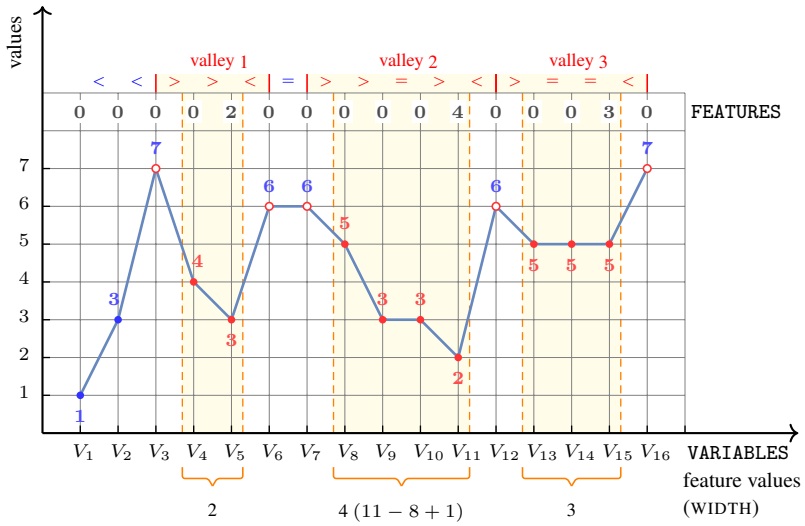


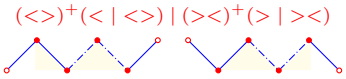
Figure 4.1532: Illustrating the WIDTH_VALLEY constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

FEATURE PATTERN

WIDTH_ZIGZAG

◁ ◀ DESCRIPTION AUTOMATON



Origin Based on the ZIGZAG pattern.

Constraint WIDTH_ZIGZAG(VARIABLES, FEATURES, DEFAULT)

Arguments

VARIABLES	:	collection(var−dvar)
FEATURES	:	collection(var−dvar)
DEFAULT	:	int

Restrictions

required(VARIABLES, var)
required(FEATURES, var)
sv = |FEATURES|
sv ≤ 3 ∨ rv ≤ 1 ⇒ FEATURES.var = 0
FEATURES.var = 0 ∨ FEATURES.var ≥ 2
FEATURES.var ≤ max(0, sv − 2)
DEFAULT = 0
where
sv = |VARIABLES|
rv =range(VARIABLES.var)

Purpose

Consider the time-series given by the VARIABLES collection. If position i does not correspond to the first position in VARIABLES where an occurrence of ZIGZAG is identified (even if this occurrence of pattern is not complete) then FEATURES$[i]$ is the default value DEFAULT; otherwise FEATURES$[i]$ gives the feature value of the corresponding occurrence of ZIGZAG.
An occurrence of the pattern ZIGZAG is the *maximal* subsequence which matches the regular expression '$(<>)^+(< \mid <>) \mid (><)^+(> \mid ><)$'.
Assume that the occurrence of the pattern ZIGZAG starts at position i and ends at position j. The feature WIDTH computes the value $j - i$.

Example Figure 4.1533 provides an example where the WIDTH_ZIGZAG $([4, 1, 3, 1, 4, 6, 1, 5, 5, 2, 7, 2, 3, 1, 6, 1], [0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 6, 0, 0, 0, 0, 0], 0)$ constraint holds.

Typical

|VARIABLES| > 3
range(VARIABLES.var) > 1

Arg. properties Functional dependency: FEATURES determined by VARIABLES and DEFAULT.

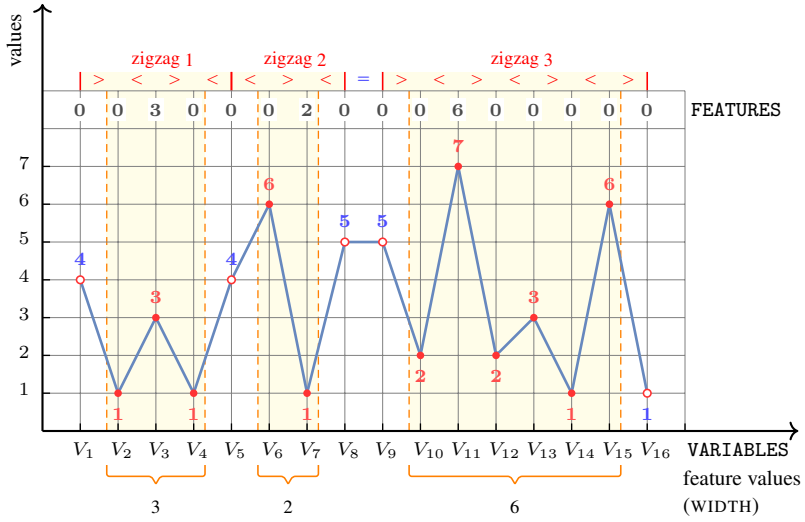


Figure 4.1533: Illustrating the WIDTH_ZIGZAG constraint of the **Example** slot

Automaton Use the decoration table 3.32 to synthesise the corresponding register automaton.

5

A Database of Parameterised Invariants

This chapter presents a database of parameterised invariants where the parameter is the sequence length. We currently have five types of invariants:

- For a single time-series constraint, linear invariants are obtained by applying the Farkas Lemma [13] to the automaton with registers that was synthesised [17] for a given time-series constraint. These invariants link consecutive register values. For those time-series constraints for which two register automata are available, we only generate invariants for the simplified automaton, i.e. the one that uses a specialised decoration table.

- For the conjunction of two time-series constraints on the same sequence of variables of length n, linear invariants of the form $e + e_0 n + e_1 \mathtt{x} + e_2 \mathtt{y} \geq 0$, with e, e_0, e_1, $e_2 \in \mathbb{Z}$, linking the result variables $\mathtt{x}$ and $\mathtt{y}$ of both time-series constraints are obtained by using the method described in [5]:

    1. Compute the intersection of the register automata corresponding to the two time-series constraints. For building this intersection we use, when available, the simplified versions of the register automata, but note that the resulting invariants can be used both with the non-simplified and the simplified register automata.
    2. Generate all possible combinations of coefficient signs of the invariants to generate, and for each combination:
        (a) Build from the intersection automaton an invariant-weighted digraph, where the weight of each arc represents the variation of the left-hand side $e + e_0 n + e_1 \mathtt{x} + e_2 \mathtt{y}$ of the desired linear invariant when the corresponding transition of the intersection automaton is triggered.
        (b) Construct an optimisation problem where the constraints prevent negative cycles in the invariant-weighted digraph, and where the optimisation criterion promotes the generation of sharp invariants. The optimal

solution to this optimisation problem are the values of the coefficients e_0, e_1, e_2.

(c) Solve a shortest-path problem from the initial state to the accepting states of an automaton derived from the intersection automaton and from the invariant-weighted digraph, giving the constant term e of the linear invariant.

3. Using a heuristic, filter out linear invariants that do not remove any infeasible points.

For the majority of linear invariants we prove their sharpness by providing two distinct points whose coordinates are possibly parameterised by the sequence length n, which both are located on the corresponding linear equality constraint and are feasible with respect to the conjunction of the two considered time-series constraints. Feasibility of a point is proved by showing that the intersection of two finite automata that accept only time series whose characteristics correspond to the coordinates of the point is not empty. Each of the two finite automata corresponds to the automata provided in the **Specialisation** slot of a time-series constraint.

- Applying the same technique used for generating linear invariants linking the result variables of a conjunction of two time-series constraints, we also generate conditional linear constraints that are valid provided the result variables are not assigned to their default value.

- For the conjunction of two time-series constraints on the same sequence of variables of length n, non-linear invariants corresponding to disjunction of linear constraints.

- For a single time-series constraint or for the conjunction of two time-series constraints on the same sequence of variables, we derive linear invariants involving the result variables `x` and `y`, as well as one of the three variables $\mathtt{o}^{<}$, $\mathtt{o}^{=}$, $\mathtt{o}^{>}$, which respectively denote the numbers of occurrences of signature variables that are assigned to '$<$', '$=$' and '$>$'. This last category of invariants is derived manually.

The database of parameterised invariants contains 917 entries: 176 entries corresponding to a single constraint and 741 entries corresponding to pairs of constraints. The total number of invariants is 2347: 759 linear invariants, 899 conditional linear invariants, 666 non-linear invariants, and 23 manually derived invariants.

There exist two complementary versions of this database of parameterised invariants, namely a machine-readable version where each invariant is represented as a Prolog fact, and a human-readable version where these Prolog facts are pretty-printed as LaTeX formulae. The machine-readable version is available from the electronic constraint catalogue given in Appendix A of this document as a stand-alone Prolog attached file, while the human-readable version is given in this chapter. Within the human-readable version, invariants are sorted by lexicographically increasing order on the tuples of lexicographically sorted constraint names. Figures that display the

invariants linking the result variables of two time-series constraints use the following conventions, where Figure 5.1 illustrates how the various types of invariants are depicted. For this illustrative purpose we use the following set of invariants, where after a linear invariant we possibly provide the coordinates of two distinct feasible points lying on the line corresponding to the linear invariant.

① $2\mathtt{x} \leq \mathtt{y}$
- $n \geq 9$: $(1, 2)$ $(2, 4)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 4\mathtt{x} + \mathtt{y} \leq 2n - 6$

③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq n - 1$
- $n \geq 11$: $(1, n-2)$ $(2, n-3)$

④ $\mathtt{y} \neq 2\mathtt{x} + 1$

⑤ $2\mathtt{x} + \mathtt{y} \leq 2n - 2$

⑥ $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \frac{n}{2} \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > n \cdot \min(1, \max(0, n-1)) - 3, \\ n \bmod 2 = 1, \\ \mathtt{y} \bmod 2 = 0 \end{array} \right)$$

⑦ $\mathtt{y} \neq 1$

⑧ $\mathtt{y} \leq 3\mathtt{x}$
- $n \geq 5$: $(0, 0)$ $(1, 3)$

⑨ $\mathtt{y} = \max(0, n-2) \wedge n > 1 \Rightarrow 2\mathtt{x} \geq n - 2 - n \bmod 2$

⑩ $n > 1 \Rightarrow 2\mathtt{x} + \mathtt{y} \leq 2n - 3$
- $n \geq 7$: $(\max(0, n-2) - 1, 3)$ $(\max(0, n-2) - 2, 5)$

⑪ $\mathtt{y} > 0 \Rightarrow \mathtt{x} \leq n - 3$
- $n \geq 5$: $(\max(0, n-2) - 1, 2)$ $(\max(0, n-2) - 1, 3)$

⑫ $\mathtt{y} \neq 1$

where:

- the invariants ①, ②, ③, and ④ are associated with the pair (NB_DIP_ON_INCREASING_SEQUENCE, SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE),
- the invariants ⑤, ⑥, and ⑦ are associated with the pair (NB_STEADY_SEQUENCE, SUM_WIDTH_DECREASING_SEQUENCE),
- the invariants ⑧, ⑨, ⑩, ⑪, and ⑫ are associated with the pair (SUM_WIDTH_SUMMIT, SUM_WIDTH_ZIGZAG).

In our figures, e.g. Figure 5.1, we use the following colour and shape scheme:

- A feasible pair of values of result variables is shown with a blue circle: light blue for a pair for which at least one of the values corresponds to the default value of one of the two time-series constraints, and blue otherwise. The convex hull of all feasible pairs of values is shown as an enclosing blue dotted line.

- A yellow circle denotes an infeasible point that is located outside the convex hull of feasible points and that is not discarded by any invariants; e.g. see the yellow point of coordinates $(3, 9)$ of Part (C) of Figure 5.1.
- A red circle denotes an infeasible point inside the convex hull that is not discarded by any non-linear invariants, but that may be discarded by some conditional linear invariants; e.g. see the red point of coordinates $(5, 8)$ for the pair (SUM_WIDTH_PLATEAU, SUM_WIDTH_ZIGZAG) and the sequence size 11.
- A diamond-shaped point denotes an infeasible point outside the convex hull, that belongs to a set of infeasible points discarded by some non-linear invariant. An illustrative example is the point of coordinates $(3, 7)$ in Part (A) of Figure 5.1, which is located outside the convex hull of feasible points and is associated with the disequality ④ $\mathtt{y} \neq 2 \cdot \mathtt{x} + 1$.
- A violet-coloured point denotes an infeasible point discarded by an invariant corresponding to a disequality between a result variable and a natural number. For instance, the violet points in Part (B) of Figure 5.1 correspond to the set of infeasible points associated with the disequality ⑦ $\mathtt{y} \neq 1$.
- An orange-coloured point denotes an infeasible point discarded by an invariant corresponding to a diagonal, horizontal line, or vertical line that starts at a specific point and extends to the border in one direction, or to a horizontal or vertical line that excludes points for which one of the coordinates is odd or even. For instance, the orange points in Part (A) of Figure 5.1 correspond to the set of infeasible points associated with ④ $\mathtt{y} \neq 2 \cdot \mathtt{x} + 1$.
- A brown-coloured point denotes an infeasible point discarded by a disjunction. For instance, the brown points in Part (B) of Figure 5.1 correspond to the set of infeasible points associated with ⑥.
- A cyan-coloured (respectively pink-coloured) half-space corresponds to infeasible points that are forbidden by linear inequalities (respectively by conditional linear inequalities). In Part (A) of Figure 5.1, the set of points that are forbidden by the linear inequality ① $2 \cdot \mathtt{x} \leq \mathtt{y}$ is coloured in cyan, while the set of points that are discarded by the conditional linear inequality ③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq n - 1$ is coloured in pink.
- A thick light violet line represents a set of infeasible points that are discarded by an implication of the form $\mathtt{x} = v \Rightarrow \ldots$ (or $\mathtt{y} = v \Rightarrow \ldots$), where v is a natural number. For instance, the thick light violet line in Part (C) of Figure 5.1 corresponds to the set of infeasible points associated with ⑨.
- For each linear invariant for which we prove sharpness, we provide two distinct points that are both feasible and are located on the line corresponding to the linear invariant. Note that the coordinates of such points may be parameterised by the sequence length n. For instance, given the conjunction NB_DIP_ON_INCREASING_SEQUENCE$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$ SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE$(\mathtt{y}, \mathtt{VARIABLES})$, we give

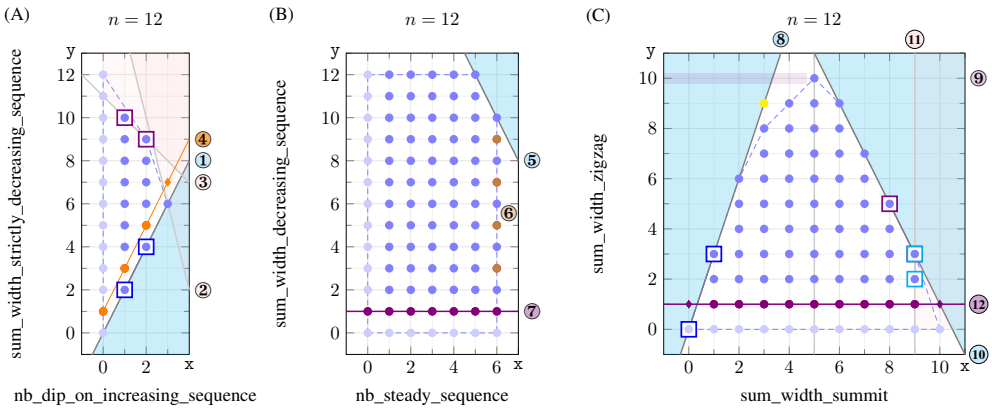


Figure 5.1: Illustration of how the different types of invariants are depicted

for the sharp invariant ③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq n - 1$ (sharp when $n \geq 11$) the two points $(1, n-2)$ and $(2, n-3)$, which (1) satisfy the equality $\mathtt{x} + \mathtt{y} = n - 1$ and (2) are feasible wrt the mentioned conjunction of time-series constraints, provided $n \geq 11$. Such points are graphically depicted by a coloured squared box; e.g. see the points of coordinates $(1, 10)$ and $(2, 9)$ in Part (A) of Figure 5.1, where $n = 12$.

MAX_MAX_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MAX_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_MIN_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_RANGE_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{var_{i-2}} - \mathtt{var_{i-1}}$

MAX_RANGE_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_RANGE_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{var_{i-1}} - \mathtt{var_{i-2}}$

MAX_RANGE_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_RANGE_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_RANGE_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_STEADY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_SURF_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$

MAX_WIDTH_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{d_i} \leq \mathtt{d_{i-1}} + 2$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$

MAX_WIDTH_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{d_i} \leq \mathtt{d_{i-1}} + 2$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$

MAX_WIDTH_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_i}$

MAX_WIDTH_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$

MIN_MAX_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MAX_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_MIN_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_RANGE_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_BUMP_ON_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_DIP_ON_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_STEADY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_SURF_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

MIN_WIDTH_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}}$

NB_DECREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{>}} = \mathtt{r_i}$

NB_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{>}} \geq \mathtt{r_i}$

NB_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{>}} \geq \mathtt{r_i} + \mathtt{r_i} > 0$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{=}} \geq \mathtt{r_i}$

NB_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{<}} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{>}} \geq \mathtt{r_i}$

NB_INCREASING

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^{<}} = \mathtt{r_i}$

NB_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$

NB_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i} + \mathtt{r_i} > 0$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i}$

NB_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 1$

NB_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [4, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-3}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i}$

NB_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [4, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-3}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i}$

NB_STEADY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} = \mathtt{r_i}$

NB_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i}$

NB_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$

NB_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-2}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{r_i}$

NB_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [4, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-3}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{r_i}$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^=} \leq |\mathtt{VARIABLES}| - \mathtt{i} - 3 * \mathtt{r_i}$

SUM_RANGE_DECREASING

$\forall \mathtt{i} \in [3, |\mathtt{VARIABLES}|] : \mathtt{r_i} - \mathtt{r_{i-2}} \geq \mathtt{var_i} - \mathtt{var_{i-2}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} - \mathtt{r_{i-1}} \geq \mathtt{var_i} - \mathtt{var_{i-1}}$

SUM_WIDTH_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{min}(2, \mathtt{r_i} - 1)$

SUM_WIDTH_DECREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{min}(2, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^=} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$

SUM_WIDTH_GORGE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$

SUM_WIDTH_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{min}(2, \mathtt{r_i} - 1)$

SUM_WIDTH_INCREASING_TERRACE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{min}(2, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^=} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$

SUM_WIDTH_INFLEXION

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$

SUM_WIDTH_PEAK

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$

SUM_WIDTH_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i} - \lfloor \mathtt{max}(0, |\mathtt{VARIABLES}| - \mathtt{i})/2 \rfloor$

SUM_WIDTH_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \mathtt{r_i} - \lfloor \mathtt{max}(0, |\mathtt{VARIABLES}| - \mathtt{i})/2 \rfloor$

SUM_WIDTH_PROPER_PLAIN

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$

SUM_WIDTH_PROPER_PLATEAU

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] \ : \ \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] \ : \ \mathtt{o_i^=} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$

SUM_WIDTH_STEADY_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$
$\mathtt{o_i^=} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$
$\mathtt{o_i^=} \leq \mathtt{max}(0, \mathtt{r_i} - 1)$

SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \leq \mathtt{max}(0, \mathtt{r_i} - 1)$

SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \leq \mathtt{max}(0, \mathtt{r_i} - 1)$

SUM_WIDTH_SUMMIT

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$

SUM_WIDTH_VALLEY

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + \mathtt{d_{i-1}} + 1$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \mathtt{min}(1, \mathtt{r_i})$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \mathtt{min}(1, \mathtt{r_i})$

SUM_WIDTH_ZIGZAG

$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \geq \mathtt{r_{i-1}}$
$\forall \mathtt{i} \in [2, |\mathtt{VARIABLES}|] : \mathtt{r_i} \leq \mathtt{r_{i-1}} + 2$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^>} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$
$\forall \mathtt{i} \in [1, |\mathtt{VARIABLES}|] : \mathtt{o_i^<} \geq \lfloor (\mathtt{r_i} + 1)/2 \rfloor$

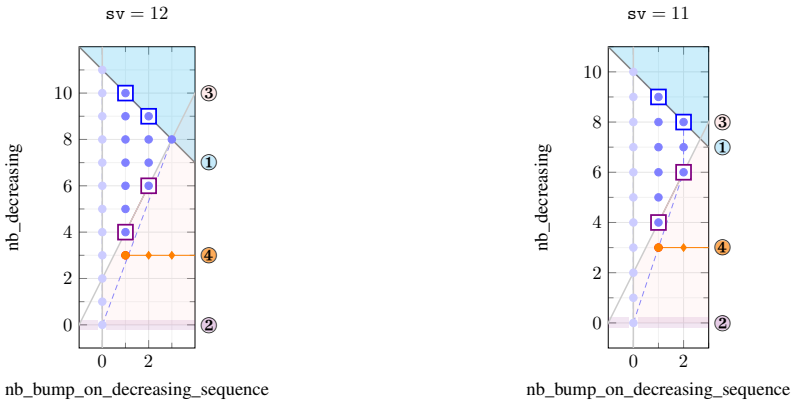
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DECREASING(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 9$: $(1, sv - 2)$ $(2, sv - 3)$

② $y = 0 \Rightarrow x = 0$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y - 2$
- $sv \geq 9$: $(1, 4)$ $(2, 6)$

④ $x < 1 \vee y \neq 3$

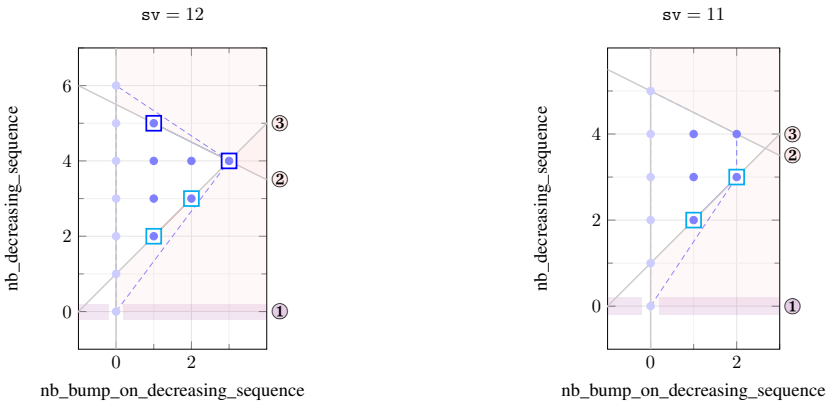


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor sv/2 \rfloor - 1)$ $(3, \lfloor sv/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 15$: $(2, \lfloor sv/2 \rfloor - 1)$ $(4, \lfloor sv/2 \rfloor - 2)$

③ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 9$: $(1, 2)$ $(2, 3)$



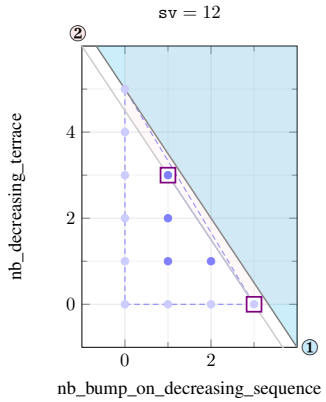
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

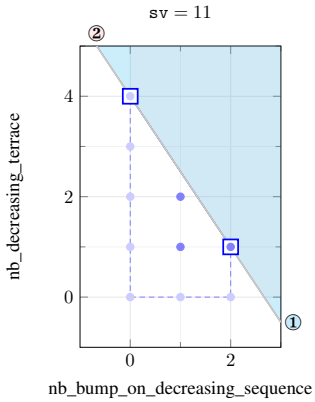
① $\mathtt{sv} > 1 \Rightarrow 3 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 2 - (\mathtt{sv} - 2) \bmod 2$

- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 9$: $(0, \lfloor(\mathtt{sv} - 2)/2\rfloor)$ $(2, \lfloor(\mathtt{sv} - 2)/2\rfloor - 3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 3 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 3$

- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 12$: $(1, \lfloor(\mathtt{sv} - 2)/2\rfloor - 2)$ $(3, \lfloor(\mathtt{sv} - 2)/2\rfloor - 5)$





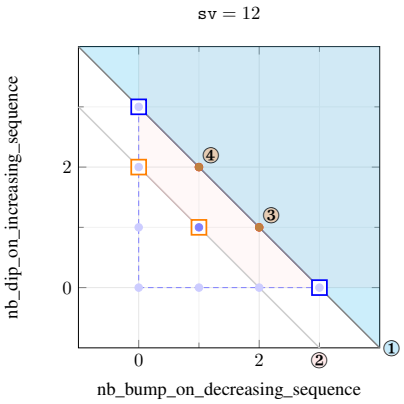
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DIP_ON_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

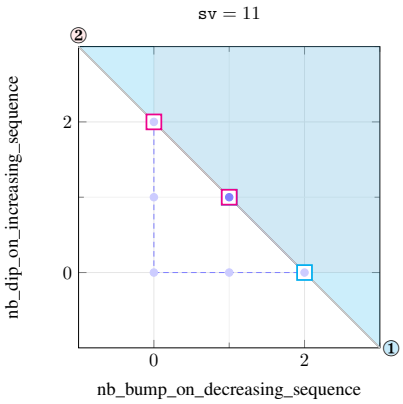
① $\mathtt{sv} > 2 \Rightarrow 3 * \mathtt{x} + 3 * \mathtt{y} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 3$
- $\mathtt{sv} \bmod 3 = 0 \wedge \mathtt{sv} \geq 6$: $(\lfloor(\mathtt{sv}-3)/3\rfloor, 0)$ $(0, \lfloor(\mathtt{sv}-3)/3\rfloor)$
- $\mathtt{sv} \bmod 3 = 1 \wedge \mathtt{sv} \geq 6$: $(\lfloor(\mathtt{sv}-3)/3\rfloor, 0)$ $(0, \lfloor(\mathtt{sv}-3)/3\rfloor)$
- $\mathtt{sv} \bmod 3 = 2 \wedge \mathtt{sv} \geq 8$: $(\lfloor(\mathtt{sv}-3)/3\rfloor, 0)$ $(\lfloor(\mathtt{sv}-3)/3\rfloor - 1, 1)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 3 * \mathtt{x} + 3 * \mathtt{y} \leq \mathtt{sv} - 5 - (\mathtt{sv} - 5) \bmod 3$
- $(\mathtt{sv}-2) \bmod 3 = 0 \wedge \mathtt{sv} \geq 11$: $(\lfloor(\mathtt{sv}-3)/3\rfloor - 1, 1)$ $(\lfloor(\mathtt{sv}-3)/3\rfloor - 2, 2)$
- $(\mathtt{sv}-2) \bmod 3 = 1 \wedge \mathtt{sv} \geq 12$: $(\lfloor(\mathtt{sv}-3)/3\rfloor - 2, 1)$ $(\lfloor(\mathtt{sv}-3)/3\rfloor - 3, 2)$
- $(\mathtt{sv}-2) \bmod 3 = 2 \wedge \mathtt{sv} \geq 13$: $(\lfloor(\mathtt{sv}-3)/3\rfloor - 2, 1)$ $(\lfloor(\mathtt{sv}-3)/3\rfloor - 3, 2)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-3)/3\rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > \max(0, \lfloor(\mathtt{sv}-3)/3\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq \max(0, \lfloor(\mathtt{sv}-3)/3\rfloor) - 1, \\ \mathtt{x} < 1, \\ \mathtt{x} > \max(0, \lfloor(\mathtt{sv}-3)/3\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

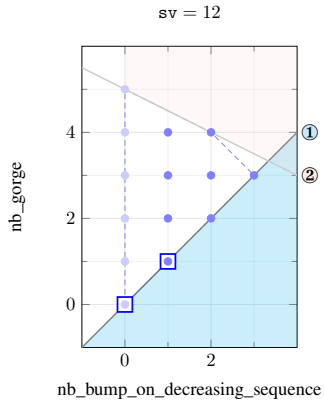


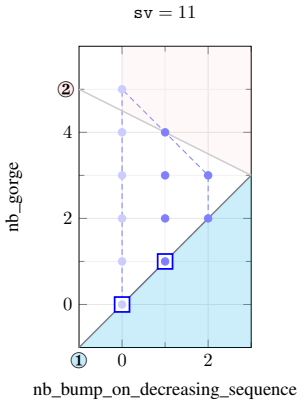


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 16: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)
- sv mod 2 = 1 ∧ sv ≥ 13: (1, ⌊(sv − 1)/2⌋ − 1) (3, ⌊(sv − 1)/2⌋ − 2)



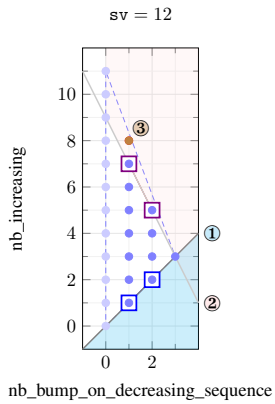


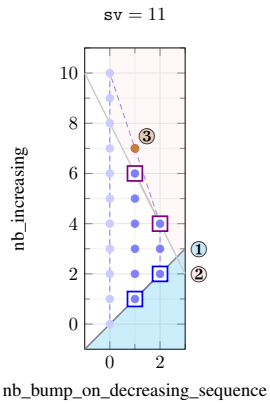
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 9: (1, 1) (2, 2)

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$
- sv ≥ 9: (1, sv − 5) (2, sv − 7)

③ $\bigvee \begin{pmatrix} y \neq \max(0, sv - 1) - 3, \\ x < 1, \\ x > \max(0, \lfloor (sv - 3)/3 \rfloor) - 1, \\ 0 = x \bmod 2 \end{pmatrix}$



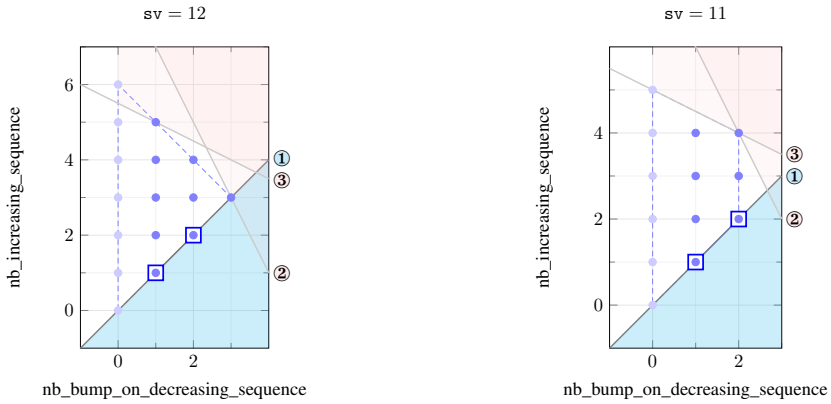


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 9: (1, 1) (2, 2)

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$

③ $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$
- sv mod $2 = 0 \wedge sv \geq 14$: (1, $\lfloor sv/2 \rfloor - 1$) (3, $\lfloor sv/2 \rfloor - 2$)
- sv mod $2 = 1 \wedge sv \geq 17$: (2, $\lfloor sv/2 \rfloor - 1$) (4, $\lfloor sv/2 \rfloor - 2$)



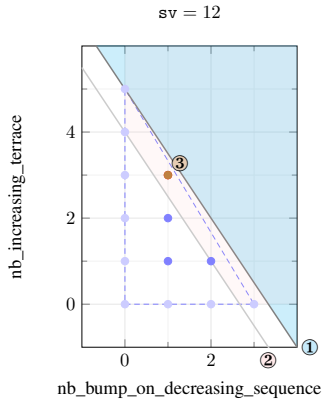
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

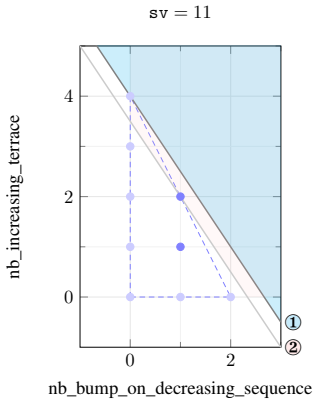
① $\mathtt{sv} > 1 \Rightarrow 3 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 2 - (\mathtt{sv} - 2) \bmod 2$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 3 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 4$

□ $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 13$: $(1, \lfloor(\mathtt{sv} - 2)/2\rfloor - 2)$ $(3, \lfloor(\mathtt{sv} - 2)/2\rfloor - 5)$

③ $$\bigvee \left(\begin{array}{l} \mathtt{y} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 2, \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{max}(0, \lfloor(\mathtt{sv} - 3)/3\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$$



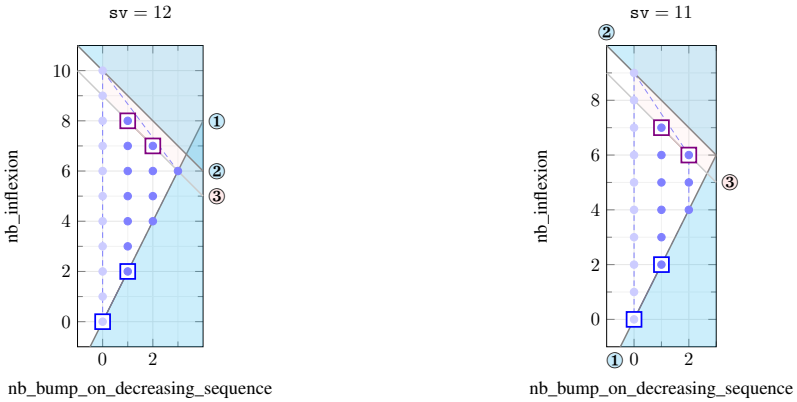


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$

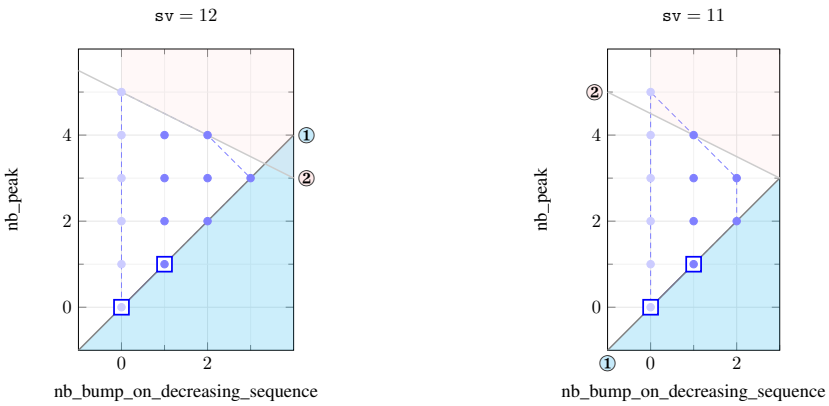
③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$
- sv ≥ 9: (1, sv − 4) (2, sv − 5)



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

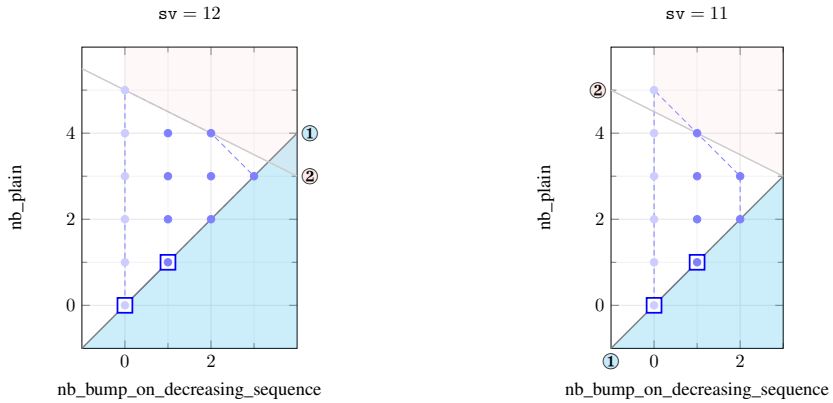
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 16: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)
- sv mod 2 = 1 ∧ sv ≥ 13: (1, ⌊(sv − 1)/2⌋ − 1) (3, ⌊(sv − 1)/2⌋ − 2)



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: (0, 0) (1, 1)

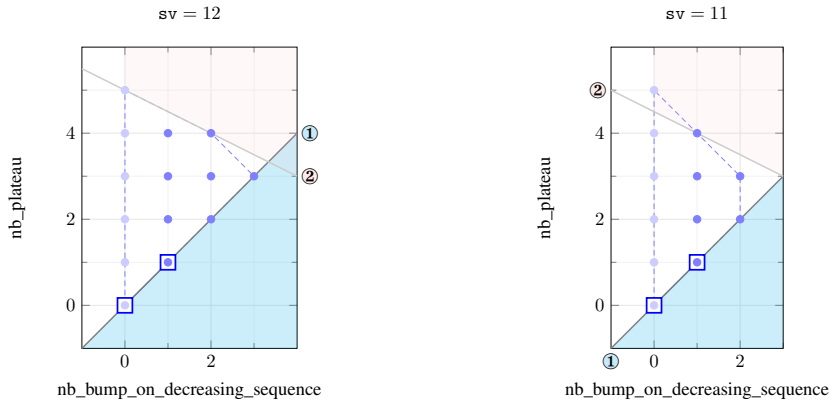
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: (0, 0) (1, 1)

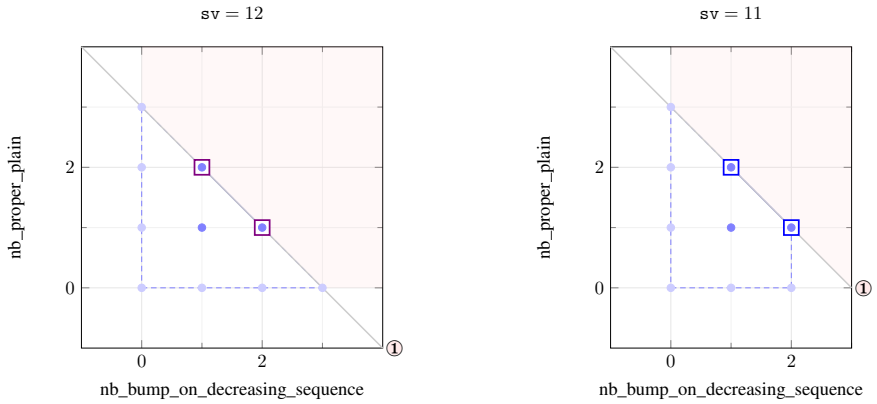
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq sv - 2 - (sv - 2) \bmod 3$

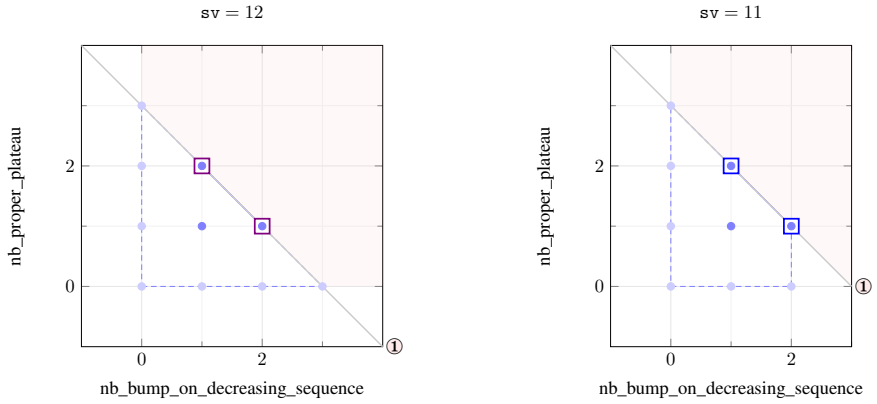
- $(sv - 2) \bmod 3 = 0 \wedge sv \geq 8$: $(\lfloor (sv - 3)/3 \rfloor, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 1, 2)$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 3)/3 \rfloor - 1, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 2, 2)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 10$: $(\lfloor (sv - 3)/3 \rfloor - 1, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 2, 2)$



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 0 \wedge sv \geq 8$: $(\lfloor (sv - 3)/3 \rfloor, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 1, 2)$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 3)/3 \rfloor - 1, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 2, 2)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 10$: $(\lfloor (sv - 3)/3 \rfloor - 1, 1)$ $(\lfloor (sv - 3)/3 \rfloor - 2, 2)$

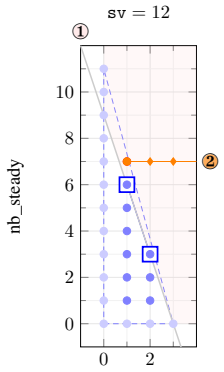


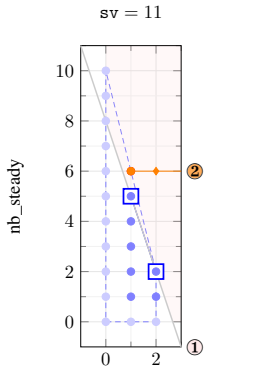
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 9$: $(1, sv - 6)$ $(2, sv - 9)$

② $x < 1 \vee y \neq \max(0, sv - 1) - 4$



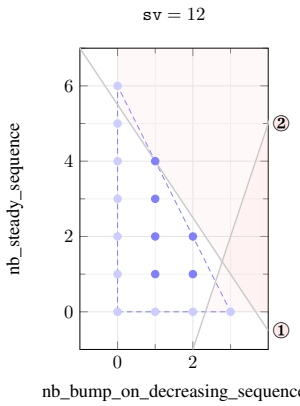


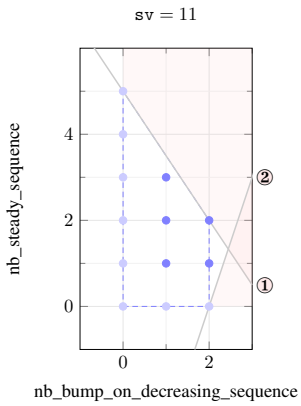
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 2 * y \leq sv - 1$

- $sv \bmod 2 = 0 \wedge sv \geq 14$: $(1, \lfloor sv/2 \rfloor - 2)$ $(3, \lfloor sv/2 \rfloor - 5)$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 5$





NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

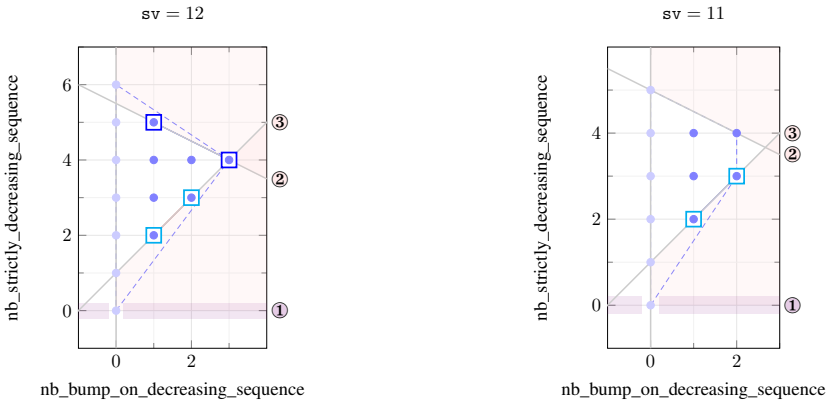
① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$

- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor sv/2 \rfloor - 1)$ $(3, \lfloor sv/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 15$: $(2, \lfloor sv/2 \rfloor - 1)$ $(4, \lfloor sv/2 \rfloor - 2)$

③ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$

- $sv \geq 9$: $(1, 2)$ $(2, 3)$



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

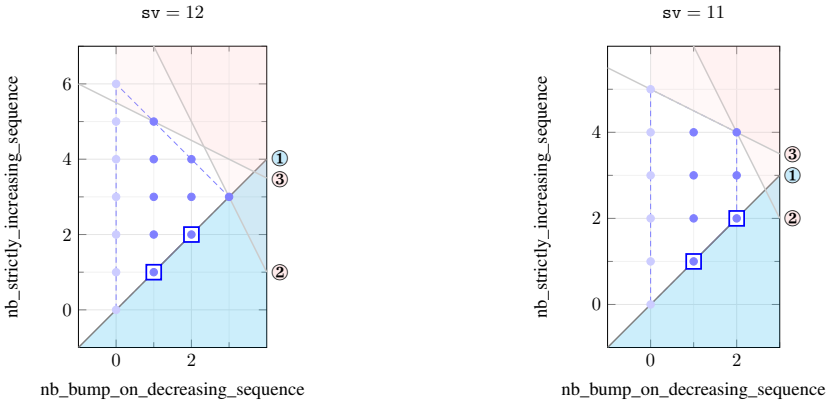
① $x \leq y$

- $sv \geq 9$: $(1, 1)$ $(2, 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$

③ $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$

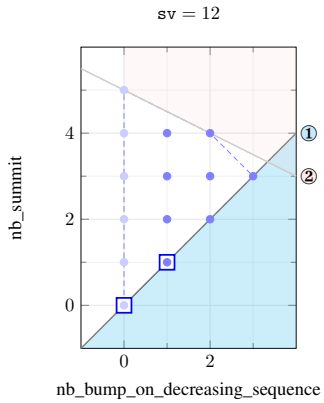
- $sv \bmod 2 = 0 \wedge sv \geq 14$: $(1, \lfloor sv/2 \rfloor - 1)$ $(3, \lfloor sv/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 17$: $(2, \lfloor sv/2 \rfloor - 1)$ $(4, \lfloor sv/2 \rfloor - 2)$

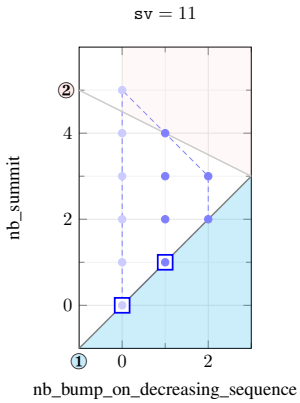


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$

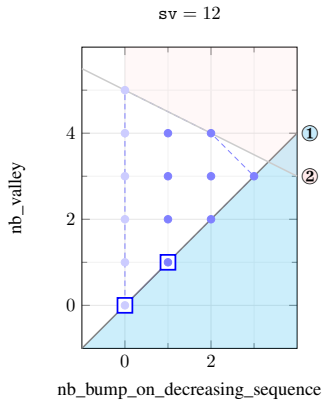


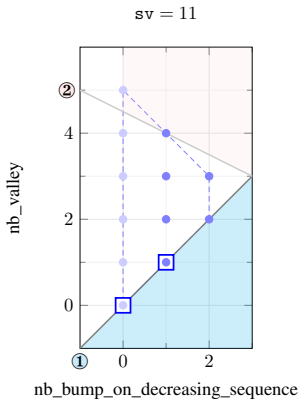


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$



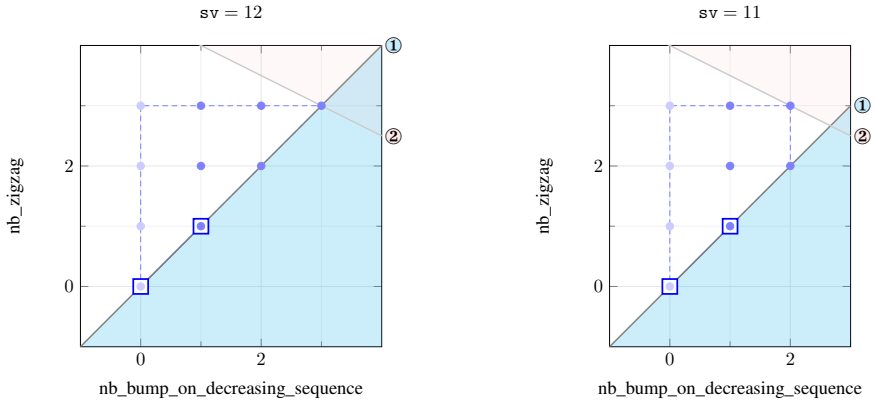


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$

□ $sv \geq 6$: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 3$



NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

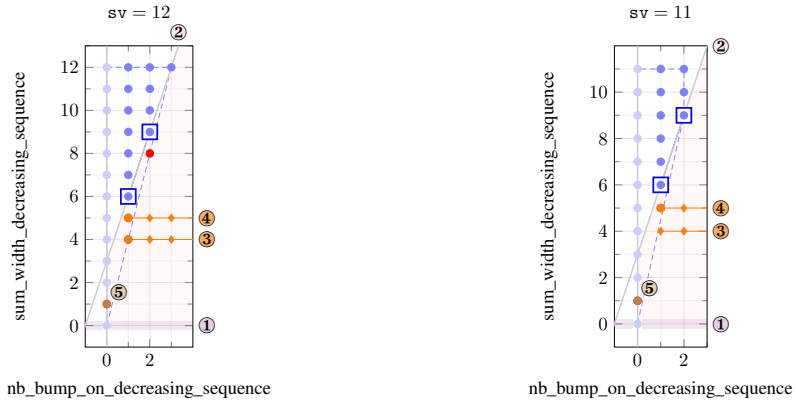
② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y - 3$

□ $sv \geq 9$: (1, 6) (2, 9)

③ $x < 1 \vee y \neq 4$

④ $x < 1 \vee y \neq 5$

⑤ $x \neq 0 \vee y \neq 1$



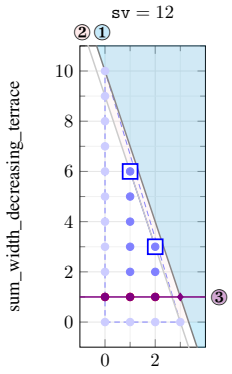
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

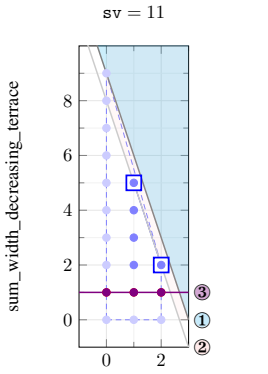
① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 11$: $(1, sv - 6)$ $(2, sv - 9)$

③ $y \neq 1$





NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

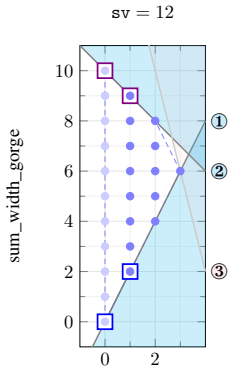
① $2 * x \leq y$

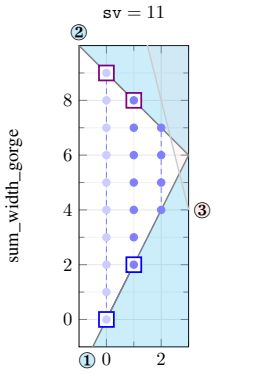
- $sv \geq 6$: $(0, 0)$ $(1, 2)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 7$: $(0, sv - 2)$ $(1, sv - 3)$

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$





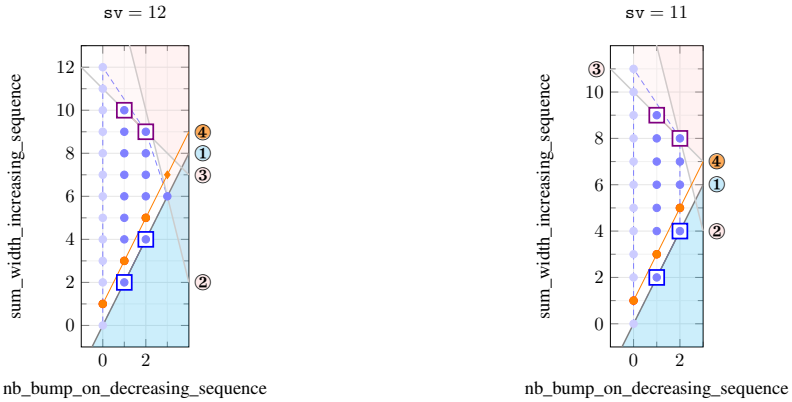
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 9: (1, 2) (2, 4)

② $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 1$
- sv ≥ 11: (1, sv − 2) (2, sv − 3)

④ $y \neq 2 * x + 1$



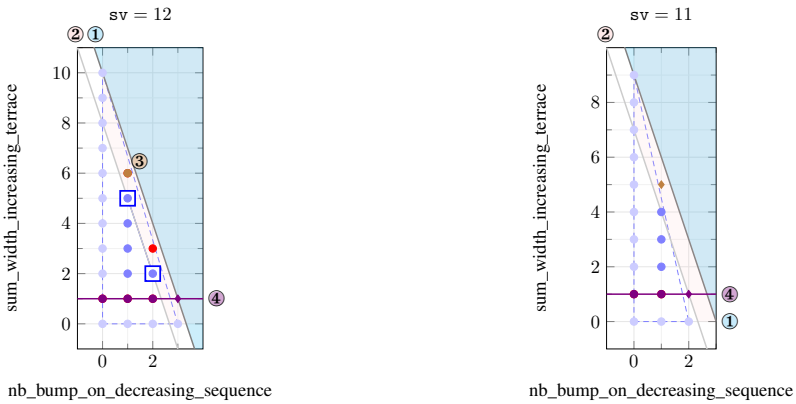
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 4$
- sv ≥ 12: (1, sv − 7) (2, sv − 10)

③ $x \neq 1 \vee y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 4 \vee sv < 7$

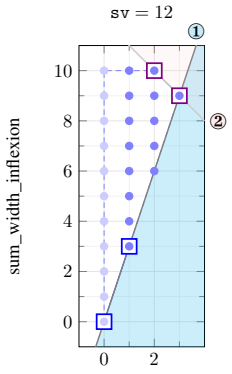
④ $y \neq 1$

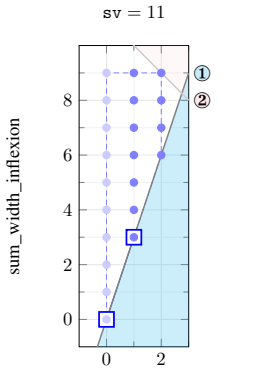


NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y$
- sv ≥ 6: (0, 0) (1, 3)

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq 4 * sv - 10 - (4 * sv - 10) \bmod 3$
- $(sv - 1) \bmod 3 = 2 \wedge sv \geq 6$: $(\lfloor (sv - 3)/3 \rfloor, sv - 3)$ $(\lfloor (sv - 3)/3 \rfloor - 1, sv - 2)$



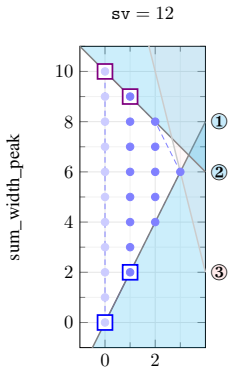


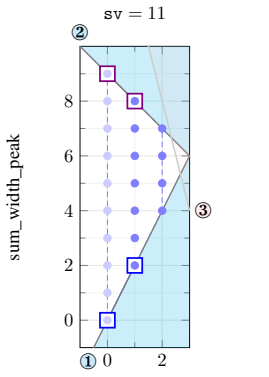
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 7: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$



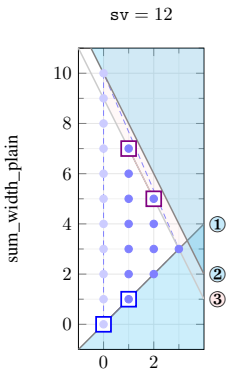


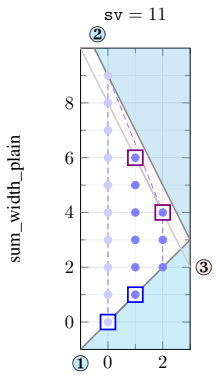
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 1)$

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$
- $sv \geq 9$: $(1, sv - 5)$ $(2, sv - 7)$



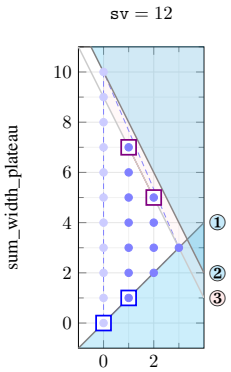


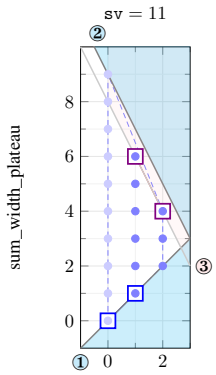
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 1)$

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$
- $sv \geq 9$: $(1, sv - 5)$ $(2, sv - 7)$





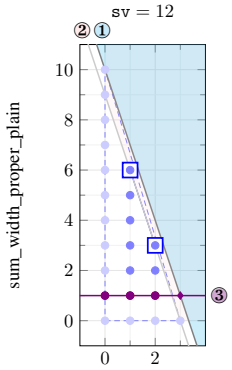
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

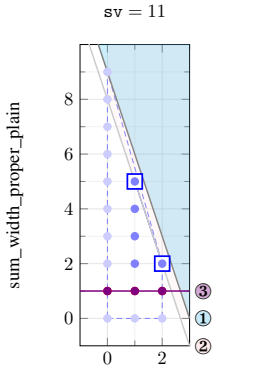
① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 11$: $(1, sv - 6)$ $(2, sv - 9)$

③ $y \neq 1$





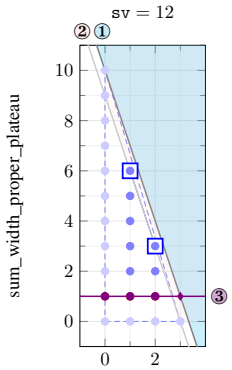
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

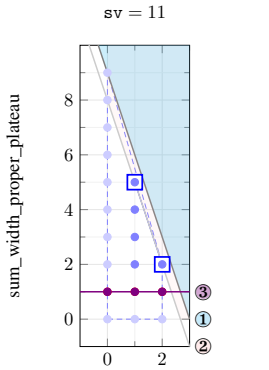
① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 11$: $(1, sv - 6)$ $(2, sv - 9)$

③ $y \neq 1$



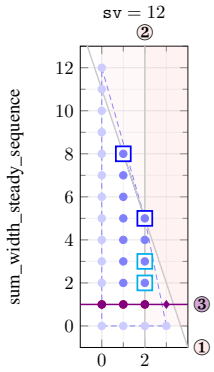


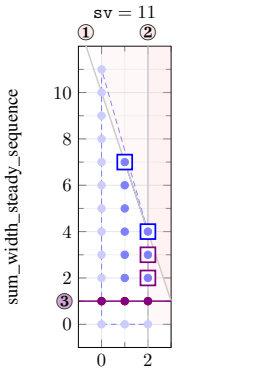
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 1$
- $sv \geq 11$: (1, sv − 4) (2, sv − 7)

② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 4 - (sv - 4) \bmod 3$
- $(sv - 1) \bmod 3 = 1 \wedge sv \geq 5$: ($\lfloor (sv - 3)/3 \rfloor$, 2) ($\lfloor (sv - 3)/3 \rfloor$, 3)
- $(sv - 1) \bmod 3 = 2 \wedge sv \geq 6$: ($\lfloor (sv - 3)/3 \rfloor - 1$, 2) ($\lfloor (sv - 3)/3 \rfloor - 1$, 3)

③ $y \neq 1$





NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

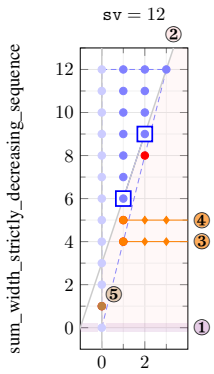
① $y = 0 \Rightarrow x = 0$

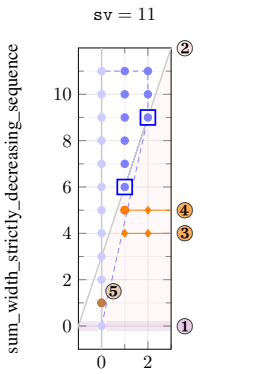
② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y - 3$
- $sv \geq 9$: (1, 6) (2, 9)

③ $x < 1 \vee y \neq 4$

④ $x < 1 \vee y \neq 5$

⑤ $x \neq 0 \vee y \neq 1$





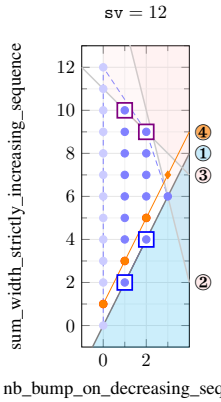
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

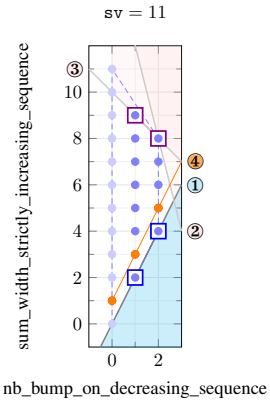
① $2 * x \leq y$
- $sv \geq 9$: $(1, 2)$ $(2, 4)$

② $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 1$
- $sv \geq 11$: $(1, sv - 2)$ $(2, sv - 3)$

④ $y \neq 2 * x + 1$



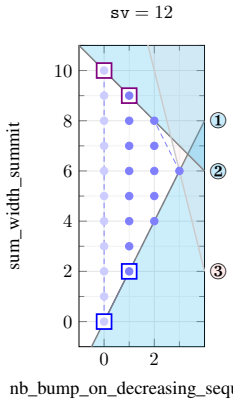


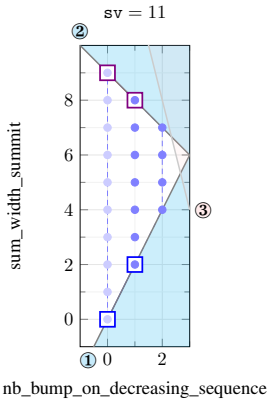
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 2)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 7$: $(0, sv - 2)$ $(1, sv - 3)$

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$



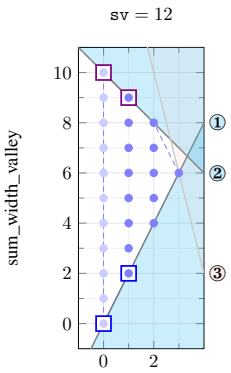


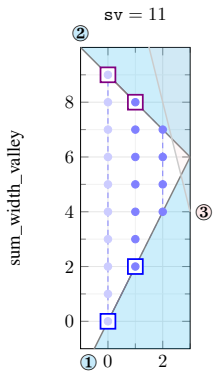
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 7: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$





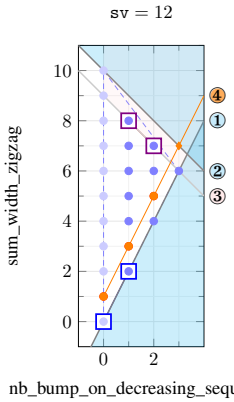
NB_BUMP_ON_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

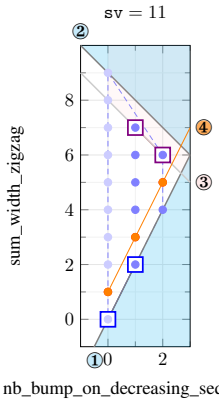
① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$
- sv ≥ 11: (1, sv − 4) (2, sv − 5)

④ $y \neq 2 * x + 1$





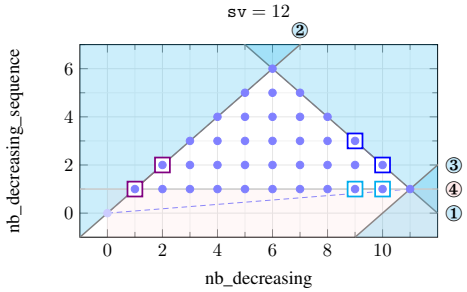
NB_DECREASING(x, VARIABLES) ∧
NB_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

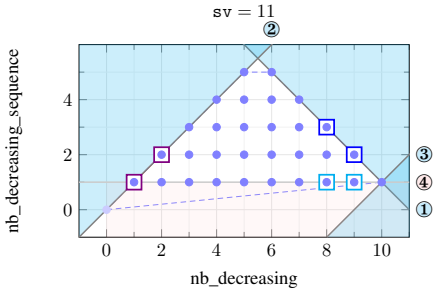
① $x + y \leq sv$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $y \leq x$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

③ $sv > 1 \Rightarrow x \leq y + sv - 2$

④ $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$



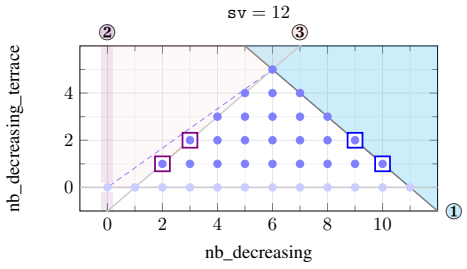


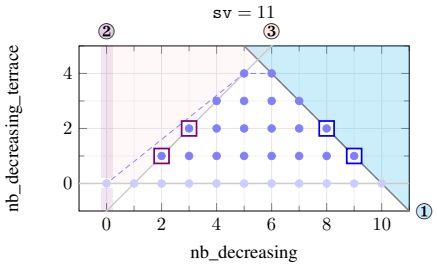
NB_DECREASING(x, VARIABLES) ∧
NB_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 6$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $x = 0 \Rightarrow y = 0$

③ $x > 0 \wedge y > 0 \Rightarrow y \leq x - 1$
- $sv \geq 6$: $(2, 1)$ $(3, 2)$



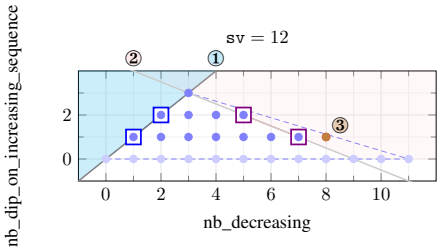


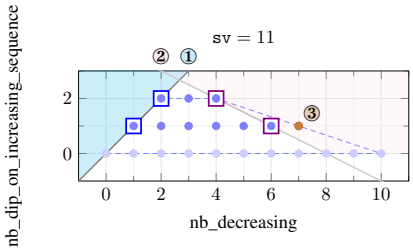
NB_DECREASING(x, VARIABLES) ∧
NB_DIP_ON_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 9$: $(1, 1)$ $(2, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 3$
- $sv \geq 9$: $(sv - 5, 1)$ $(sv - 7, 2)$

③ $\bigvee \begin{pmatrix} x \neq \max(0, sv - 1) - 3, \\ y < 1, \\ y > \max(0, \lfloor (sv - 3)/3 \rfloor) - 1, \\ 0 = y \bmod 2 \end{pmatrix}$

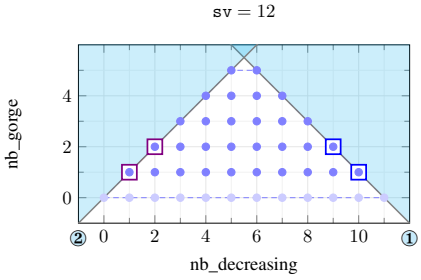


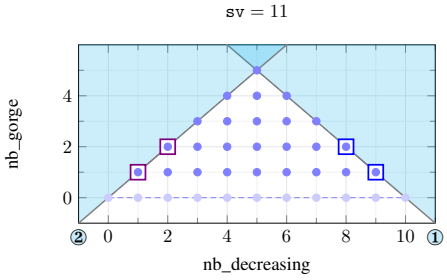


NB_DECREASING(x, VARIABLES) ∧
NB_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

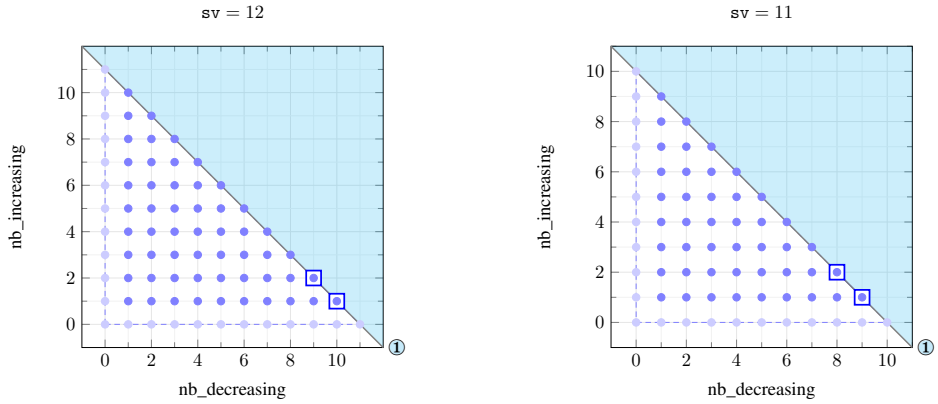




NB_DECREASING(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

- $sv \geq 3$: $(sv - 2, 1)$ $(sv - 3, 2)$



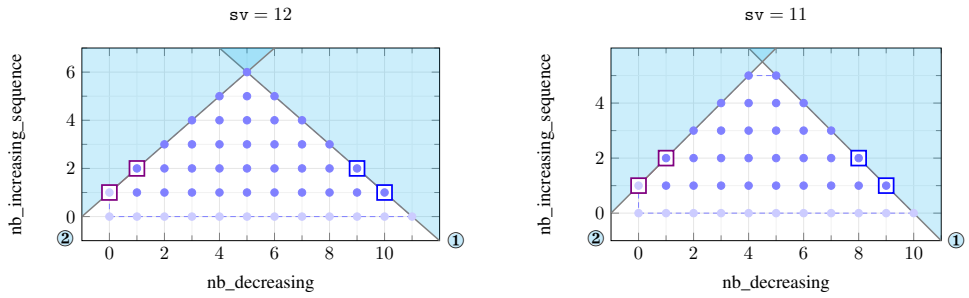
NB_DECREASING(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x + 1$

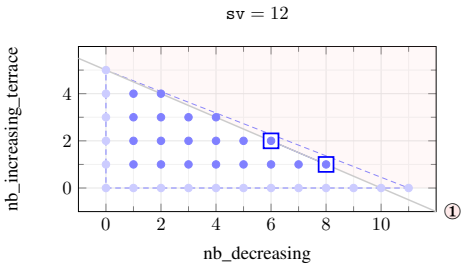
- $sv \geq 4$: $(0, 1)$ $(1, 2)$

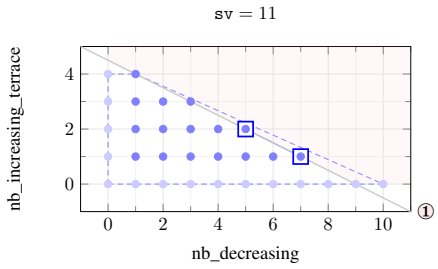


NB_DECREASING(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$

- $sv \geq 6$: $(sv - 4, 1)$ $(sv - 6, 2)$





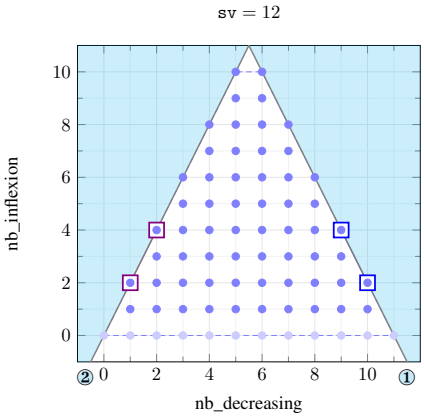
NB_DECREASING(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

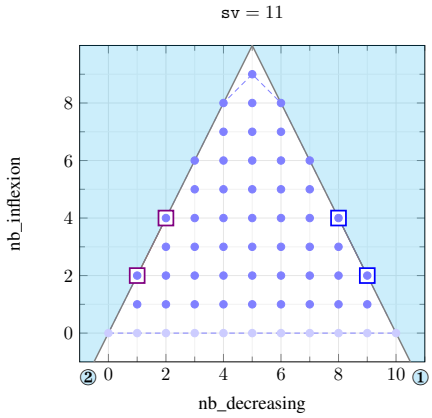
① $2 * x + y \leq 2 * sv - 2$

- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \leq 2 * x$

- $sv \geq 6$: $(1, 2)$ $(2, 4)$

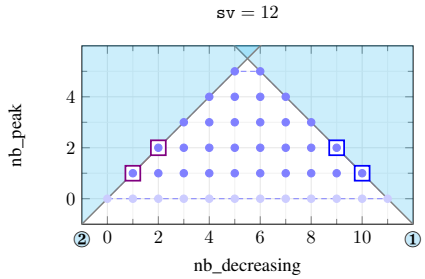


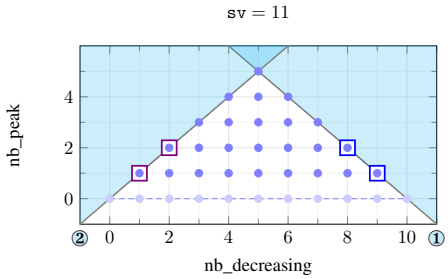


NB_DECREASING(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

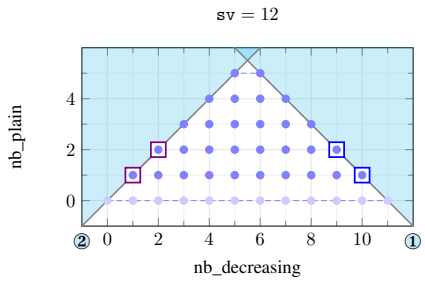


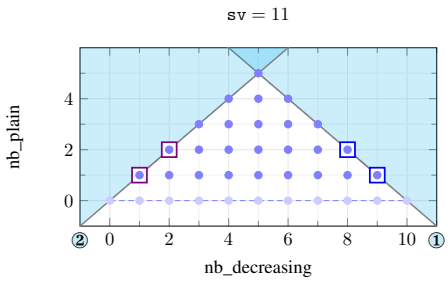


NB_DECREASING(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

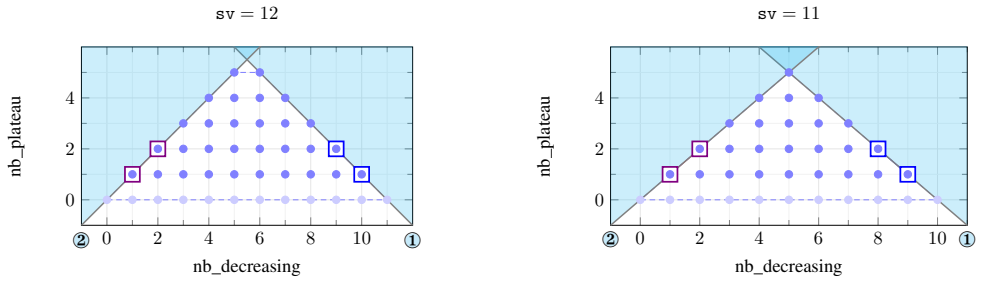




NB_DECREASING(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

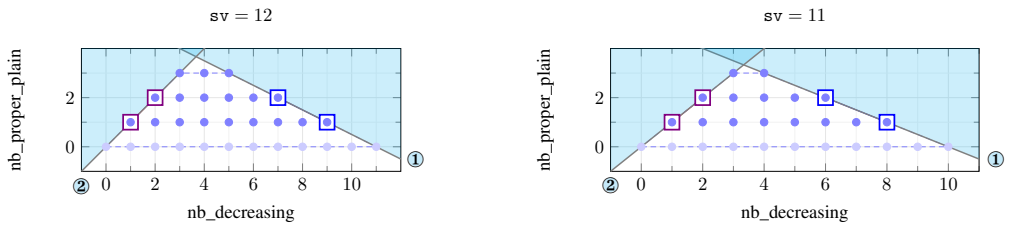
② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$



NB_DECREASING(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- $sv \geq 7$: $(sv - 3, 1)$ $(sv - 5, 2)$

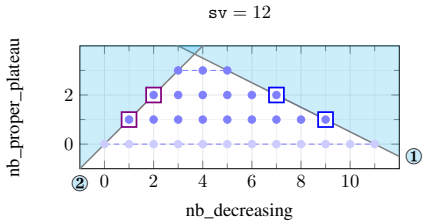
② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

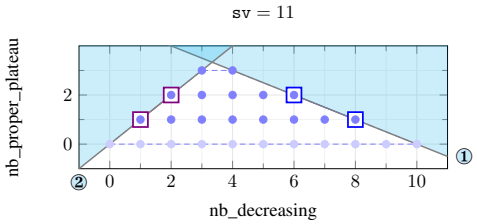


NB_DECREASING(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- $sv \geq 7$: $(sv - 3, 1)$ $(sv - 5, 2)$

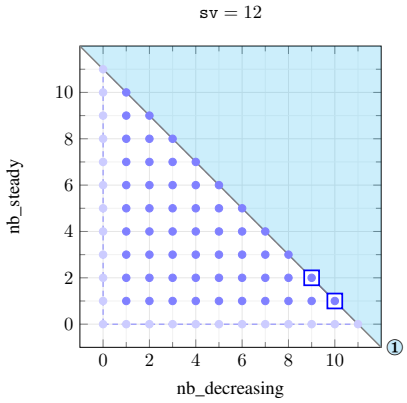
② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

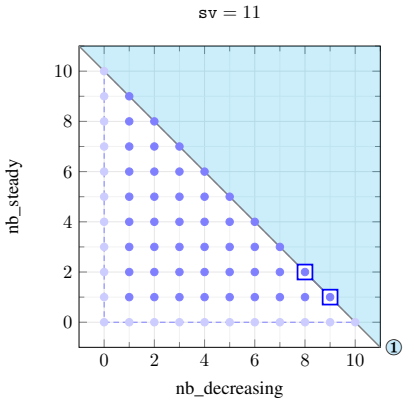




NB_DECREASING(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

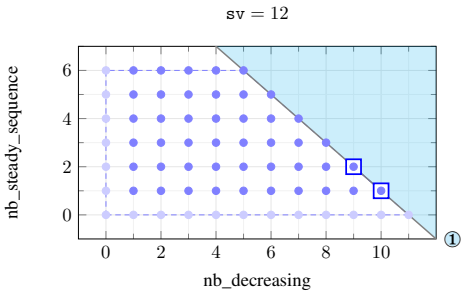
① $x + y \leq sv - 1$
- $sv \geq 3$: $(sv - 2, 1)$ $(sv - 3, 2)$

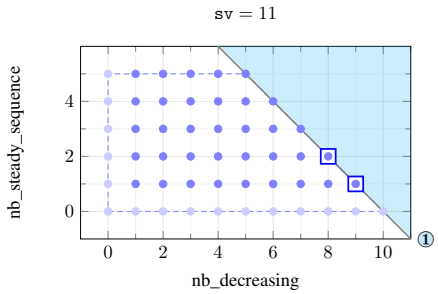




NB_DECREASING(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$





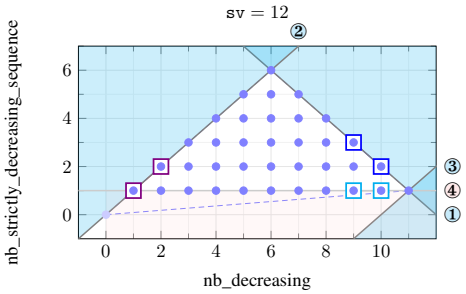
NB_DECREASING(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

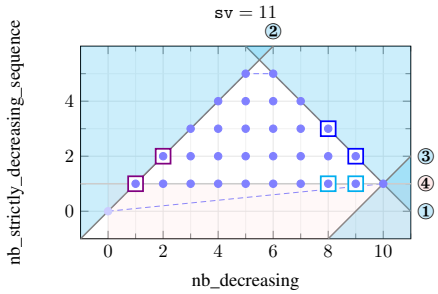
① $x + y \leq sv$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $y \leq x$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

③ $sv > 1 \Rightarrow x \leq y + sv - 2$

④ $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$

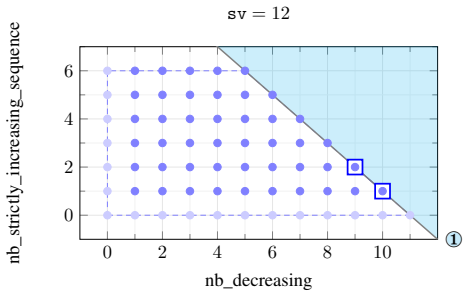


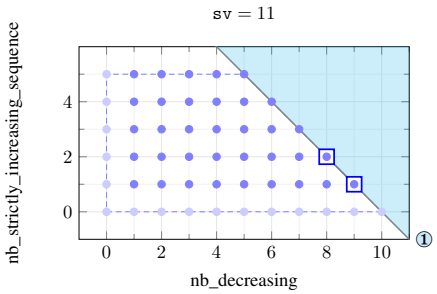


NB_DECREASING(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$





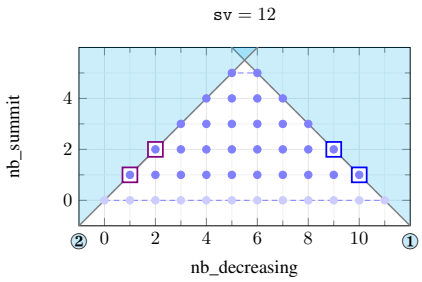
NB_DECREASING(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

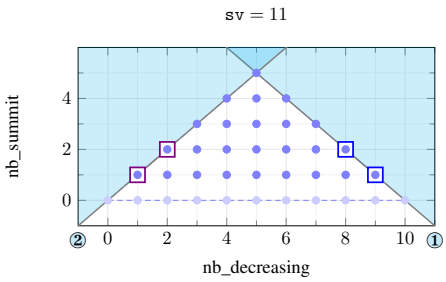
① $x + y \leq sv - 1$

- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$

- $sv \geq 5$: $(1, 1)$ $(2, 2)$

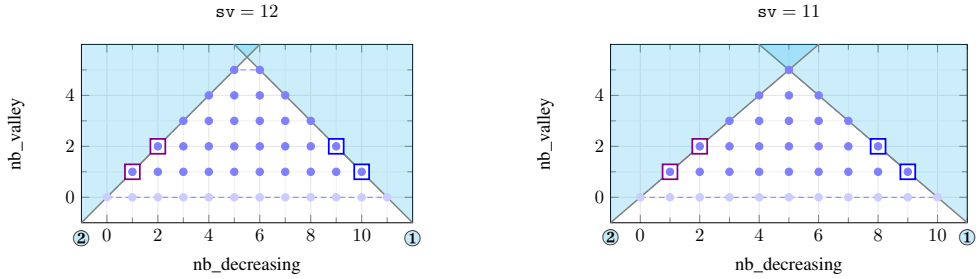




NB_DECREASING(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

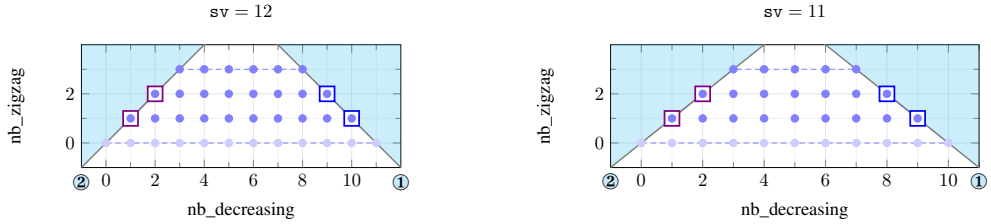


NB_DECREASING(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 7$: $(sv - 2, 1)$ $(sv - 3, 2)$

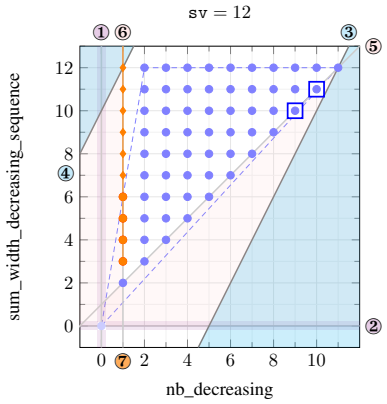
② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

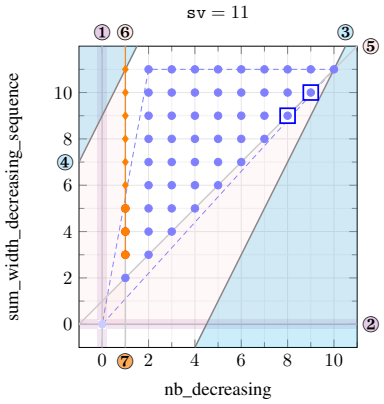
③ $$\bigvee \left(\begin{array}{l} y \neq \max(0, \lfloor (sv - 1)/3 \rfloor) - 0, \\ x < \max(0, \lfloor (sv - 1)/3 \rfloor) + 1, \\ x > 1 * \max(0, sv - 1) - \max(0, \lfloor (sv - 1)/3 \rfloor) - 1, \\ 1 \neq sv \bmod 3 \end{array} \right)$$



NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x = 0 \Rightarrow y = 0$
② $y = 0 \Rightarrow x = 0$
③ $sv > 1 \Rightarrow 2 * x \leq y + sv - 2$
④ $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$
⑤ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
□ $sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv - 2)$
⑥ $y > 0 \Rightarrow x \geq 1$
⑦ $x \neq 1 \vee y < 3$





NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

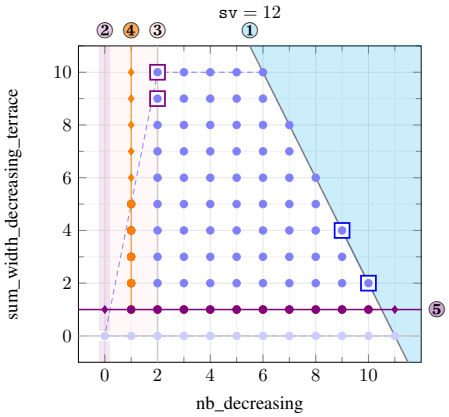
① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

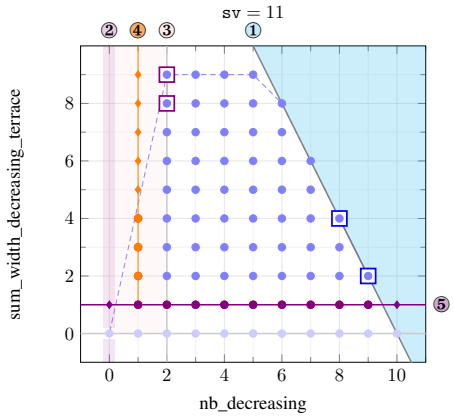
② $x = 0 \Rightarrow y = 0$

③ $x > 0 \wedge y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

④ $x \neq 1 \vee y < 1$

⑤ $y \neq 1$

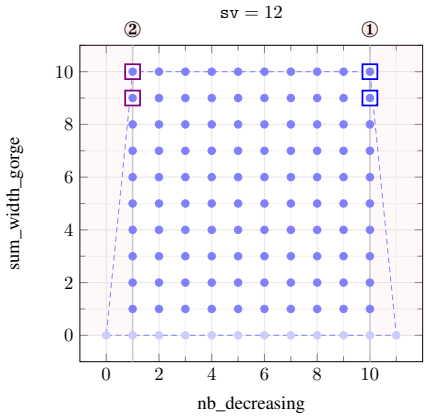


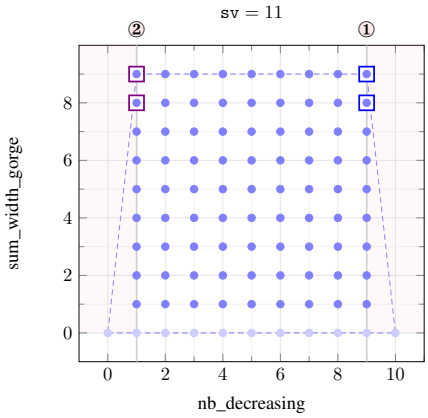


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

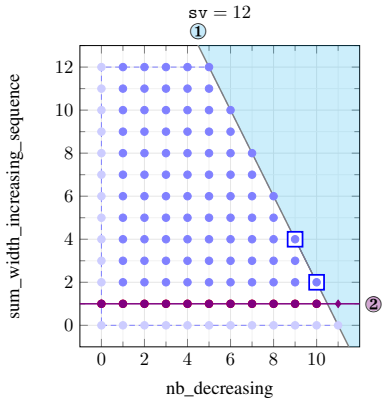


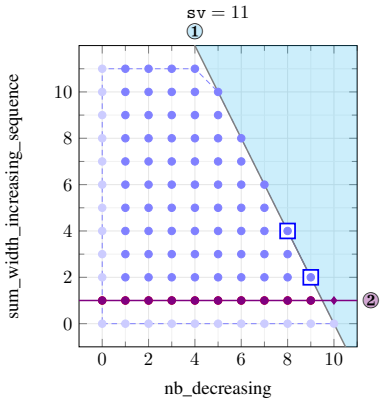


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$



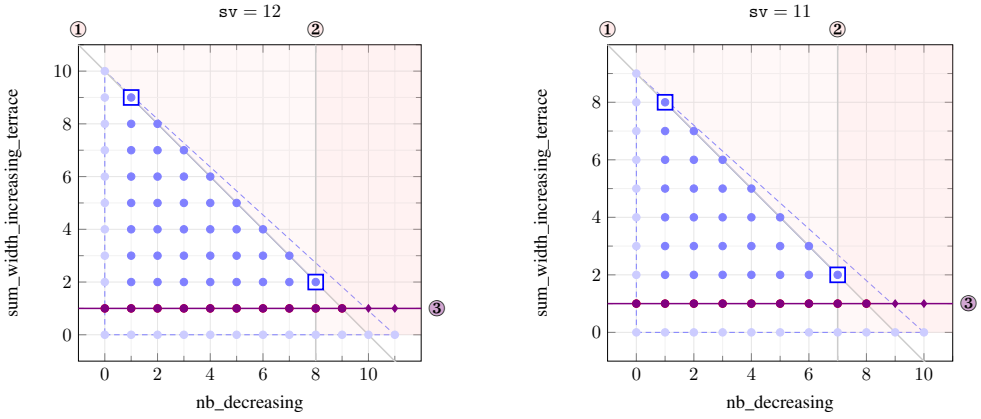


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 4, 2)$ $(1, sv - 3)$

② $y > 0 \Rightarrow x \leq sv - 4$

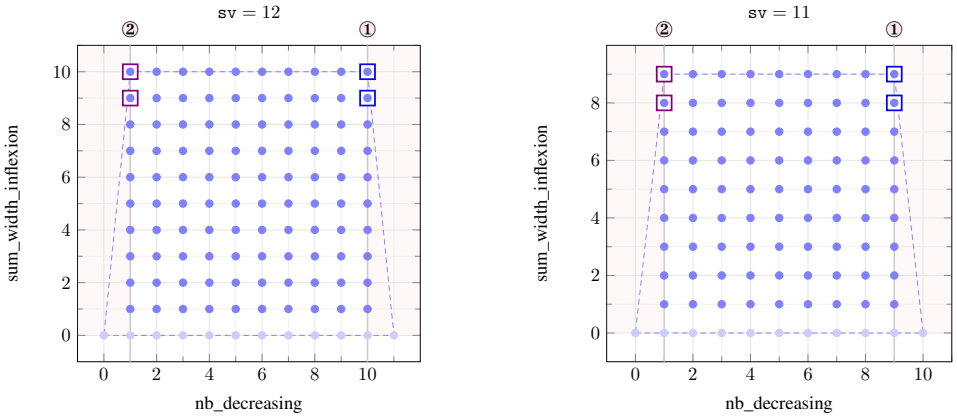
③ $y \neq 1$



NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

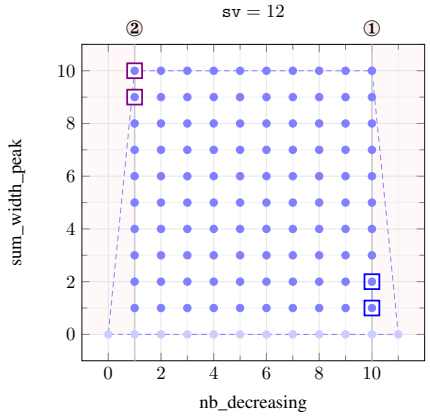
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

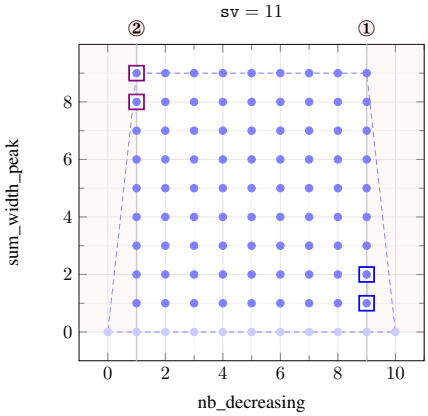


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 2, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

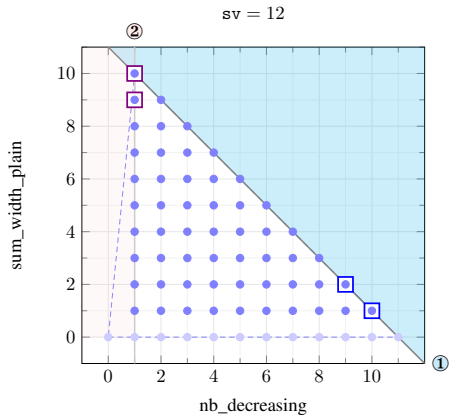


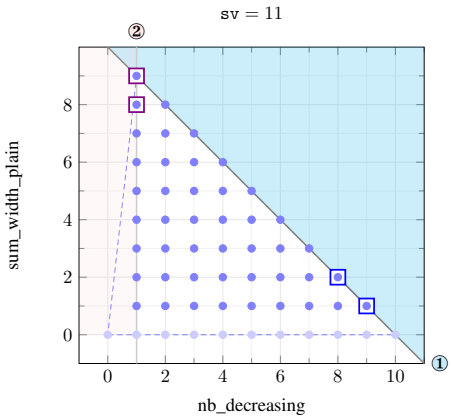


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

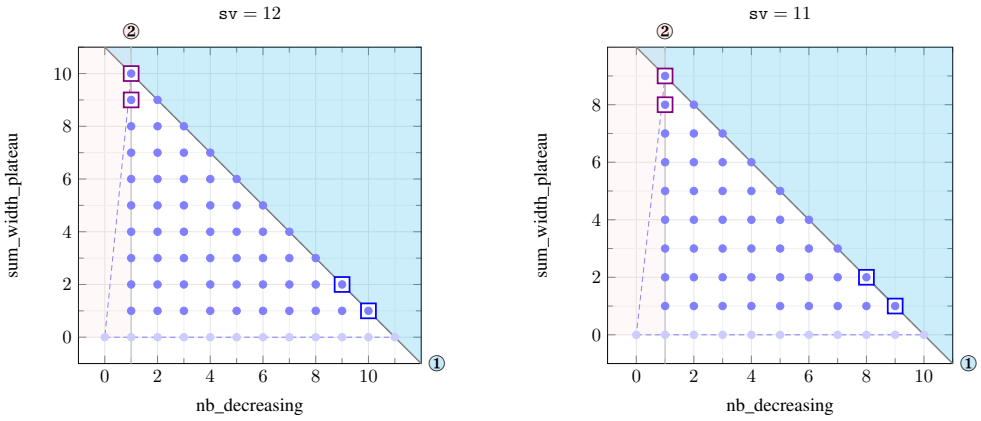




NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



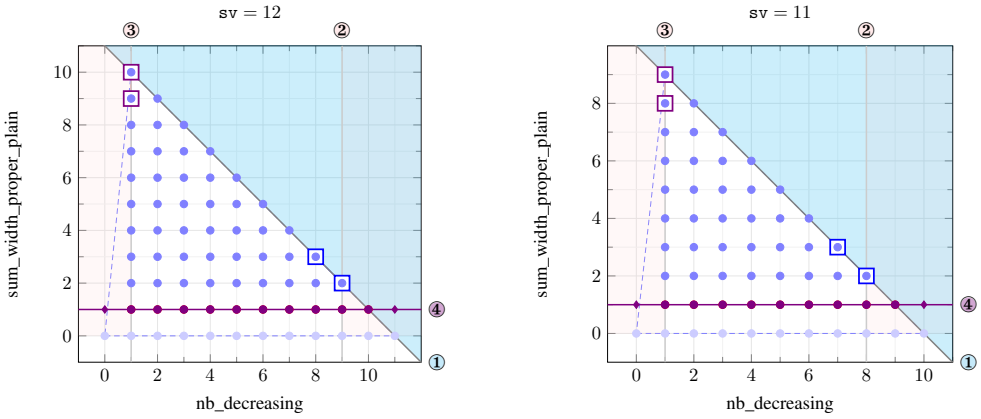
NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 4, 3)$

② $y > 0 \Rightarrow x \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$



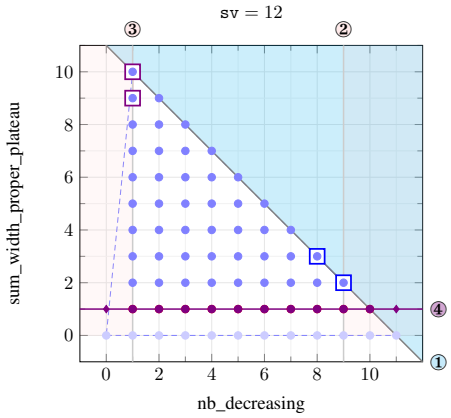
NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

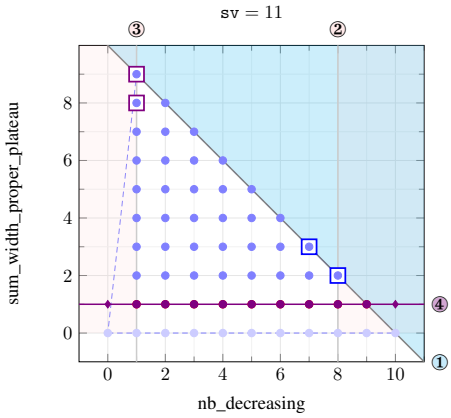
① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 4, 3)$

② $y > 0 \Rightarrow x \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$

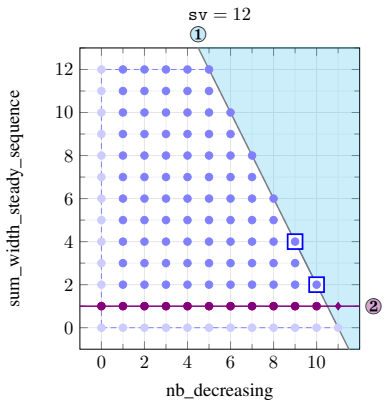


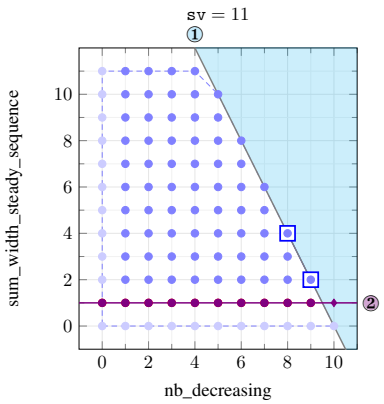


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$





NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

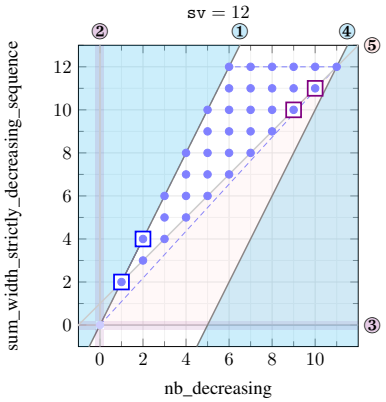
① $y \leq 2 * x$
- $sv \geq 4$: $(1, 2)$ $(2, 4)$

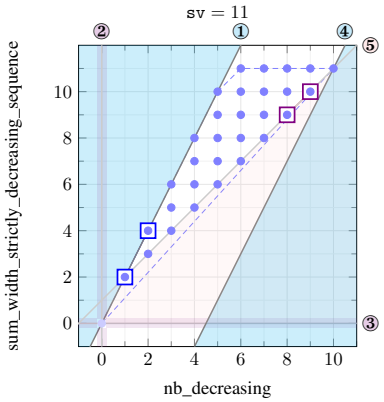
② $x = 0 \Rightarrow y = 0$

③ $y = 0 \Rightarrow x = 0$

④ $sv > 1 \Rightarrow 2 * x \leq y + sv - 2$

⑤ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv - 2)$

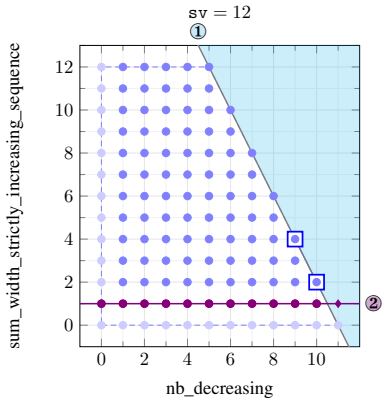


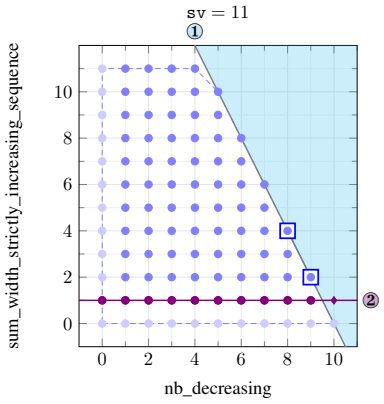


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$

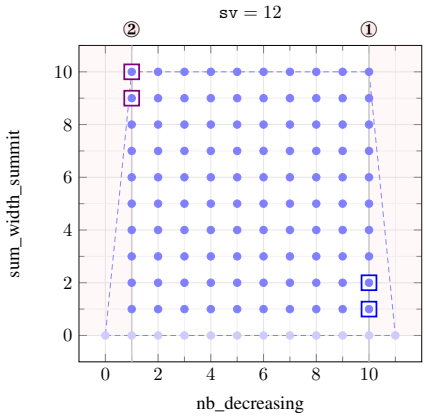


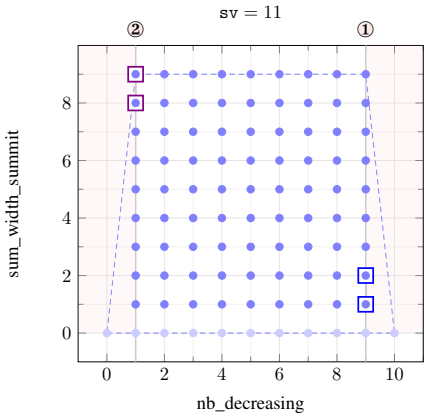


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 2, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

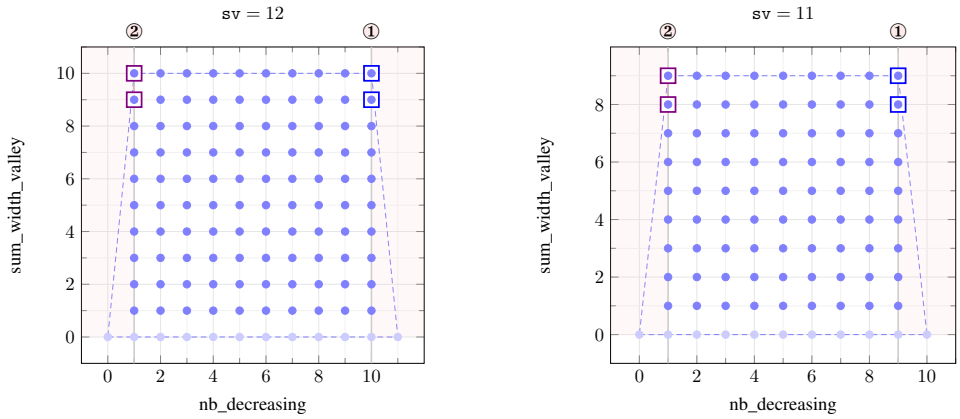




NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$

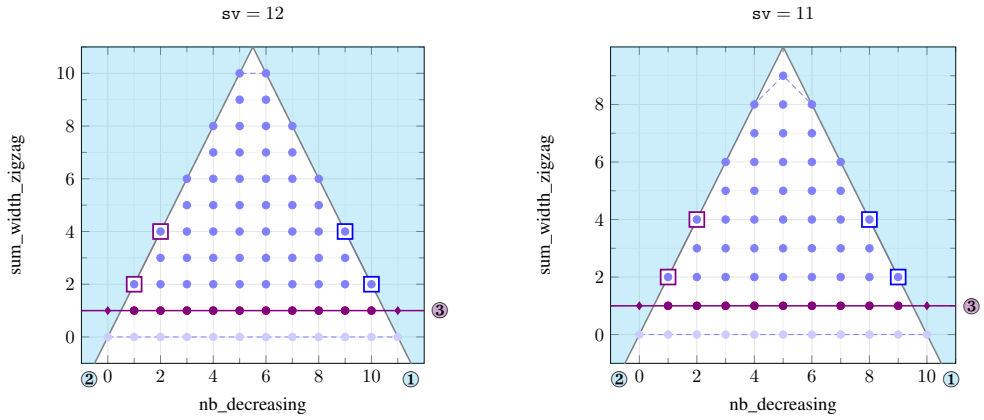


NB_DECREASING(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \leq 2 * x$
- $sv \geq 6$: $(1, 2)$ $(2, 4)$

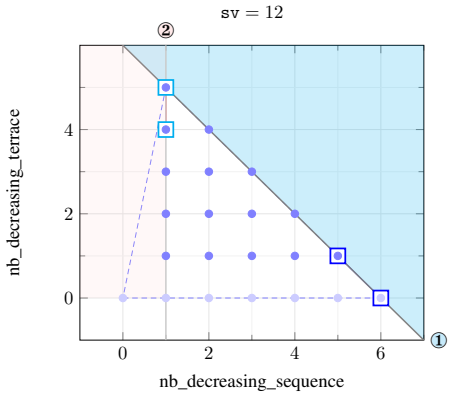
③ $y \neq 1$

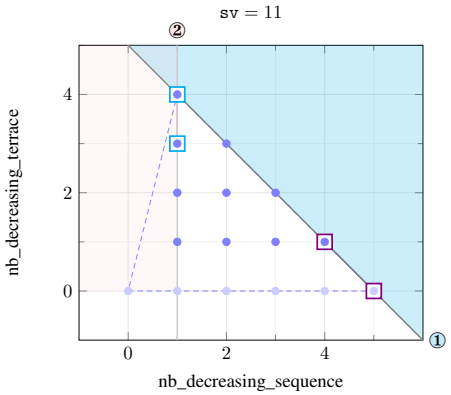


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, \lfloor (sv - 2)/2 \rfloor)$ $(1, \lfloor (sv - 2)/2 \rfloor - 1)$



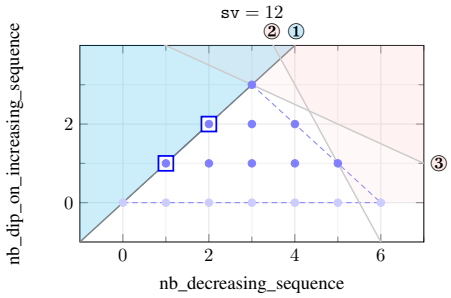


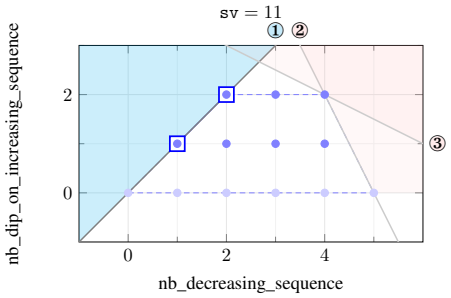
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_DIP_ON_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 9$: $(1, 1)$ $(2, 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 14$: $(\lfloor sv/2 \rfloor - 1, 1)$ $(\lfloor sv/2 \rfloor - 2, 3)$
- $sv \bmod 2 = 1 \wedge sv \geq 17$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 2, 4)$

③ $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 3$

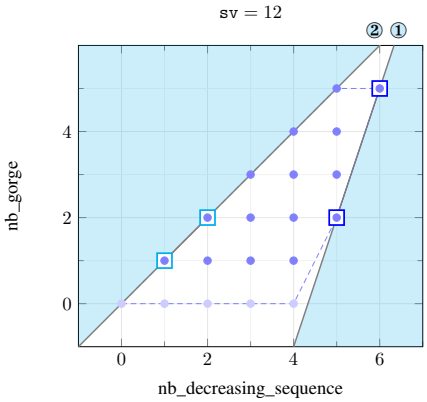


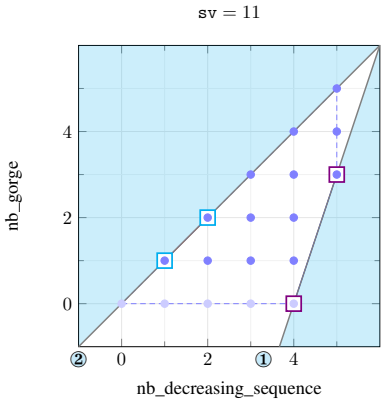


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv + 1$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 3)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 2)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 5)$

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

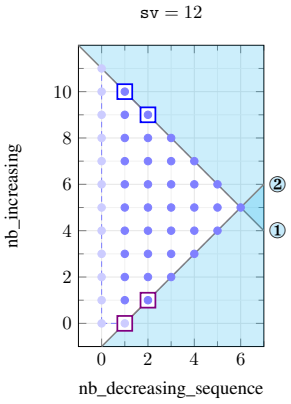


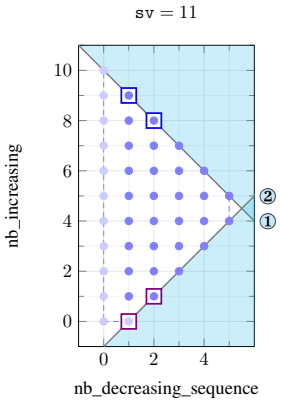


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 4: (1, sv − 2) (2, sv − 3)

② $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)



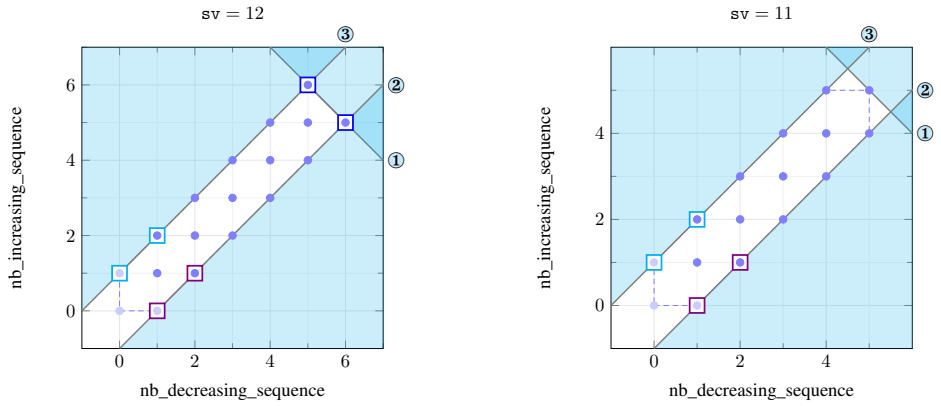


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor sv/2 \rfloor - 1, \lfloor sv/2 \rfloor)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

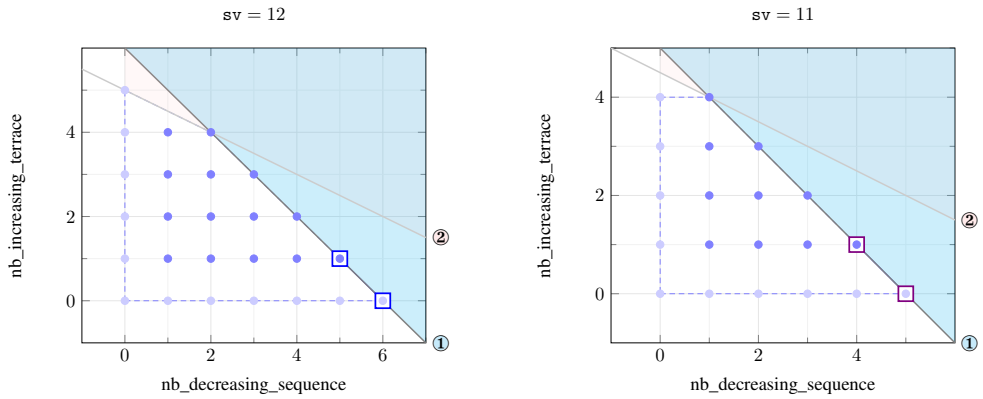
③ $y \leq x + 1$
- $sv \geq 4$: $(0, 1)$ $(1, 2)$



NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$

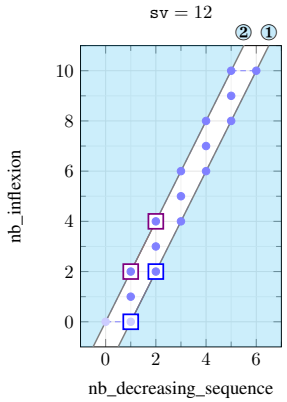
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$

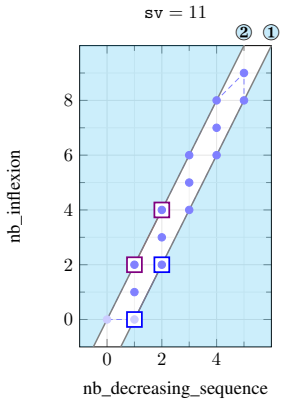


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

② $y \leq 2 * x$
- sv ≥ 6: (1, 2) (2, 4)

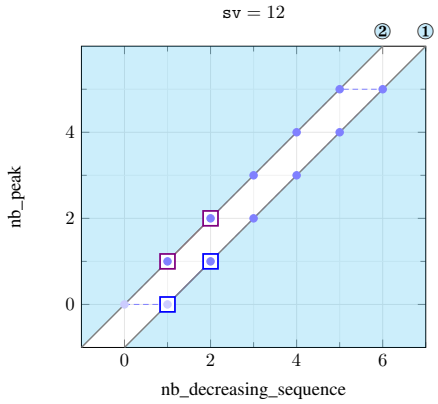


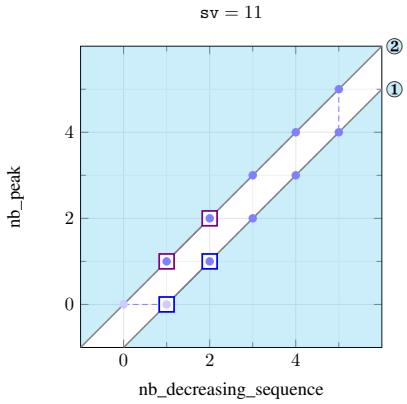


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

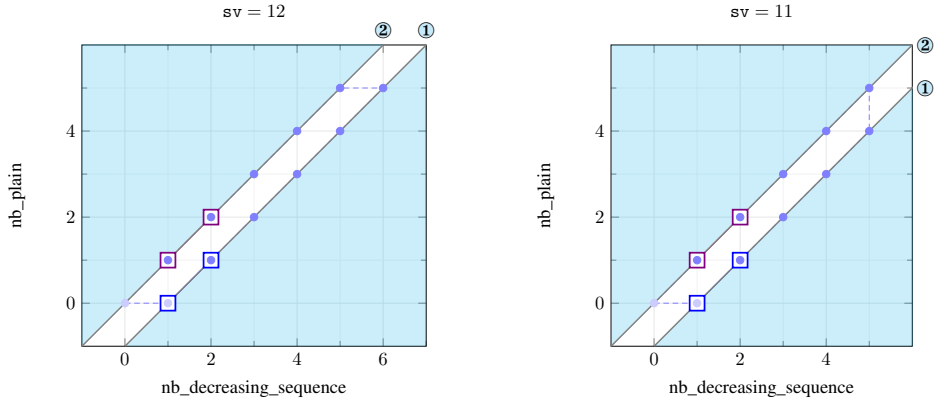




NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

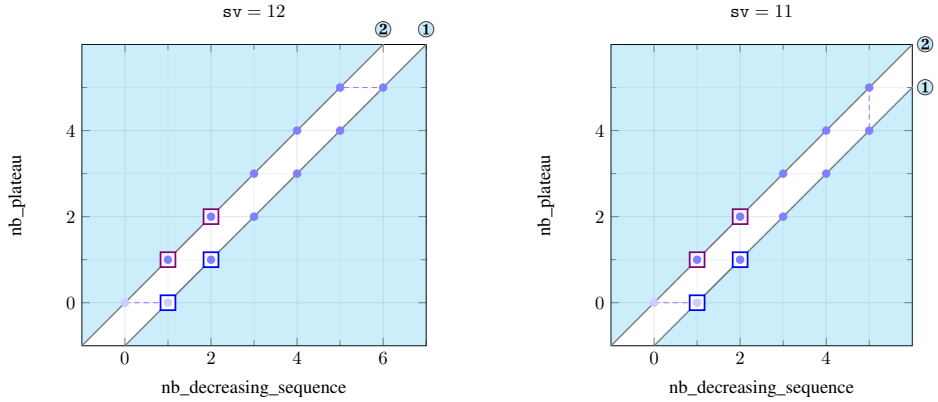
② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)



NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

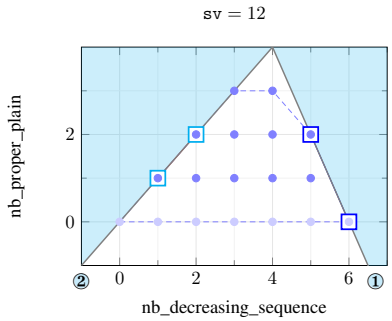
② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

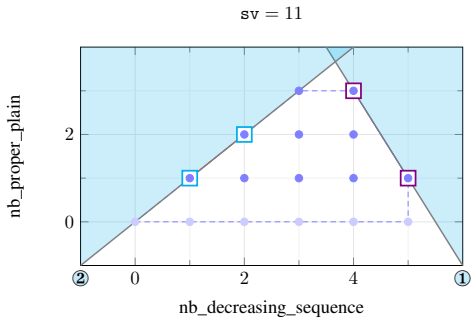


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 2)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊sv/2⌋, 1) (⌊sv/2⌋ − 1, 3)

② $y \leq x$
- sv ≥ 7: (1, 1) (2, 2)

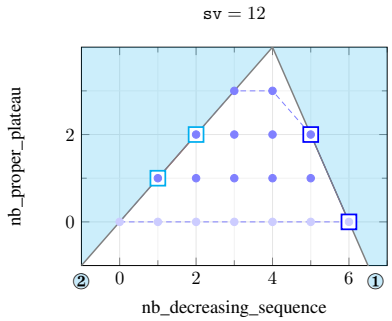


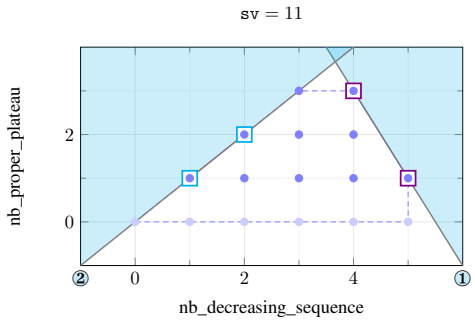


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 2)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊sv/2⌋, 1) (⌊sv/2⌋ − 1, 3)

② $y \leq x$
- sv ≥ 7: (1, 1) (2, 2)

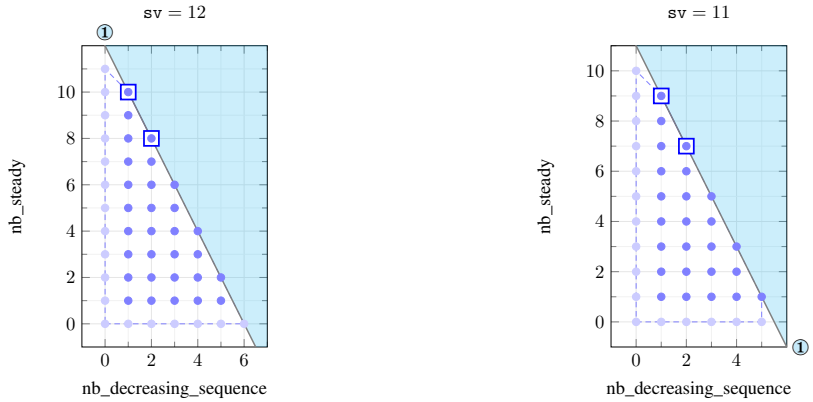




NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$

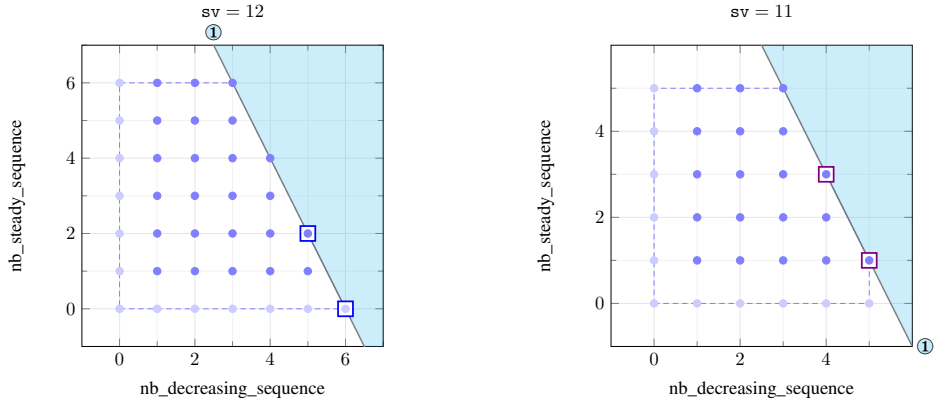
- sv ≥ 4: (1, sv − 2) (2, sv − 4)



NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$

- sv mod 2 = 0 ∧ sv ≥ 4: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 2)
- sv mod 2 = 1 ∧ sv ≥ 7: (⌊sv/2⌋, 1) (⌊sv/2⌋ − 1, 3)

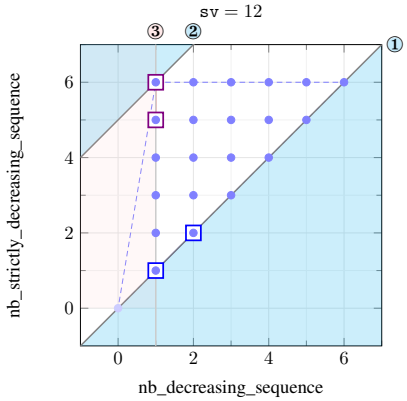


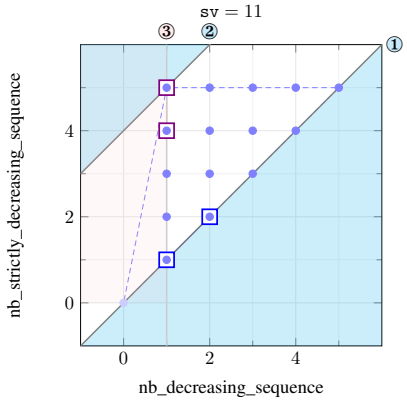
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 4: (1, 1) (2, 2)

② $sv > 1 \Rightarrow 2 * y \leq 2 * x + sv - 2 - (sv - 2) \bmod 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, ⌊sv/2⌋) (1, ⌊sv/2⌋ − 1)

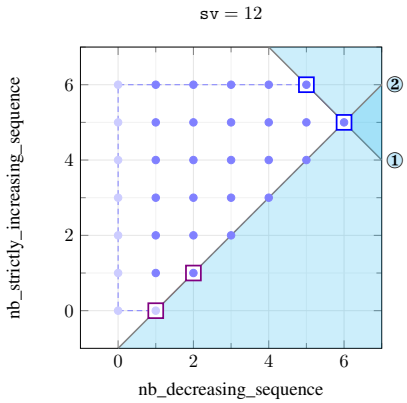


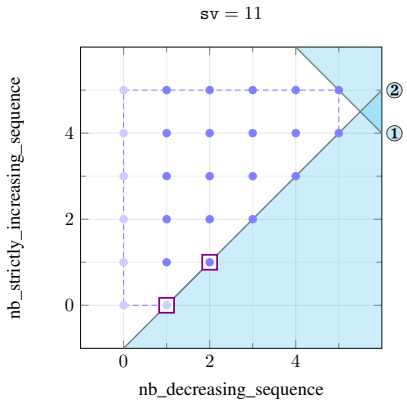


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 2: (⌊sv/2⌋, ⌊sv/2⌋ − 1) (⌊sv/2⌋ − 1, ⌊sv/2⌋)

② $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

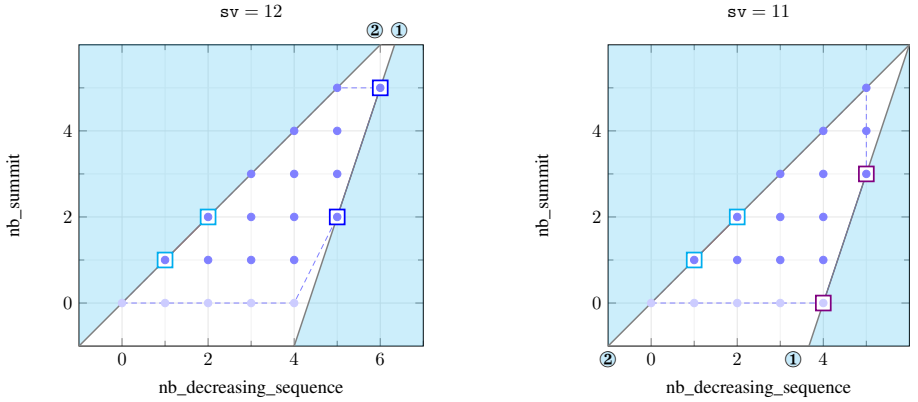




NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv + 1$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 3)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 2)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 5)$

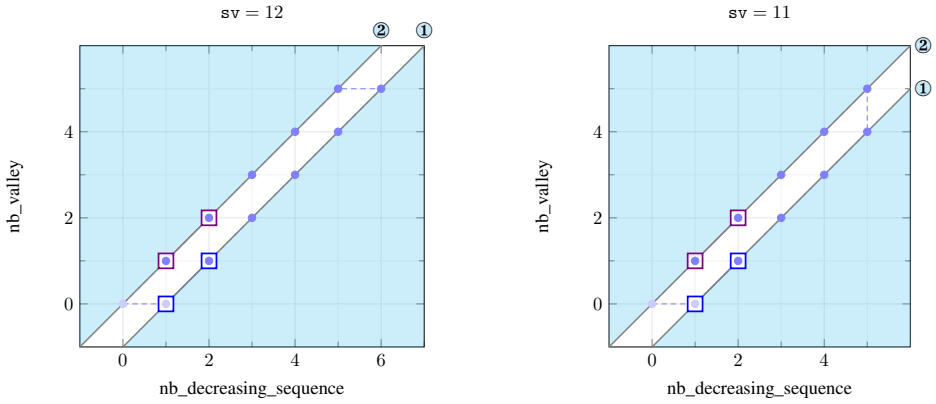
② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)



NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

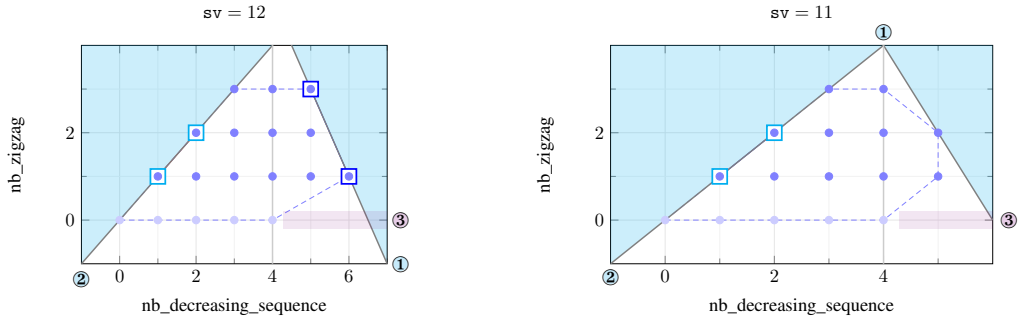


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv + 1$
- sv mod 2 = 0 ∧ sv ≥ 10: (⌊sv/2⌋, 1) (⌊sv/2⌋ − 1, 3)
- sv mod 2 = 1 ∧ sv ≥ 13: (⌊sv/2⌋, 2) (⌊sv/2⌋ − 1, 4)

② $y \leq x$
- sv ≥ 7: (1, 1) (2, 2)

③ $y = 0 \Rightarrow 3 * x \leq sv + 1 - (sv + 1) \bmod 3$

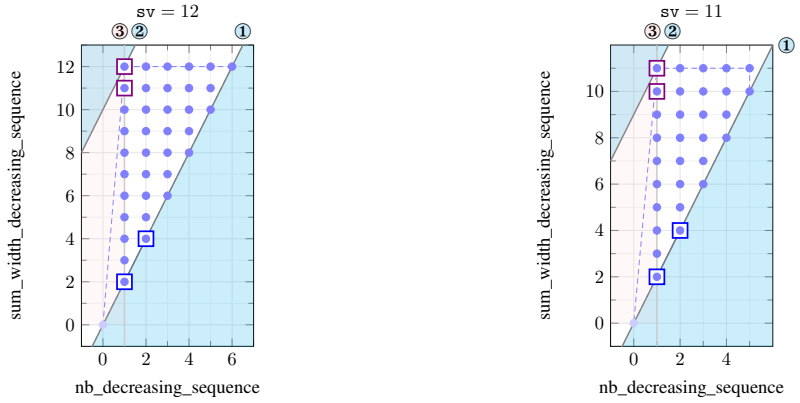


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv) (1, sv − 1)



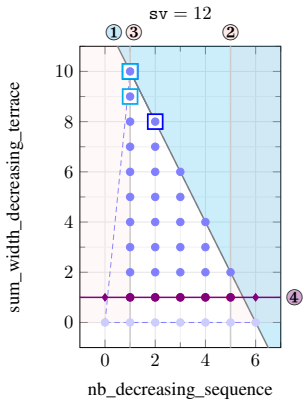
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

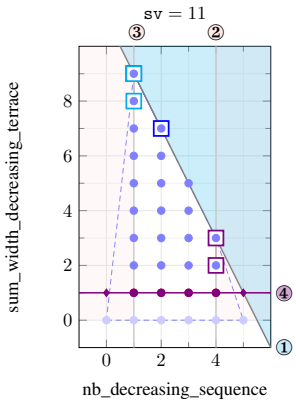
① $2 * x + y \leq sv$
- $sv \geq 6$: $(1, sv - 2)$ $(2, sv - 4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$



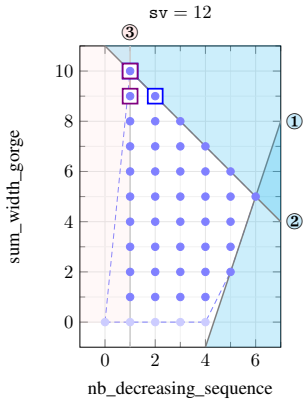


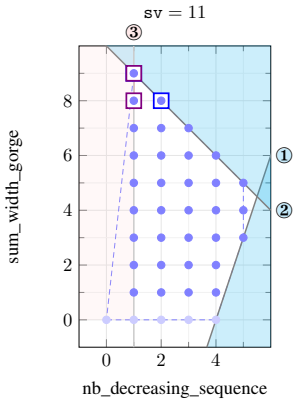
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv + 1$

② $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



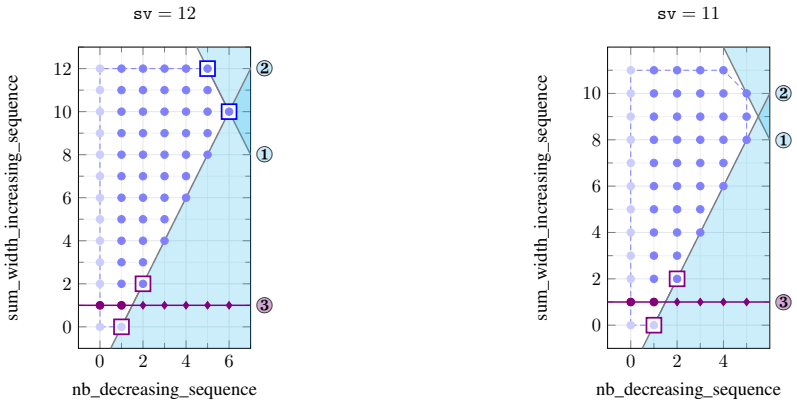


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $2 * x \leq y + 2$
- $sv \geq 4$: $(1, 0)$ $(2, 2)$

③ $y \neq 1$



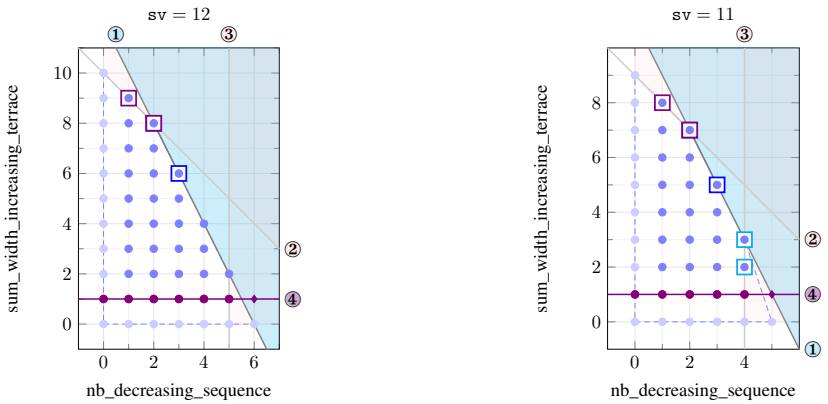
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \geq 8$: $(2, sv - 4)$ $(3, sv - 6)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(1, sv - 3)$ $(2, sv - 4)$

③ $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

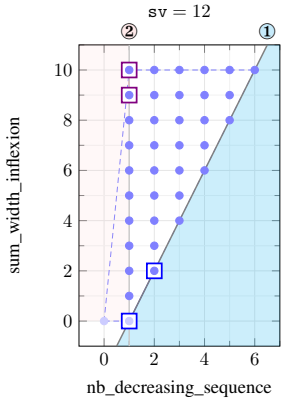
④ $y \neq 1$

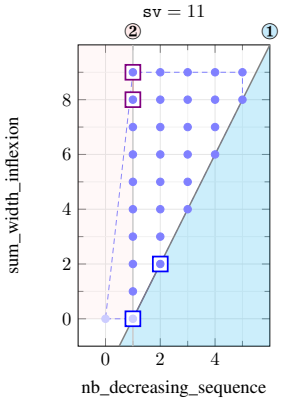


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

② $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



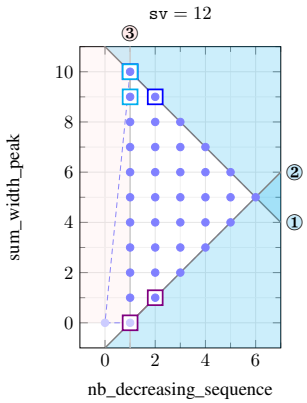


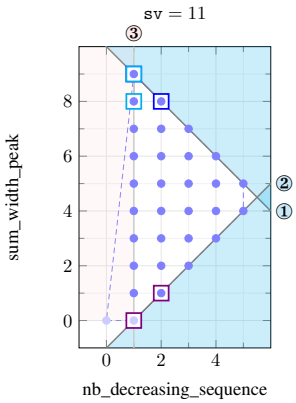
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 4: (1, sv − 2) (2, sv − 3)

② $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



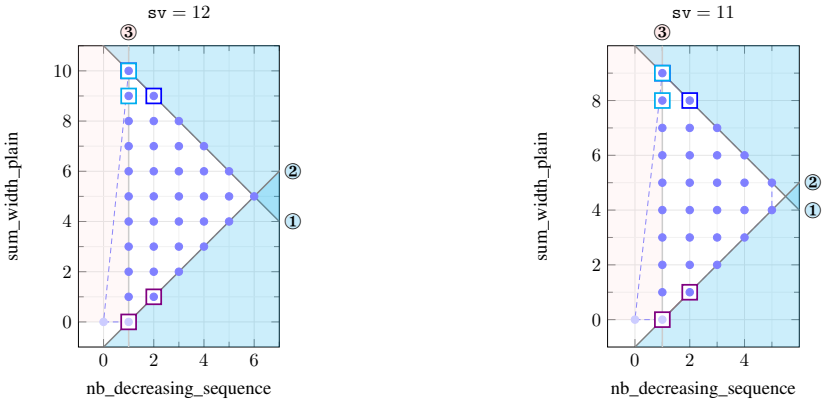


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

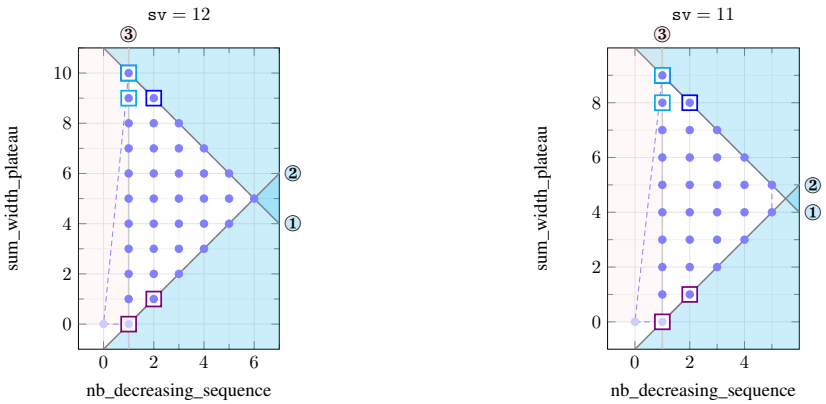


NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



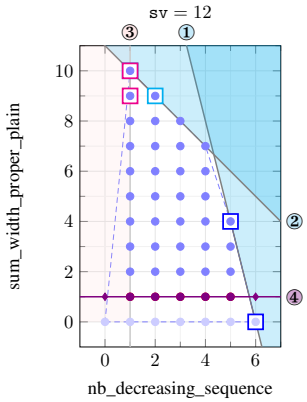
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

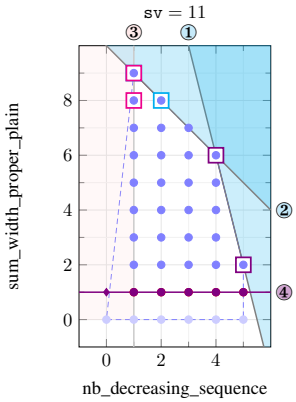
① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- sv ≥ 5: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





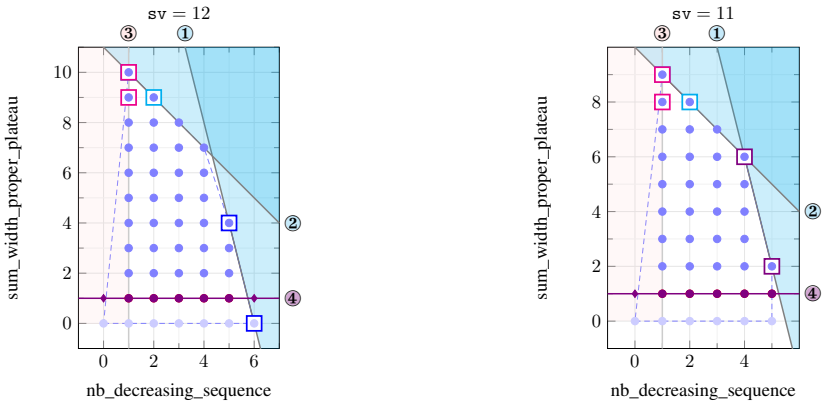
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- sv ≥ 5: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv - 2)$ $(1, sv - 3)$

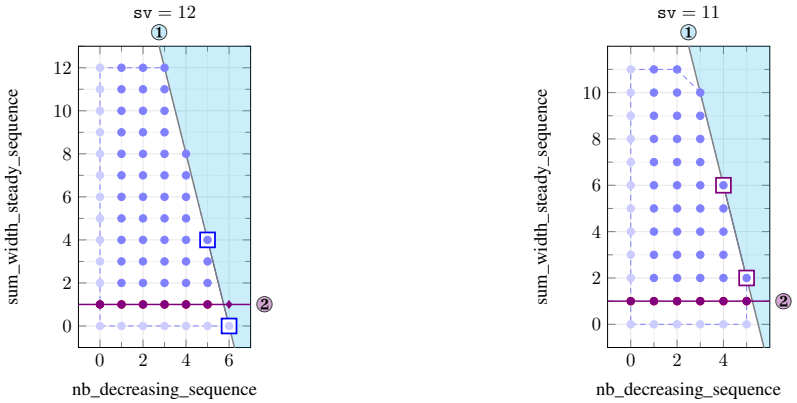
④ $y \neq 1$



NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 4: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 7: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $y \neq 1$

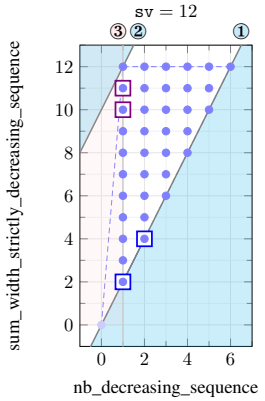


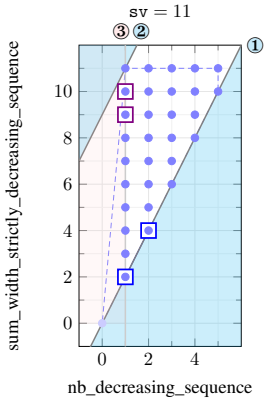
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 1) (1, sv − 2)



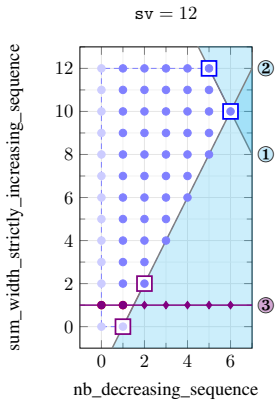


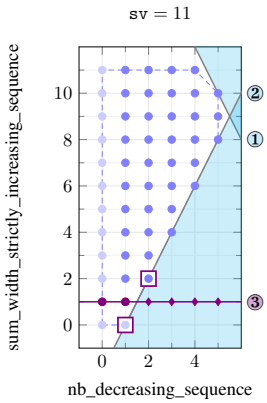
NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 2: (⌊sv/2⌋, sv − 2) (⌊sv/2⌋ − 1, sv)

② $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

③ $y \neq 1$





NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

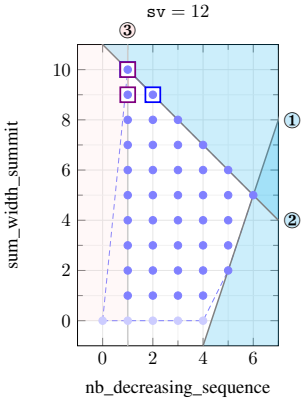
① $3 * x \leq y + sv + 1$

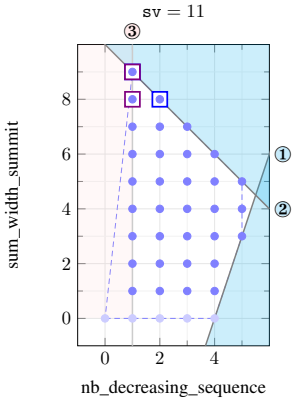
② $x + y \leq sv - 1$

- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

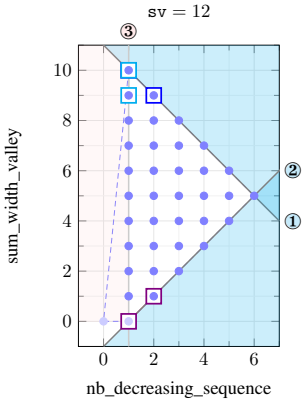
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

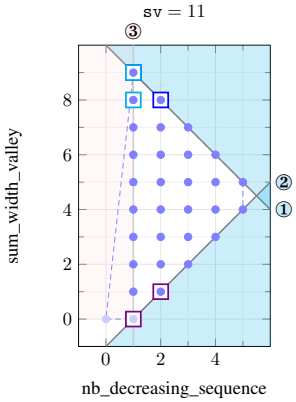
② $x \leq y + 1$

- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

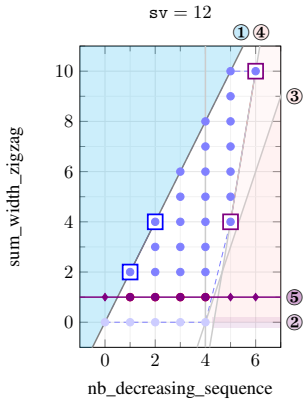
① $y \leq 2 * x$
- $sv \geq 6$: (1, 2) (2, 4)

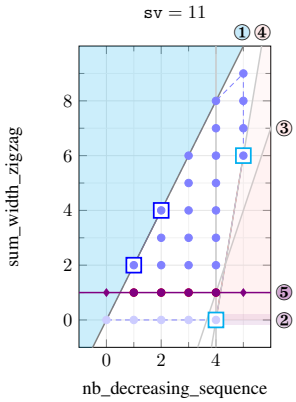
② $y = 0 \Rightarrow 3 * x \leq sv + 1 - (sv + 1) \bmod 3$

③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv + 2$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv - 8)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, sv - 5)$ $(\lfloor sv/2 \rfloor - 1, sv - 11)$

⑤ $y \neq 1$





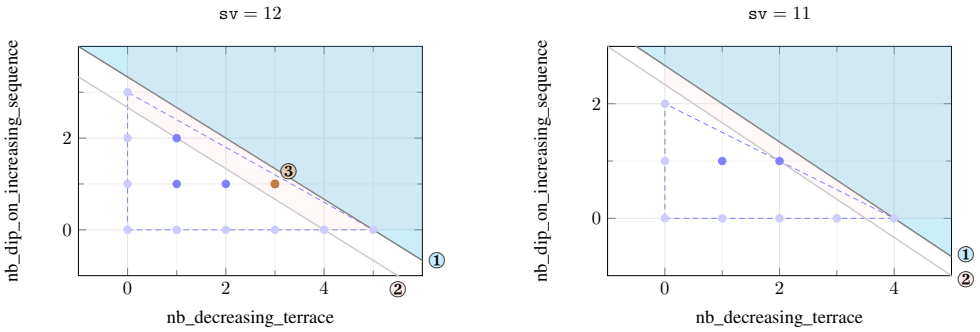
NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_DIP_ON_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + 3 * \mathtt{y} \leq \mathtt{sv} - 2 - (\mathtt{sv} - 2) \bmod 2$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + 3 * \mathtt{y} \leq \mathtt{sv} - 4$

- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 13$: $(\lfloor(\mathtt{sv}-2)/2\rfloor - 2, 1)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 5, 3)$

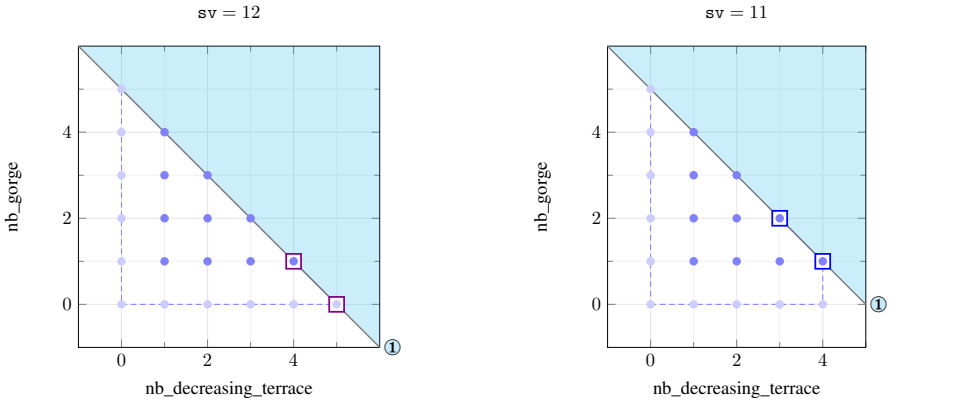
③ $$\bigvee \begin{pmatrix} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 2, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{max}(0, \lfloor(\mathtt{sv}-3)/3\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{pmatrix}$$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 2$

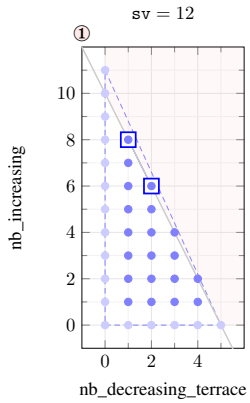
- $(\mathtt{sv} - 1) \bmod 2 = 0 \wedge \mathtt{sv} \geq 5$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 1)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, 2)$
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 0)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, 1)$

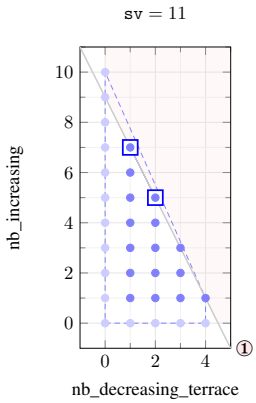


NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$

- $sv \geq 6$: $(1, sv - 4)$ $(2, sv - 6)$



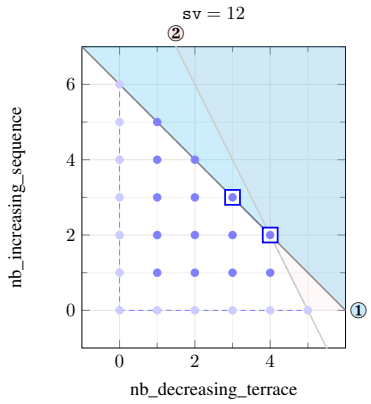


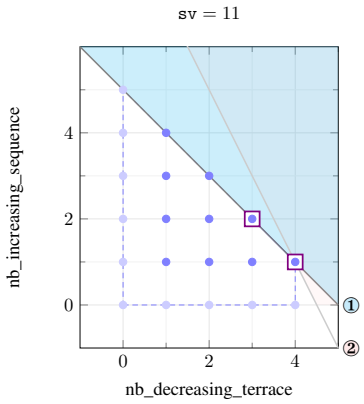
NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$

- $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 2, 3)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$





NB_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
NB_INCREASING_TERRACE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 2 - (\mathtt{sv} - 2) \bmod 2$
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 0)$ $(0, \lfloor(\mathtt{sv}-2)/2\rfloor)$
- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 5$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 0)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, 1)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$
- $(\mathtt{sv}-1) \bmod 2 = 0 \wedge \mathtt{sv} \geq 7$: $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, 1)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 2, 2)$
- $(\mathtt{sv}-1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 8$: $(\lfloor(\mathtt{sv}-2)/2\rfloor - 2, 1)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 3, 2)$

③ $\mathtt{o}^{=} \geq \mathtt{x} + \mathtt{y}$

④ $\bigvee \begin{cases} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 3, \\ \mathtt{y} < 3, \\ \mathtt{y} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{cases}$

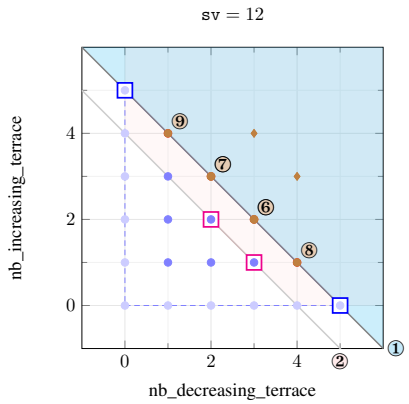
⑤ $\bigvee \begin{cases} \mathtt{y} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 3, \\ \mathtt{x} < 3, \\ \mathtt{x} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{cases}$

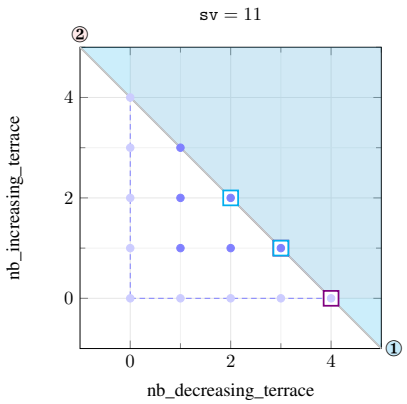
⑥ $\bigvee \begin{cases} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 2, \\ \mathtt{y} < 2, \\ \mathtt{y} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{cases}$

⑦ $\bigvee \begin{cases} \mathtt{y} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 2, \\ \mathtt{x} < 2, \\ \mathtt{x} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{x} \bmod 2 \end{cases}$

⑧ $\bigvee \begin{cases} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{cases}$

⑨ $\bigvee \begin{cases} \mathtt{y} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{x} < 1, \\ \mathtt{x} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{cases}$

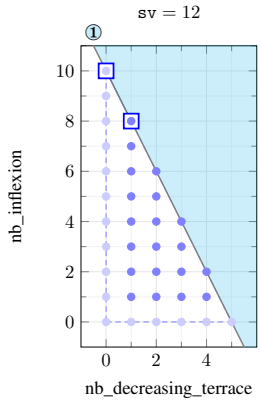


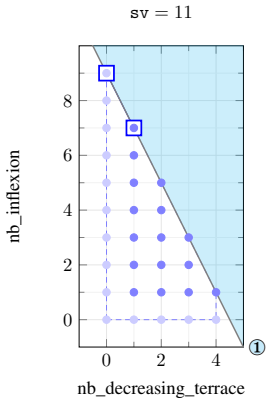


NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$

- $\mathtt{sv} \geq 4$: $(0, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 4)$

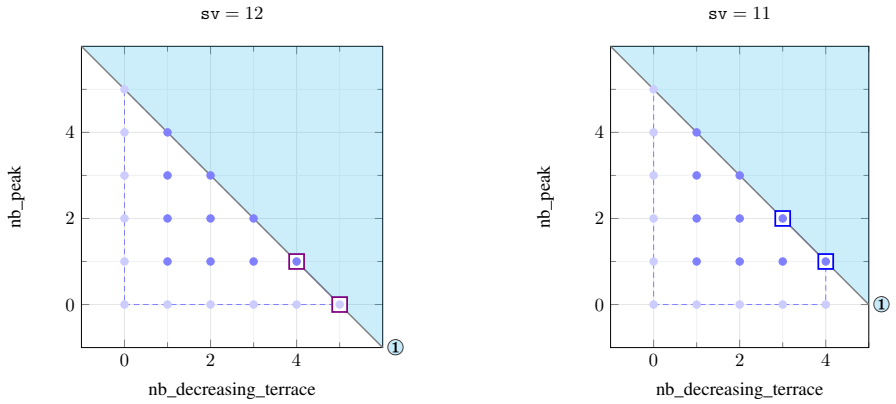




NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

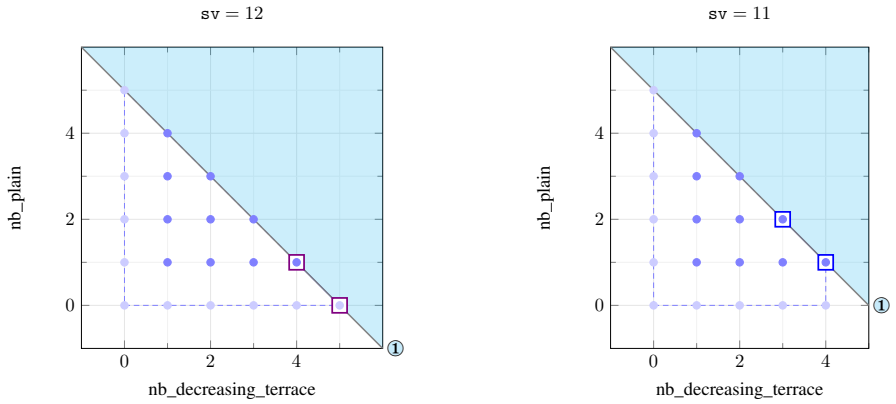
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

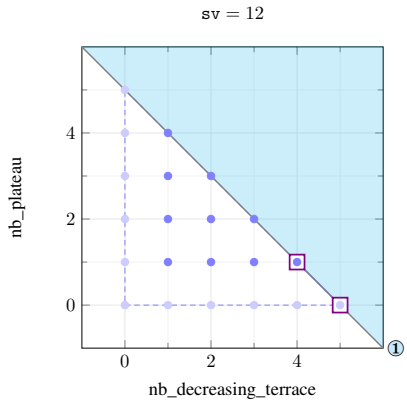
① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

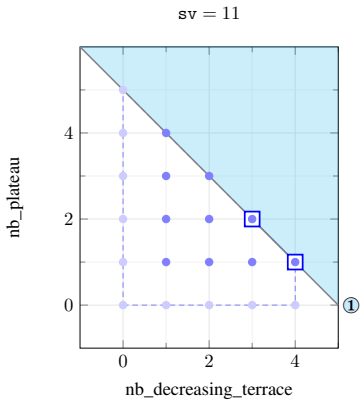
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$

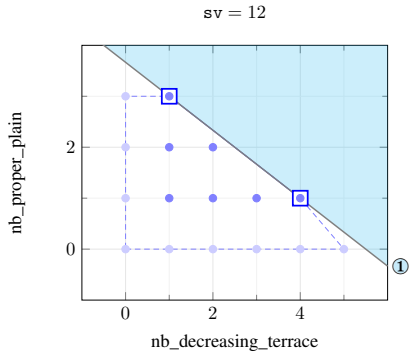


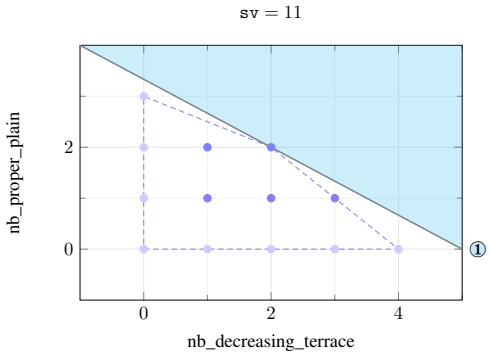


NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 3 * y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 10$: $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 4, 3)$

② $o^{=} \geq x + y$

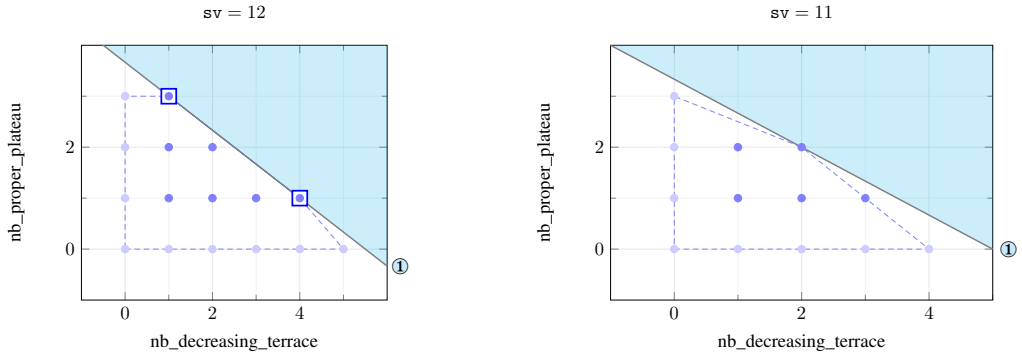




NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 3 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-2)/2 \rfloor - 1, 1)$ $(\lfloor (sv-2)/2 \rfloor - 4, 3)$

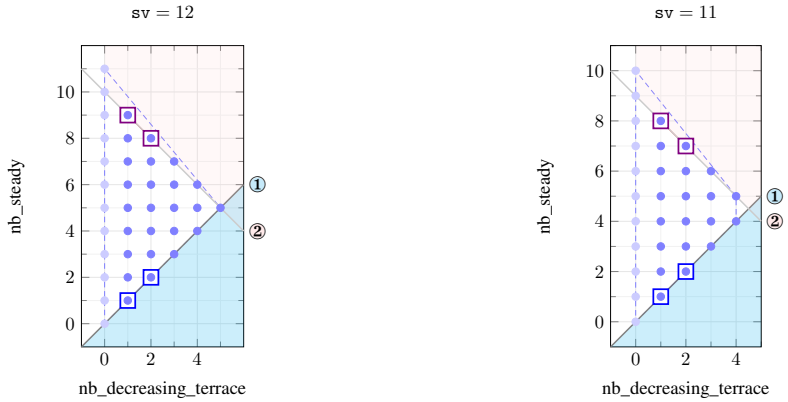
② $o^{=} \geq x + y$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (1, 1) (2, 2)

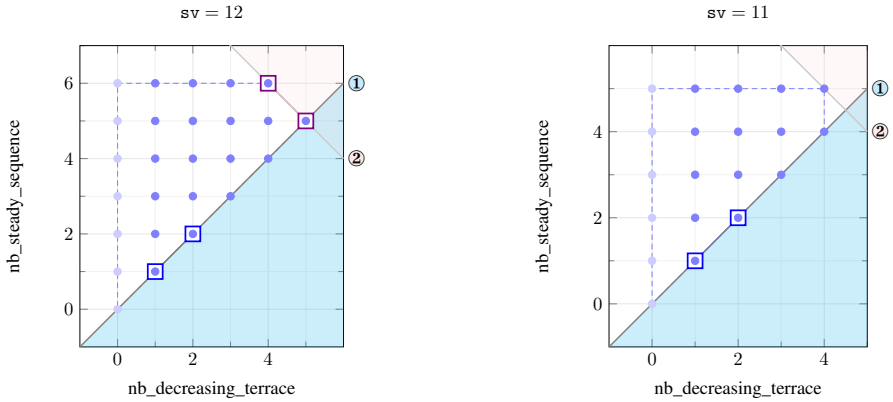
② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- sv ≥ 6: (1, sv − 3) (2, sv − 4)



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (1, 1) (2, 2)

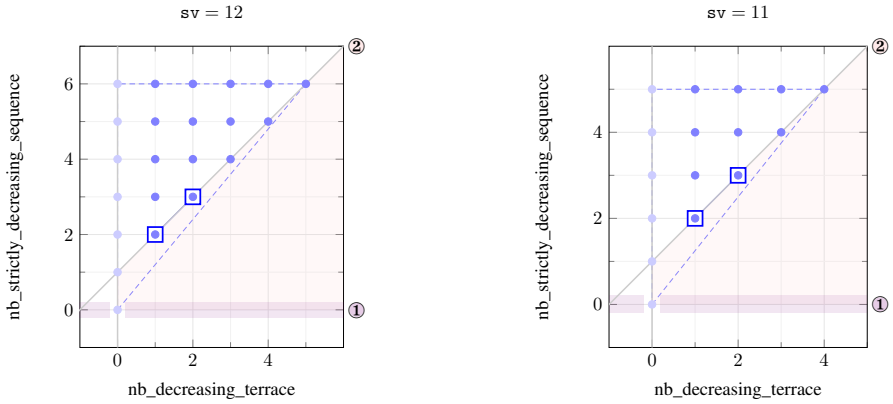
② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 4: $(\lfloor (sv-2)/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor (sv-2)/2 \rfloor - 1, \lfloor sv/2 \rfloor)$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

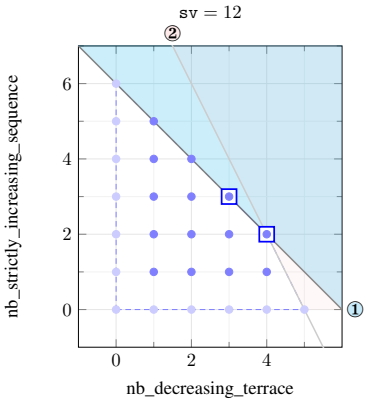
② $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- sv ≥ 6: (1, 2) (2, 3)

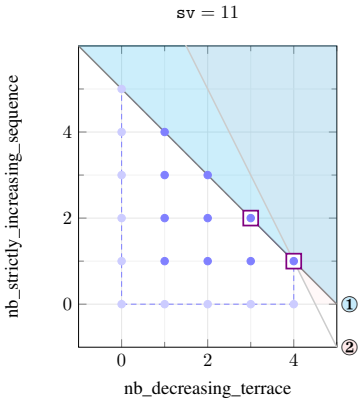


NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor (sv-2)/2 \rfloor - 1, 2)$ $(\lfloor (sv-2)/2 \rfloor - 2, 3)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv-2)/2 \rfloor, 1)$ $(\lfloor (sv-2)/2 \rfloor - 1, 2)$

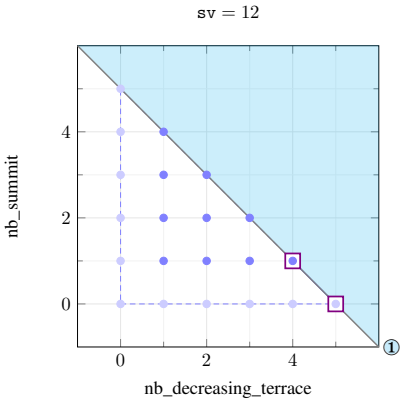
② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$

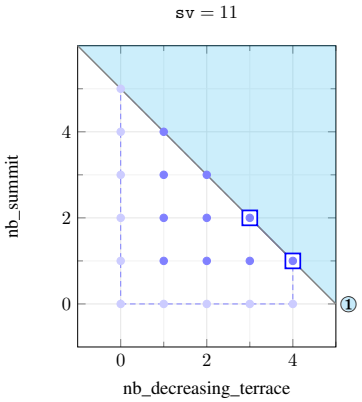




NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

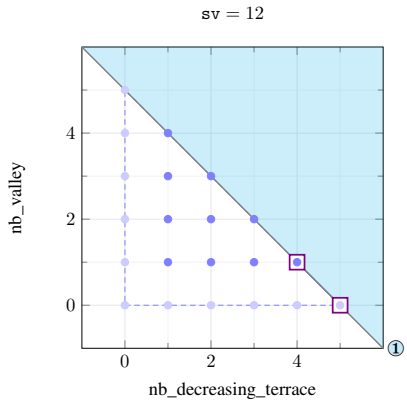
① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv-1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv-2)/2 \rfloor, 1)$ $(\lfloor (sv-2)/2 \rfloor - 1, 2)$
- $(sv-1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor, 0)$ $(\lfloor (sv-2)/2 \rfloor - 1, 1)$

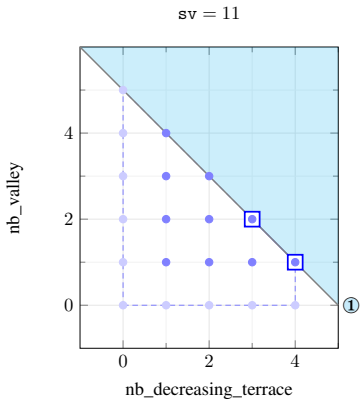




NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$

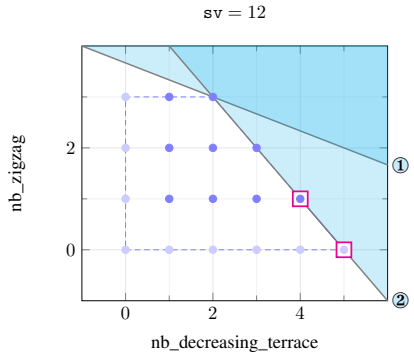


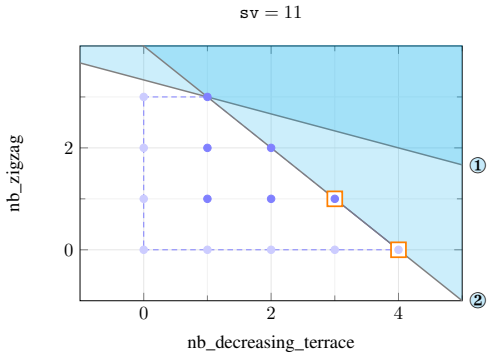


NB_DECREASING_TERRACE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + 3 * y \leq sv - 1$
- $sv \bmod 3 = 0 \wedge sv \geq 24$: $(2, \lfloor (sv - 1)/3 \rfloor)$ $(5, \lfloor (sv - 1)/3 \rfloor - 1)$
- $sv \bmod 3 = 1 \wedge sv \geq 16$: $(0, \lfloor (sv - 1)/3 \rfloor)$ $(3, \lfloor (sv - 1)/3 \rfloor - 1)$
- $sv \bmod 3 = 2 \wedge sv \geq 20$: $(1, \lfloor (sv - 1)/3 \rfloor)$ $(4, \lfloor (sv - 1)/3 \rfloor - 1)$

② $sv > 1 \Rightarrow 2 * x + 2 * y \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$





NB_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{y} = 0 \Rightarrow \mathtt{x} = 0$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} - 2$

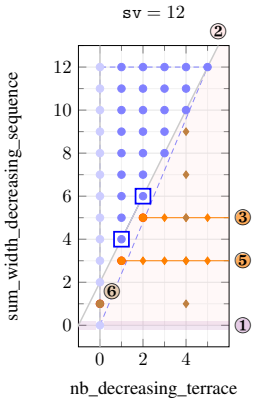
- $\mathtt{sv} \geq 6$: $(1, 4)$ $(2, 6)$

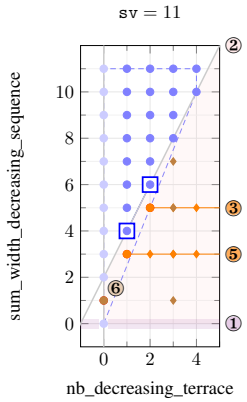
③ $\mathtt{x} < 2 \vee \mathtt{y} \neq 5$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor (\mathtt{sv} - 2)/2 \rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑤ $\mathtt{x} < 1 \vee \mathtt{y} \neq 3$

⑥ $\mathtt{x} \neq 0 \vee \mathtt{y} \neq 1$

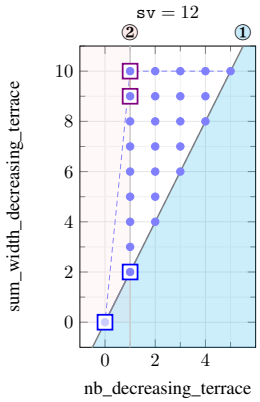


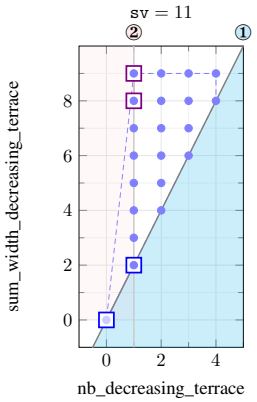


NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)





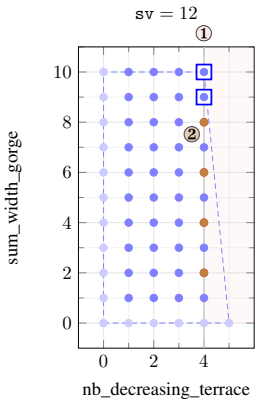
NB_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_GORGE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

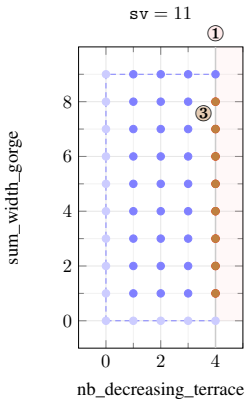
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

□ $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv}-2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\bigvee \left\{ \begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right.$

③ $\bigvee \left\{ \begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor), \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right.$





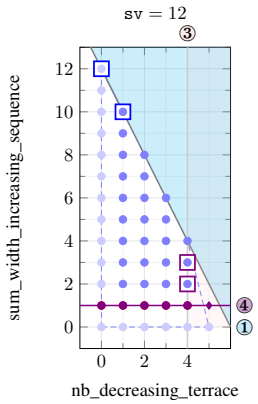
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

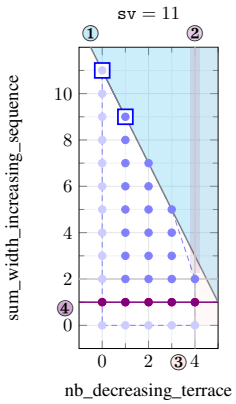
① $2 * x + y \leq sv$
- $sv \geq 6$: $(0, sv)$ $(1, sv - 2)$

② $x = \max(0, \lfloor (sv - 2)/2 \rfloor) \wedge sv \bmod 2 = 1 \wedge sv > 3 \Rightarrow y \leq 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

④ $y \neq 1$





NB_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INCREASING_TERRACE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 0)$ $(0, \mathtt{sv}-2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 9$: $(1, \mathtt{sv}-5)$ $(2, \mathtt{sv}-7)$

③ $\mathtt{o}^{=} \geq \mathtt{x} + \lfloor(\mathtt{y}+1)/2\rfloor$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \min(1, \max(0, \mathtt{sv}-3)) - 4, \\ \mathtt{sv} < 7 \end{array} \right)$

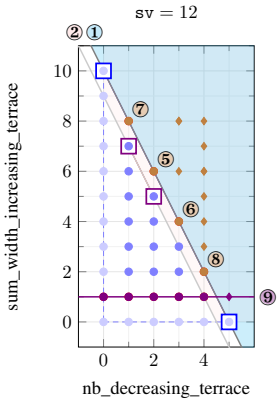
⑤ $\mathtt{x} \neq 2 \vee \mathtt{y} \neq (\mathtt{sv}-2) * \min(1, \max(0, \mathtt{sv}-3)) - 4 \vee \mathtt{sv} < 7$

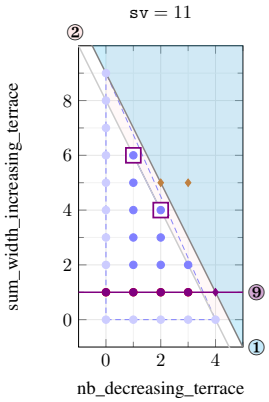
⑥ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 2, \\ \mathtt{y} < 3, \\ \mathtt{y} > (\mathtt{sv}-2) * \min(1, \max(0, \mathtt{sv}-3)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

⑦ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \min(1, \max(0, \mathtt{sv}-3)) - 2, \\ \mathtt{x} < 1, \\ \mathtt{x} > \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑧ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \max(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \min(1, \max(0, \mathtt{sv}-3)) - 2, \\ 1 = \mathtt{sv} \bmod 2 \end{array} \right)$

⑨ $\mathtt{y} \neq 1$





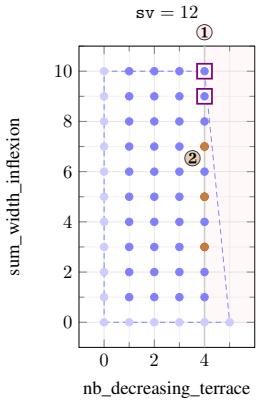
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

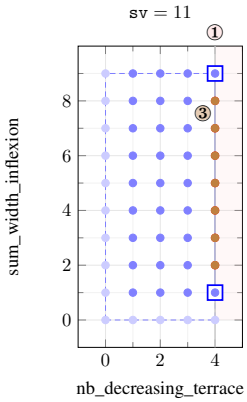
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

- □ $(sv - 1) \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor, sv - 2)$ $(\lfloor (sv-2)/2 \rfloor, 1)$
- □ $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor - 1, sv - 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv - 3)$

② $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor) - 1, \\ y < 3, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

③ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor), \\ y < 2, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1, \\ 0 = sv \bmod 2 \end{array} \right)$





NB_DECREASING_TERRACE($\mathtt{x}, \mathtt{VARIABLES}$) $\wedge$
SUM_WIDTH_PEAK($\mathtt{y}, \mathtt{VARIABLES}$) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

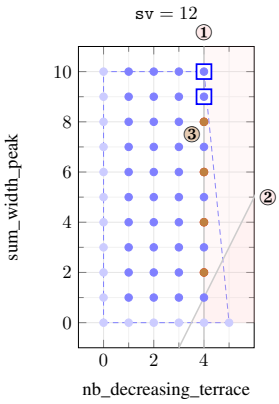
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

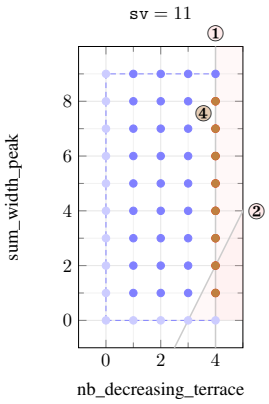
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 5$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor), \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right)$

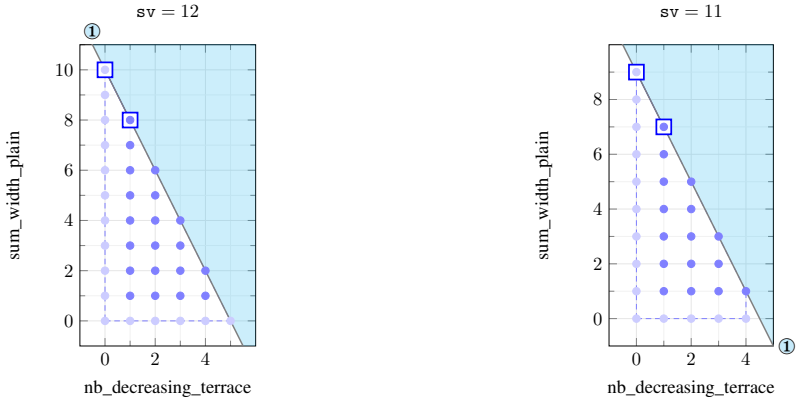




NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$

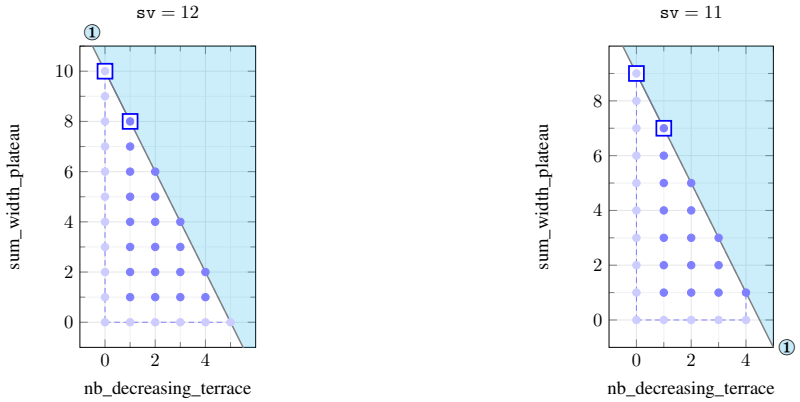
- $\mathtt{sv} \geq 4$: $(0, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 4)$



NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$

- $\mathtt{sv} \geq 4$: $(0, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 4)$



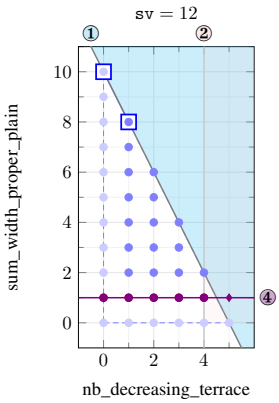
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

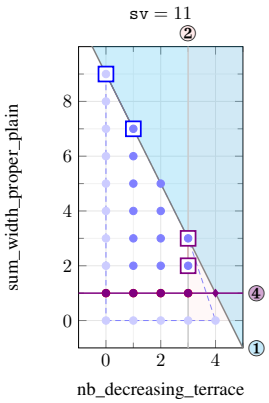
① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

④ $y \neq 1$





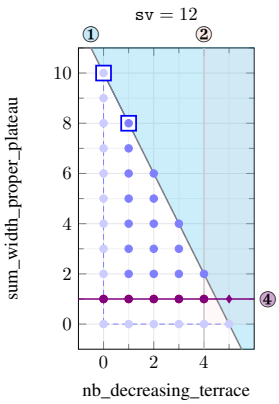
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

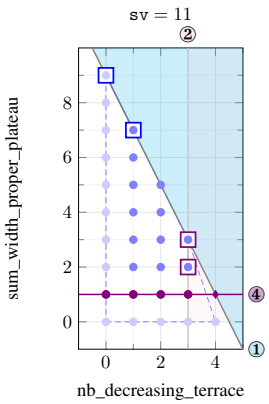
① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

④ $y \neq 1$



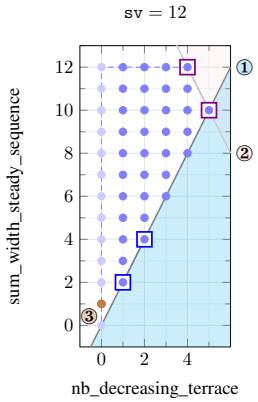


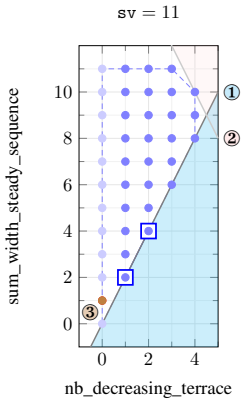
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 6$: (1, 2) (2, 4)

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor, sv-2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv)$

③ $x \neq 0 \vee y \neq 1$





NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y - 2$

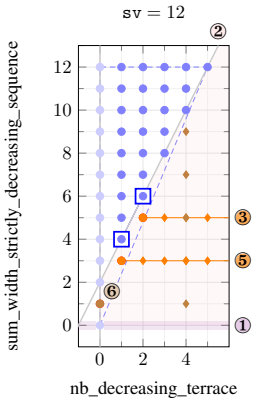
- $sv \geq 6$: (1, 4) (2, 6)

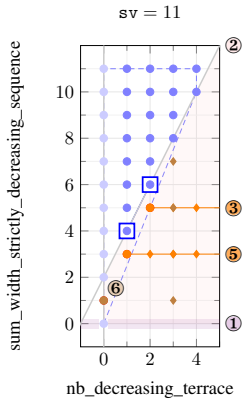
③ $x < 2 \vee y \neq 5$

④ $\bigvee \begin{pmatrix} x \neq \max(0, \lfloor (sv - 2)/2 \rfloor) - 1, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{pmatrix}$

⑤ $x < 1 \vee y \neq 3$

⑥ $x \neq 0 \vee y \neq 1$





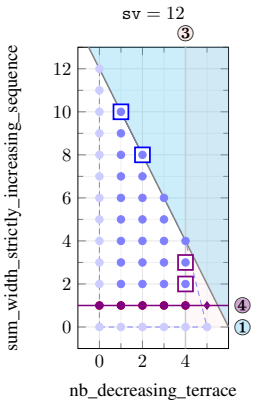
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

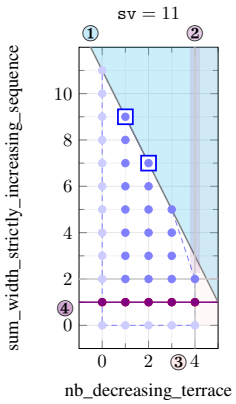
① $2 * x + y \leq sv$
- $sv \geq 8$: $(1, sv - 2)$ $(2, sv - 4)$

② $x = \max(0, \lfloor (sv - 2)/2 \rfloor) \wedge sv \bmod 2 = 1 \wedge sv > 3 \Rightarrow y \leq 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

④ $y \neq 1$





NB_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_SUMMIT($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

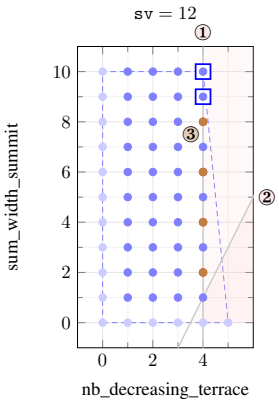
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

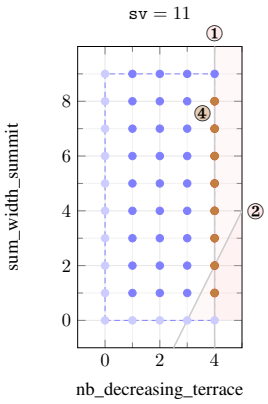
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 5$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor), \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right)$





NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

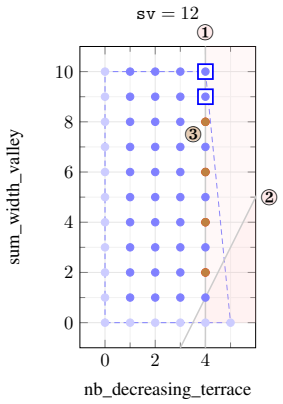
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

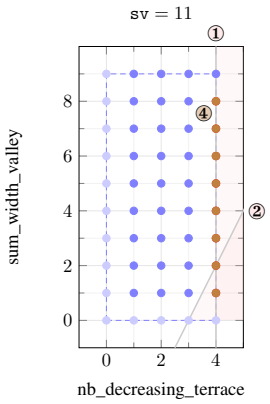
□ $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor - 1, sv - 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv - 3)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 5$

③ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor) - 1, \\ y < 2, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ 1 = sv \bmod 2, \\ 1 = y \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1, \\ 0 = sv \bmod 2 \end{array} \right)$



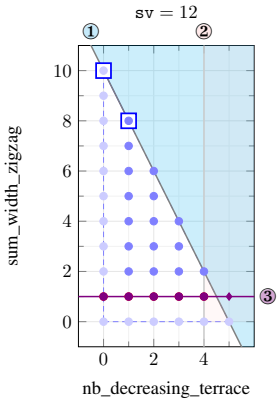


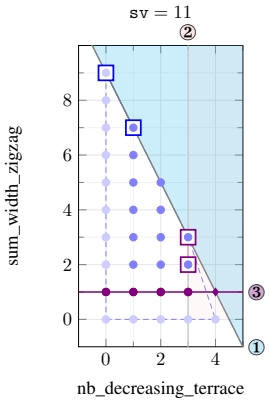
NB_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

③ $y \neq 1$

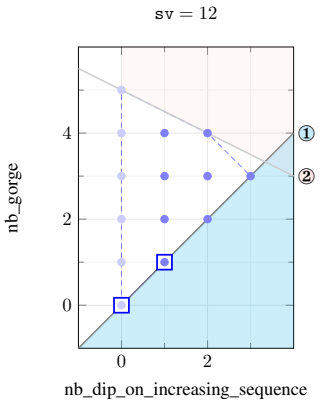


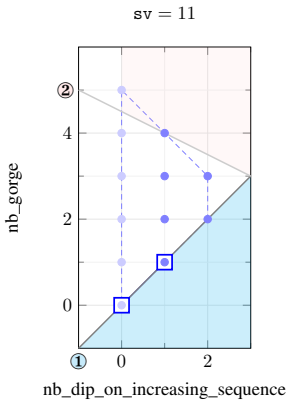


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 1)$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv - 1)/2 \rfloor - 1)$ $(4, \lfloor (sv - 1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv - 1)/2 \rfloor - 1)$ $(3, \lfloor (sv - 1)/2 \rfloor - 2)$





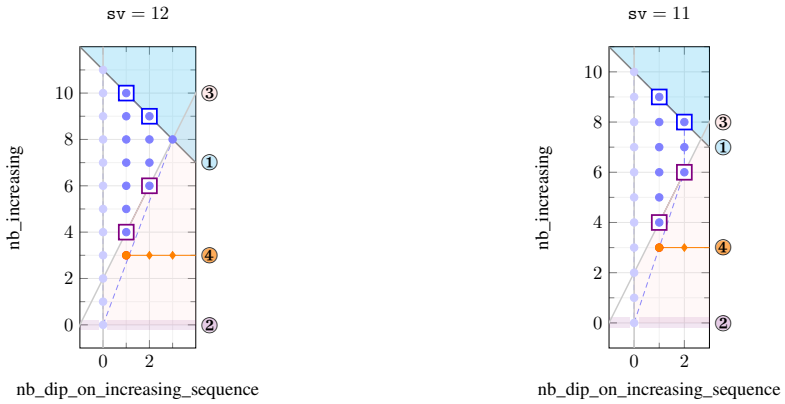
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 9$: $(1, sv - 2)$ $(2, sv - 3)$

② $y = 0 \Rightarrow x = 0$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y - 2$
- $sv \geq 9$: $(1, 4)$ $(2, 6)$

④ $x < 1 \vee y \neq 3$

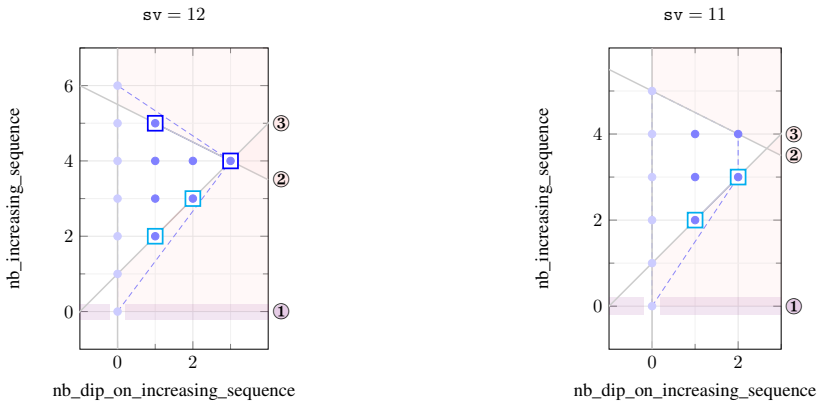


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor sv/2 \rfloor - 1)$ $(3, \lfloor sv/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 15$: $(2, \lfloor sv/2 \rfloor - 1)$ $(4, \lfloor sv/2 \rfloor - 2)$

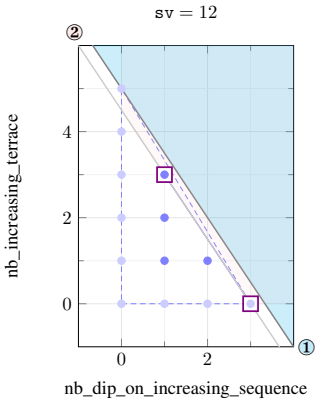
③ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 9$: $(1, 2)$ $(2, 3)$

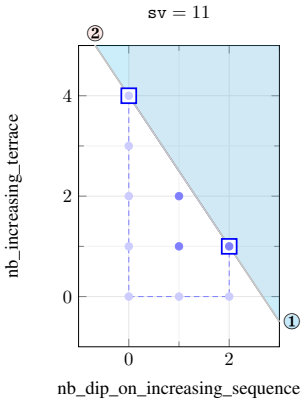


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 3 * x + 2 * y \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 9$: $(0, \lfloor (sv-2)/2 \rfloor)$ $(2, \lfloor (sv-2)/2 \rfloor - 3)$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + 2 * y \leq sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor (sv-2)/2 \rfloor - 2)$ $(3, \lfloor (sv-2)/2 \rfloor - 5)$



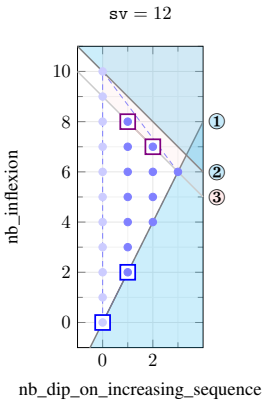


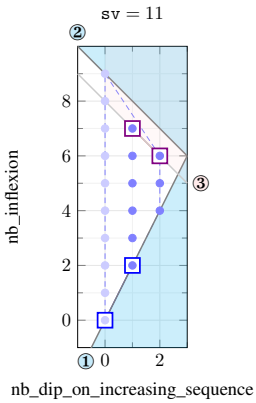
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 2)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$
- $sv \geq 9$: $(1, sv - 4)$ $(2, sv - 5)$

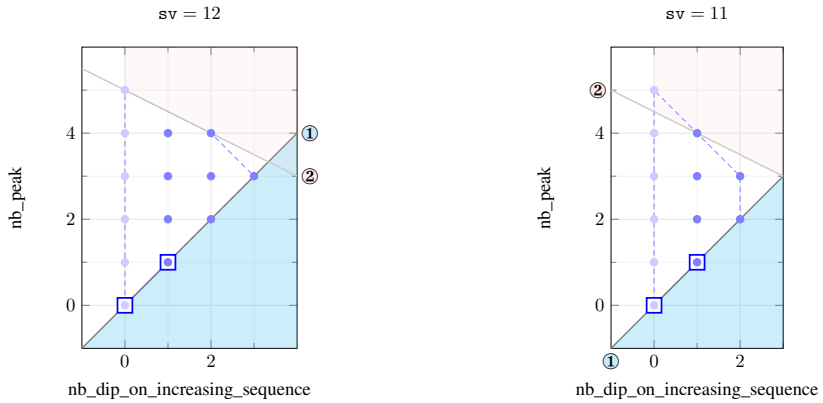




NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

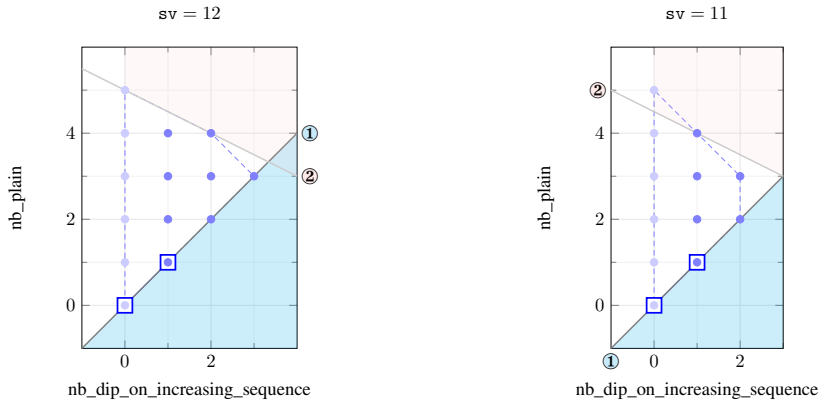
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod $2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- sv mod $2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

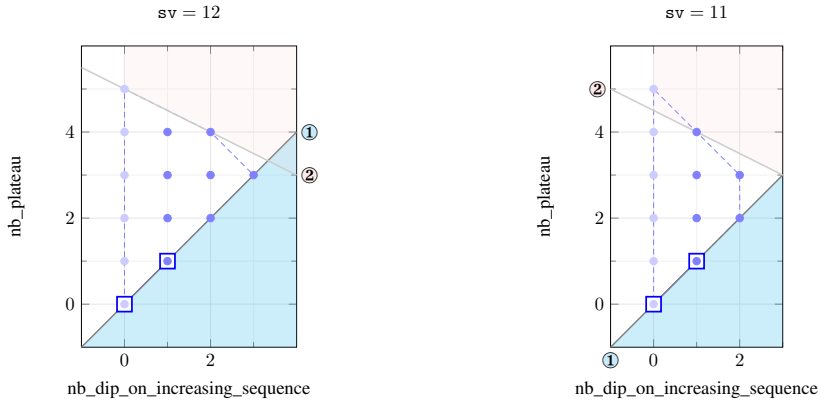
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod $2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$
- sv mod $2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 2)$



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

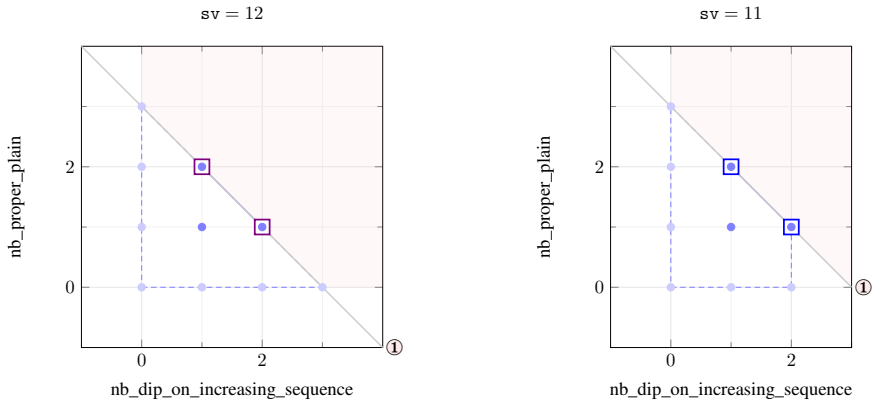
① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 16: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)
- sv mod 2 = 1 ∧ sv ≥ 13: (1, ⌊(sv − 1)/2⌋ − 1) (3, ⌊(sv − 1)/2⌋ − 2)



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

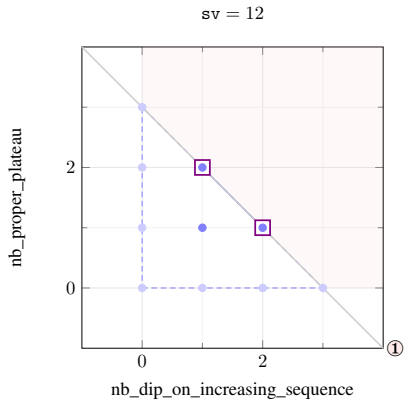
① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq sv - 2 - (sv - 2) \bmod 3$
- (sv − 2) mod 3 = 0 ∧ sv ≥ 8: (⌊(sv − 3)/3⌋, 1) (⌊(sv − 3)/3⌋ − 1, 2)
- (sv − 2) mod 3 = 1 ∧ sv ≥ 9: (⌊(sv − 3)/3⌋ − 1, 1) (⌊(sv − 3)/3⌋ − 2, 2)
- (sv − 2) mod 3 = 2 ∧ sv ≥ 10: (⌊(sv − 3)/3⌋ − 1, 1) (⌊(sv − 3)/3⌋ − 2, 2)

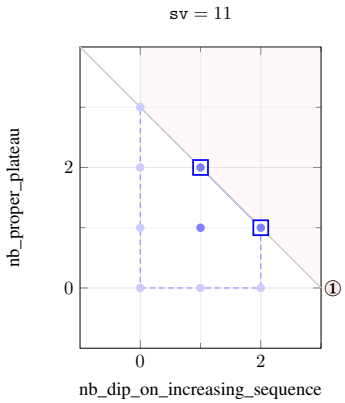


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 0 \wedge sv \geq 8$: $(\lfloor (sv-3)/3 \rfloor, 1)$ $(\lfloor (sv-3)/3 \rfloor - 1, 2)$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv-3)/3 \rfloor - 1, 1)$ $(\lfloor (sv-3)/3 \rfloor - 2, 2)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 10$: $(\lfloor (sv-3)/3 \rfloor - 1, 1)$ $(\lfloor (sv-3)/3 \rfloor - 2, 2)$



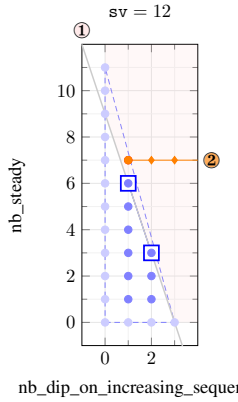


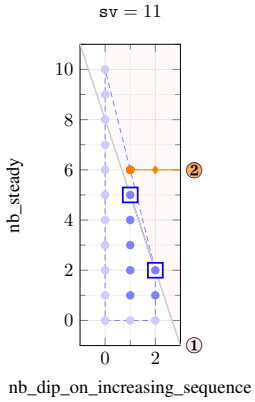
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 9$: $(1, sv - 6)$ $(2, sv - 9)$

② $x < 1 \vee y \neq \max(0, sv - 1) - 4$

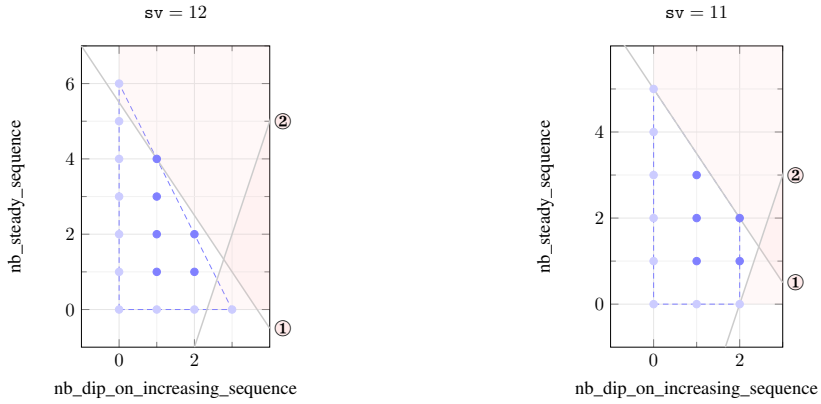




NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (1, ⌊sv/2⌋ − 2) (3, ⌊sv/2⌋ − 5)

② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 5$

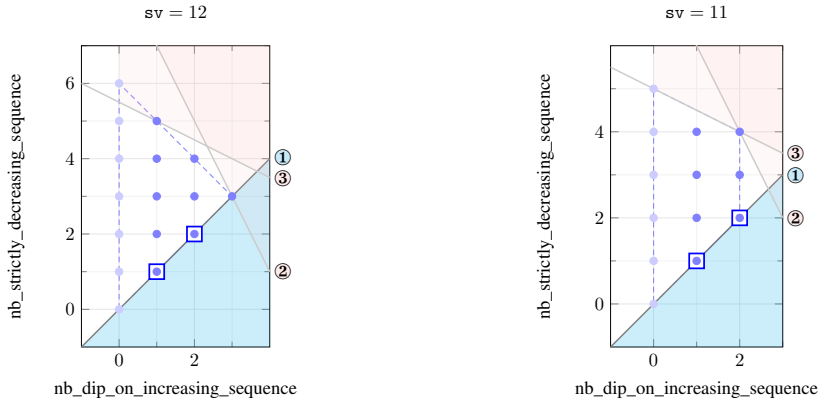


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 9: (1, 1) (2, 2)

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$

③ $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (1, ⌊sv/2⌋ − 1) (3, ⌊sv/2⌋ − 2)
- sv mod 2 = 1 ∧ sv ≥ 17: (2, ⌊sv/2⌋ − 1) (4, ⌊sv/2⌋ − 2)

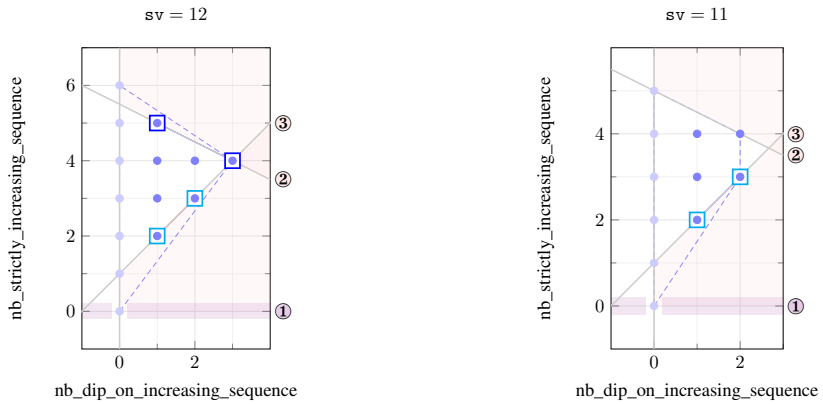


NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor sv/2 \rfloor - 1)$ $(3, \lfloor sv/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 15$: $(2, \lfloor sv/2 \rfloor - 1)$ $(4, \lfloor sv/2 \rfloor - 2)$

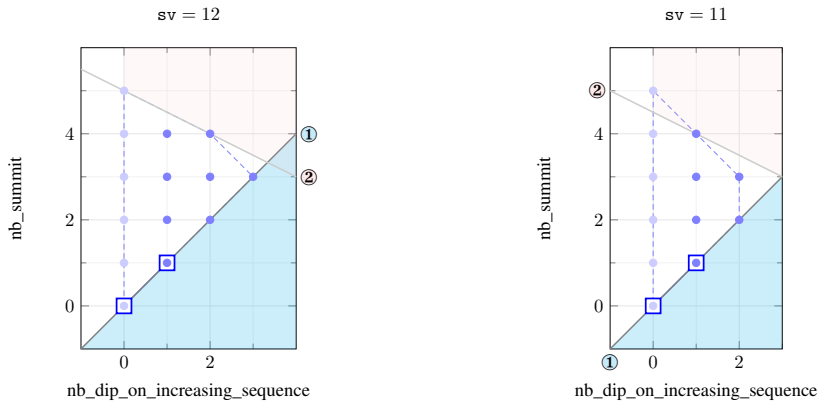
③ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 9$: $(1, 2)$ $(2, 3)$



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 6$: $(0, 0)$ $(1, 1)$

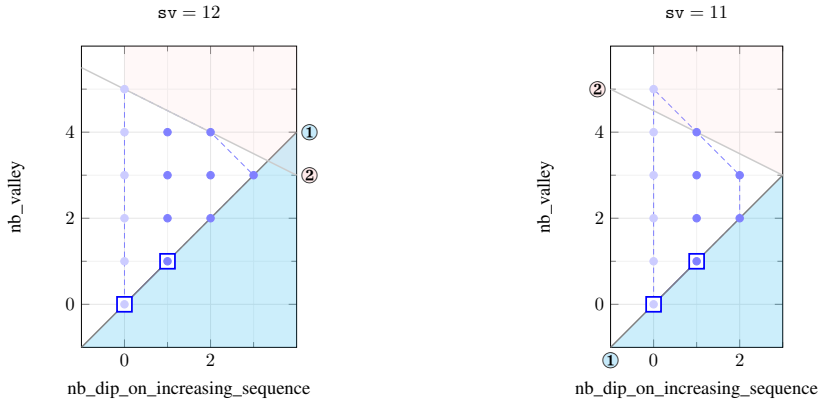
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 16$: $(2, \lfloor (sv - 1)/2 \rfloor - 1)$ $(4, \lfloor (sv - 1)/2 \rfloor - 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(1, \lfloor (sv - 1)/2 \rfloor - 1)$ $(3, \lfloor (sv - 1)/2 \rfloor - 2)$



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

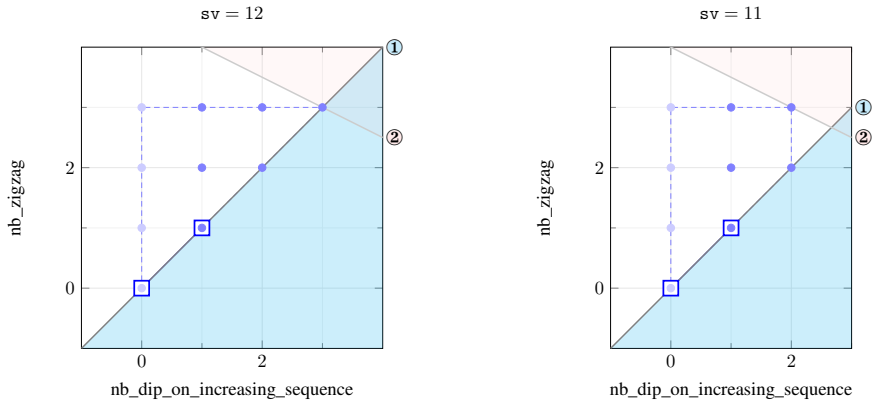
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 16: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)
- sv mod 2 = 1 ∧ sv ≥ 13: (1, ⌊(sv − 1)/2⌋ − 1) (3, ⌊(sv − 1)/2⌋ − 2)



NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq sv - 3$



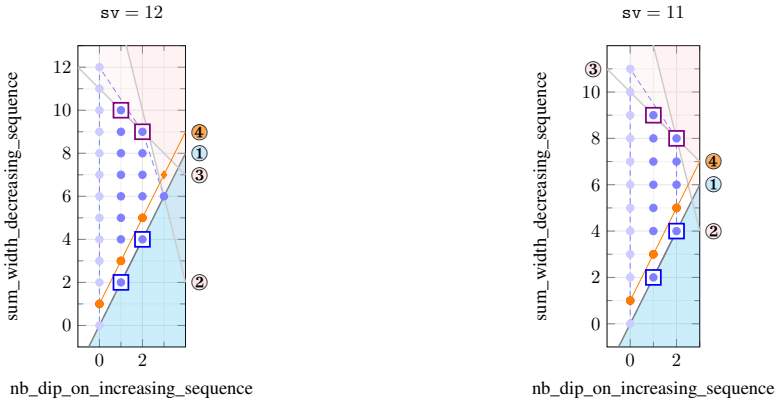
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 9: (1, 2) (2, 4)

② $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 1$
- sv ≥ 11: (1, sv − 2) (2, sv − 3)

④ $y \neq 2 * x + 1$



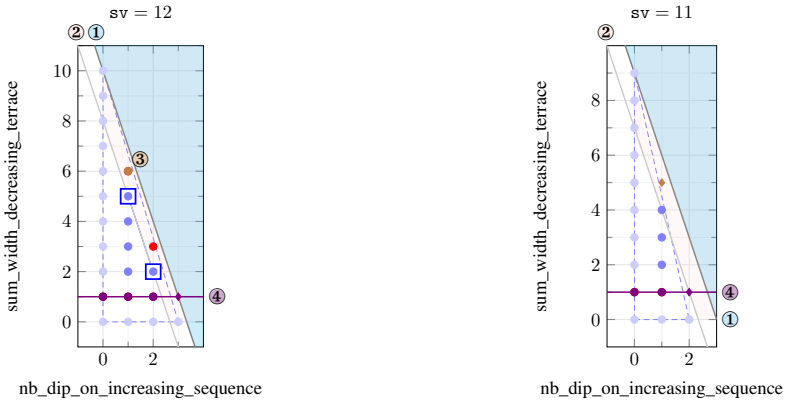
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 4$
- sv ≥ 12: (1, sv − 7) (2, sv − 10)

③ $x \neq 1 \vee y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 4 \vee sv < 7$

④ $y \neq 1$

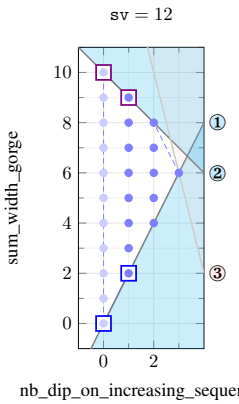


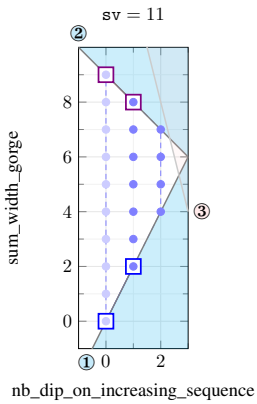
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 7: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$





NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

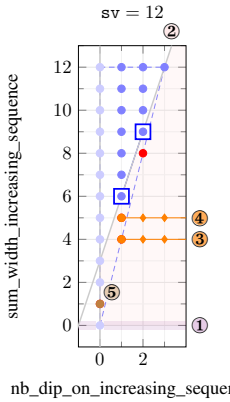
① $y = 0 \Rightarrow x = 0$

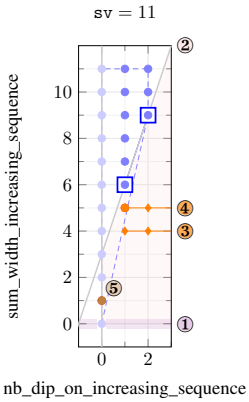
② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y - 3$
- sv ≥ 9: (1, 6) (2, 9)

③ $x < 1 \vee y \neq 4$

④ $x < 1 \vee y \neq 5$

⑤ $x \neq 0 \vee y \neq 1$





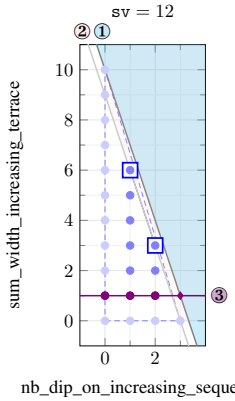
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

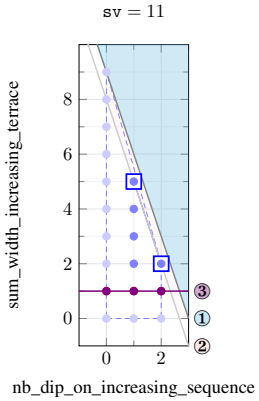
① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 11$: $(1, sv - 6)$ $(2, sv - 9)$

③ $y \neq 1$





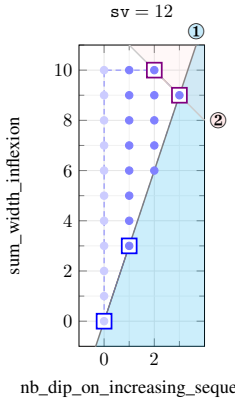
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

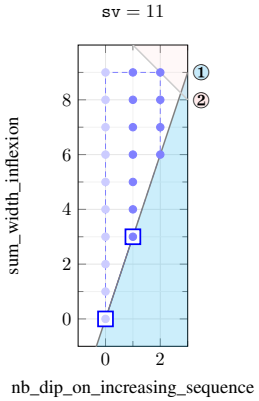
① $3 * x \leq y$

- $sv \geq 6$: $(0, 0)$ $(1, 3)$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + 3 * y \leq 4 * sv - 10 - (4 * sv - 10) \bmod 3$

- $(sv - 1) \bmod 3 = 2 \wedge sv \geq 6$: $(\lfloor (sv - 3)/3 \rfloor, sv - 3)$ $(\lfloor (sv - 3)/3 \rfloor - 1, sv - 2)$



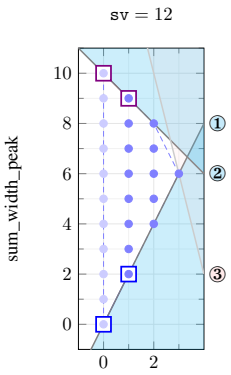


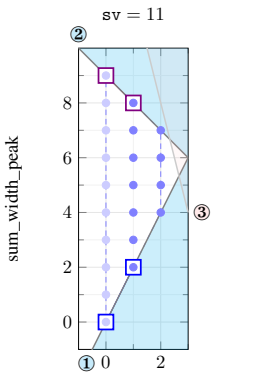
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 7: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$



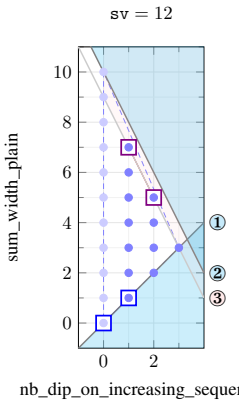


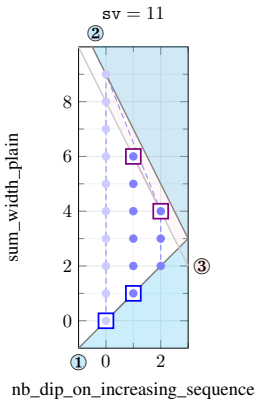
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$
- sv ≥ 9: (1, sv − 5) (2, sv − 7)



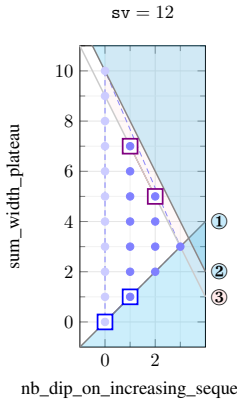


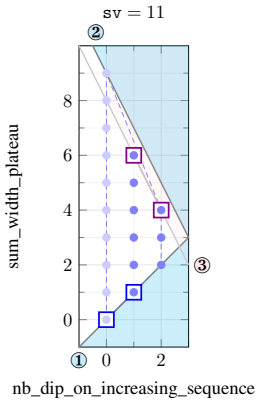
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (0, 0) (1, 1)

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 3$
- sv ≥ 9: (1, sv − 5) (2, sv − 7)



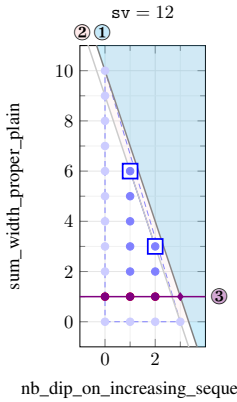


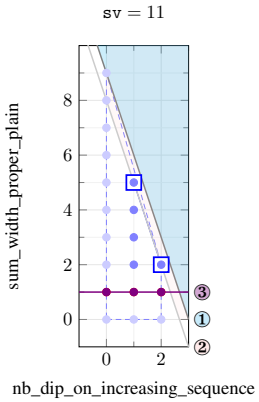
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$
- sv ≥ 11: (1, sv − 6) (2, sv − 9)

③ $y \neq 1$





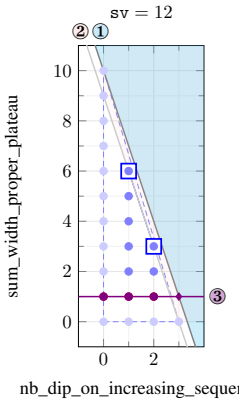
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

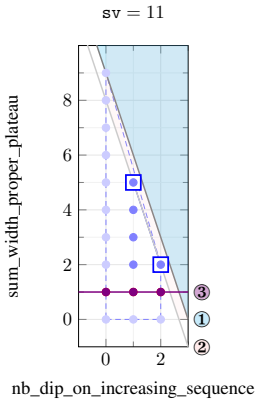
① $sv > 1 \Rightarrow 3 * x + y \leq sv - 2$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 3$

- $sv \geq 11$: $(1, sv - 6)$ $(2, sv - 9)$

③ $y \neq 1$





NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

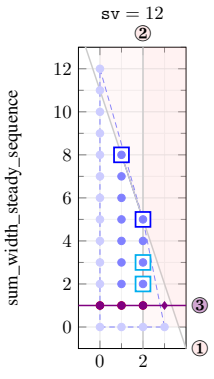
① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 1$

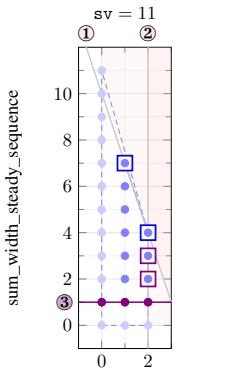
- $sv \geq 11$: $(1, sv - 4)$ $(2, sv - 7)$

② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 4 - (sv - 4) \bmod 3$

- $(sv - 1) \bmod 3 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 3)/3 \rfloor, 2)$ $(\lfloor (sv - 3)/3 \rfloor, 3)$
- $(sv - 1) \bmod 3 = 2 \wedge sv \geq 6$: $(\lfloor (sv - 3)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 3)/3 \rfloor - 1, 3)$

③ $y \neq 1$





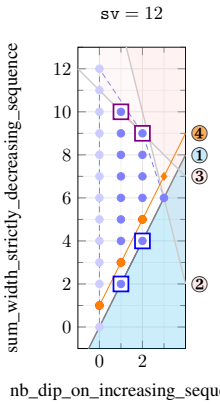
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

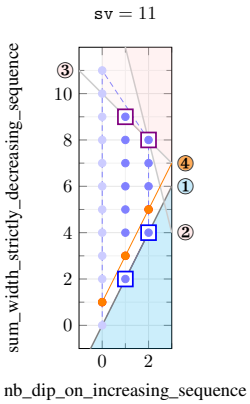
① $2 * x \leq y$
- sv ≥ 9: (1, 2) (2, 4)

② $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 1$
- sv ≥ 11: (1, sv − 2) (2, sv − 3)

④ $y \neq 2 * x + 1$





NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

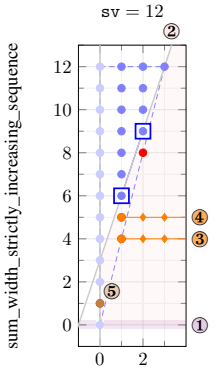
② $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y - 3$

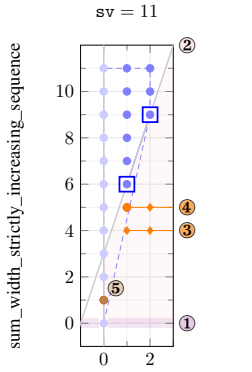
- $sv \geq 9$: (1, 6) (2, 9)

③ $x < 1 \vee y \neq 4$

④ $x < 1 \vee y \neq 5$

⑤ $x \neq 0 \vee y \neq 1$





NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

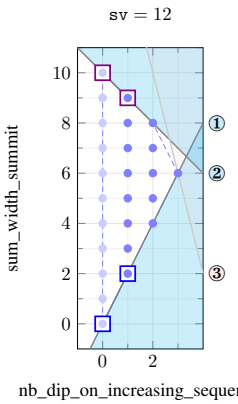
① $2 * x \leq y$

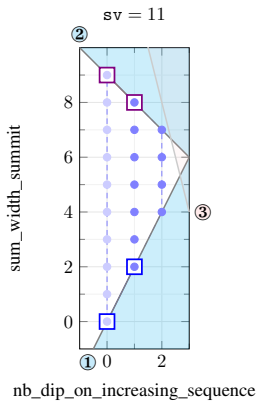
- $sv \geq 6$: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 7$: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$



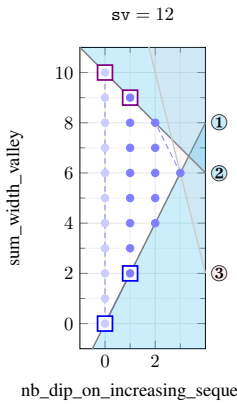


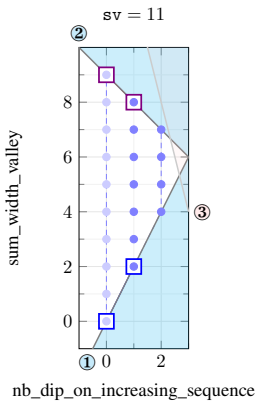
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 7: (0, sv − 2) (1, sv − 3)

③ $x > 0 \wedge y > 0 \Rightarrow 4 * x + y \leq 2 * sv - 6$





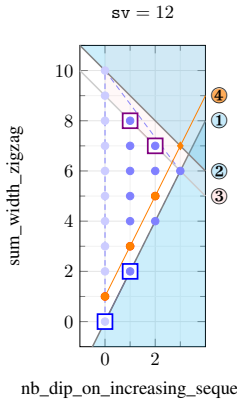
NB_DIP_ON_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

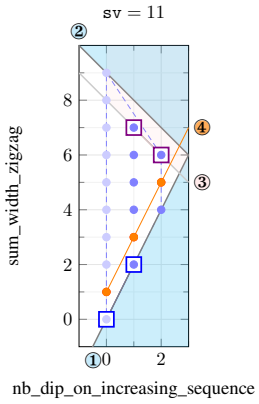
① $2 * x \leq y$
- sv ≥ 6: (0, 0) (1, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$
- sv ≥ 11: (1, sv − 4) (2, sv − 5)

④ $y \neq 2 * x + 1$

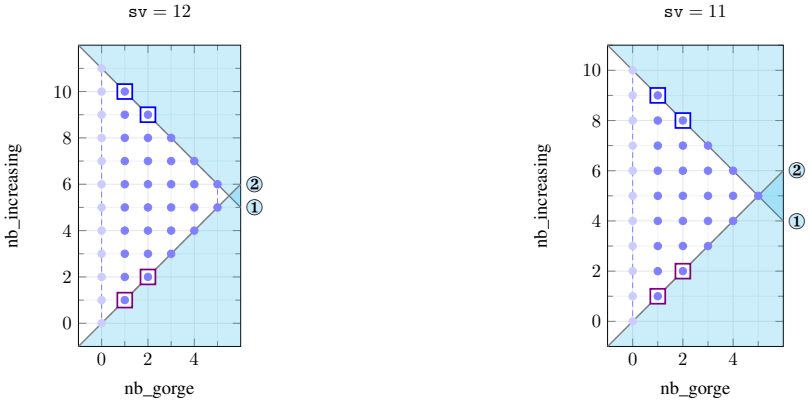




NB_GORGE(x, VARIABLES) ∧
NB_INCREASING(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

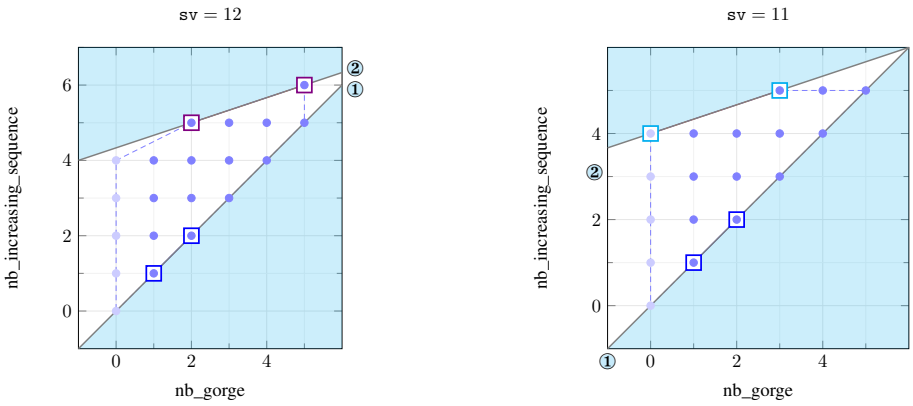
② $x \leq y$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$



NB_GORGE(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

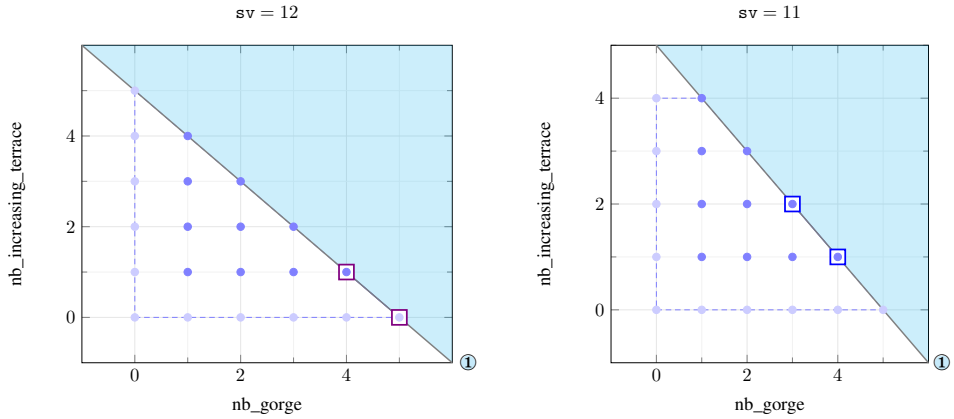
① $x \leq y$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

② $3 * y \leq x + sv + 1$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor (sv - 1)/2 \rfloor, \lfloor sv/2 \rfloor)$ $(\lfloor (sv - 1)/2 \rfloor - 3, \lfloor sv/2 \rfloor - 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor (sv - 1)/2 \rfloor - 2, \lfloor sv/2 \rfloor)$ $(\lfloor (sv - 1)/2 \rfloor - 5, \lfloor sv/2 \rfloor - 1)$



NB_GORGE(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

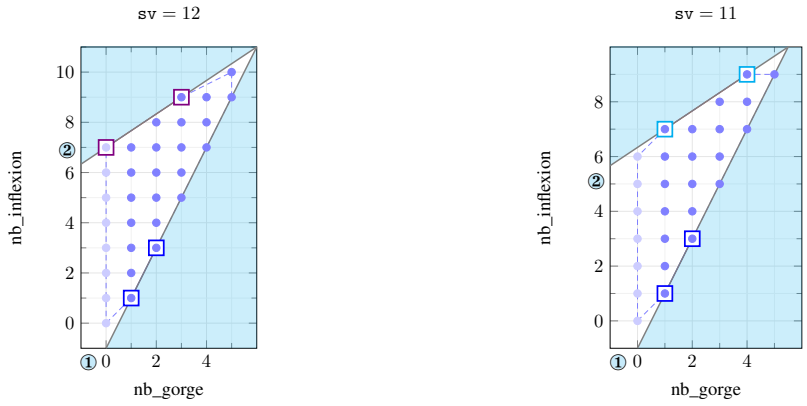
① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 7$: $(\lfloor (sv-1)/2 \rfloor - 1, 1)$ $(\lfloor (sv-1)/2 \rfloor - 2, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-1)/2 \rfloor, 0)$ $(\lfloor (sv-1)/2 \rfloor - 1, 1)$



NB_GORGE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- $sv \geq 5$: $(1, 1)$ $(2, 3)$

② $sv > 1 \Rightarrow 3 * y \leq 2 * x + 2 * sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(\lfloor (sv-1)/2 \rfloor - 2, sv - 3)$ $(\lfloor (sv-1)/2 \rfloor - 5, sv - 5)$
- $sv \bmod 2 = 1 \wedge sv \geq 9$: $(\lfloor (sv-1)/2 \rfloor - 1, sv - 2)$ $(\lfloor (sv-1)/2 \rfloor - 4, sv - 4)$

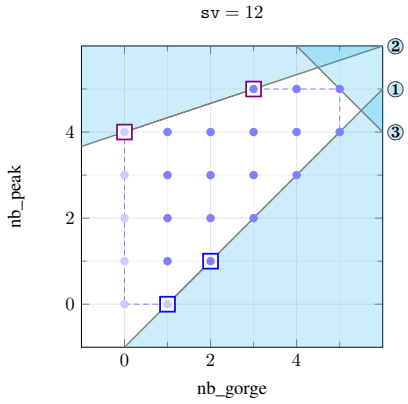


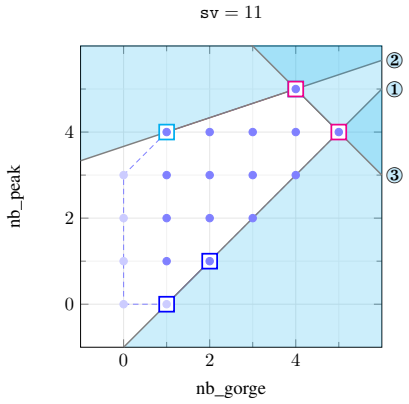
NB_GORGE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $3 * y \leq x + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋ − 2, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 5, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 4, ⌊(sv − 1)/2⌋ − 1)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋)

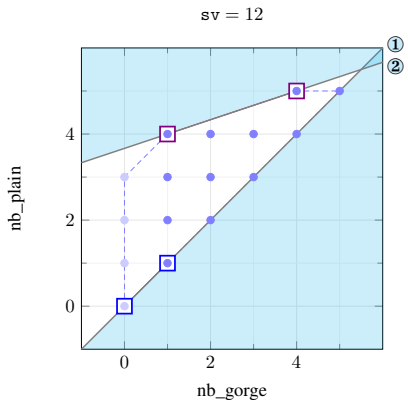


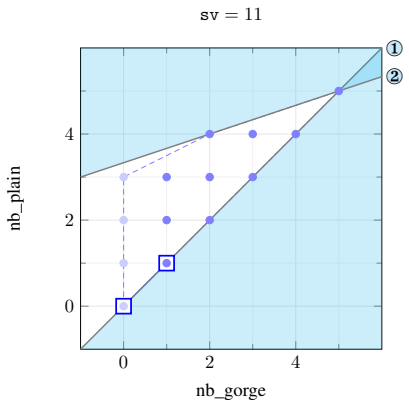


NB_GORGE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

② $3 * y \leq x + sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 4, ⌊(sv − 1)/2⌋ − 1)



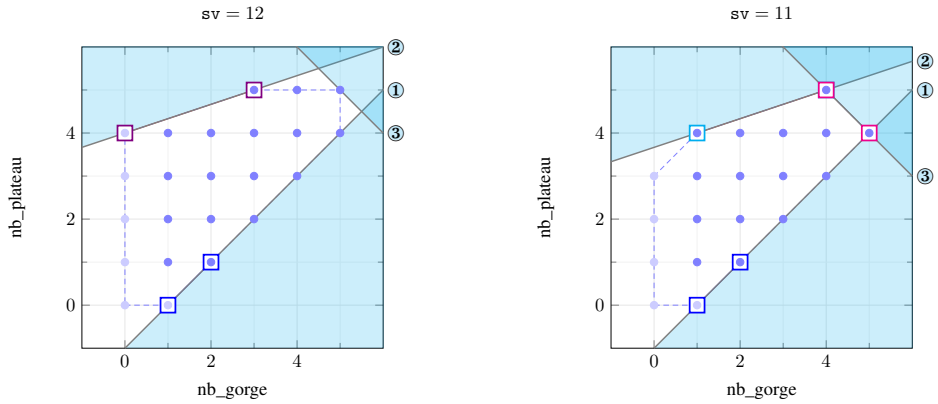


NB_GORGE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

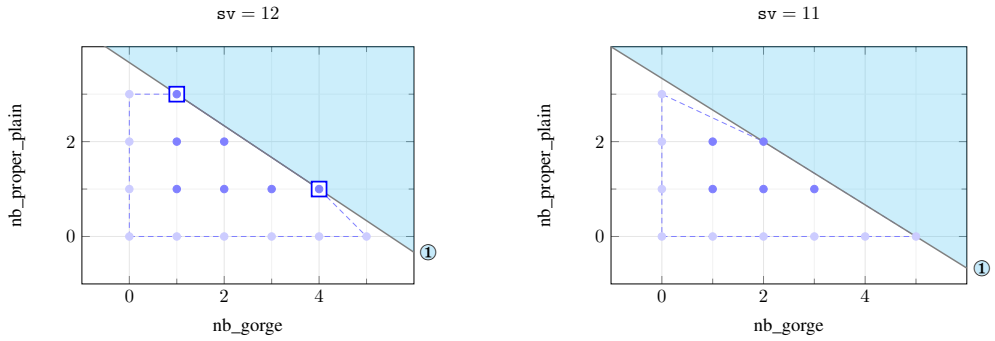
② $3 * y \leq x + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋ − 2, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 5, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 4, ⌊(sv − 1)/2⌋ − 1)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋)



NB_GORGE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 3 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: (⌊(sv − 1)/2⌋ − 1, 1) (⌊(sv − 1)/2⌋ − 4, 3)

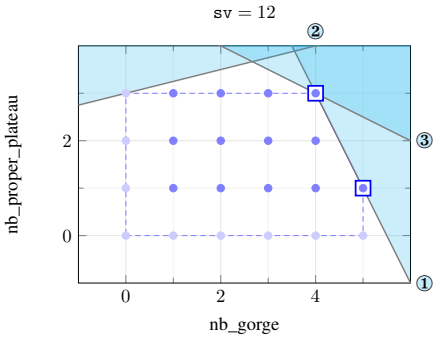


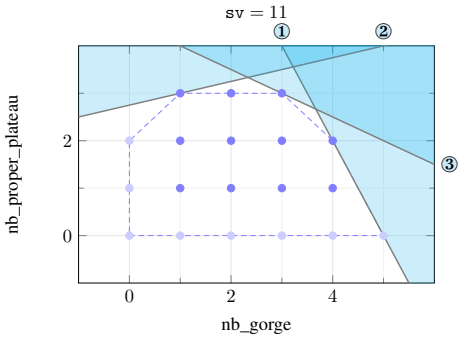
NB_GORGE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

② $4 * y \leq x + sv$

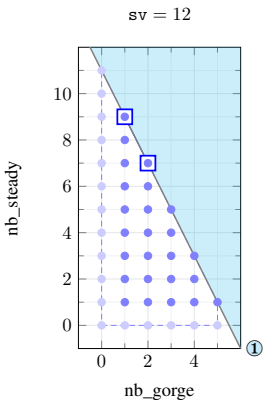
③ $sv > 1 \Rightarrow x + 2 * y \leq sv - 2$

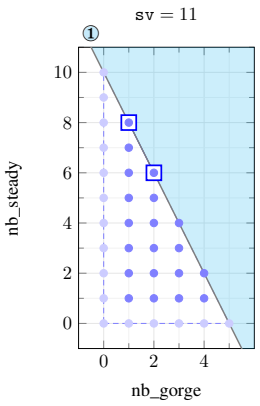




NB_GORGE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 3) (2, sv − 5)

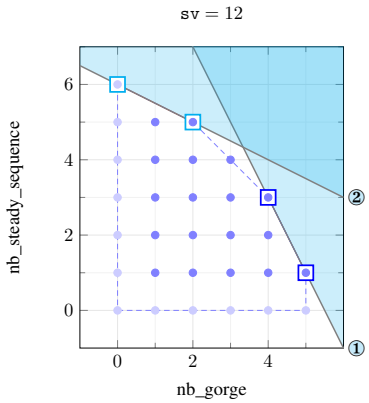


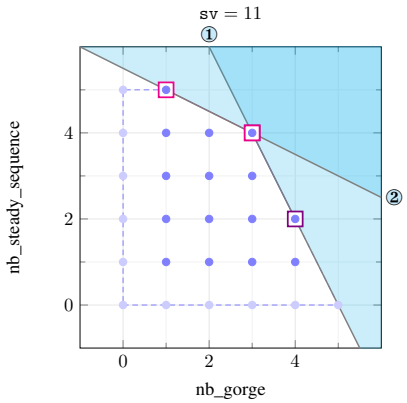


NB_GORGE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

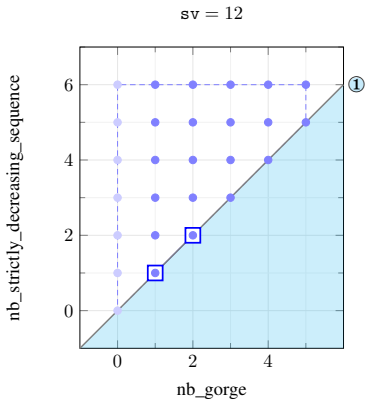
② $x + 2 * y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(0, \lfloor sv/2 \rfloor)$ $(2, \lfloor sv/2 \rfloor - 1)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(1, \lfloor sv/2 \rfloor)$ $(3, \lfloor sv/2 \rfloor - 1)$

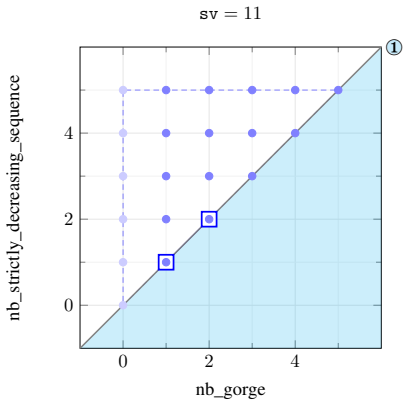




NB_GORGE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

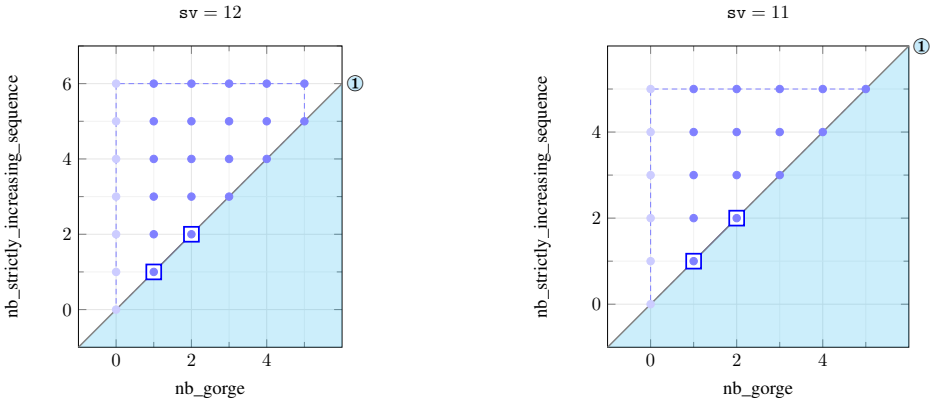
① $x \leq y$
- sv ≥ 5: (1, 1) (2, 2)





NB_GORGE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 5: (1, 1) (2, 2)

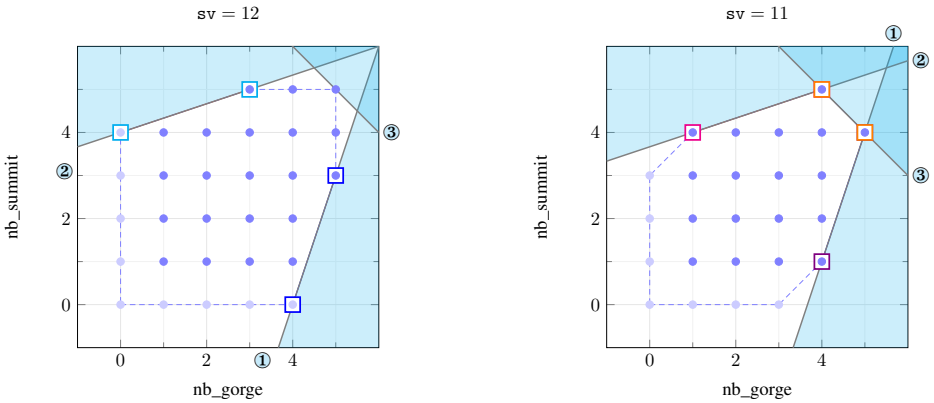


NB_GORGE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 2) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋ − 5)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋ − 4)

② $3 * y \leq x + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋ − 2, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 5, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 4, ⌊(sv − 1)/2⌋ − 1)

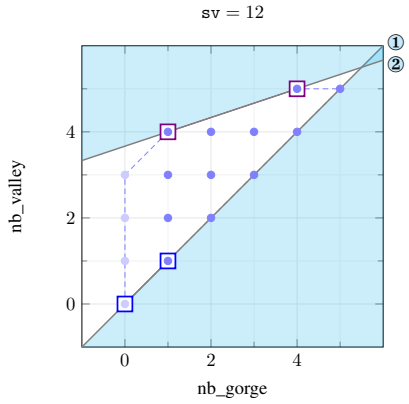
③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋)

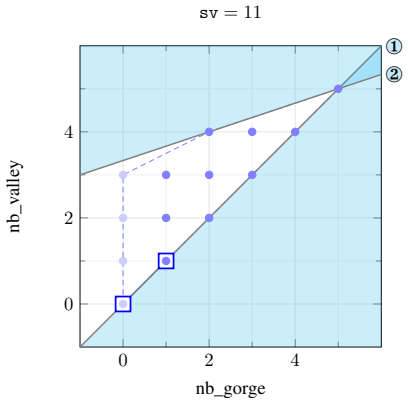


NB_GORGE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

② $3 * y \leq x + sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 10$: $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor)$ $(\lfloor (sv-1)/2 \rfloor - 4, \lfloor (sv-1)/2 \rfloor - 1)$



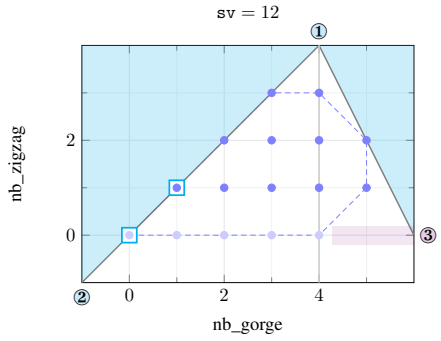


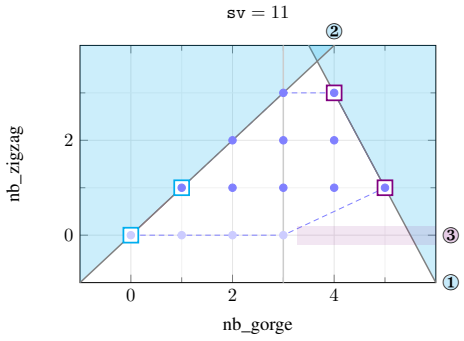
NB_GORGE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \bmod 2 = 0 \wedge sv \geq 14$: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 4)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

② $y \leq x$
- $sv \geq 4$: $(0, 0)$ $(1, 1)$

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$





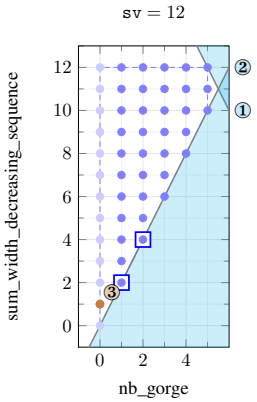
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

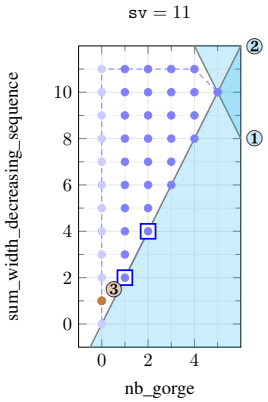
① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$





NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

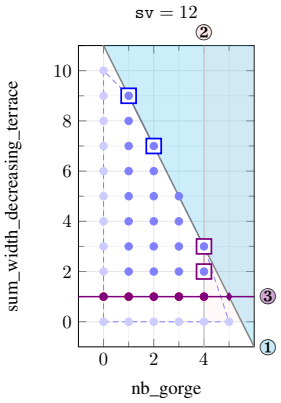
① $2 * x + y \leq sv - 1$

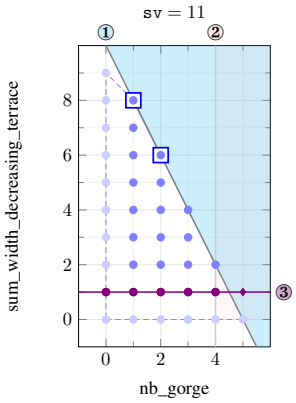
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$



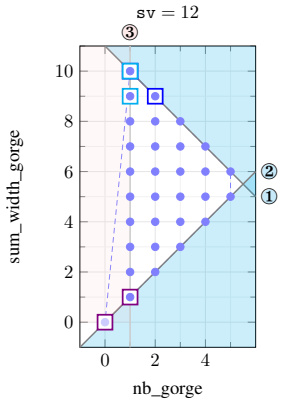


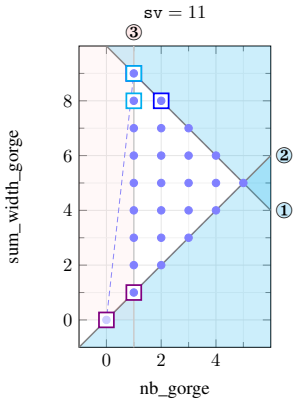
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$



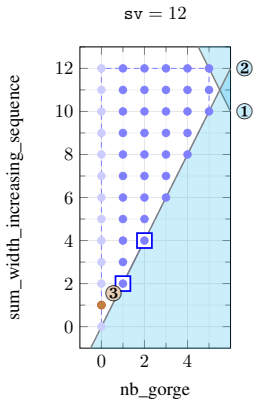


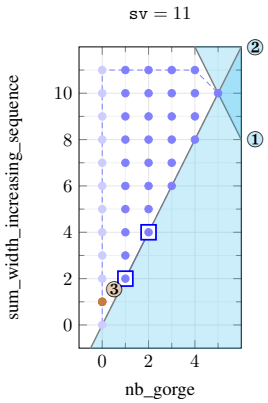
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



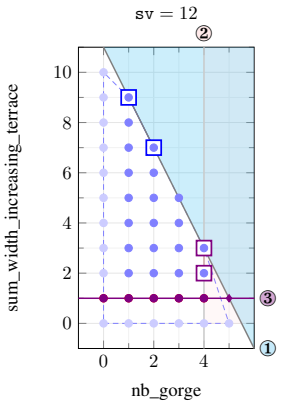


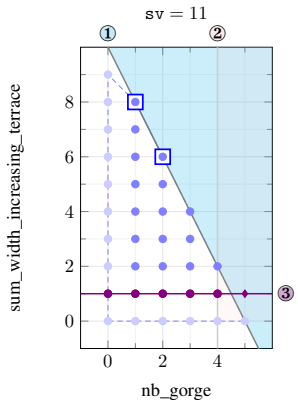
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

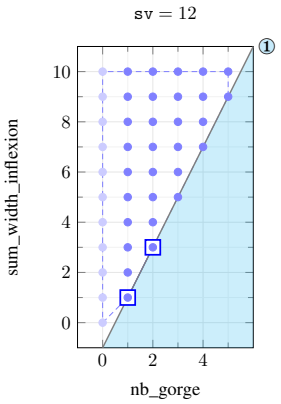
③ $y \neq 1$

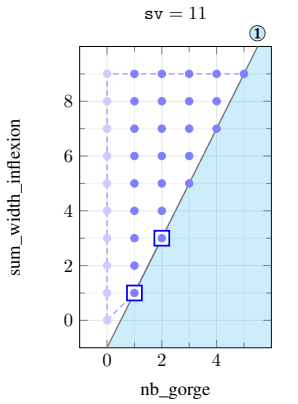




NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- $sv \geq 5$: $(1, 1)$ $(2, 3)$

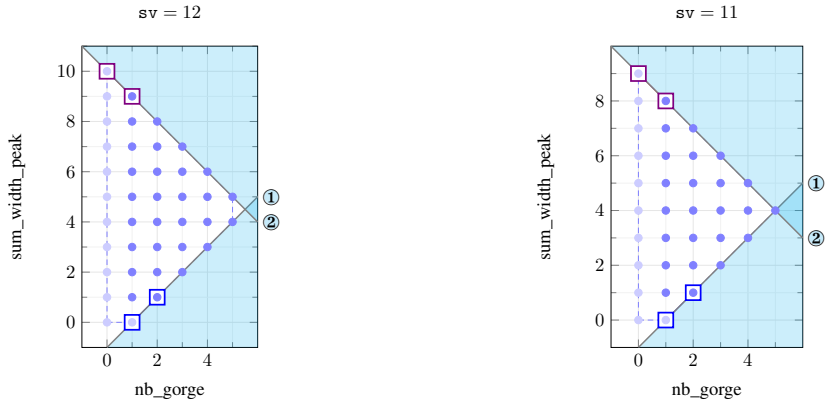




NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 3: (0, sv − 2) (1, sv − 3)



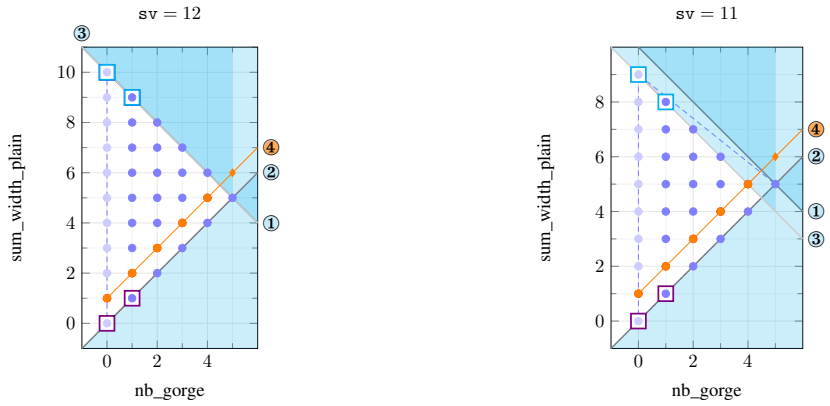
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: (0, sv − 2) (1, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

③ $x < \lfloor (sv - 1)/2 \rfloor \Rightarrow x + y \leq sv - 2$
- sv ≥ 6: (0, sv − 2) (1, sv − 3)

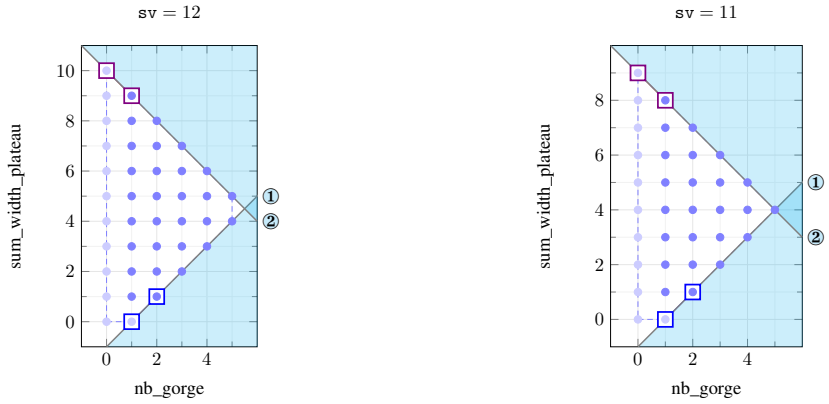
④ $y \neq x + 1$



NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$

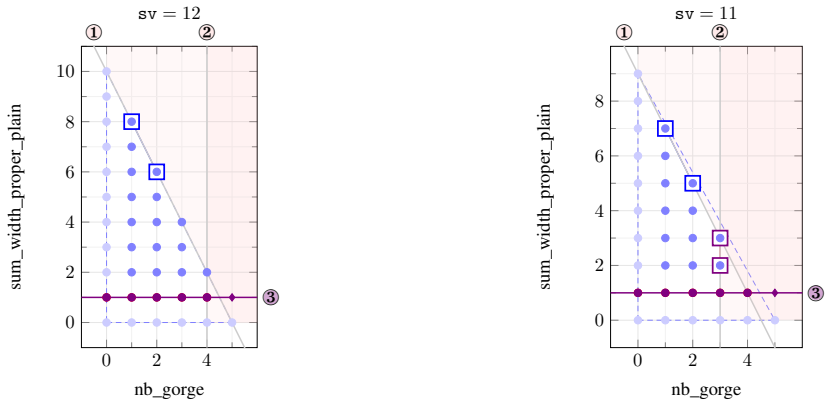


NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 8$: $(1, sv - 4)$ $(2, sv - 6)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 1)/2 \rfloor - 2, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 2, 3)$

③ $y \neq 1$

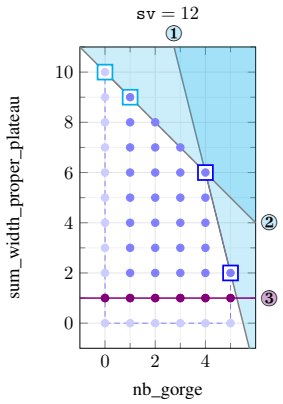


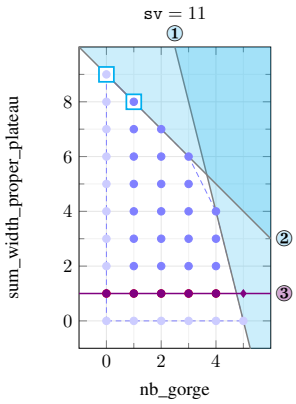
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$



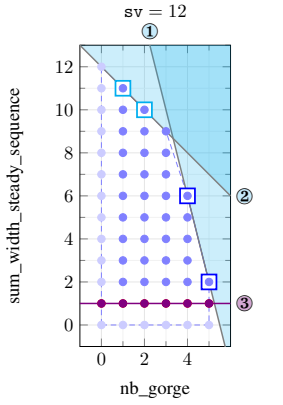


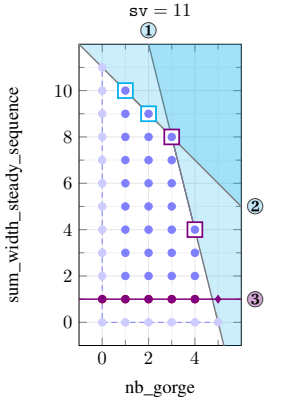
NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $x + y \leq sv$
- sv ≥ 8: $(1, sv-1)$ $(2, sv-2)$

③ $y \neq 1$



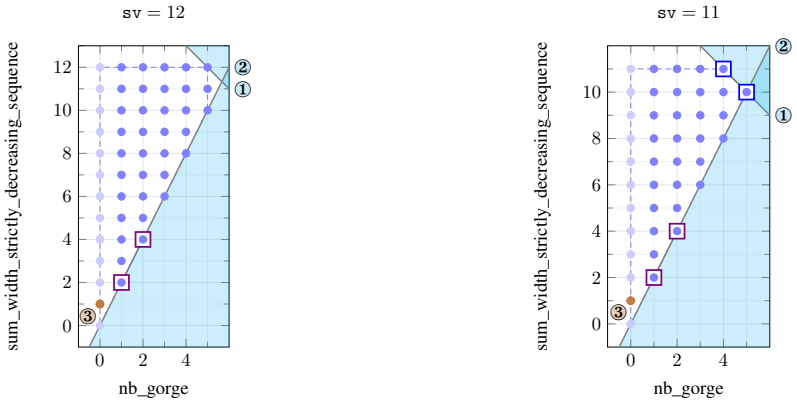


NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 3$: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$

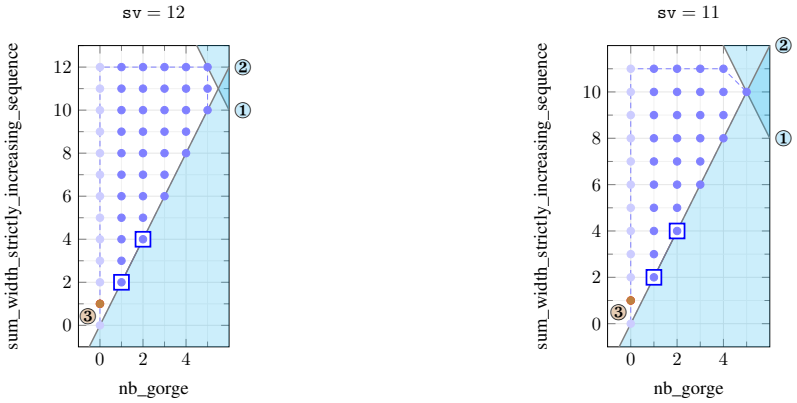


NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$

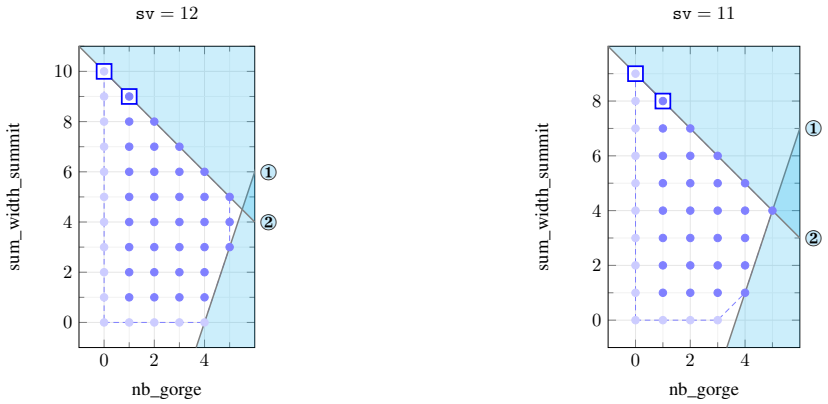


NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

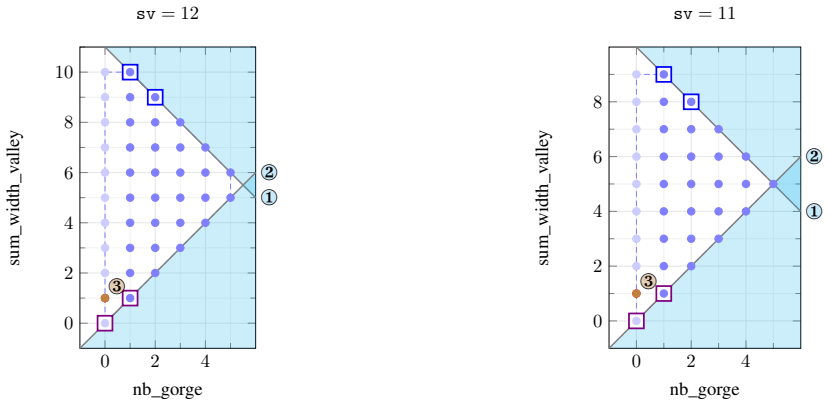
① $x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$

- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $x \neq 0 \vee y \neq 1$



NB_GORGE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 3 * x$
- sv ≥ 5: (0, 0) (1, 3)

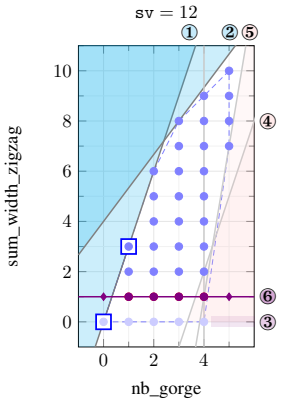
② $3 * y \leq 4 * x + sv$

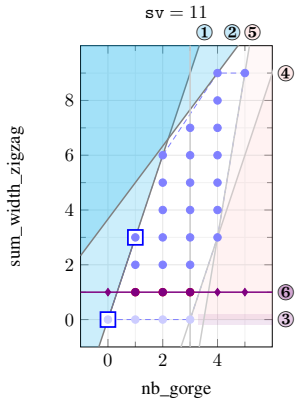
③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

⑤ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, sv − 5) (⌊(sv − 1)/2⌋ − 1, sv − 11)
- sv mod 2 = 1 ∧ sv ≥ 17: (⌊(sv − 1)/2⌋ − 1, sv − 8) (⌊(sv − 1)/2⌋ − 2, sv − 14)

⑥ $y \neq 1$





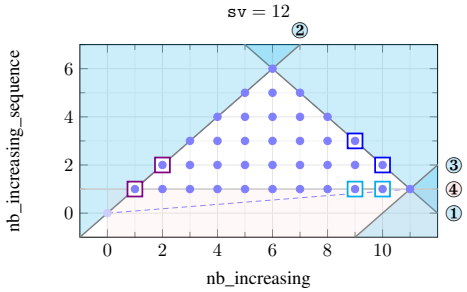
NB_INCREASING(x, VARIABLES) ∧
NB_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

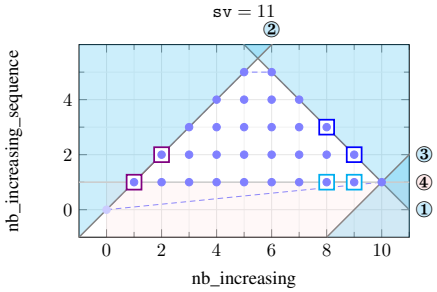
① $x + y \leq sv$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $y \leq x$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

③ $sv > 1 \Rightarrow x \leq y + sv - 2$

④ $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$



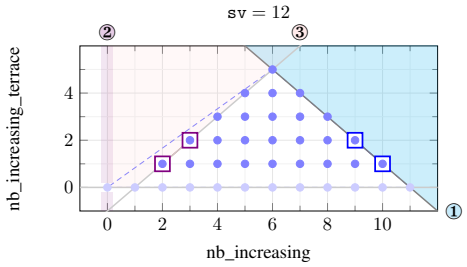


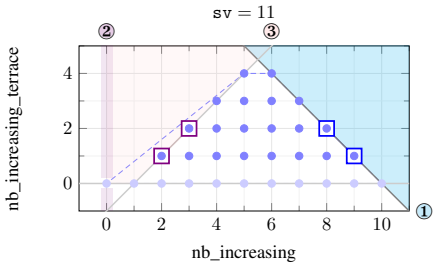
NB_INCREASING(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 6$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $x = 0 \Rightarrow y = 0$

③ $x > 0 \wedge y > 0 \Rightarrow y \leq x - 1$
- $sv \geq 6$: $(2, 1)$ $(3, 2)$

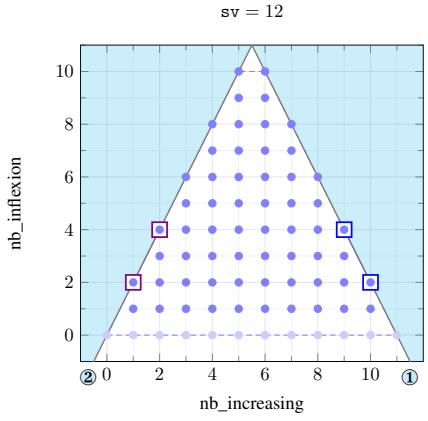


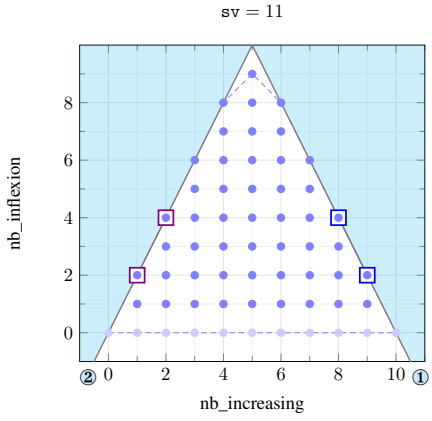


NB_INCREASING(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \leq 2 * x$
- $sv \geq 6$: $(1, 2)$ $(2, 4)$

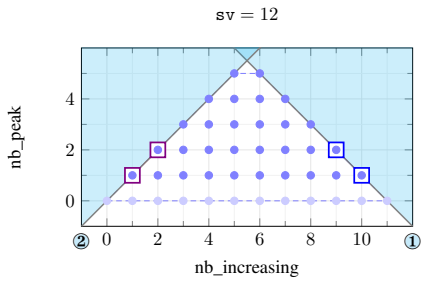


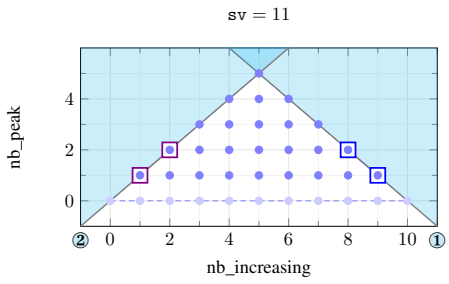


NB_INCREASING(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

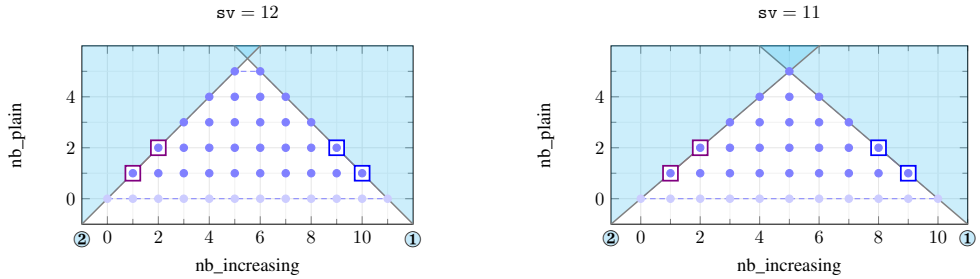




NB_INCREASING(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

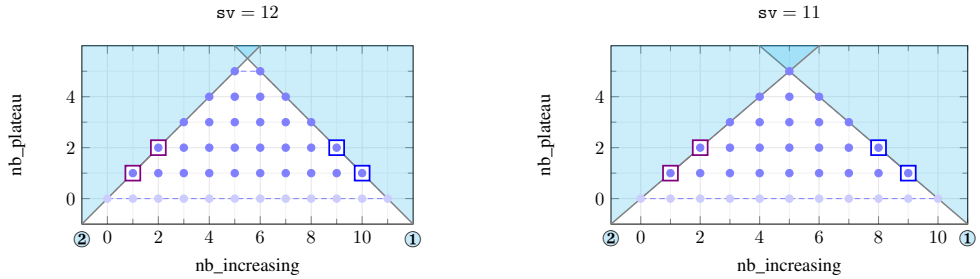
② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

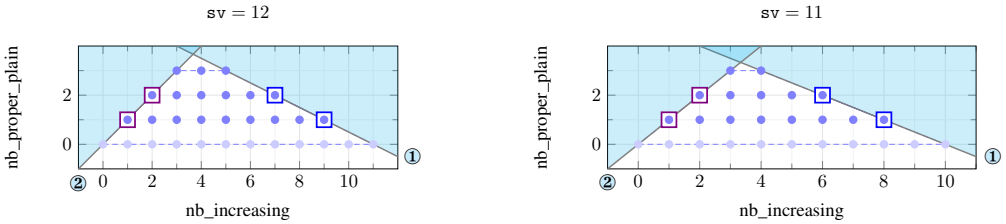
② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- $sv \geq 7$: $(sv - 3, 1)$ $(sv - 5, 2)$

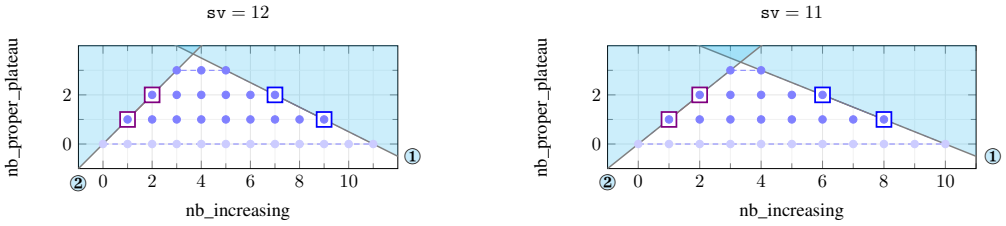
② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- $sv \geq 7$: $(sv - 3, 1)$ $(sv - 5, 2)$

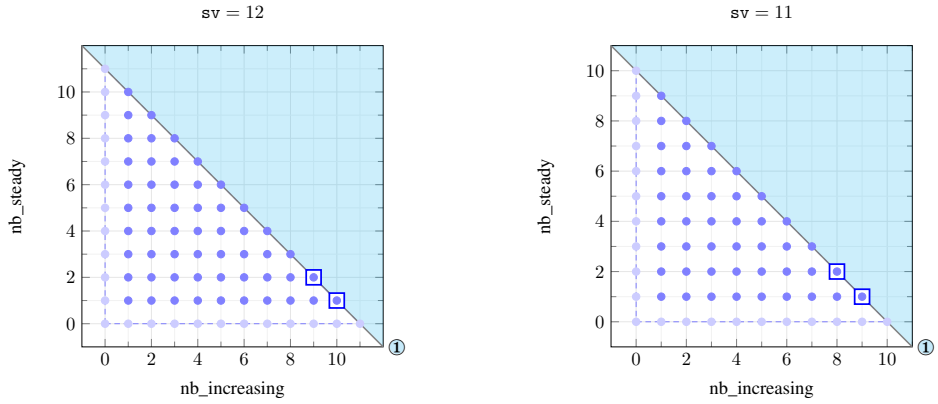
② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

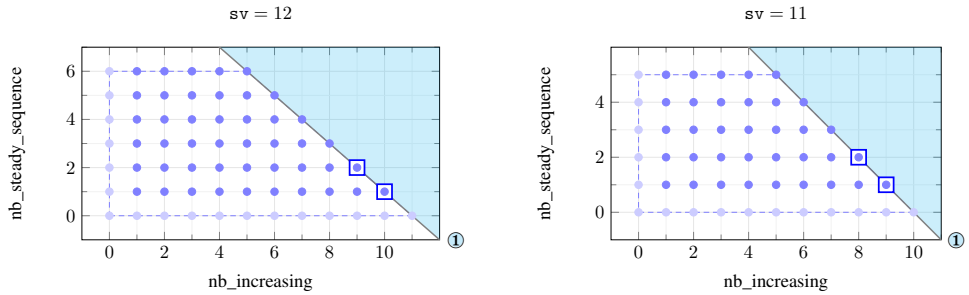
- $sv \geq 3$: $(sv - 2, 1)$ $(sv - 3, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

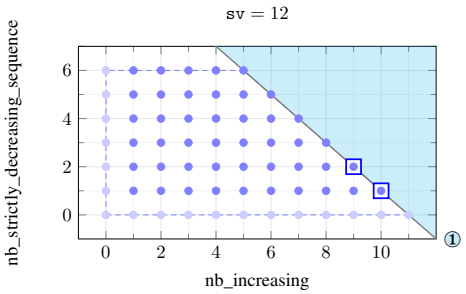
① $x + y \leq sv - 1$

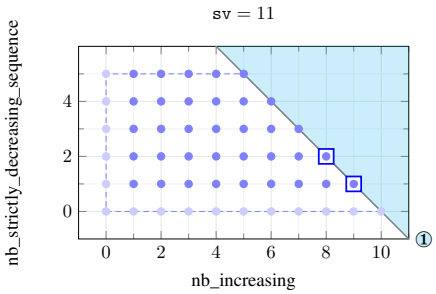
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$





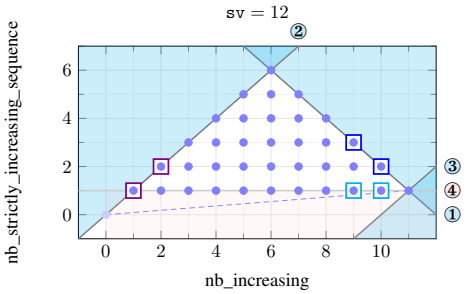
NB_INCREASING(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

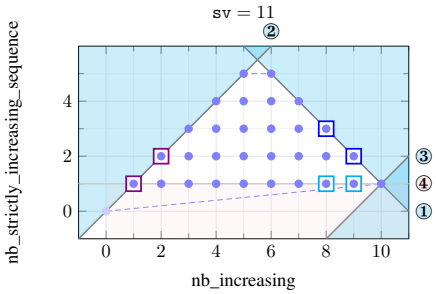
① $x + y \leq sv$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $y \leq x$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

③ $sv > 1 \Rightarrow x \leq y + sv - 2$

④ $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$

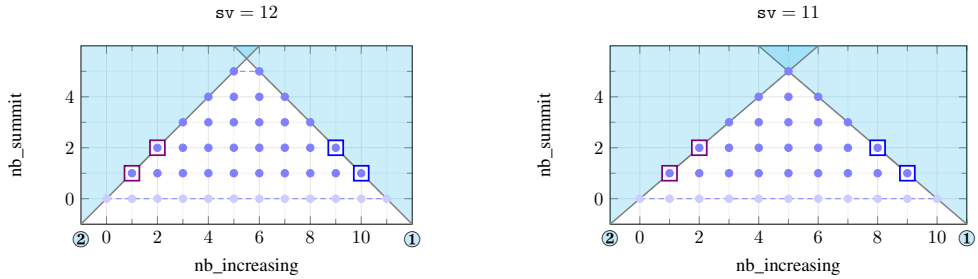




NB_INCREASING(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

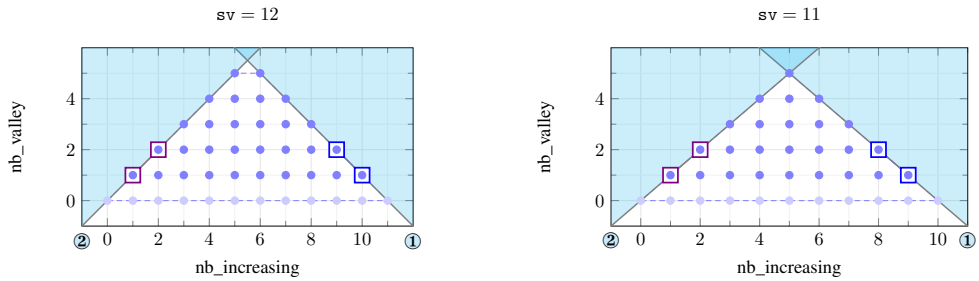
② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$



NB_INCREASING(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 5$: $(1, 1)$ $(2, 2)$

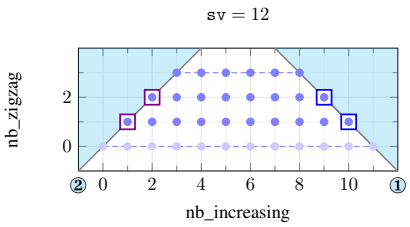


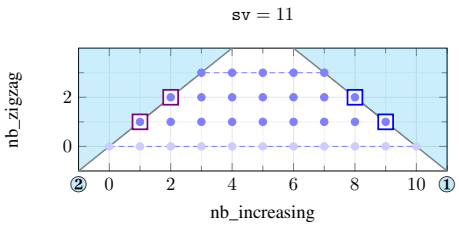
NB_INCREASING(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 7$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

③ $\bigvee \left(\begin{array}{l} y \neq \max(0, \lfloor (sv-1)/3 \rfloor) - 0, \\ x < \max(0, \lfloor (sv-1)/3 \rfloor) + 1, \\ x > 1 * \max(0, sv-1) - \max(0, \lfloor (sv-1)/3 \rfloor) - 1, \\ 1 \neq sv \bmod 3 \end{array} \right)$

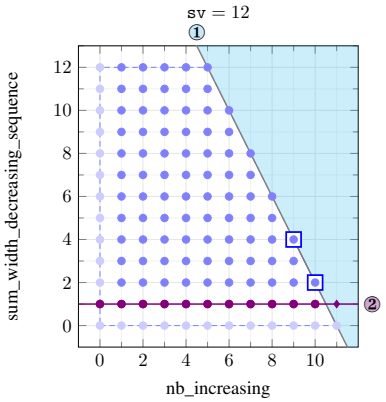


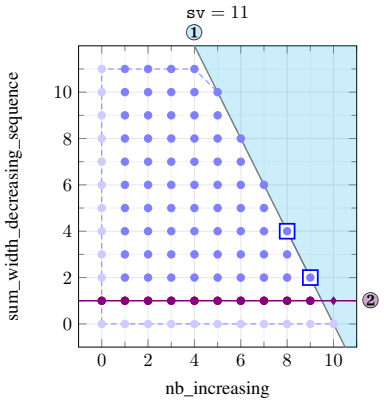


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$



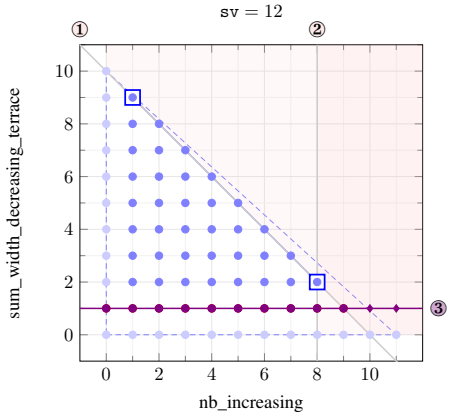


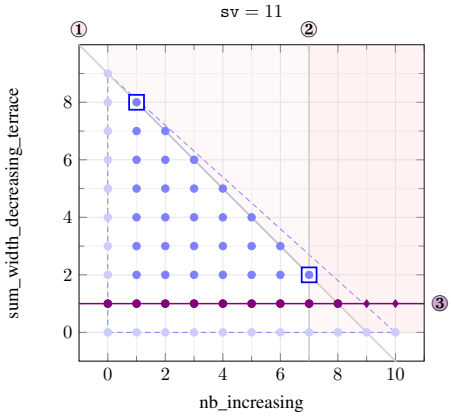
NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 4, 2)$ $(1, sv - 3)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$

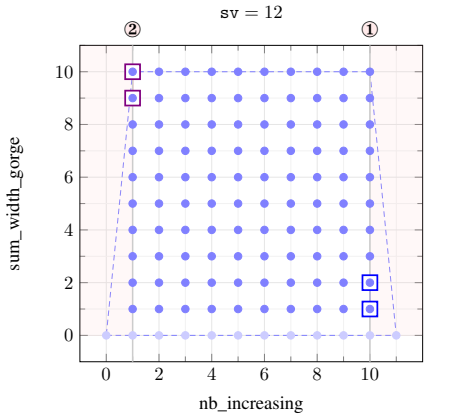


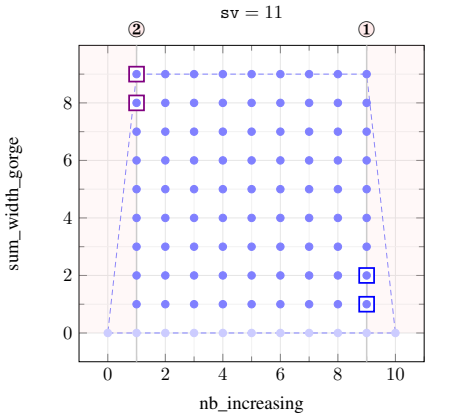


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 2, 2)$

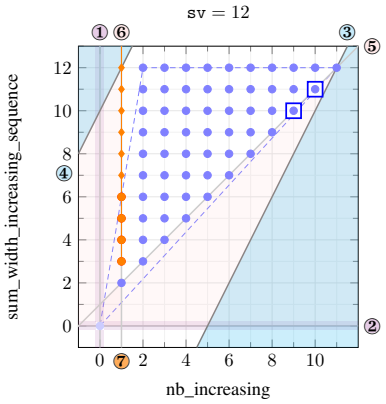
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

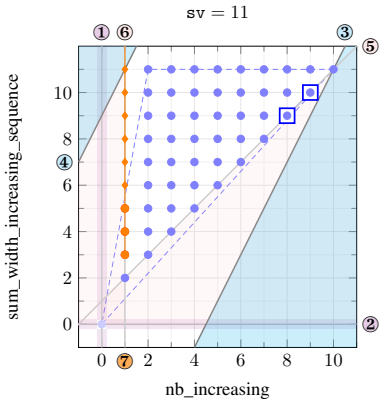




NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x = 0 \Rightarrow y = 0$
② $y = 0 \Rightarrow x = 0$
③ $sv > 1 \Rightarrow 2 * x \leq y + sv - 2$
④ $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$
⑤ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
□ $sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv - 2)$
⑥ $y > 0 \Rightarrow x \geq 1$
⑦ $x \neq 1 \vee y < 3$





NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

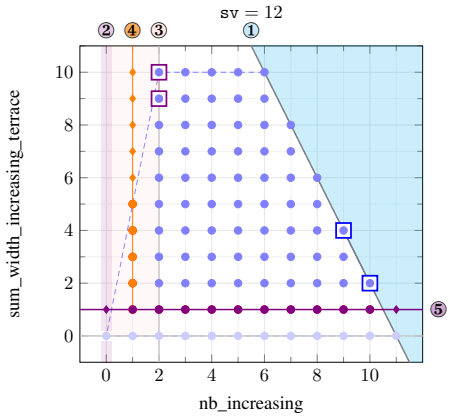
① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

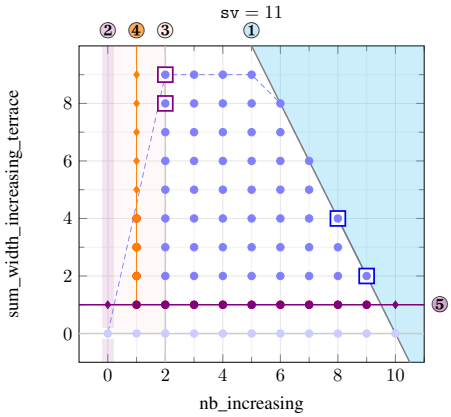
② $x = 0 \Rightarrow y = 0$

③ $x > 0 \wedge y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

④ $x \neq 1 \vee y < 1$

⑤ $y \neq 1$

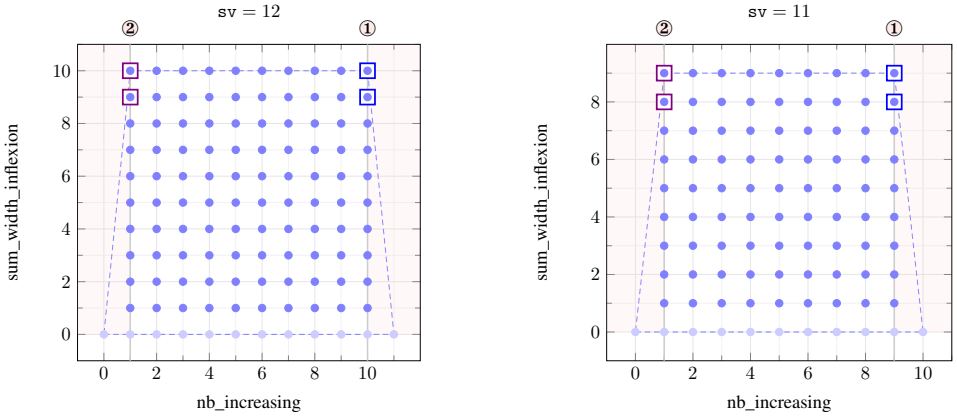




NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

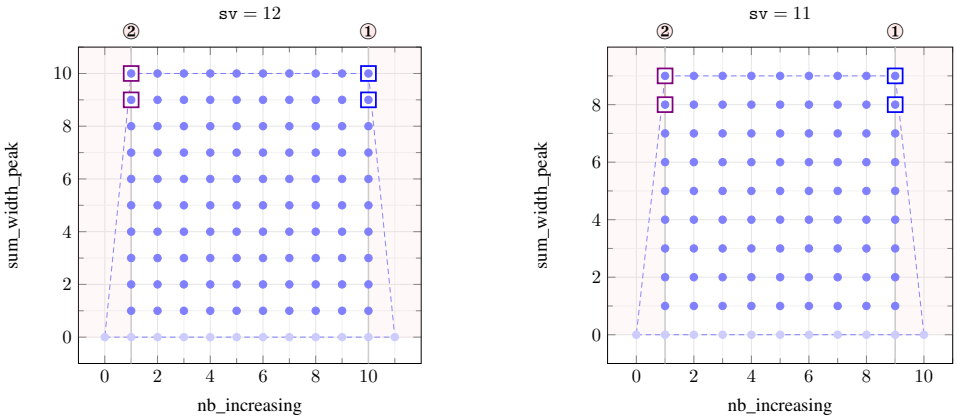
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

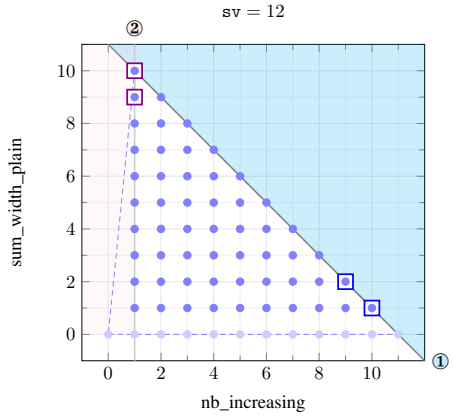
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$

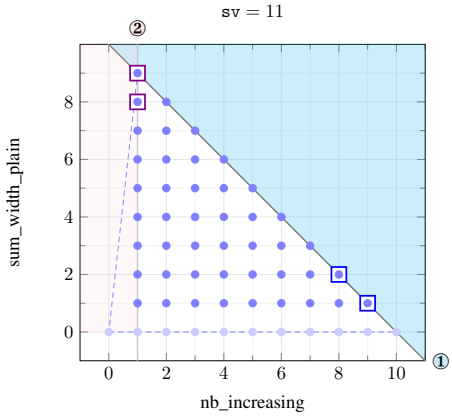


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

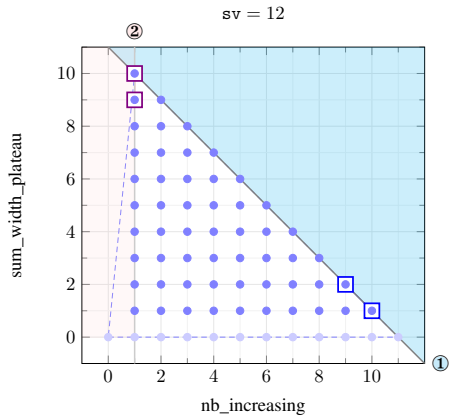


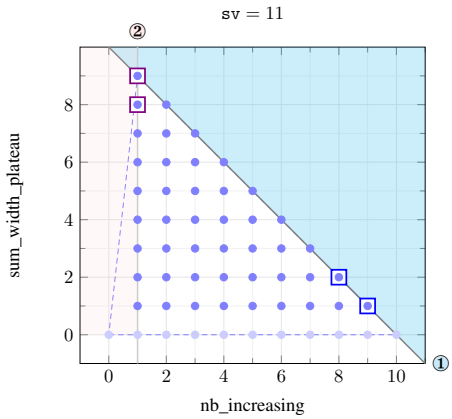


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





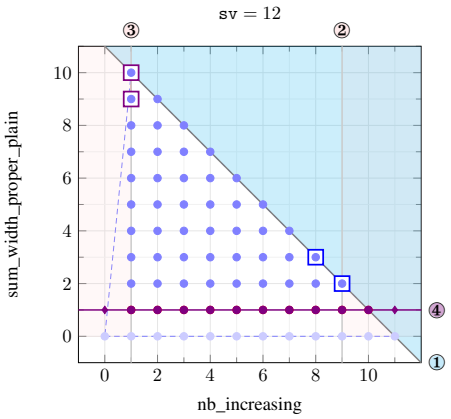
NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

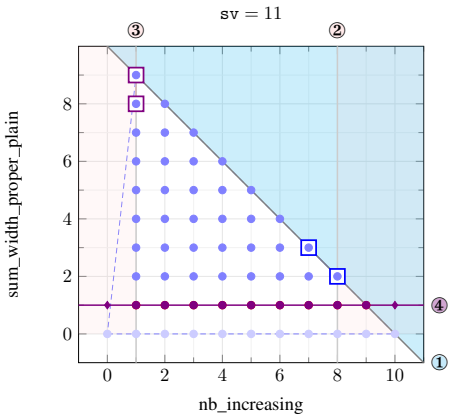
① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 4, 3)$

② $y > 0 \Rightarrow x \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





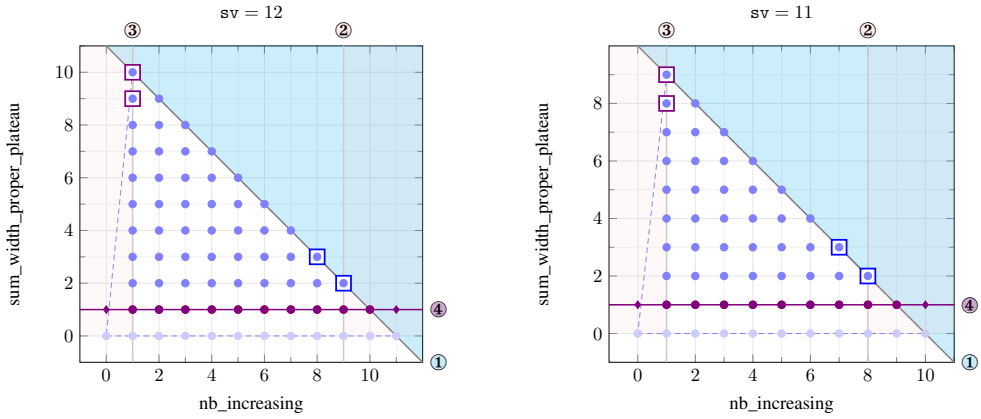
NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 4, 3)$

② $y > 0 \Rightarrow x \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

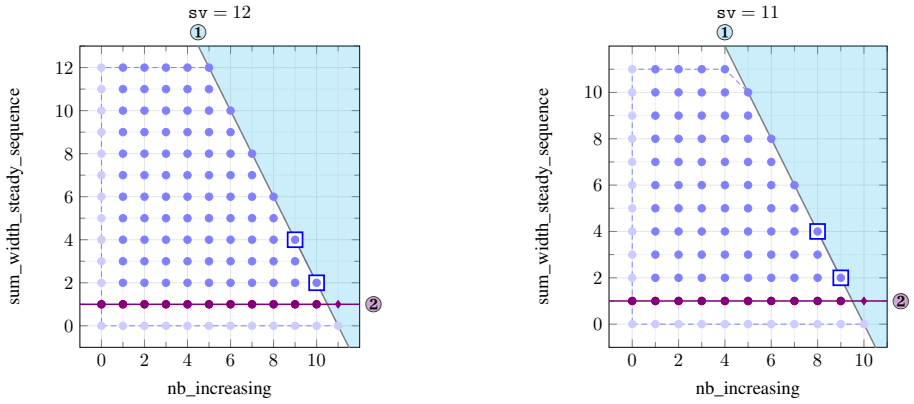
④ $y \neq 1$



NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

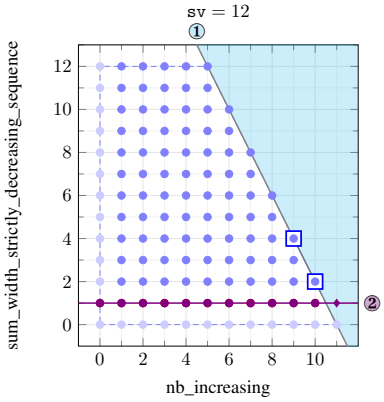
② $y \neq 1$

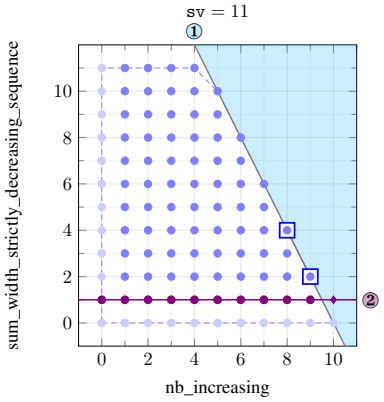


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$





NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

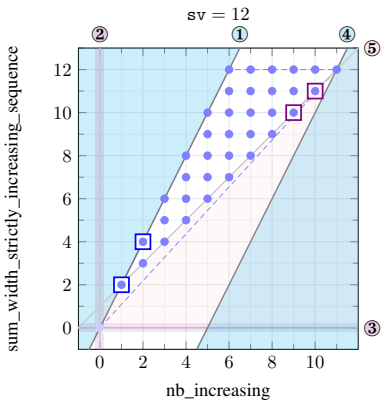
① $y \leq 2 * x$
- $sv \geq 4$: $(1, 2)$ $(2, 4)$

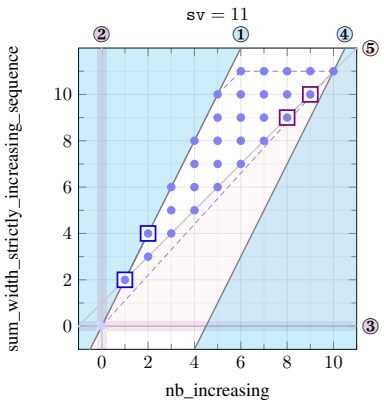
② $x = 0 \Rightarrow y = 0$

③ $y = 0 \Rightarrow x = 0$

④ $sv > 1 \Rightarrow 2 * x \leq y + sv - 2$

⑤ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv - 2)$

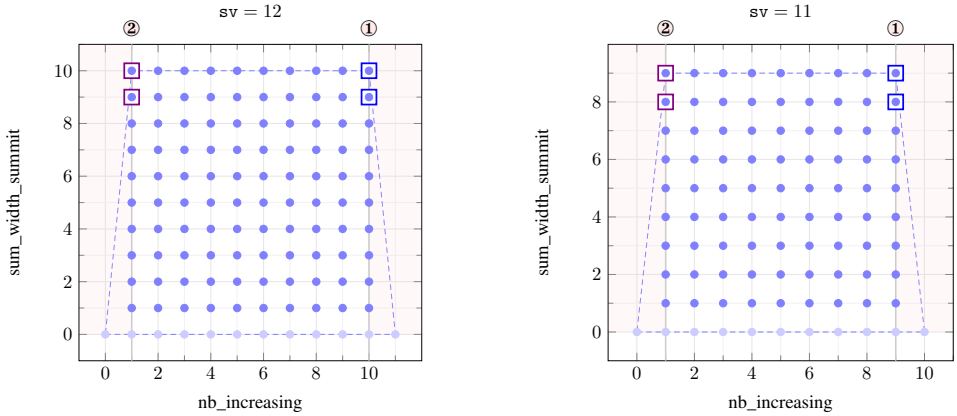




NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 2, sv - 3)$

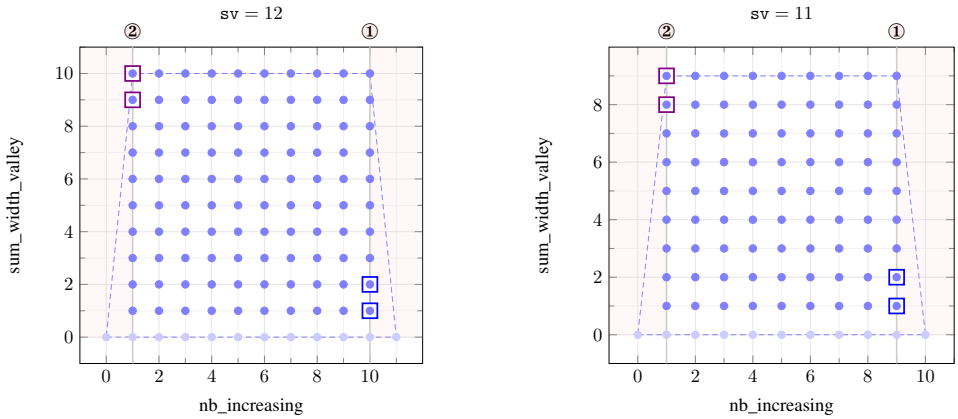
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$



NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 2, 2)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

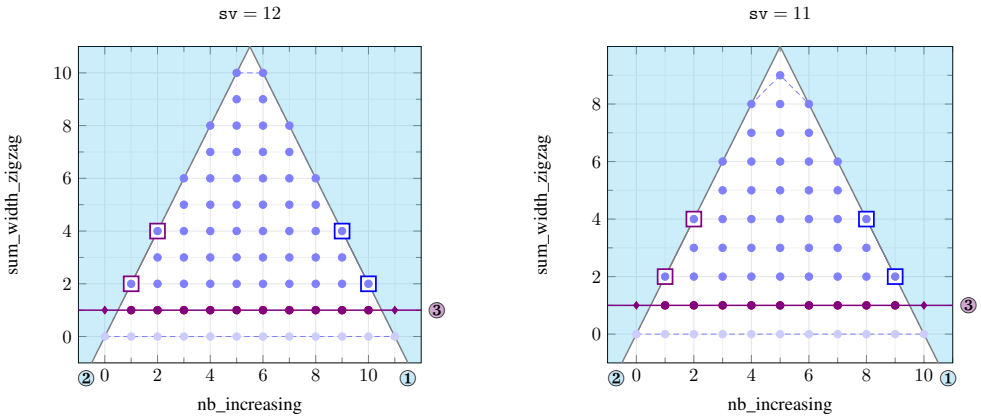


NB_INCREASING(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \leq 2 * x$
- $sv \geq 6$: $(1, 2)$ $(2, 4)$

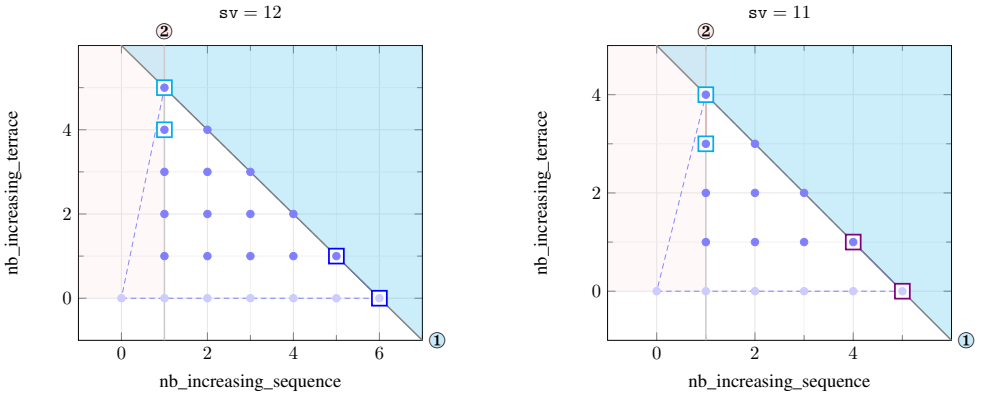
③ $y \neq 1$



NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 1)$

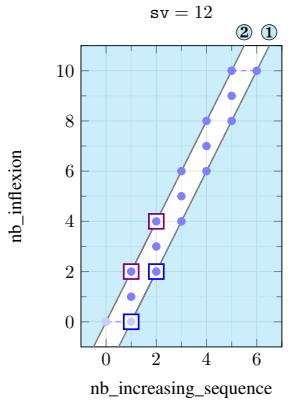
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, \lfloor (sv - 2)/2 \rfloor)$ $(1, \lfloor (sv - 2)/2 \rfloor - 1)$

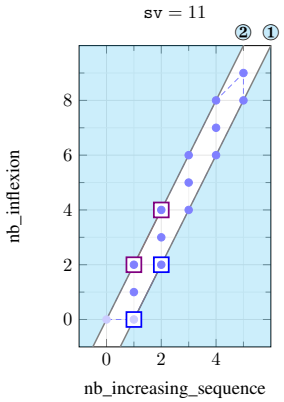


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

② $y \leq 2 * x$
- sv ≥ 6: (1, 2) (2, 4)

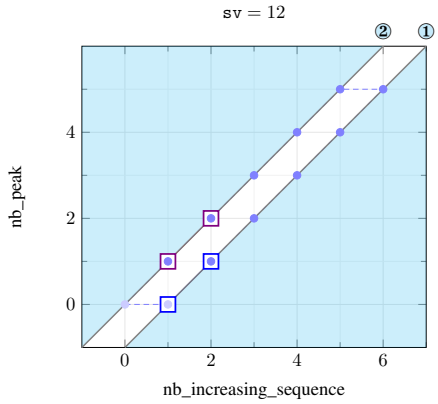


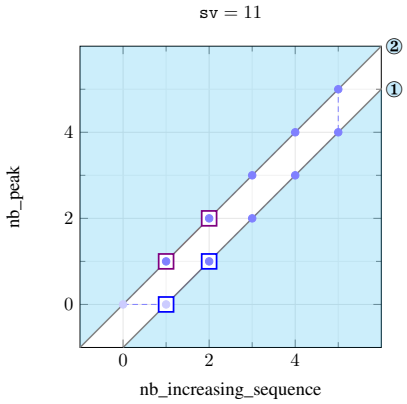


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

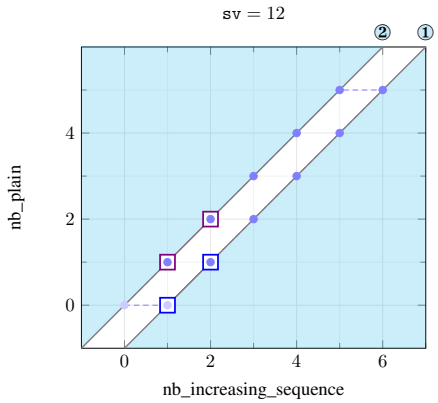


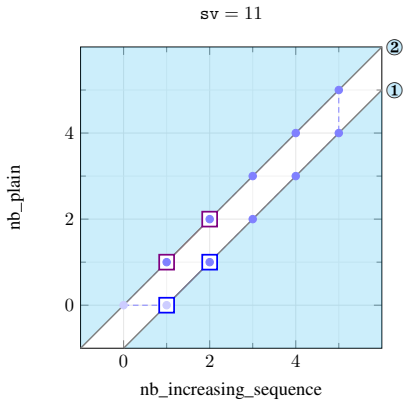


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

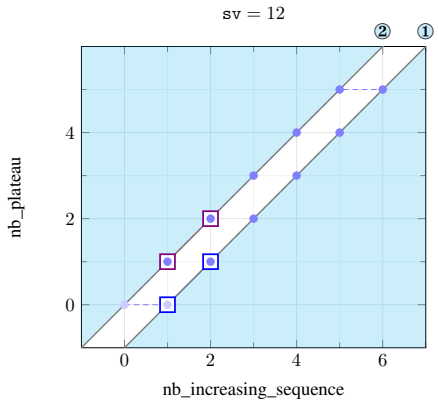


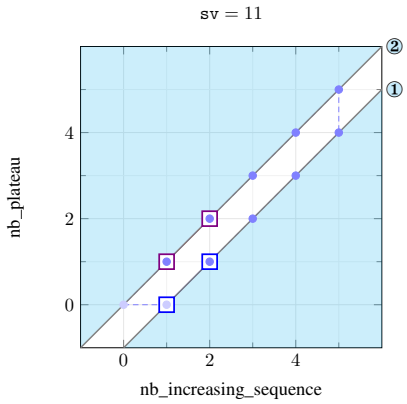


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

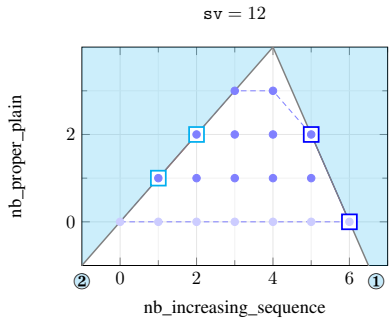


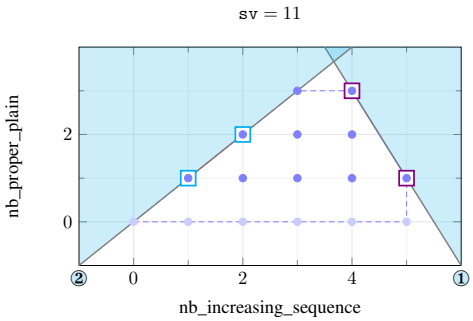


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, 1)$ $(\lfloor sv/2 \rfloor - 1, 3)$

② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

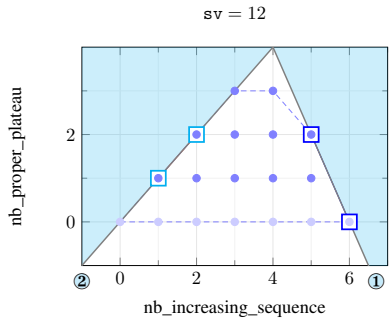


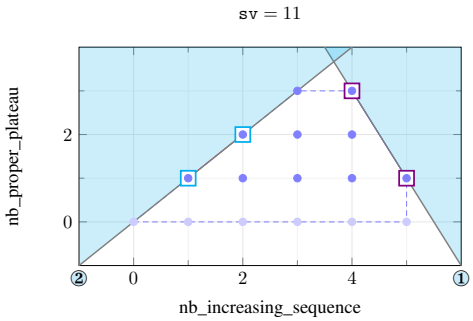


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, 1)$ $(\lfloor sv/2 \rfloor - 1, 3)$

② $y \leq x$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

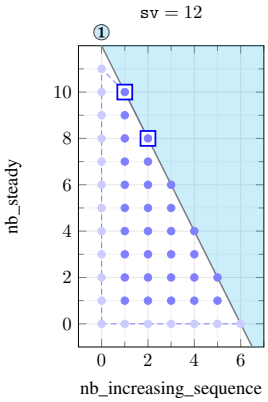


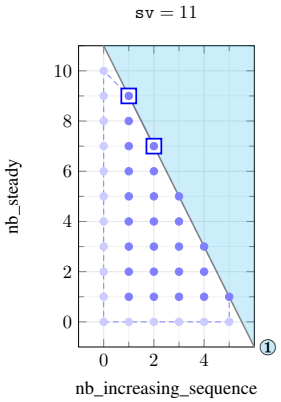


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$

- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 4)$

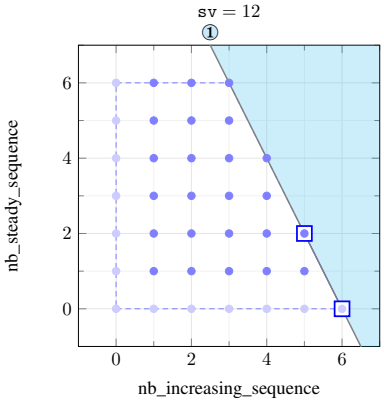


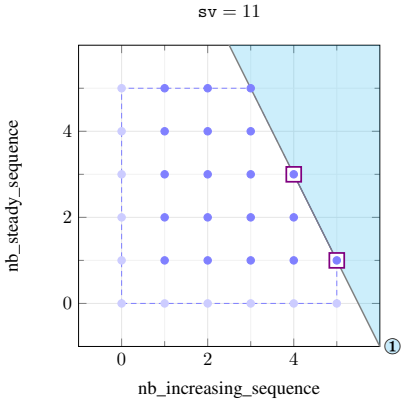


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$

- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 2)$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor, 1)$ $(\lfloor sv/2 \rfloor - 1, 3)$

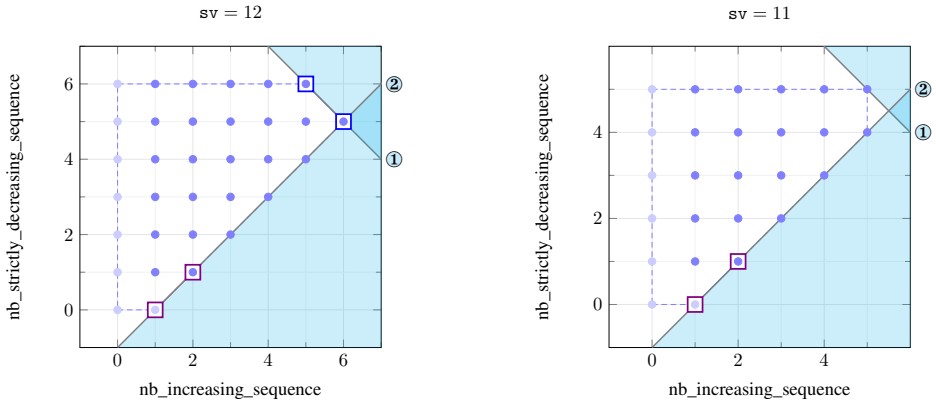




NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor sv/2 \rfloor - 1, \lfloor sv/2 \rfloor)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

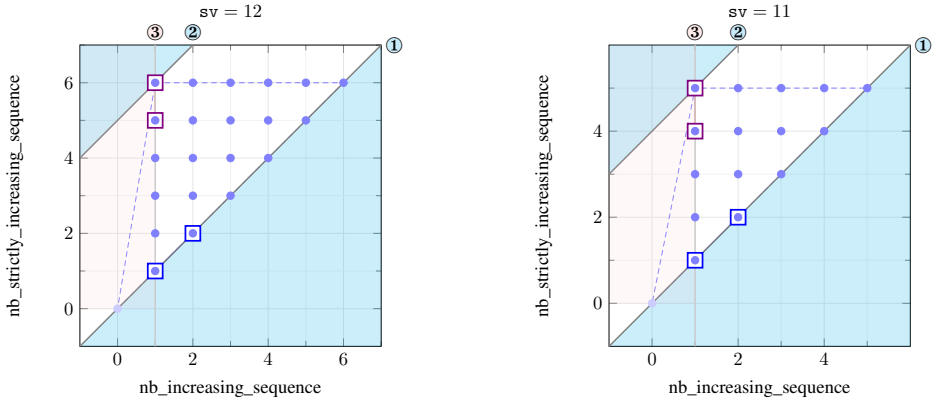


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

② $sv > 1 \Rightarrow 2 * y \leq 2 * x + sv - 2 - (sv - 2) \bmod 2$

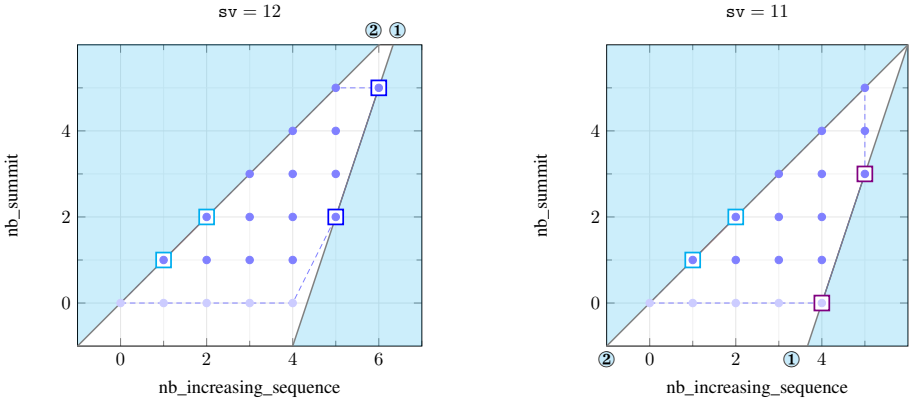
③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, \lfloor sv/2 \rfloor)$ $(1, \lfloor sv/2 \rfloor - 1)$



NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv + 1$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 3)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 2)$ $(\lfloor sv/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 5)$

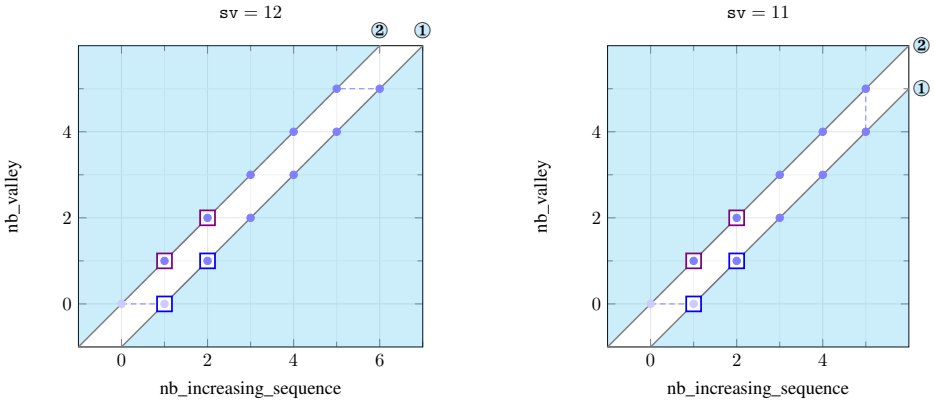
② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)



NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

② $y \leq x$
- sv ≥ 5: (1, 1) (2, 2)

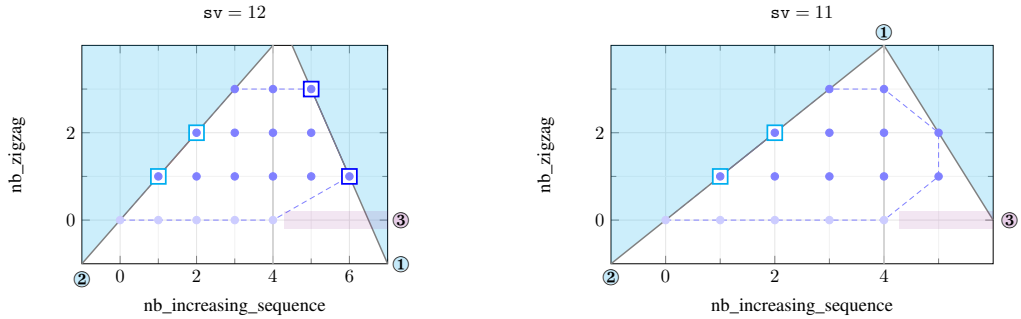


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv + 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor sv/2 \rfloor, 1)$ $(\lfloor sv/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 4)$

② $y \leq x$
- sv ≥ 7: (1, 1) (2, 2)

③ $y = 0 \Rightarrow 3 * x \leq sv + 1 - (sv + 1) \bmod 3$

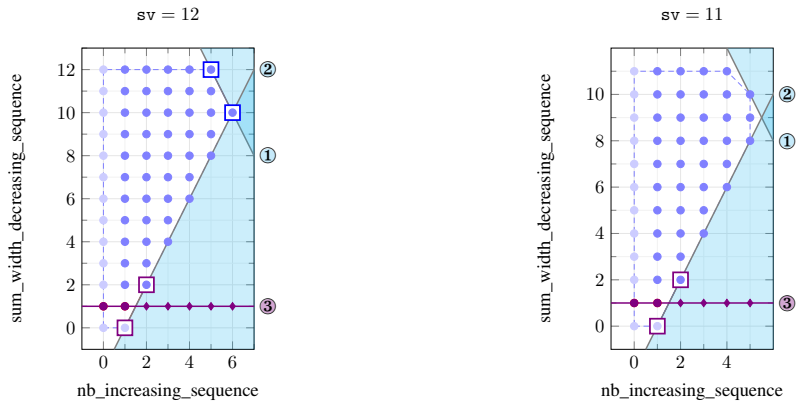


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 2: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

③ $y \neq 1$



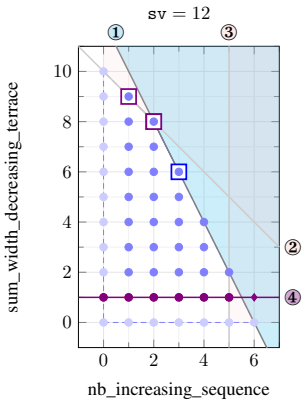
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

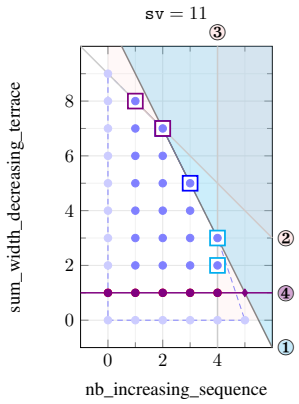
① $2 * x + y \leq sv$
- $sv \geq 8$: $(2, sv - 4)$ $(3, sv - 6)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(1, sv - 3)$ $(2, sv - 4)$

③ $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

④ $y \neq 1$



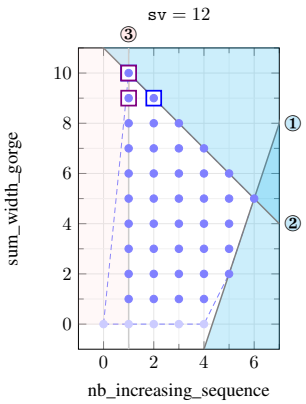


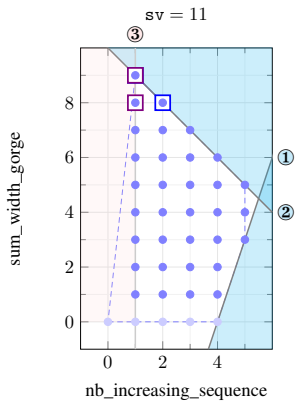
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv + 1$

② $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



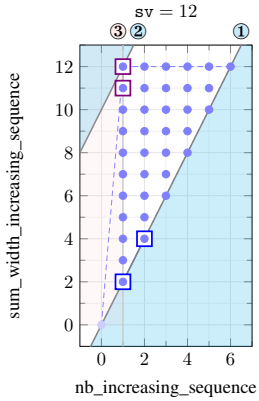


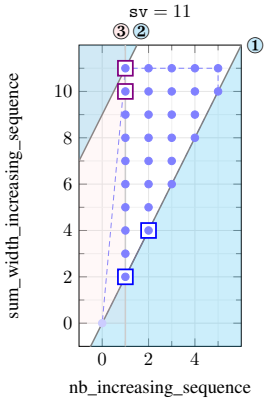
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv) (1, sv − 1)





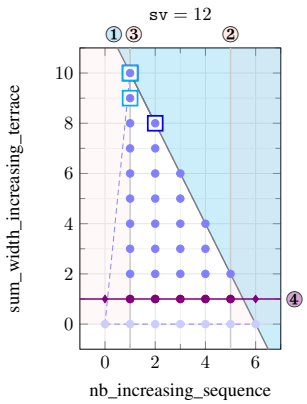
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

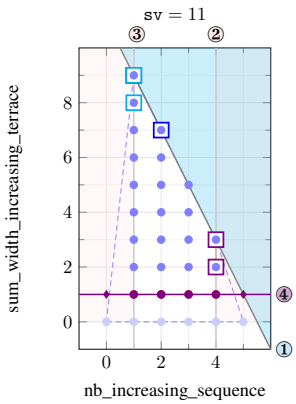
① $2 * x + y \leq sv$
- sv ≥ 6: (1, sv − 2) (2, sv − 4)

② $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- sv mod 2 = 1 ∧ sv ≥ 5: (⌊sv/2⌋ − 1, 2) (⌊sv/2⌋ − 1, 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)

④ $y \neq 1$

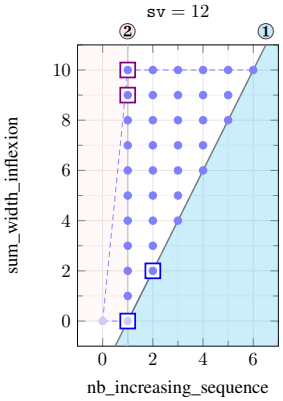


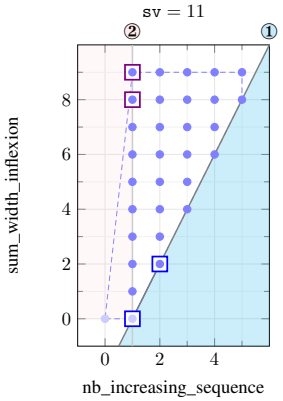


NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 2$
- sv ≥ 4: (1, 0) (2, 2)

② $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



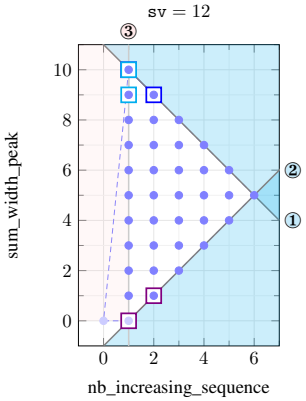


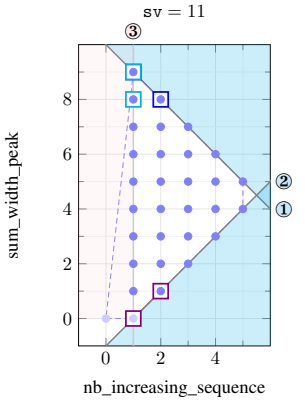
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 4: (1, sv − 2) (2, sv − 3)

② $x \leq y + 1$
- sv ≥ 4: (1, 0) (2, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



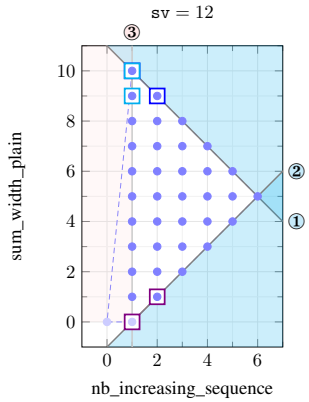


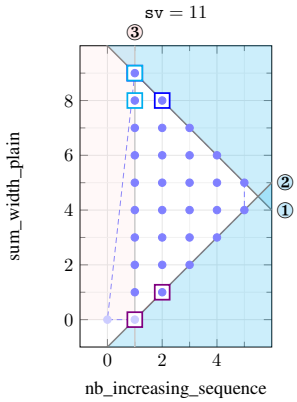
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



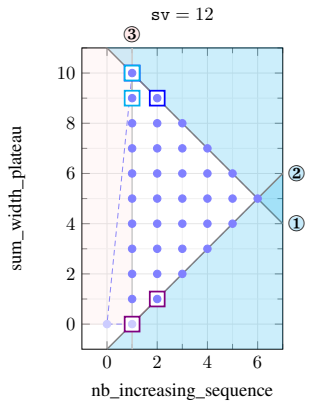


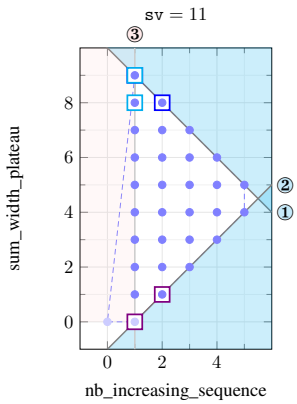
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y + 1$
- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





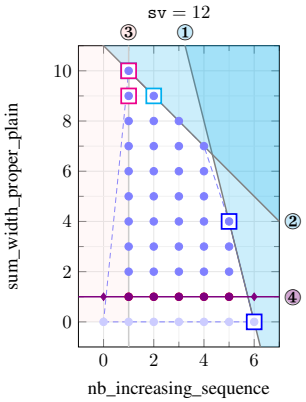
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

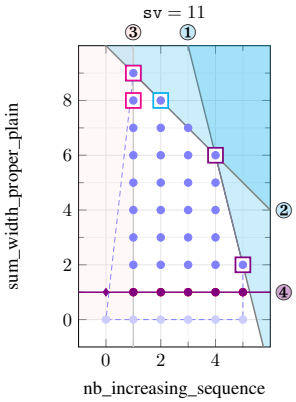
① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- sv ≥ 5: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





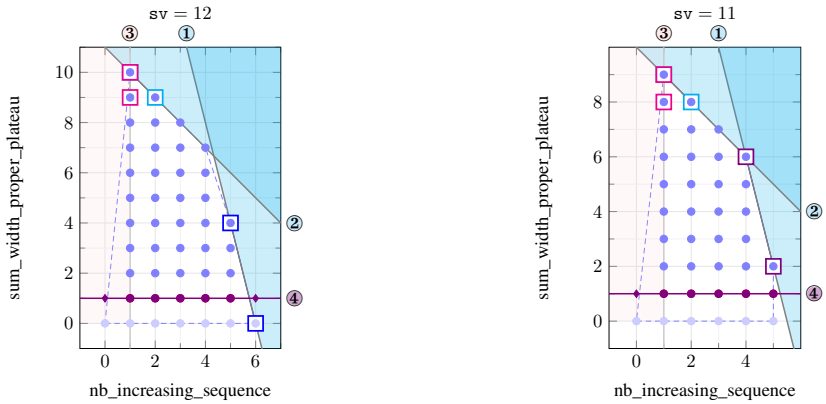
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- sv ≥ 5: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv - 2)$ $(1, sv - 3)$

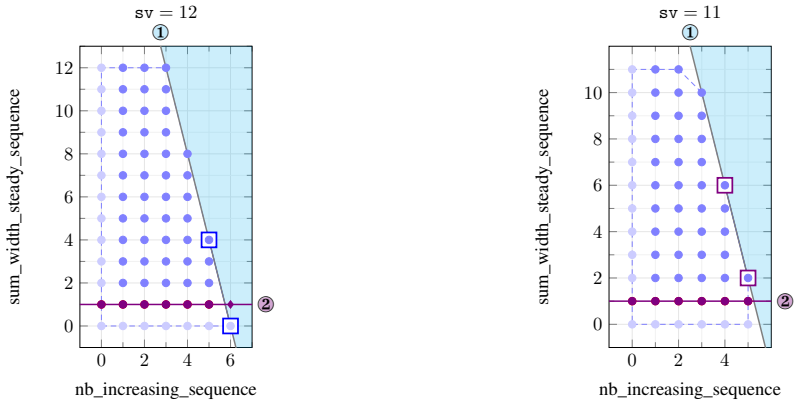
④ $y \neq 1$



NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 4: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 7: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $y \neq 1$

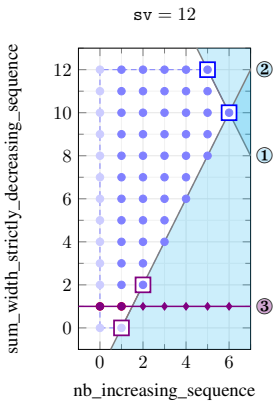


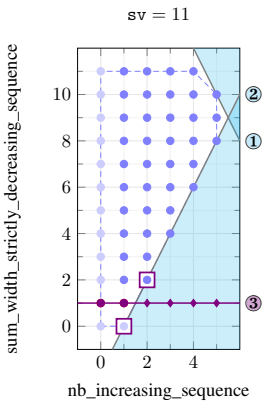
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $2 * x \leq y + 2$
- $sv \geq 4$: $(1, 0)$ $(2, 2)$

③ $y \neq 1$



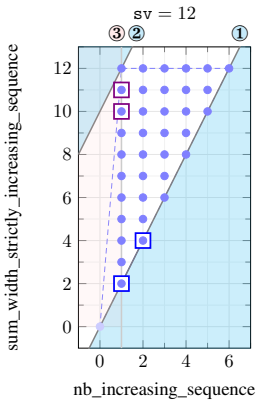


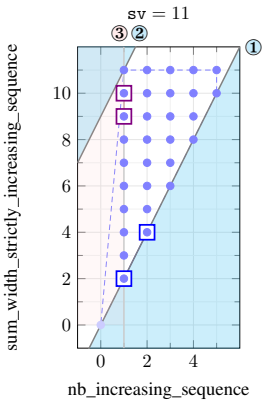
NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 4$: $(1, 2)$ $(2, 4)$

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 1)$ $(1, sv - 2)$





NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

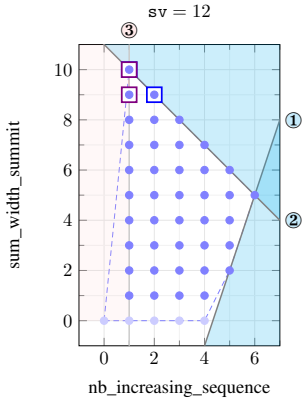
① $3 * x \leq y + sv + 1$

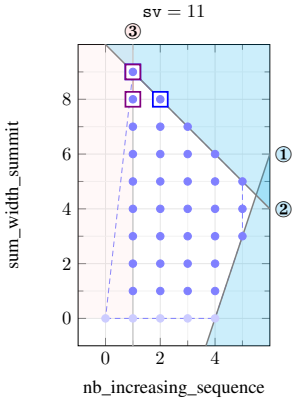
② $x + y \leq sv - 1$

- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

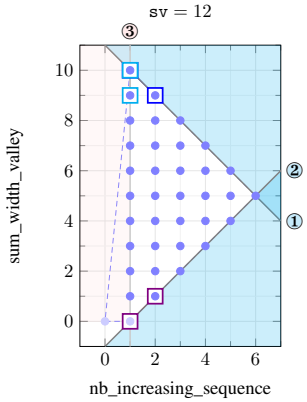
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

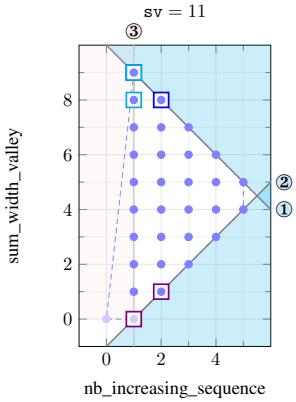
② $x \leq y + 1$

- $sv \geq 4$: $(1, 0)$ $(2, 1)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

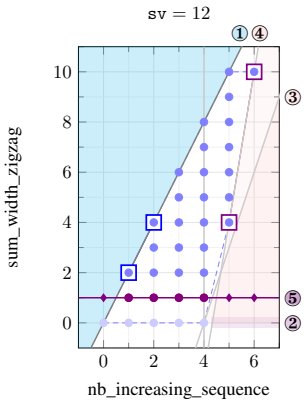
① $y \leq 2 * x$
- $sv \geq 6$: $(1, 2)$ $(2, 4)$

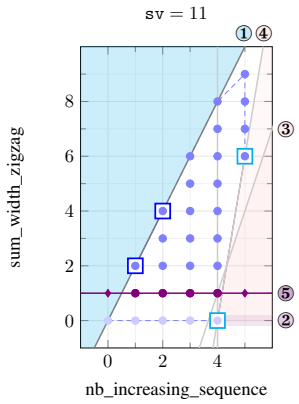
② $y = 0 \Rightarrow 3 * x \leq sv + 1 - (sv + 1) \bmod 3$

③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv + 2$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv - 8)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, sv - 5)$ $(\lfloor sv/2 \rfloor - 1, sv - 11)$

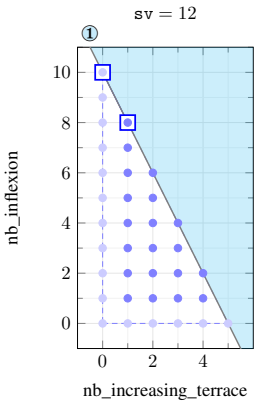
⑤ $y \neq 1$

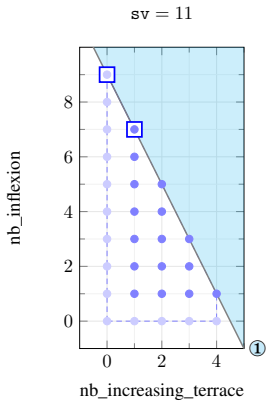




NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 4)$

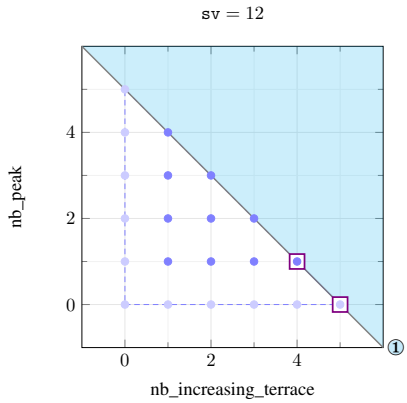


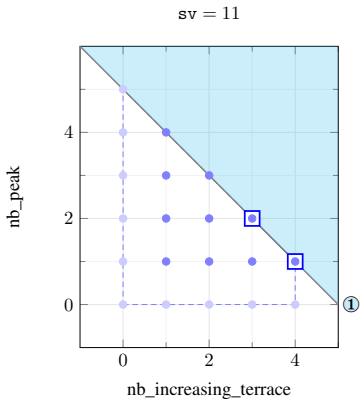


NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$

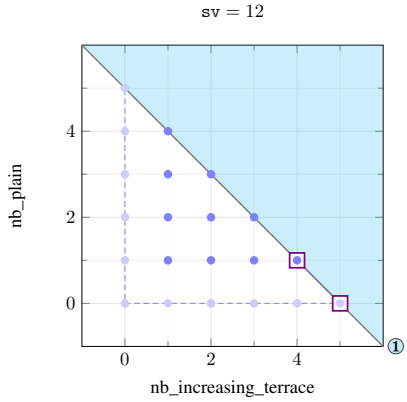


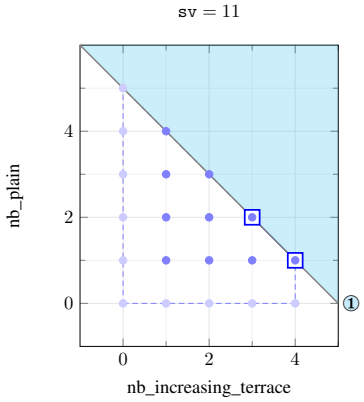


NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

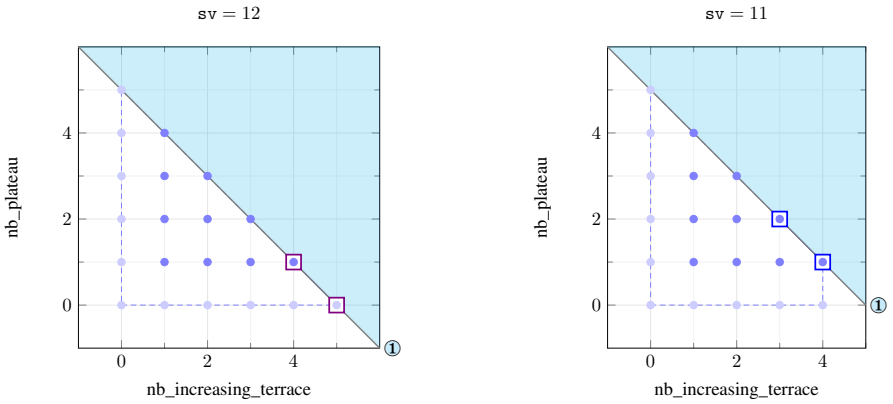
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$





NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

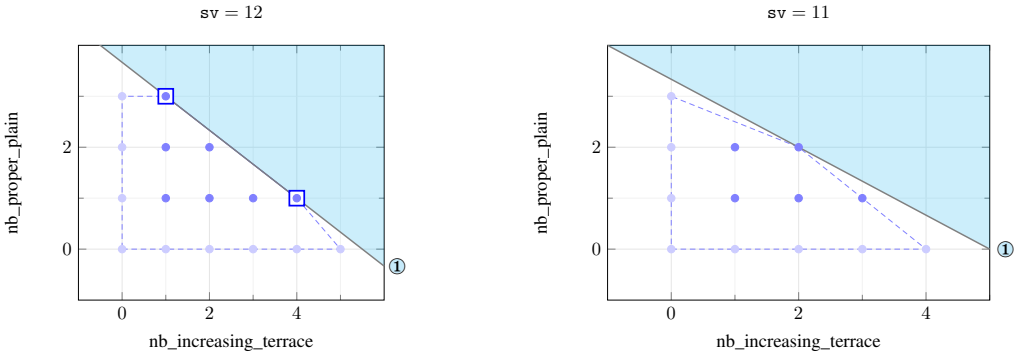
① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 3 * y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 10$: $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 4, 3)$

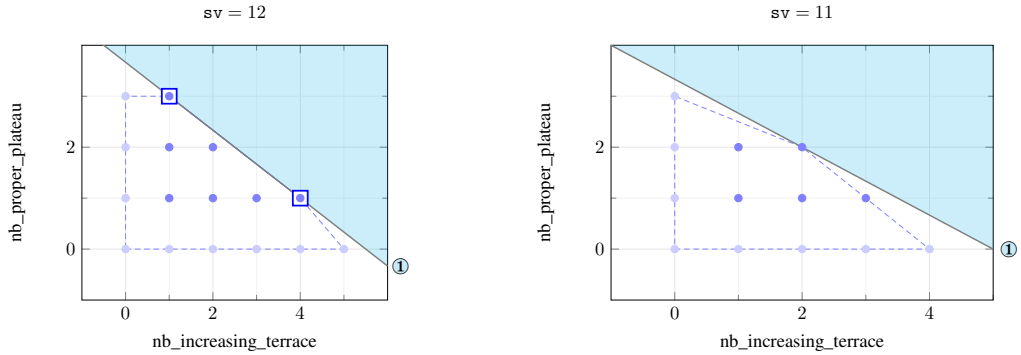
② $o^{=} \geq x + y$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 3 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-2)/2 \rfloor - 1, 1)$ $(\lfloor (sv-2)/2 \rfloor - 4, 3)$

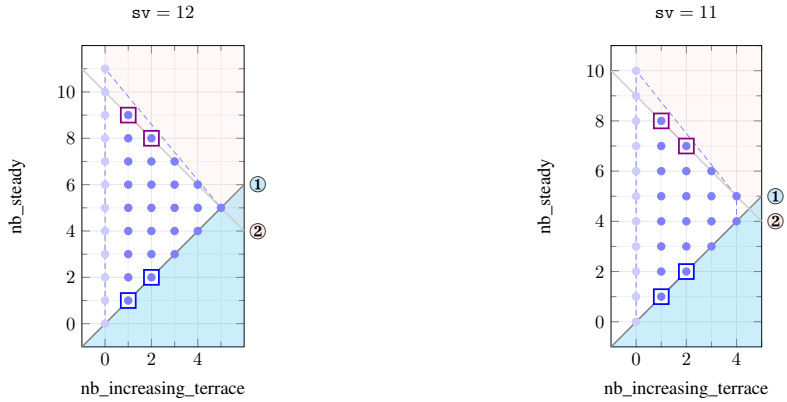
② $o^{=} \geq x + y$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (1, 1) (2, 2)

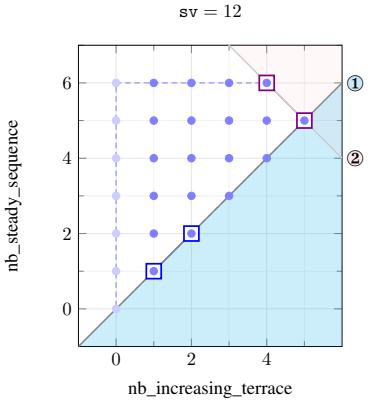
② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- sv ≥ 6: (1, sv − 3) (2, sv − 4)

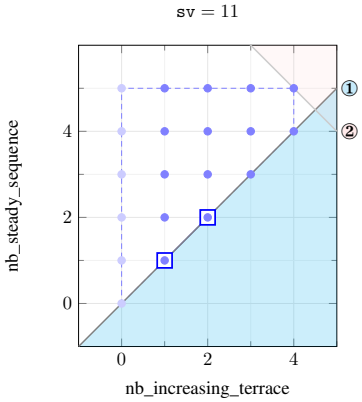


NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 6: (1, 1) (2, 2)

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 4: $(\lfloor (sv-2)/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor (sv-2)/2 \rfloor - 1, \lfloor sv/2 \rfloor)$

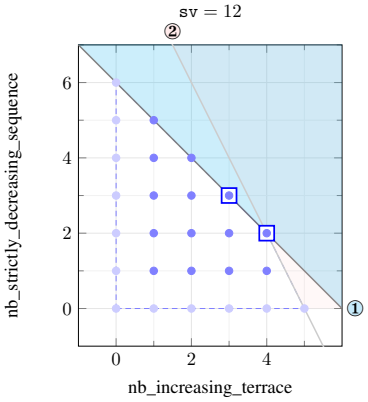


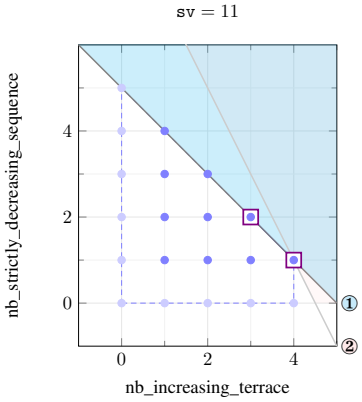


NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- sv mod 2 = 0 ∧ sv ≥ 6: $(\lfloor (sv-2)/2 \rfloor - 1, 2)$ $(\lfloor (sv-2)/2 \rfloor - 2, 3)$
- sv mod 2 = 1 ∧ sv ≥ 5: $(\lfloor (sv-2)/2 \rfloor, 1)$ $(\lfloor (sv-2)/2 \rfloor - 1, 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$



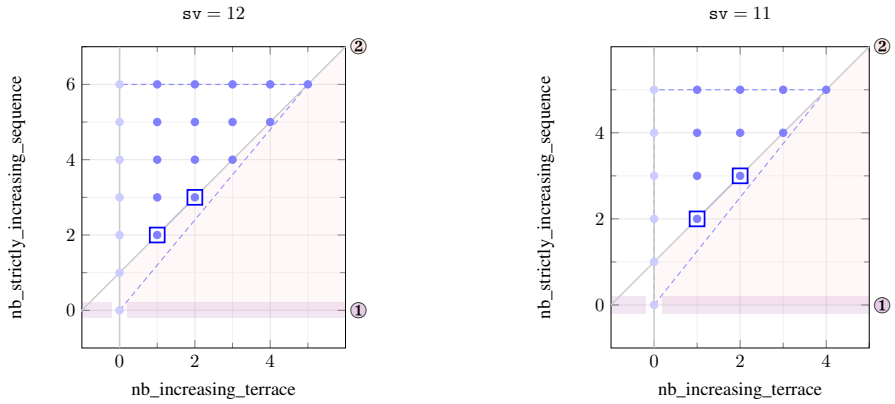


NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$

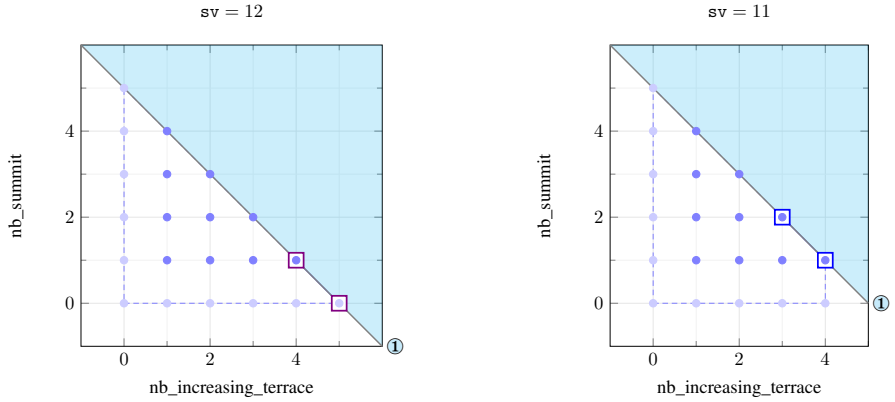
- $sv \geq 6$: (1, 2) (2, 3)



NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

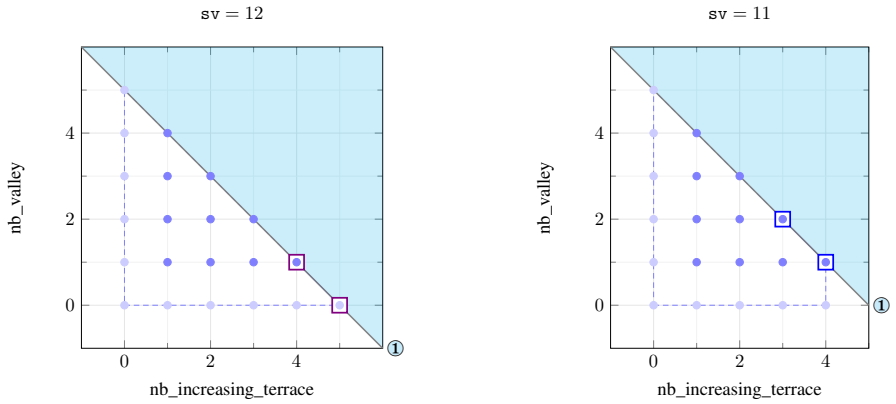
- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

- $(sv - 1) \bmod 2 = 0 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 1)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



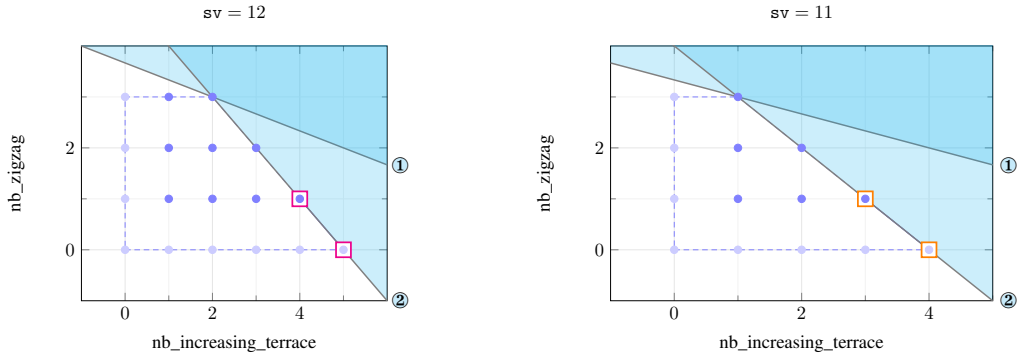
NB_INCREASING_TERRACE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + 3 * y \leq sv - 1$

- $sv \bmod 3 = 0 \wedge sv \geq 24$: $(2, \lfloor (sv - 1)/3 \rfloor)$ $(5, \lfloor (sv - 1)/3 \rfloor - 1)$
- $sv \bmod 3 = 1 \wedge sv \geq 16$: $(0, \lfloor (sv - 1)/3 \rfloor)$ $(3, \lfloor (sv - 1)/3 \rfloor - 1)$
- $sv \bmod 3 = 2 \wedge sv \geq 20$: $(1, \lfloor (sv - 1)/3 \rfloor)$ $(4, \lfloor (sv - 1)/3 \rfloor - 1)$

② $sv > 1 \Rightarrow 2 * x + 2 * y \leq sv - 2 - (sv - 2) \bmod 2$

- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor, 0)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 1)$



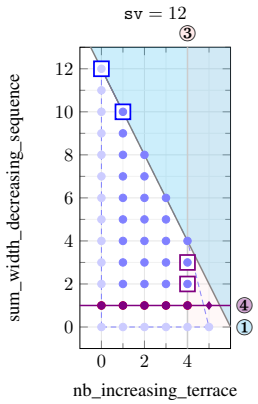
NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

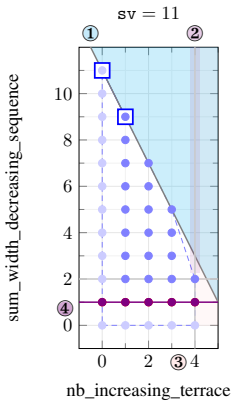
① $2 * x + y \leq sv$
- $sv \geq 6$: $(0, sv)$ $(1, sv - 2)$

② $x = \max(0, \lfloor (sv - 2)/2 \rfloor) \wedge sv \bmod 2 = 1 \wedge sv > 3 \Rightarrow y \leq 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

④ $y \neq 1$





NB_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_DECREASING_TERRACE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\text{VARIABLES}|$

① $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv}-2)/2\rfloor, 0)$ $(0, \mathtt{sv}-2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 9$: $(1, \mathtt{sv}-5)$ $(2, \mathtt{sv}-7)$

③ $\mathtt{o}^{=} \geq \mathtt{x} + \lfloor(\mathtt{y}+1)/2\rfloor$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4, \\ \mathtt{sv} < 7 \end{array} \right)$

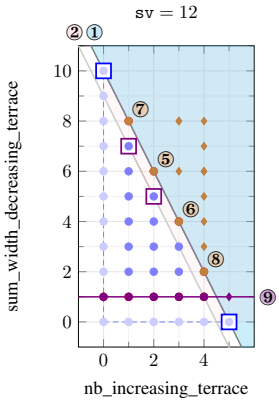
⑤ $\mathtt{x} \neq 2 \vee \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4 \vee \mathtt{sv} < 7$

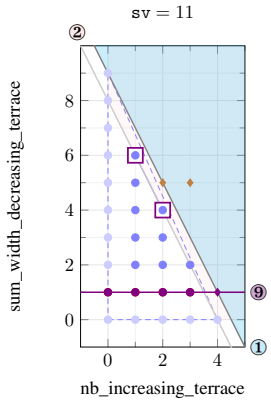
⑥ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 2, \\ \mathtt{y} < 3, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

⑦ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑧ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv}-2)/2\rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ 1 = \mathtt{sv} \bmod 2 \end{array} \right)$

⑨ $\mathtt{y} \neq 1$





NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

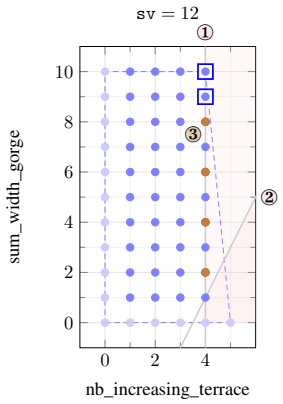
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

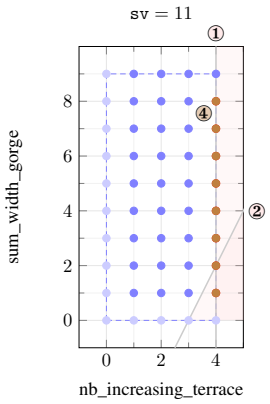
□ $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 5$

③ $\bigvee \left\{ \begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right.$

④ $\bigvee \left\{ \begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor), \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right.$





NB_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INCREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{y} = 0 \Rightarrow \mathtt{x} = 0$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} - 2$

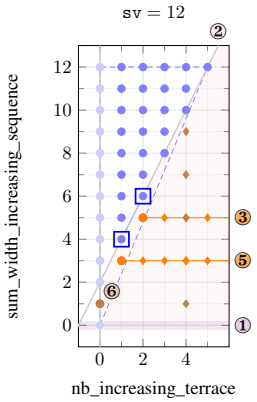
□ $\mathtt{sv} \geq 6$: $(1, 4)$ $(2, 6)$

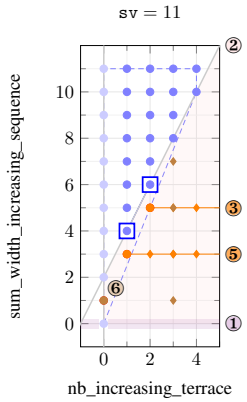
③ $\mathtt{x} < 2 \vee \mathtt{y} \neq 5$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor (\mathtt{sv} - 2)/2 \rfloor) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑤ $\mathtt{x} < 1 \vee \mathtt{y} \neq 3$

⑥ $\mathtt{x} \neq 0 \vee \mathtt{y} \neq 1$

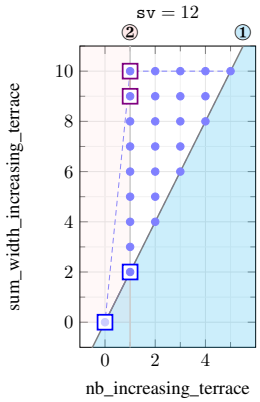


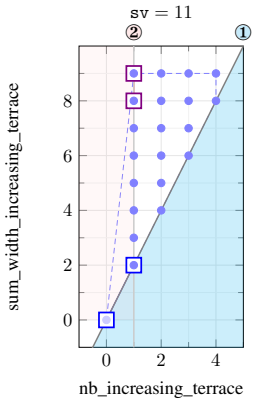


NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)





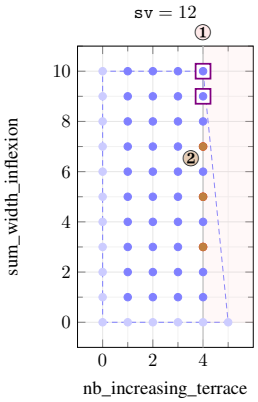
NB_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INFLEXION($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

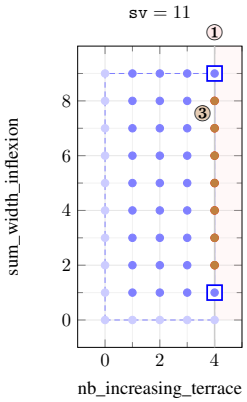
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

- $(\mathtt{sv} - 1) \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor, 1)$
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 1, \\ \mathtt{y} < 3, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor), \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right)$





NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

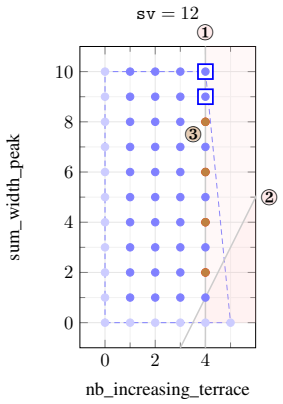
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

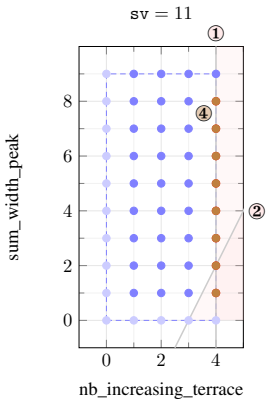
□ $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor - 1, sv - 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv - 3)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 5$

③ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor) - 1, \\ y < 2, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ 1 = sv \bmod 2, \\ 1 = y \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1, \\ 0 = sv \bmod 2 \end{array} \right)$

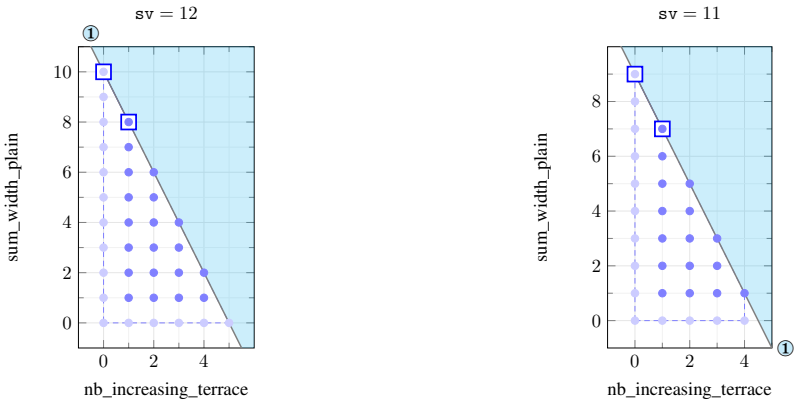




NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

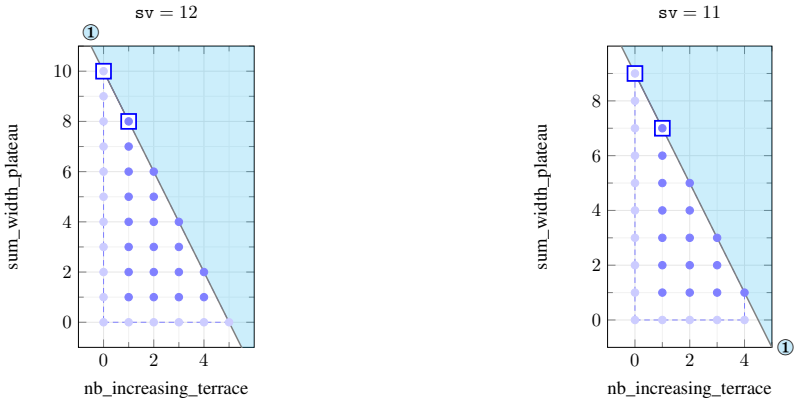
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 4)$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 4)$



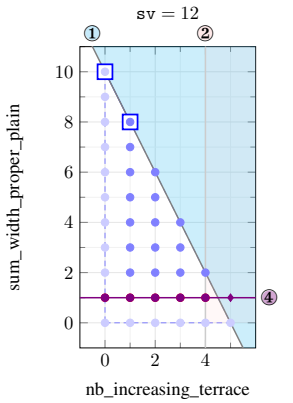
NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

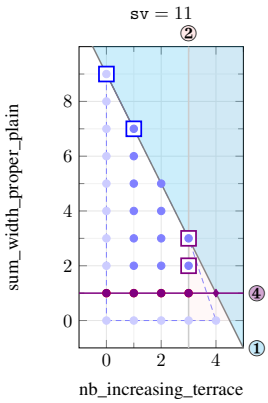
① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv-2)$ $(1, sv-4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv-4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv-2)/2 \rfloor - 1, 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y+1)/2 \rfloor$

④ $y \neq 1$





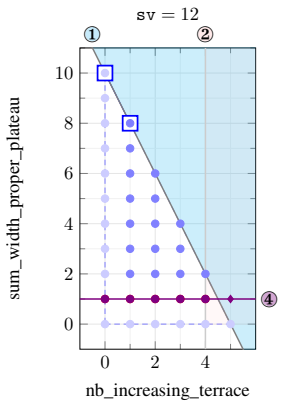
NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

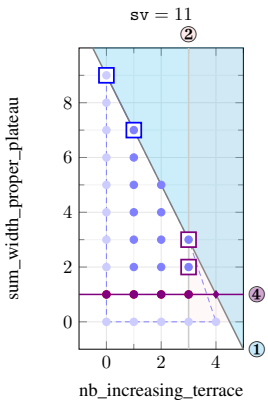
① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv-2)$ $(1, sv-4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv-4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv-2)/2 \rfloor - 1, 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y+1)/2 \rfloor$

④ $y \neq 1$



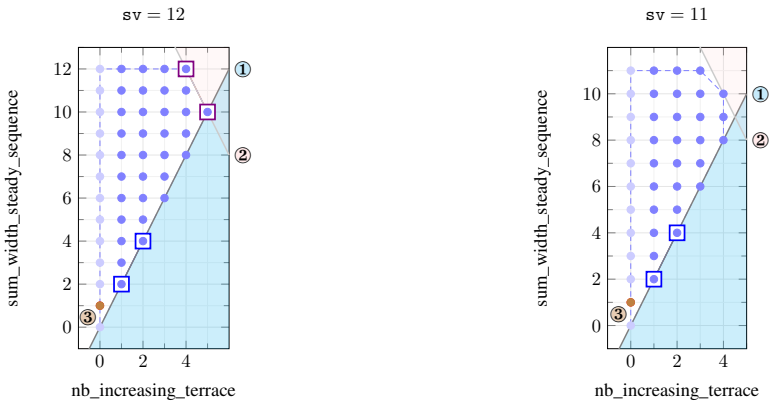


NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 6$: $(1, 2)$ $(2, 4)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor, sv-2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv)$

③ $x \neq 0 \vee y \neq 1$



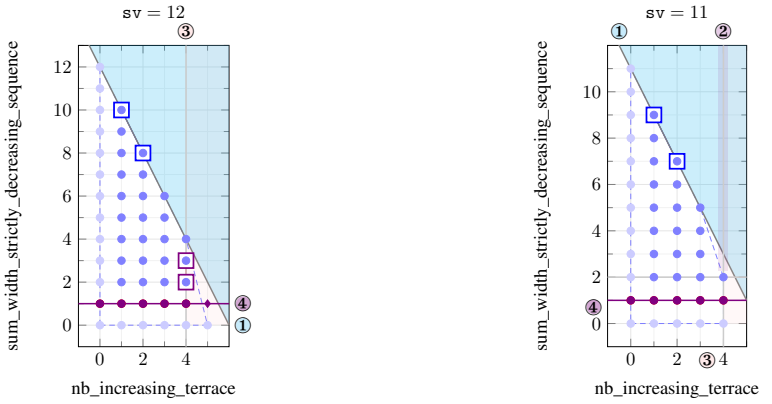
NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \geq 8$: $(1, sv-2)$ $(2, sv-4)$

② $x = \max(0, \lfloor (sv-2)/2 \rfloor) \wedge sv \bmod 2 = 1 \wedge sv > 3 \Rightarrow y \leq 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv-3) \bmod 2$
- $(sv-1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor - 1, 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, 3)$

④ $y \neq 1$



NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow x = 0$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y - 2$

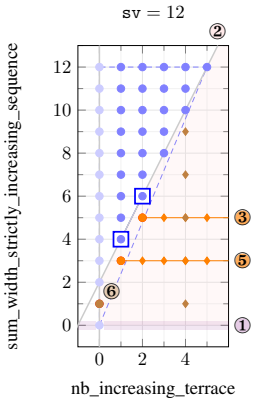
□ $sv \geq 6$: (1, 4) (2, 6)

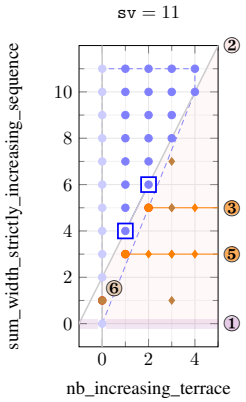
③ $x < 2 \vee y \neq 5$

④ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv - 2)/2 \rfloor) - 1, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{array} \right)$

⑤ $x < 1 \vee y \neq 3$

⑥ $x \neq 0 \vee y \neq 1$





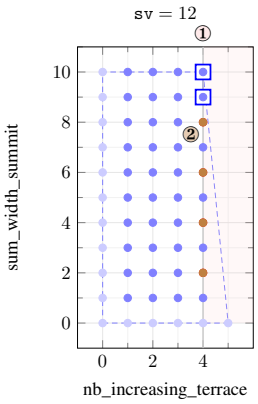
NB_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_SUMMIT($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

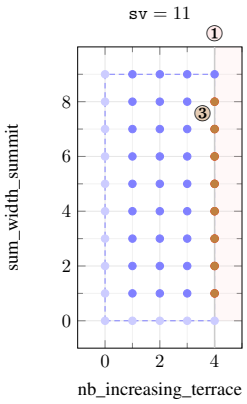
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 2)$ $(\lfloor(\mathtt{sv} - 2)/2\rfloor - 1, \mathtt{sv} - 3)$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ 1 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor(\mathtt{sv} - 2)/2\rfloor), \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ 0 = \mathtt{sv} \bmod 2 \end{array} \right)$





NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

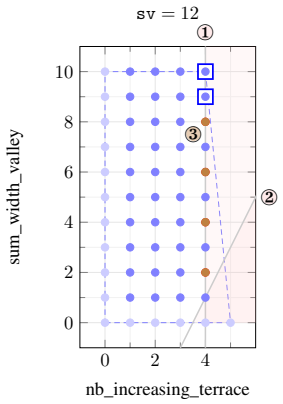
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

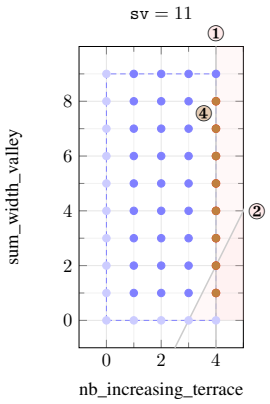
□ $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor (sv-2)/2 \rfloor - 1, sv - 2)$ $(\lfloor (sv-2)/2 \rfloor - 1, sv - 3)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 5$

③ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor) - 1, \\ y < 2, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ 1 = sv \bmod 2, \\ 1 = y \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} x \neq \max(0, \lfloor (sv-2)/2 \rfloor), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1, \\ 0 = sv \bmod 2 \end{array} \right)$



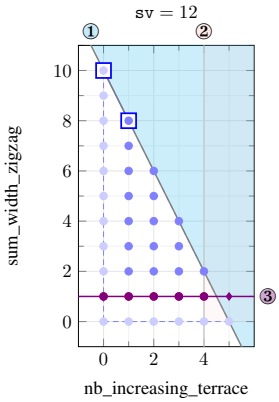


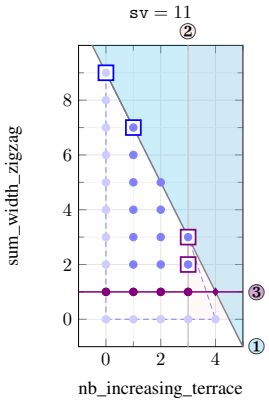
NB_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(\lfloor (sv - 2)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 2)/2 \rfloor - 1, 3)$

③ $y \neq 1$

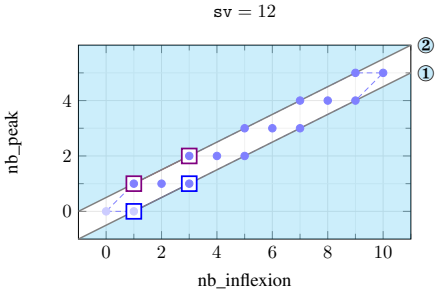


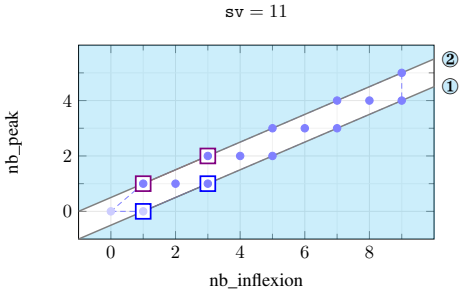


NB_INFLEXION(x, VARIABLES) ∧
NB_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- $sv \geq 5$: $(1, 0)$ $(3, 1)$

② $2 * y \leq x + 1$
- $sv \geq 5$: $(1, 1)$ $(3, 2)$

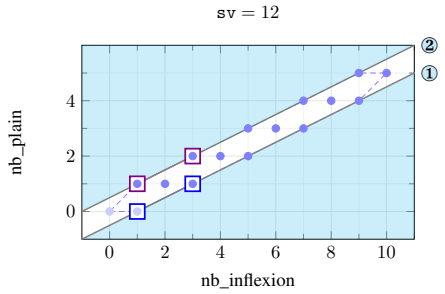


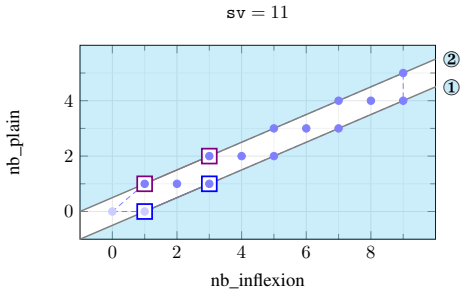


NB_INFLEXION(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- sv ≥ 5: (1, 0) (3, 1)

② $2 * y \leq x + 1$
- sv ≥ 5: (1, 1) (3, 2)

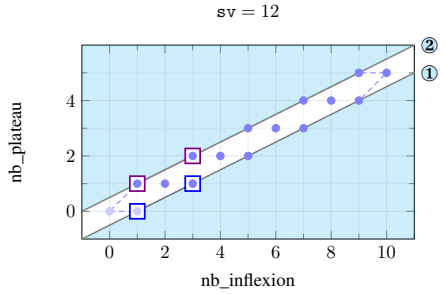


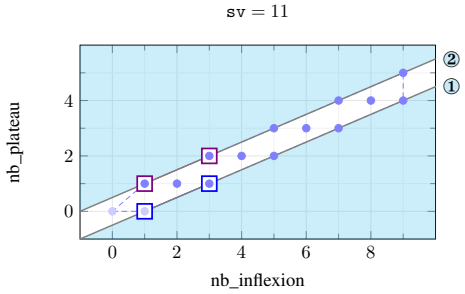


NB_INFLEXION(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- sv ≥ 5: (1, 0) (3, 1)

② $2 * y \leq x + 1$
- sv ≥ 5: (1, 1) (3, 2)

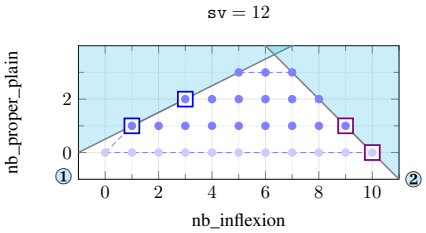


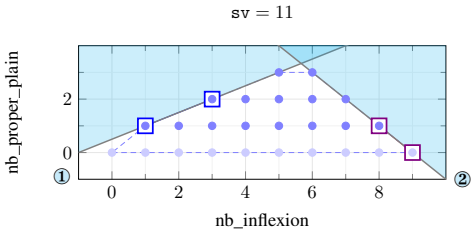


NB_INFLEXION(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * y \leq x + 1$
- sv ≥ 7: (1, 1) (3, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 4: (sv − 2, 0) (sv − 3, 1)

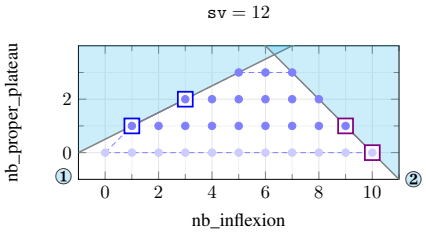


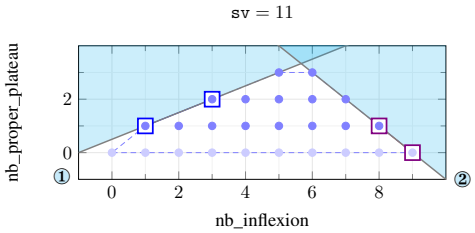


NB_INFLEXION(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * y \leq x + 1$
- sv ≥ 7: (1, 1) (3, 2)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 4: (sv − 2, 0) (sv − 3, 1)

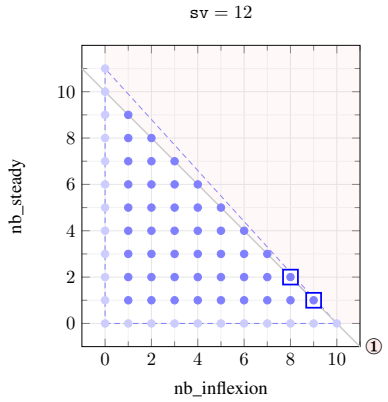


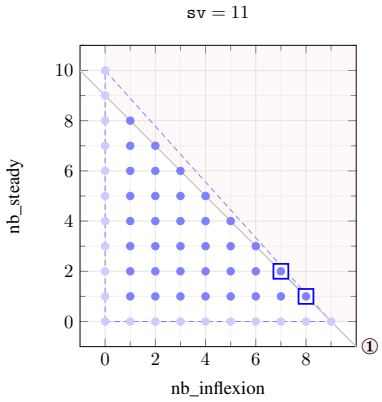


NB_INFLEXION(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$

- $sv \geq 4$: $(sv - 3, 1)$ $(sv - 4, 2)$

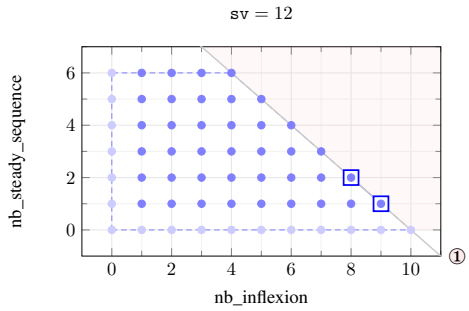


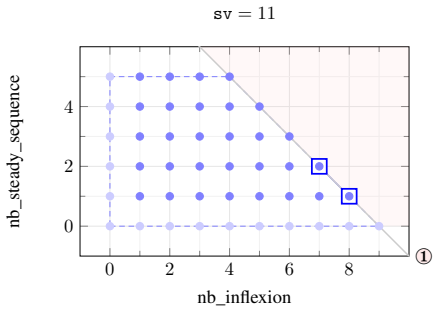


NB_INFLEXION(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$

- $sv \geq 4$: $(sv - 3, 1)$ $(sv - 4, 2)$





NB_INFLEXION($\mathtt{x}$, VARIABLES) $\wedge$
NB_STRICTLY_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

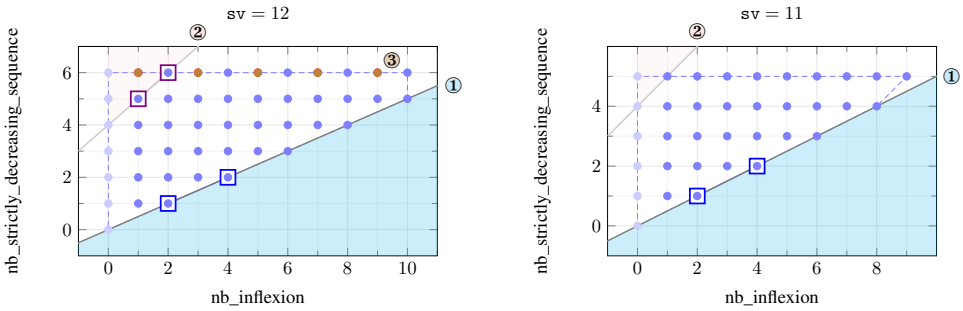
① $\mathtt{x} \leq 2 * \mathtt{y}$

- $\mathtt{sv} \geq 6$: (2, 1) (4, 2)

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{y} \leq 2 * \mathtt{x} + \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$

- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 4$: (1, $\lfloor \mathtt{sv}/2 \rfloor - 1$) (2, $\lfloor \mathtt{sv}/2 \rfloor$)

③ $$\bigvee \begin{pmatrix} \mathtt{y} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{max}(0, \mathtt{sv} - 2) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{x} \bmod 2 \end{pmatrix}$$

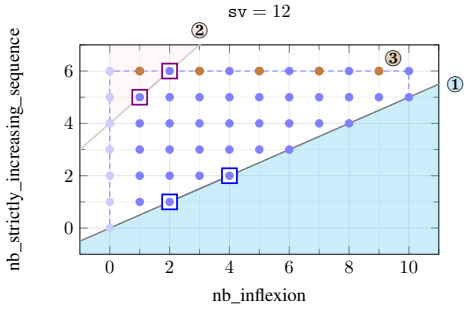


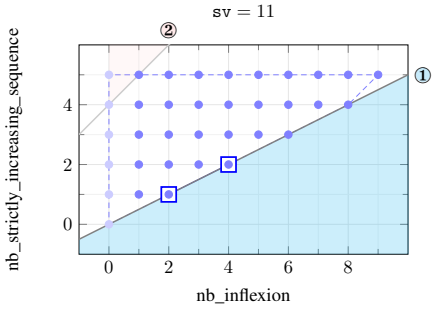
NB_INFLEXION(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y$
- $sv \geq 6$: (2, 1) (4, 2)

② $x > 0 \wedge y > 0 \Rightarrow 2 * y \leq 2 * x + sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(1, \lfloor sv/2 \rfloor - 1)$ $(2, \lfloor sv/2 \rfloor)$

③ $\bigvee \begin{pmatrix} y \neq \lfloor sv/2 \rfloor, \\ x < 1, \\ x > \max(0, sv - 2) - 1, \\ 1 = sv \bmod 2, \\ 0 = x \bmod 2 \end{pmatrix}$

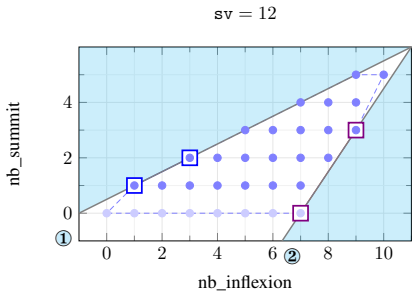


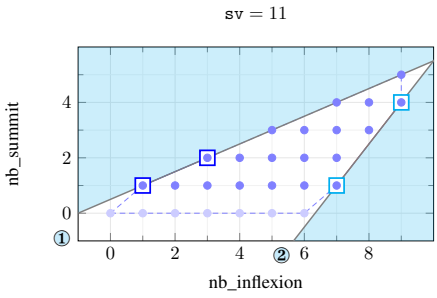


NB_INFLEXION(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $2 * y \leq x + 1$
- $sv \geq 5$: (1, 1) (3, 2)

② $sv > 1 \Rightarrow 3 * x \leq 2 * y + 2 * sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(sv - 3, \lfloor (sv - 1)/2 \rfloor - 2)$ $(sv - 5, \lfloor (sv - 1)/2 \rfloor - 5)$
- $sv \bmod 2 = 1 \wedge sv \geq 9$: $(sv - 2, \lfloor (sv - 1)/2 \rfloor - 1)$ $(sv - 4, \lfloor (sv - 1)/2 \rfloor - 4)$

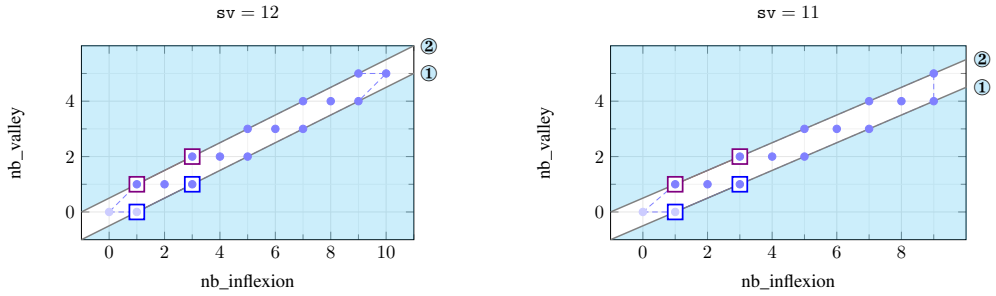




NB_INFLEXION(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- $sv \geq 5$: (1, 0) (3, 1)

② $2 * y \leq x + 1$
- $sv \geq 5$: (1, 1) (3, 2)

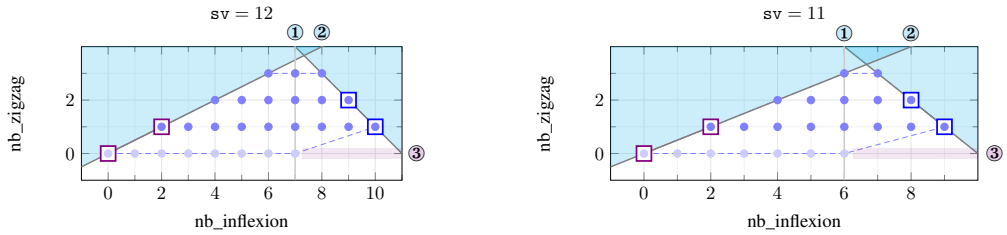


NB_INFLEXION(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 7$: (sv − 2, 1) (sv − 3, 2)

② $2 * y \leq x$
- $sv \geq 4$: (0, 0) (2, 1)

③ $y = 0 \wedge sv > 3 \Rightarrow 3 * x \leq 2 * sv - 3 - (2 * sv - 3) \bmod 3$



NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- □ $sv \bmod 2 = 0 \wedge sv \geq 4$: $(sv-2, sv-2)$ $(sv-4, sv-4)$
- □ $sv \bmod 2 = 1 \wedge sv \geq 5$: $(sv-3, sv-3)$ $(sv-5, sv-5)$

② $x = \max(0, sv-2) \wedge sv \bmod 2 = 1 \Rightarrow y \leq sv-1$

③ $x > 0 \Rightarrow y \geq 2$
- □ $sv \geq 4$: $(1, 2)$ $(2, 2)$

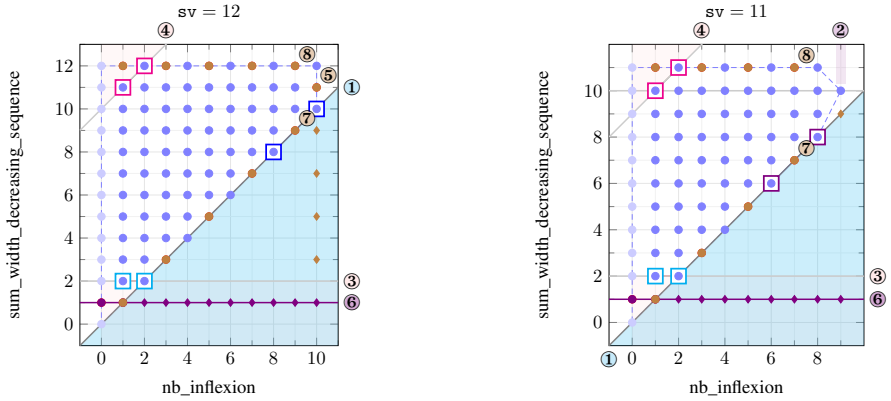
④ $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- □ $sv \geq 4$: $(1, sv-1)$ $(2, sv)$

⑤ $$\bigvee \begin{pmatrix} x \neq \max(0, sv-2), \\ y < 3, \\ y > sv * \min(1, \max(0, sv-1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{pmatrix}$$

⑥ $y \neq 1$

⑦ $y \neq x \vee 0 = x \bmod 2$

⑧ $$\bigvee \begin{pmatrix} y \neq sv * \min(1, \max(0, sv-1)), \\ x < 1, \\ x > \max(0, sv-2) - 1, \\ 0 = x \bmod 2 \end{pmatrix}$$

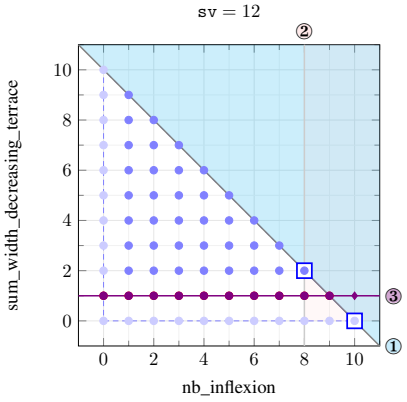


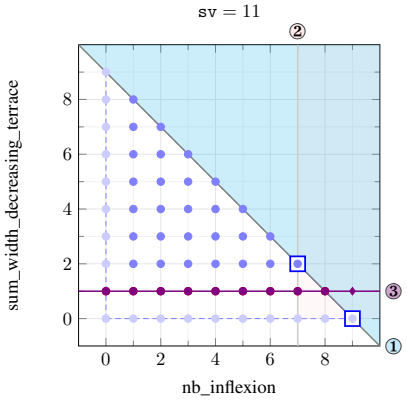
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$



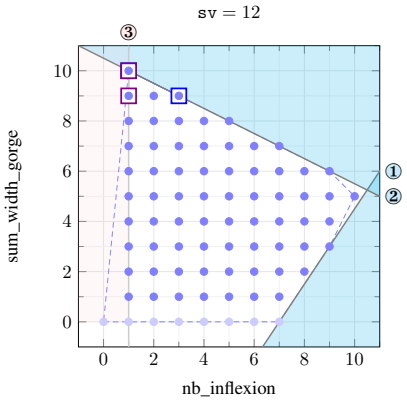


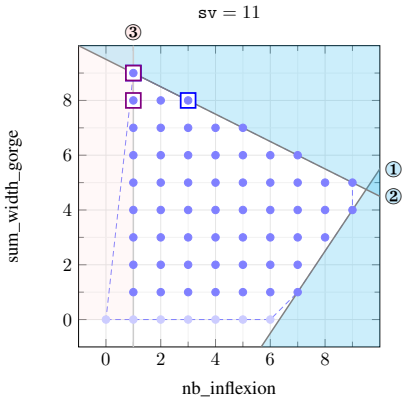
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow 3 * x \leq 2 * y + 2 * sv - 3$

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- $sv \geq 5$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv mod 2 = 0 ∧ sv ≥ 4: (sv − 2, sv − 2) (sv − 4, sv − 4)
- sv mod 2 = 1 ∧ sv ≥ 5: (sv − 3, sv − 3) (sv − 5, sv − 5)

② $x = \max(0, sv - 2) \wedge sv \bmod 2 = 1 \Rightarrow y \leq sv - 1$

③ $x > 0 \Rightarrow y \geq 2$
- sv ≥ 4: (1, 2) (2, 2)

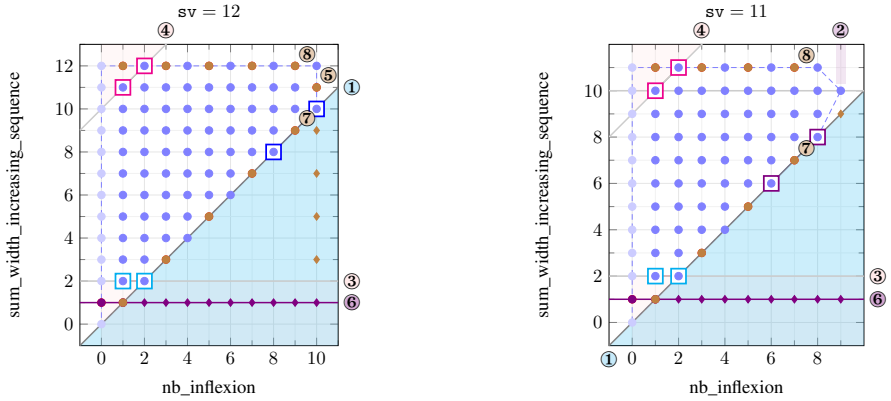
④ $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- sv ≥ 4: (1, sv − 1) (2, sv)

⑤ $\bigvee \left(\begin{array}{l} x \neq \max(0, sv - 2), \\ y < 3, \\ y > sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

⑥ $y \neq 1$

⑦ $y \neq x \vee 0 = x \bmod 2$

⑧ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > \max(0, sv - 2) - 1, \\ 0 = x \bmod 2 \end{array} \right)$

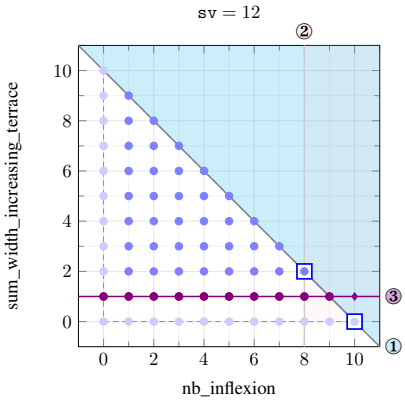


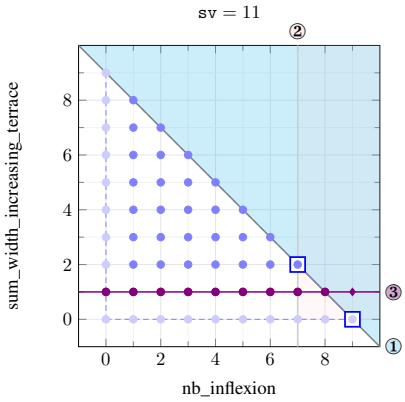
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$

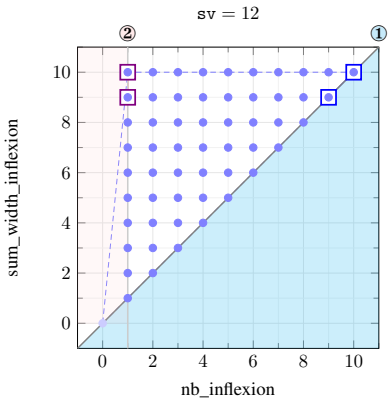


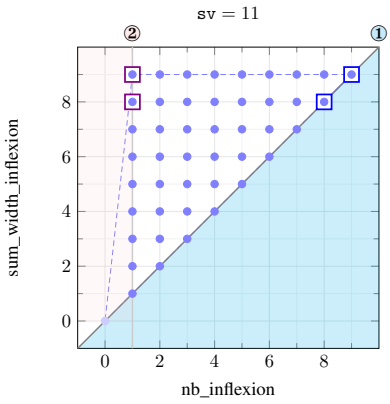


NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$



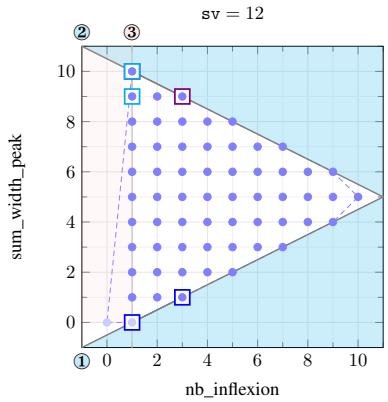


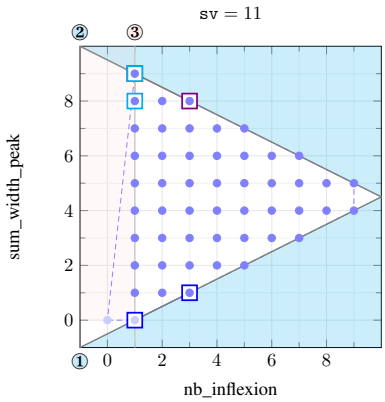
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- sv ≥ 5: (1, 0) (3, 1)

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- sv ≥ 5: (1, sv − 2) (3, sv − 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



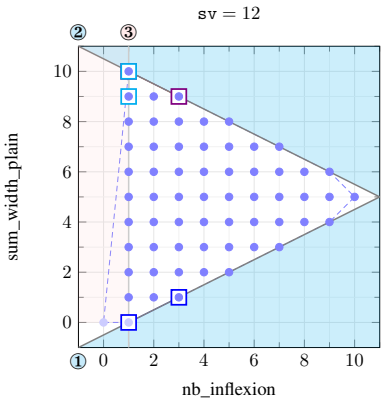


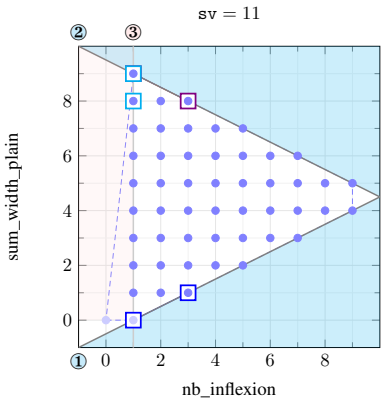
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- $sv \geq 5$: $(1, 0)$ $(3, 1)$

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- $sv \geq 5$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



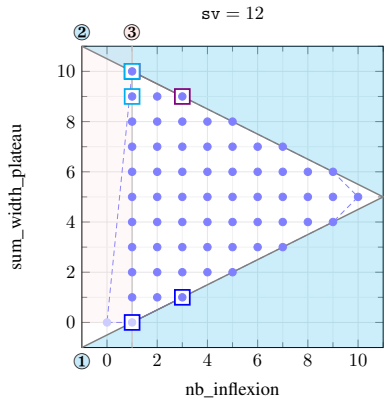


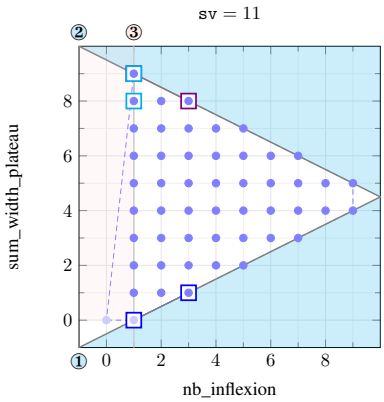
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$
- sv ≥ 5: (1, 0) (3, 1)

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- sv ≥ 5: (1, sv − 2) (3, sv − 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)





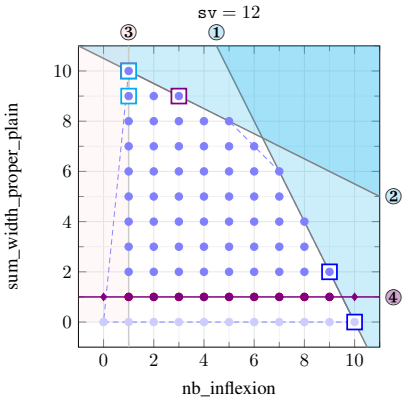
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

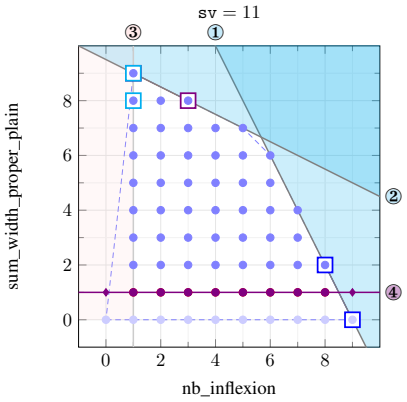
① $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 3, 2)$

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- $sv \geq 7$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





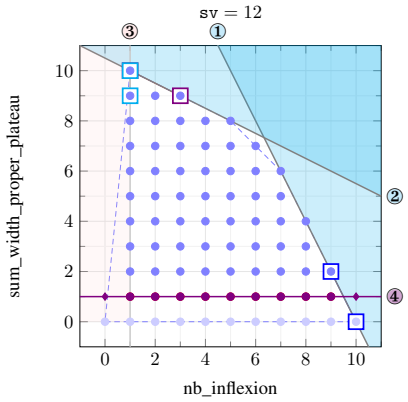
NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

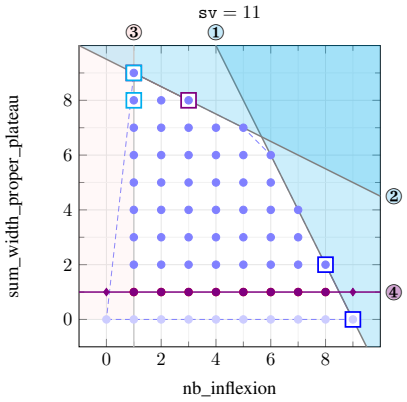
① $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 3, 2)$

② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$
- $sv \geq 7$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$

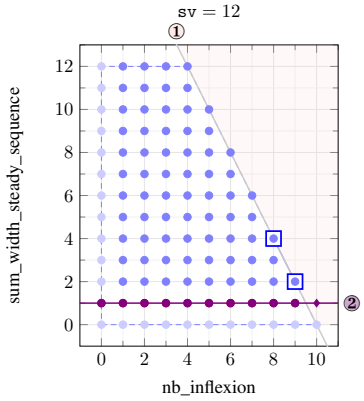


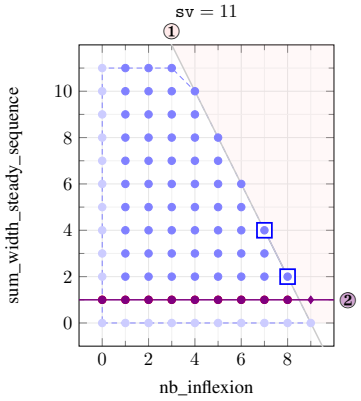


NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 4$: $(sv - 3, 2)$ $(sv - 4, 4)$

② $y \neq 1$





NB_INFLEXION($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} \leq \mathtt{y}$
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\mathtt{sv}-2, \mathtt{sv}-2)$ $(\mathtt{sv}-4, \mathtt{sv}-4)$
- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 5$: $(\mathtt{sv}-3, \mathtt{sv}-3)$ $(\mathtt{sv}-5, \mathtt{sv}-5)$

② $\mathtt{x} = \mathtt{max}(0, \mathtt{sv}-2) \wedge \mathtt{sv} \bmod 2 = 1 \Rightarrow \mathtt{y} \leq \mathtt{sv}-1$

③ $\mathtt{x} > 0 \Rightarrow \mathtt{y} \geq 2$
- $\mathtt{sv} \geq 4$: $(1, 2)$ $(2, 2)$

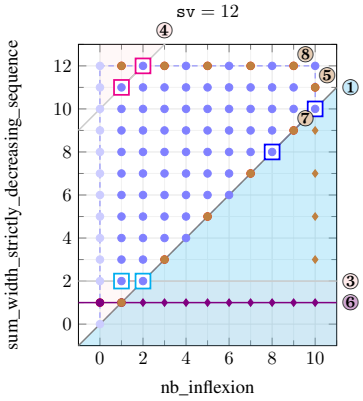
④ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 2$
- $\mathtt{sv} \geq 4$: $(1, \mathtt{sv}-1)$ $(2, \mathtt{sv})$

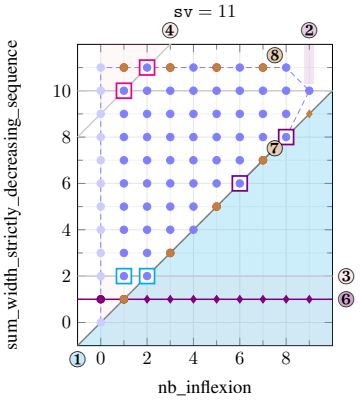
⑤ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv}-2), \\ \mathtt{y} < 3, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑥ $\mathtt{y} \neq 1$

⑦ $\mathtt{y} \neq \mathtt{x} \vee 0 = \mathtt{x} \bmod 2$

⑧ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)), \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{max}(0, \mathtt{sv}-2) - 1, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$





NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv mod 2 = 0 ∧ sv ≥ 4: (sv − 2, sv − 2) (sv − 4, sv − 4)
- sv mod 2 = 1 ∧ sv ≥ 5: (sv − 3, sv − 3) (sv − 5, sv − 5)

② $x = \max(0, sv - 2) \wedge sv \bmod 2 = 1 \Rightarrow y \leq sv - 1$

③ $x > 0 \Rightarrow y \geq 2$
- sv ≥ 4: (1, 2) (2, 2)

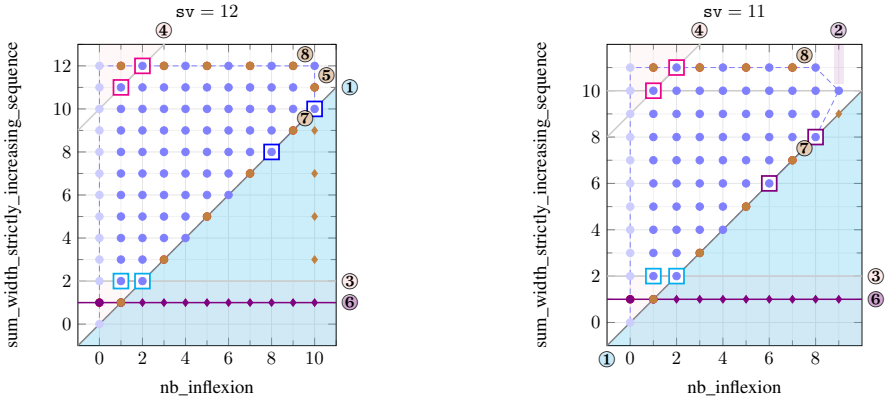
④ $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- sv ≥ 4: (1, sv − 1) (2, sv)

⑤ $\bigvee \left(\begin{array}{l} x \neq \max(0, sv - 2), \\ y < 3, \\ y > sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

⑥ $y \neq 1$

⑦ $y \neq x \vee 0 = x \bmod 2$

⑧ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > \max(0, sv - 2) - 1, \\ 0 = x \bmod 2 \end{array} \right)$



NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

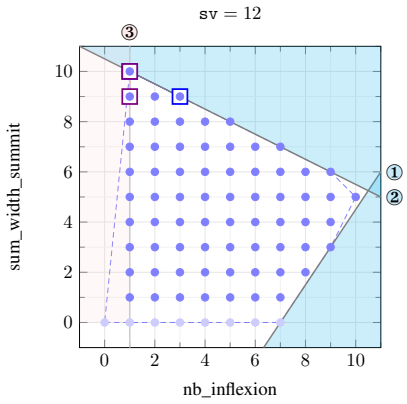
① $sv > 1 \Rightarrow 3 * x \leq 2 * y + 2 * sv - 3$

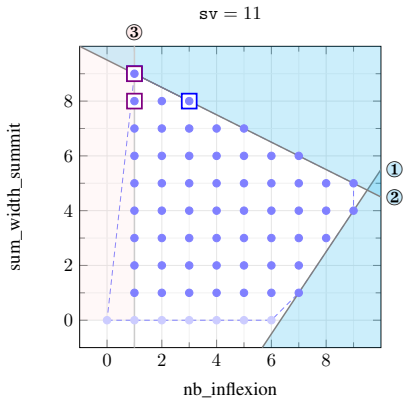
② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$

- $sv \geq 5$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq 2 * y + 1$

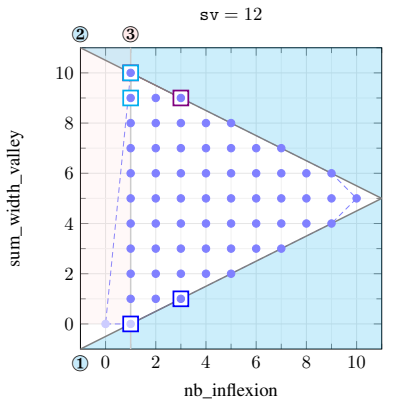
- $sv \geq 5$: $(1, 0)$ $(3, 1)$

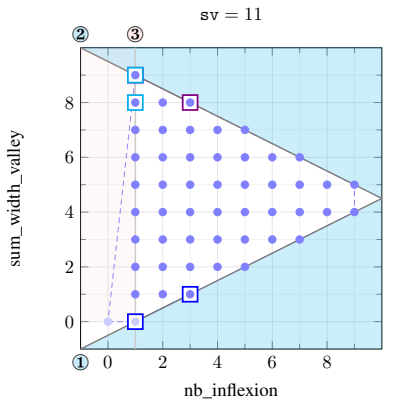
② $sv > 1 \Rightarrow x + 2 * y \leq 2 * sv - 3$

- $sv \geq 5$: $(1, sv - 2)$ $(3, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





NB_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

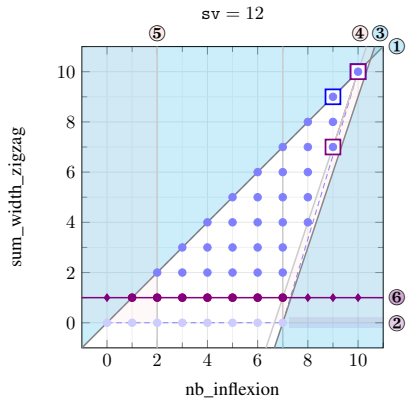
② $y = 0 \wedge sv > 3 \Rightarrow 3 * x \leq 2 * sv - 3 - (2 * sv - 3) \bmod 3$

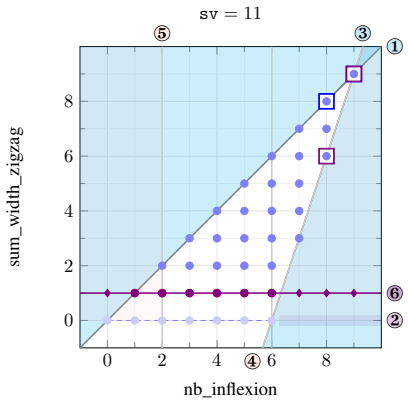
③ $sv > 1 \Rightarrow 3 * x \leq y + 2 * sv - 4 + (sv \bmod 3 = 0)$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + 2 * sv - 4$
- $sv \geq 7$: $(sv-2, sv-2)$ $(sv-3, sv-5)$

⑤ $y > 0 \Rightarrow x \geq 2$

⑥ $y \neq 1$



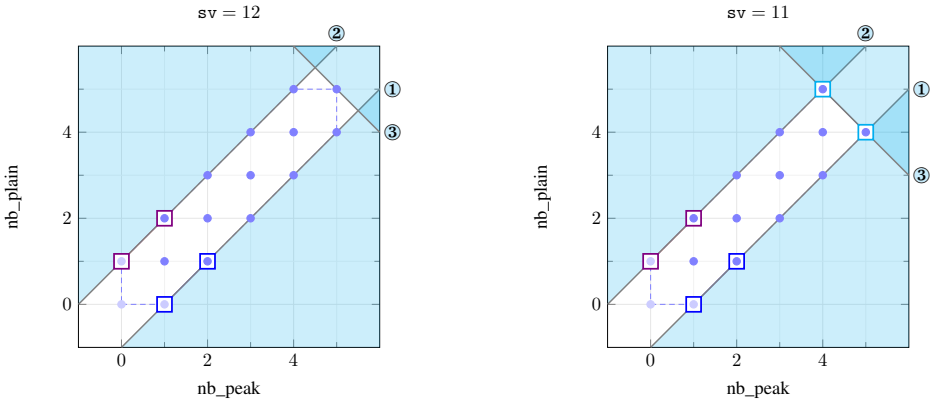


NB_PEAK(x, VARIABLES) ∧
NB_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $y \leq x + 1$
- sv ≥ 5: (0, 1) (1, 2)

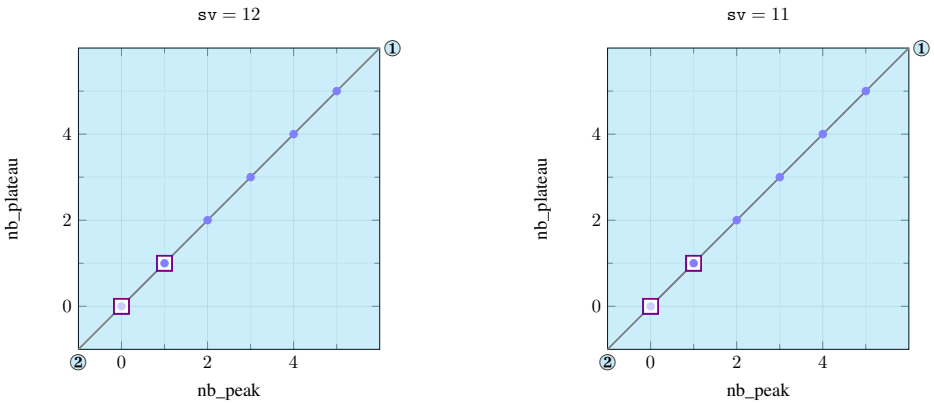
③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor)$



NB_PEAK(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

② $y \leq x$
- sv ≥ 3: (0, 0) (1, 1)

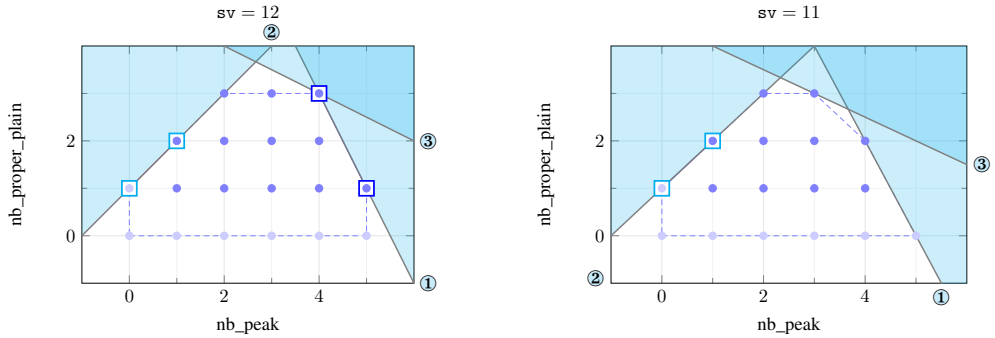


NB_PEAK(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

② $y \leq x + 1$
- sv ≥ 7: (0, 1) (1, 2)

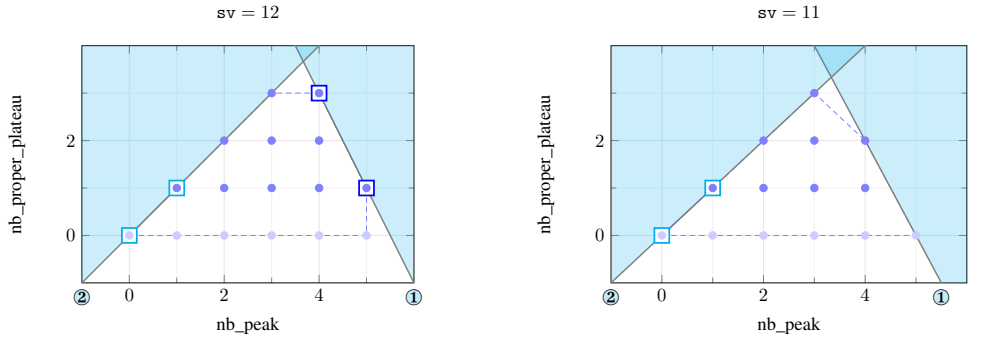
③ $sv > 1 \Rightarrow x + 2 * y \leq sv - 2$



NB_PEAK(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

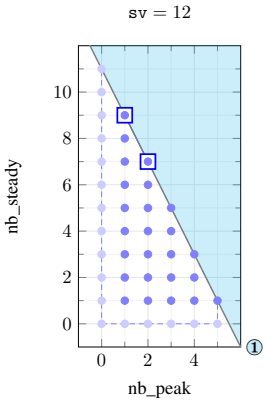
② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

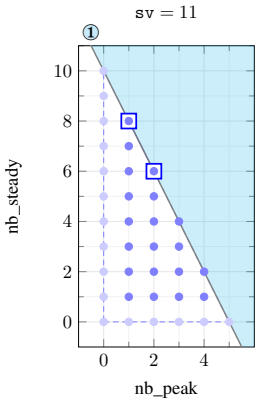


NB_PEAK(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 3)$ $(2, sv - 5)$

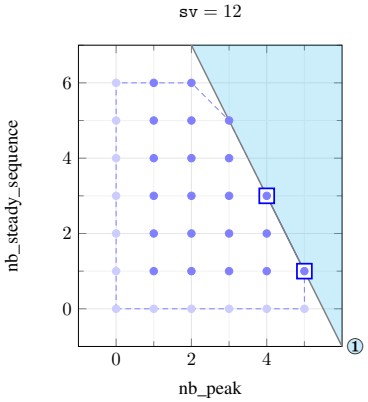


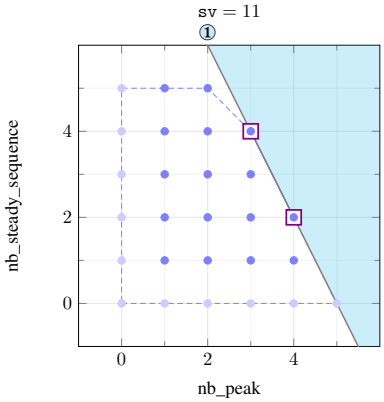


NB_PEAK(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

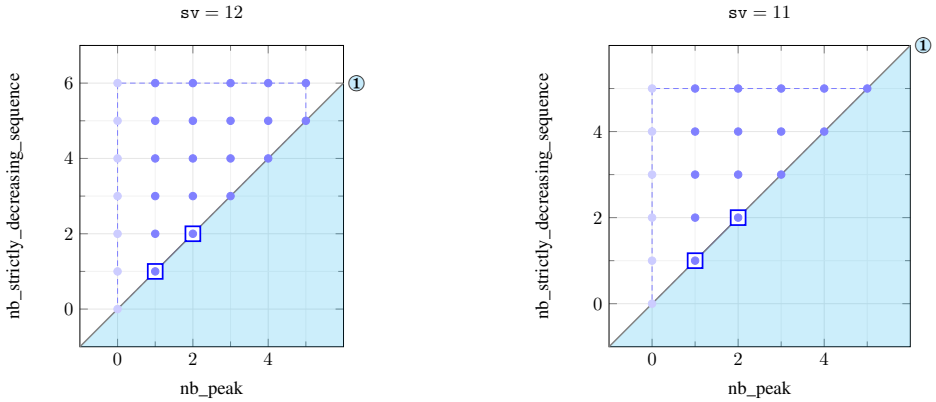
- $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor, 1)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$
- $sv \bmod 2 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 2, 4)$





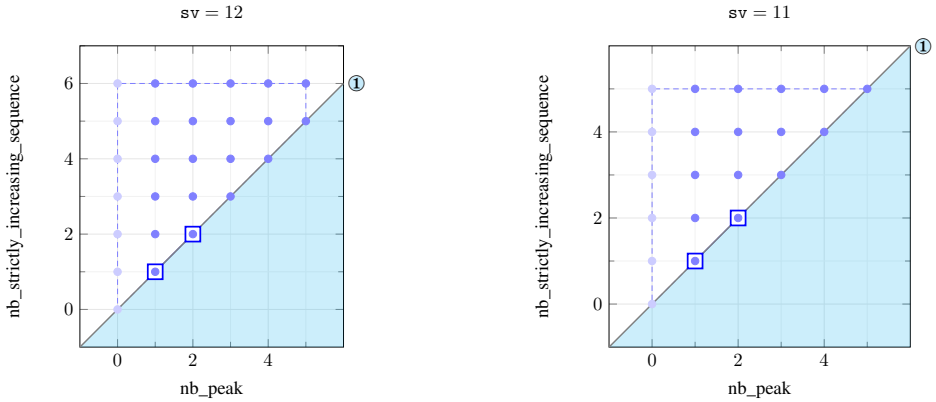
NB_PEAK(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① x ≤ y
- sv ≥ 5: (1, 1) (2, 2)



NB_PEAK(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

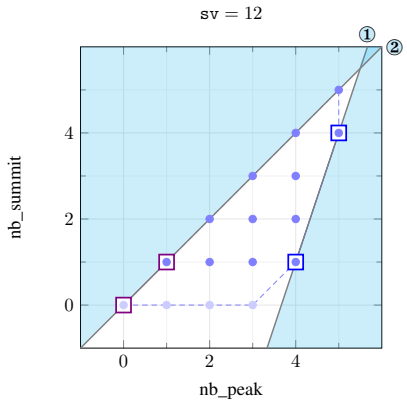
① x ≤ y
- sv ≥ 5: (1, 1) (2, 2)

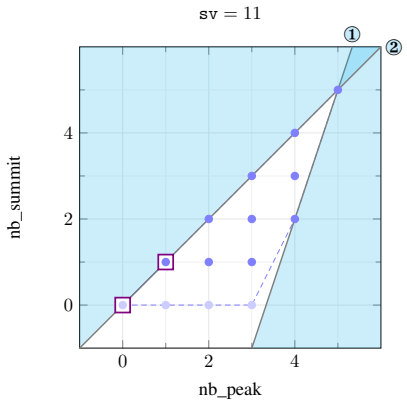


NB_PEAK(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 4)$

② $y \leq x$
- sv ≥ 3: $(0, 0)$ $(1, 1)$



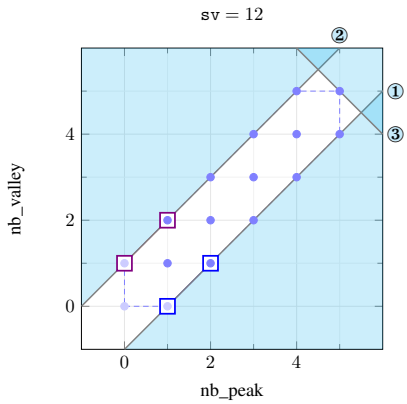


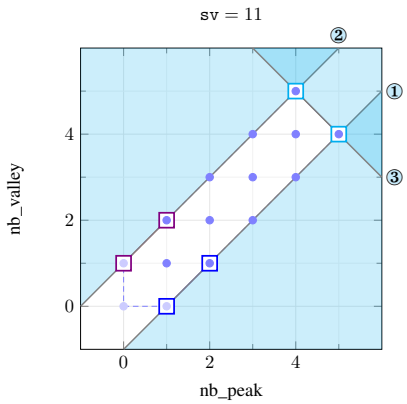
NB_PEAK(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: $(1, 0)$ $(2, 1)$

② $y \leq x + 1$
- sv ≥ 5: $(0, 1)$ $(1, 2)$

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor)$



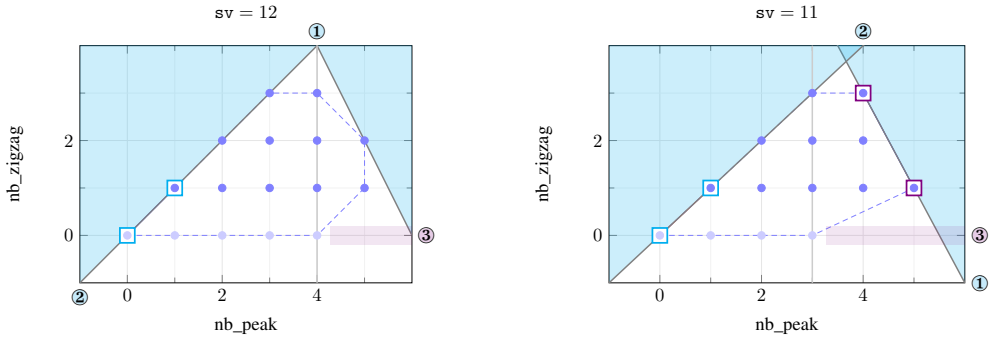


NB_PEAK(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, 2) (⌊(sv − 1)/2⌋ − 1, 4)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊(sv − 1)/2⌋, 1) (⌊(sv − 1)/2⌋ − 1, 3)

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

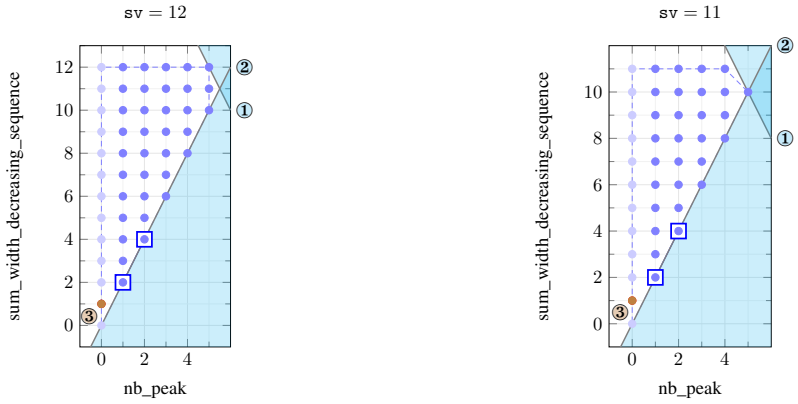


NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- sv ≥ 5: (1, 2) (2, 4)

③ $x \neq 0 \vee y \neq 1$

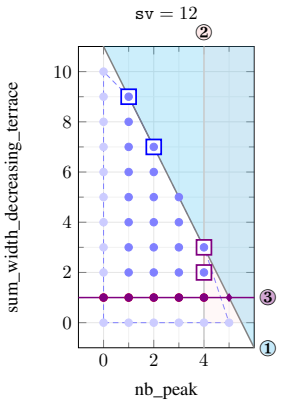


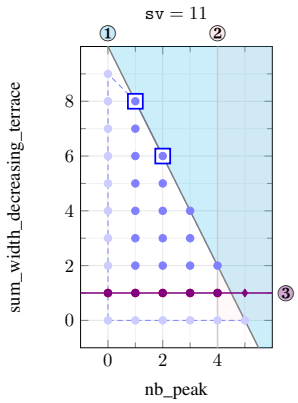
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$

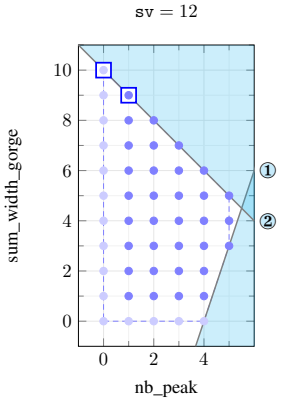


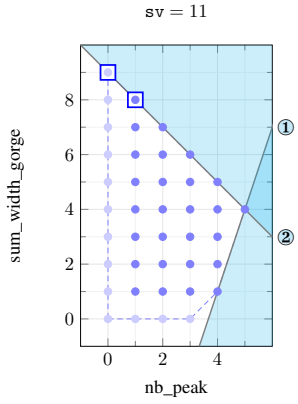


NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



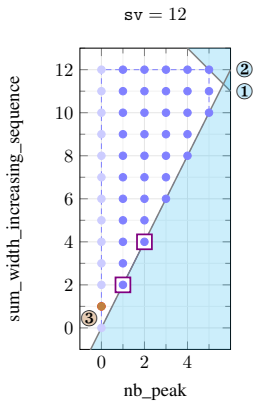


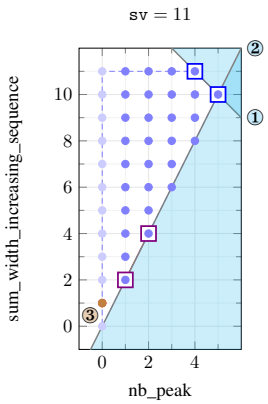
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 3$: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



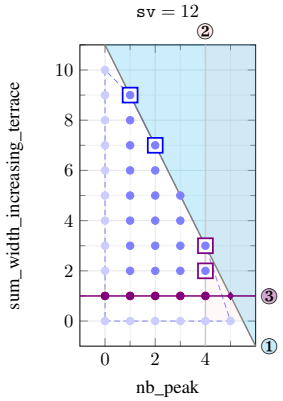


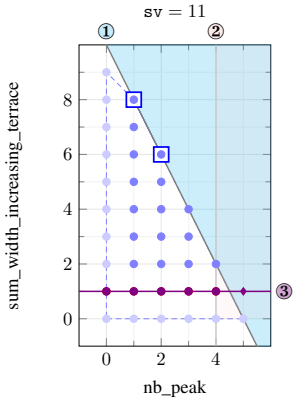
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv-3)$ $(2, sv-5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv-1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

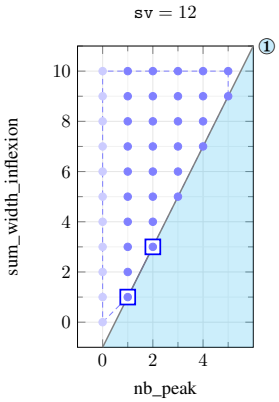
③ $y \neq 1$

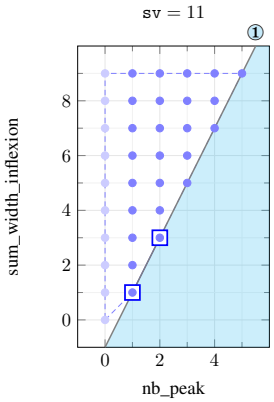




NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- sv ≥ 5: (1, 1) (2, 3)



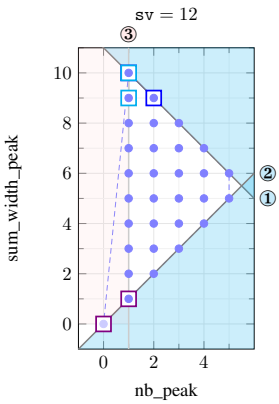


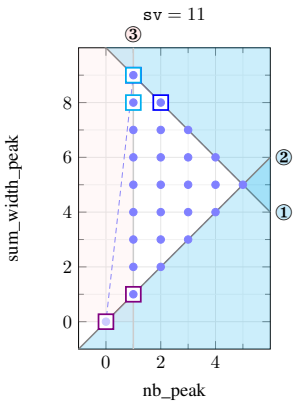
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 2) (2, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 2) (1, sv − 3)

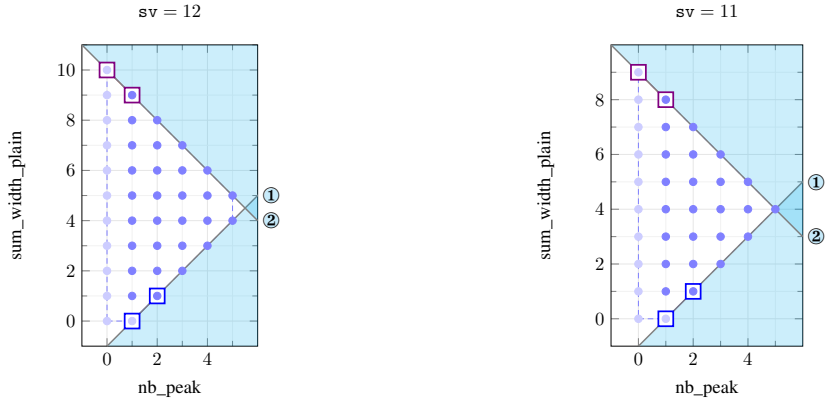




NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 3: (0, sv − 2) (1, sv − 3)

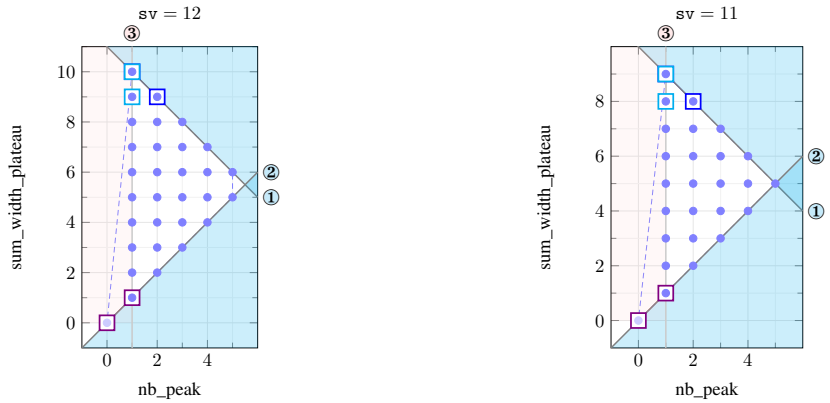


NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 2) (2, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 2) (1, sv − 3)

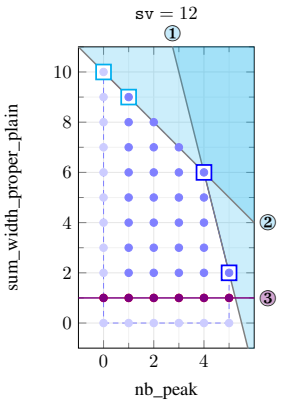


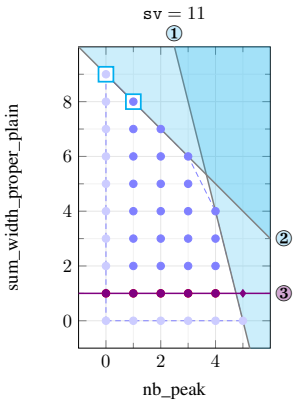
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$





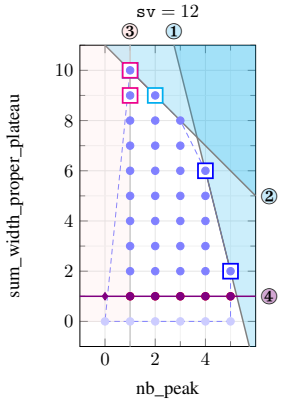
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

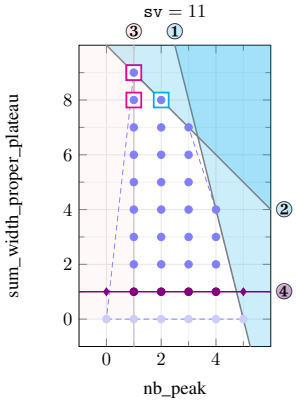
① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $x + y \leq sv - 1$
- sv ≥ 7: $(1, sv-2)$ $(2, sv-3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv-2)$ $(1, sv-3)$

④ $y \neq 1$

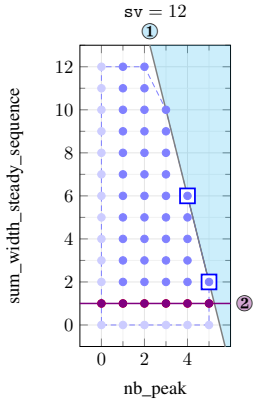


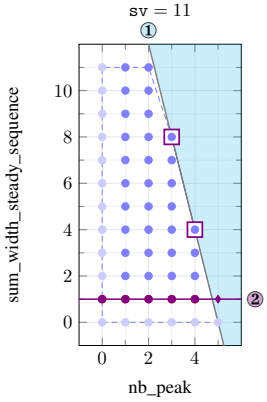


NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 6: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 9: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

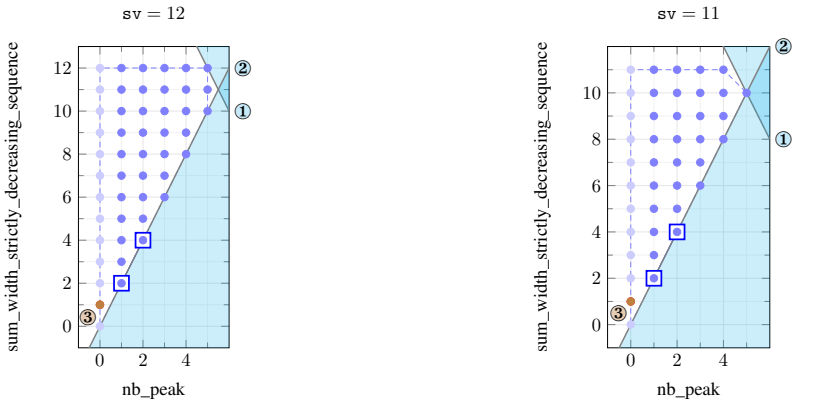
② $y \neq 1$





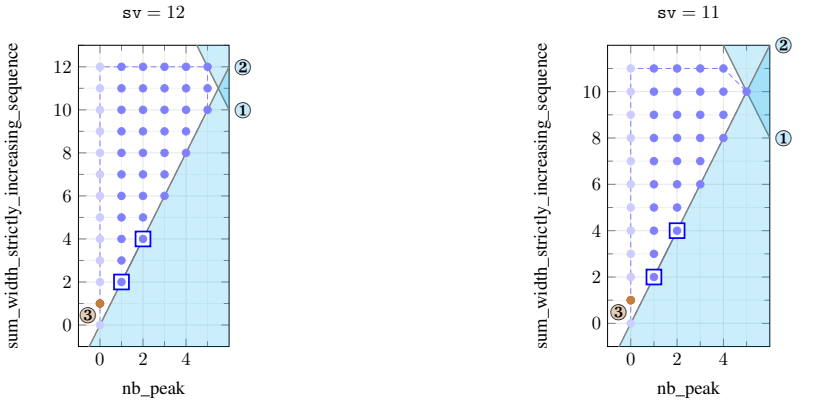
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

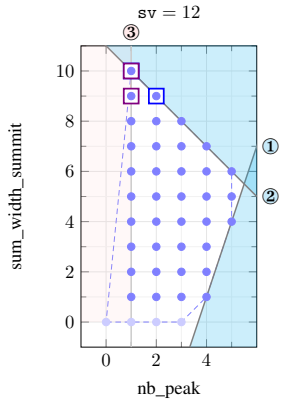
① $3 * x \leq y + sv - 1$

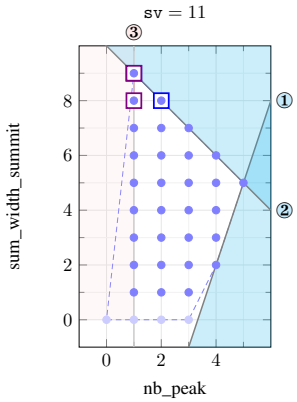
② $x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$





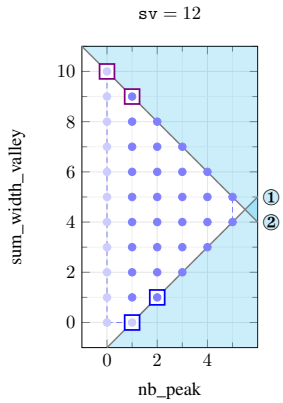
NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

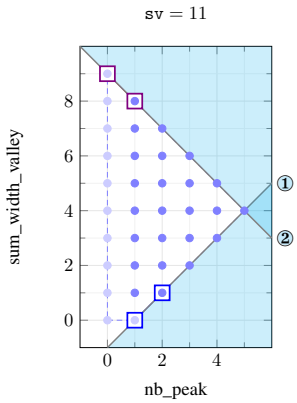
① $x \leq y + 1$

- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$





NB_PEAK(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- sv ≥ 7: (1, 3) (2, 5)

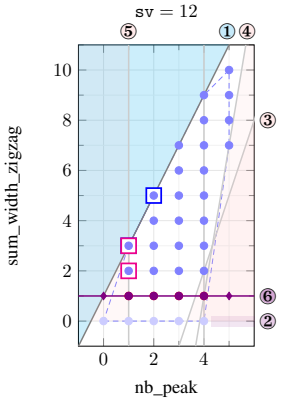
② $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

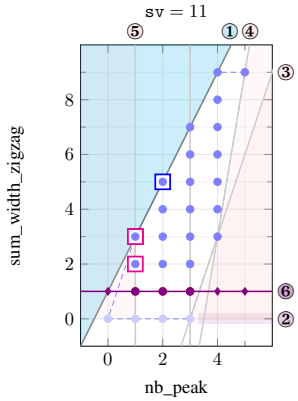
③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, sv − 5) (⌊(sv − 1)/2⌋ − 1, sv − 11)
- sv mod 2 = 1 ∧ sv ≥ 17: (⌊(sv − 1)/2⌋ − 1, sv − 8) (⌊(sv − 1)/2⌋ − 2, sv − 14)

⑤ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, 2) (1, 3)

⑥ $y \neq 1$



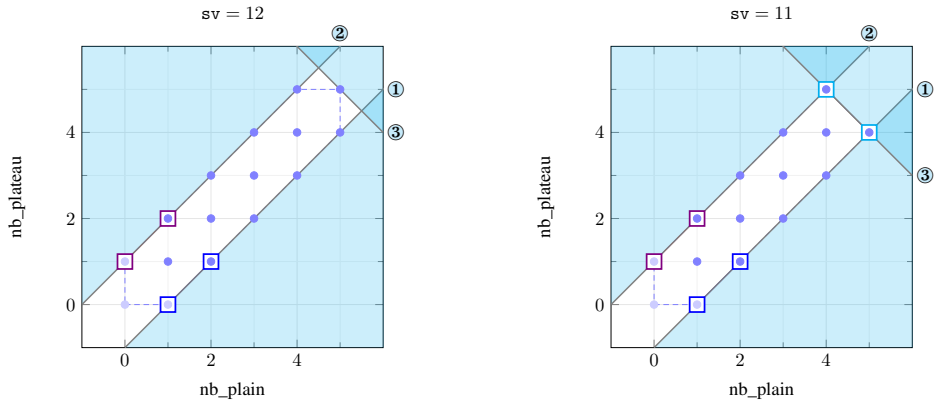


NB_PLAIN(x, VARIABLES) ∧
NB_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $y \leq x + 1$
- sv ≥ 5: (0, 1) (1, 2)

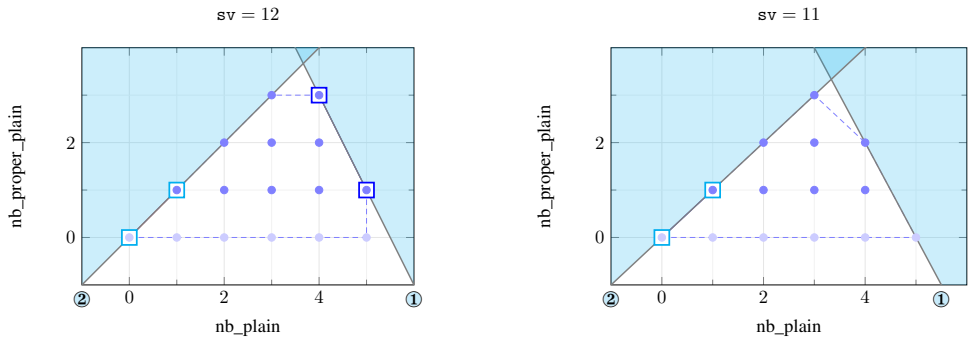
③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor)$



NB_PLAIN(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

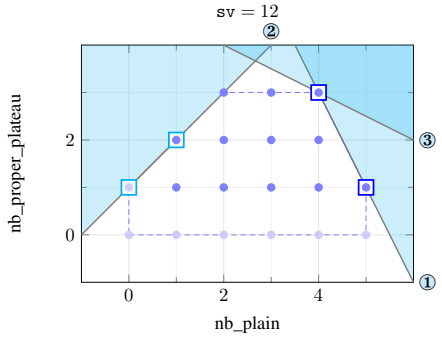


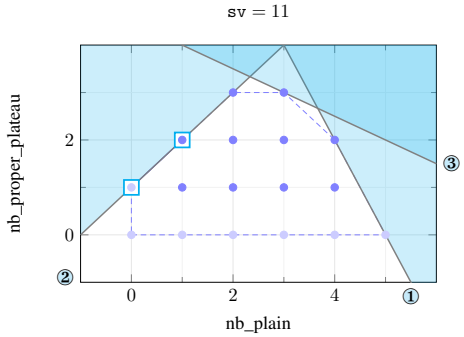
NB_PLAIN(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

② $y \leq x + 1$
- sv ≥ 7: (0, 1) (1, 2)

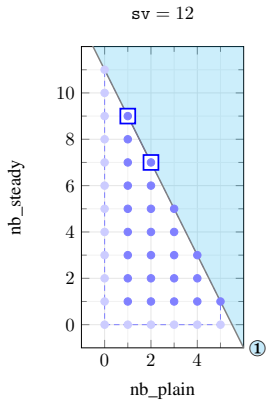
③ $sv > 1 \Rightarrow x + 2 * y \leq sv - 2$

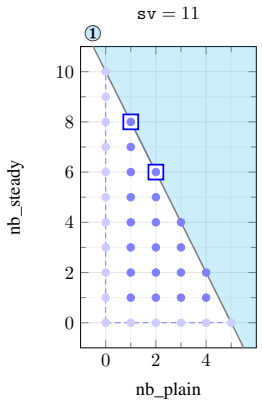




NB_PLAIN(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

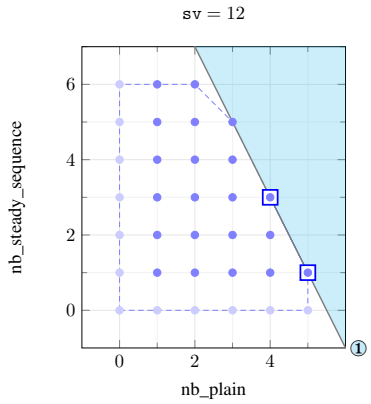
① $2 * x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 3) (2, sv − 5)

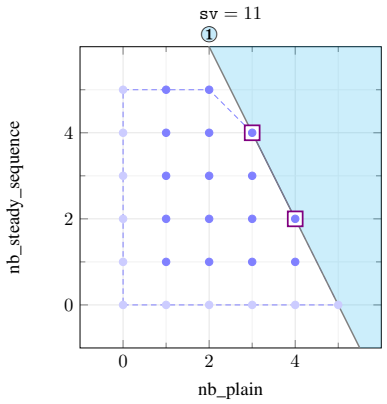




NB_PLAIN(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

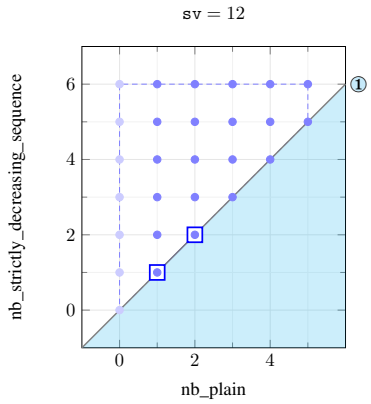
① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 6: (⌊(sv − 1)/2⌋, 1) (⌊(sv − 1)/2⌋ − 1, 3)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋ − 1, 2) (⌊(sv − 1)/2⌋ − 2, 4)

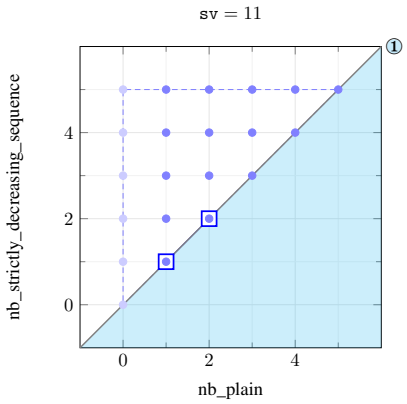




NB_PLAIN(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

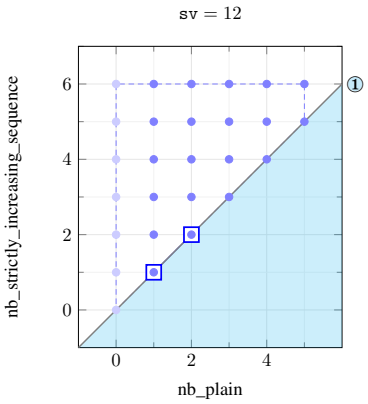
① $x \leq y$
- sv ≥ 5: (1, 1) (2, 2)

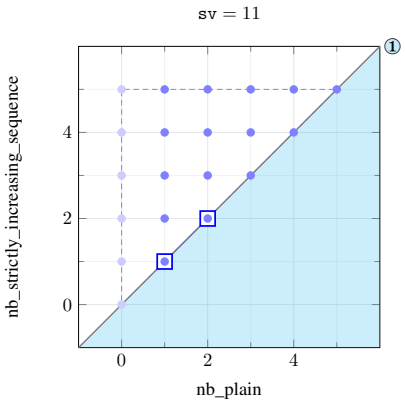




NB_PLAIN(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 5: (1, 1) (2, 2)



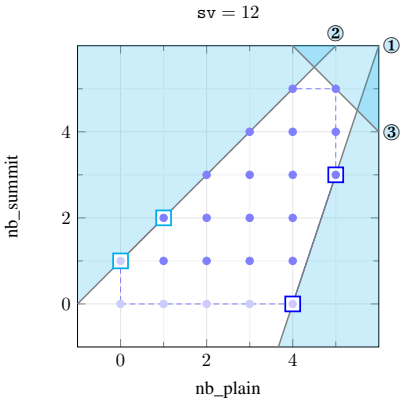


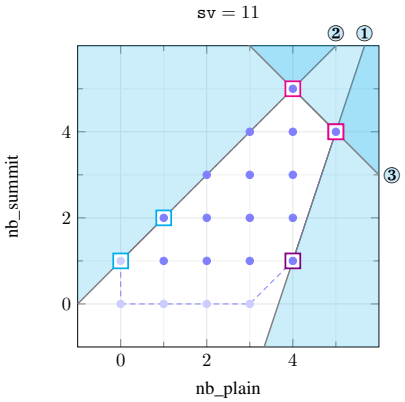
NB_PLAIN(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 2) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋ − 5)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋ − 4)

② $y \leq x + 1$
- sv ≥ 5: (0, 1) (1, 2)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋)

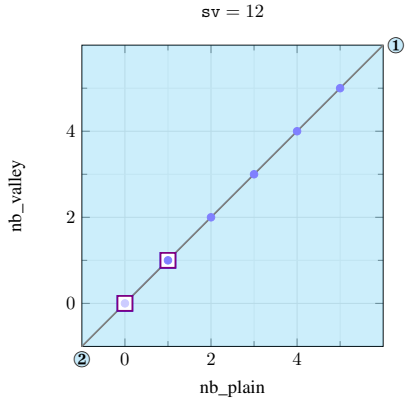


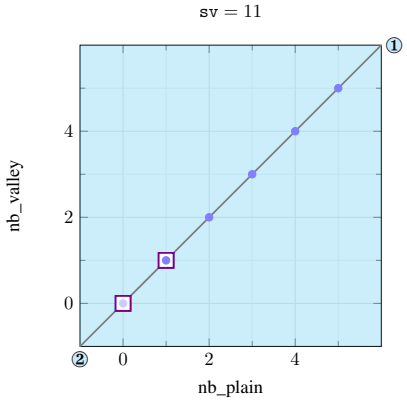


NB_PLAIN(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

② $y \leq x$
- sv ≥ 3: (0, 0) (1, 1)



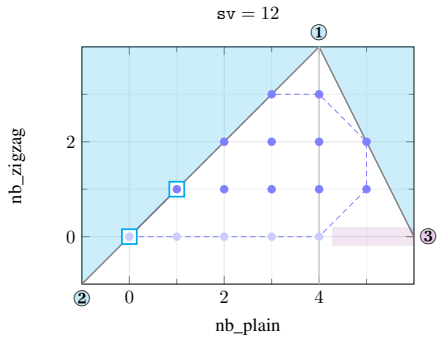


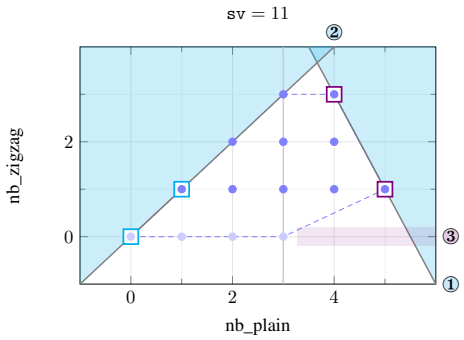
NB_PLAIN(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, 2) (⌊(sv − 1)/2⌋ − 1, 4)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊(sv − 1)/2⌋, 1) (⌊(sv − 1)/2⌋ − 1, 3)

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$



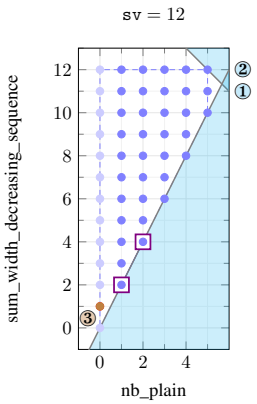


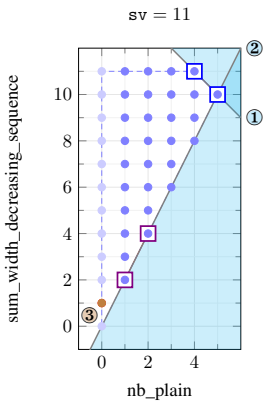
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 3$: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



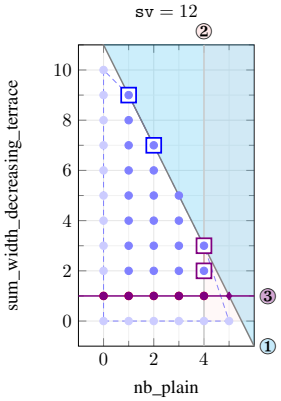


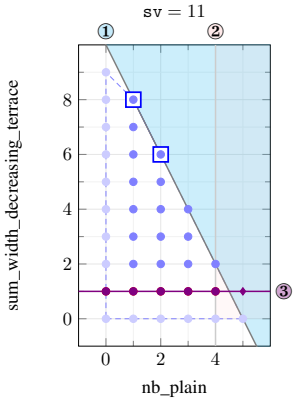
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv-3)$ $(2, sv-5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv-1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

③ $y \neq 1$





NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

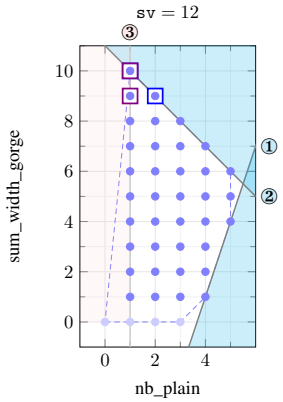
① $3 * x \leq y + sv - 1$

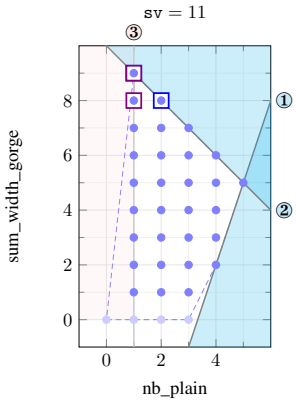
② $x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$





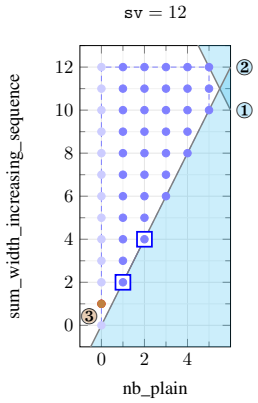
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

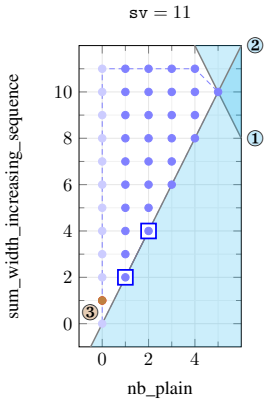
① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



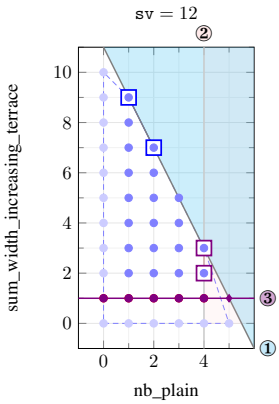


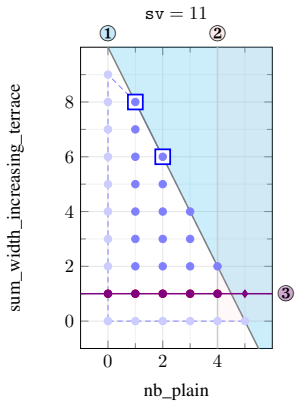
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

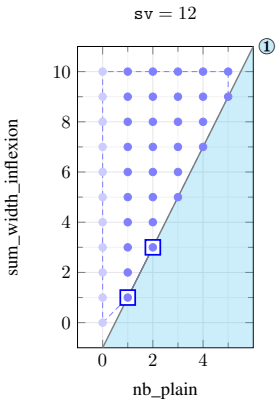
③ $y \neq 1$

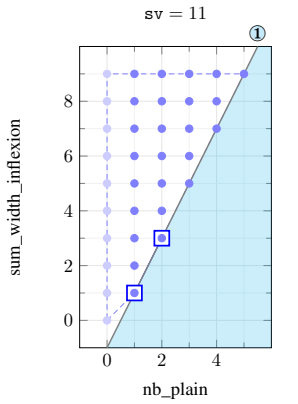




NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- $sv \geq 5$: $(1, 1)$ $(2, 3)$

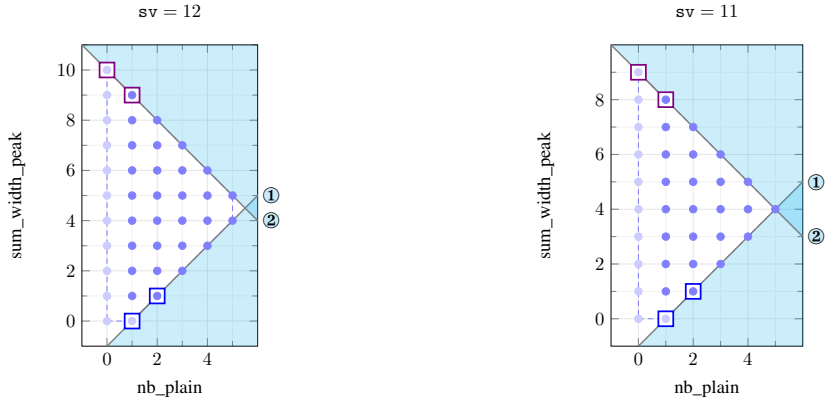




NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 3: (0, sv − 2) (1, sv − 3)

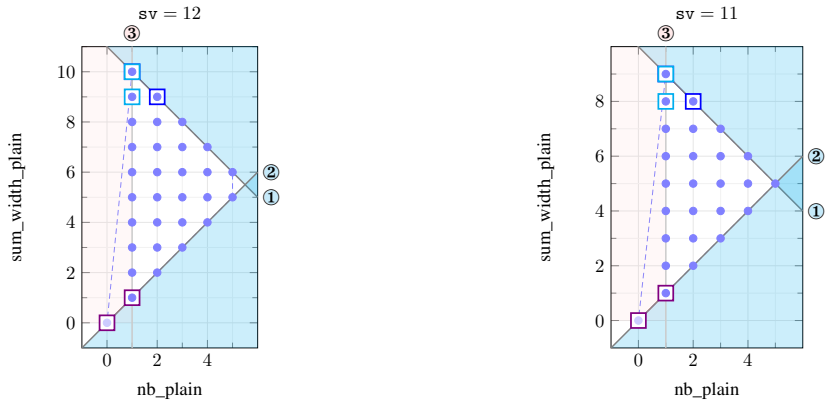


NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 2) (2, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

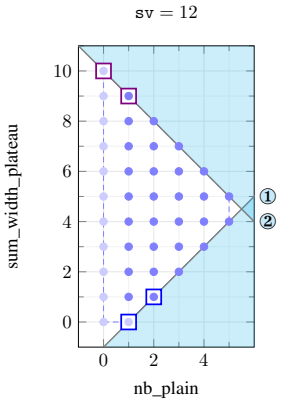
③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 2) (1, sv − 3)

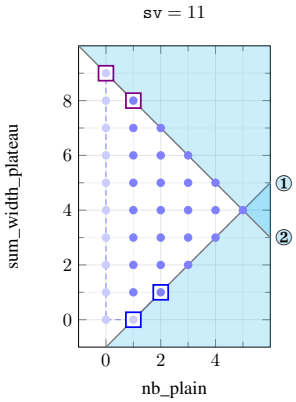


NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$





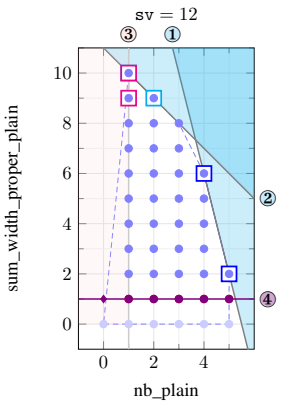
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

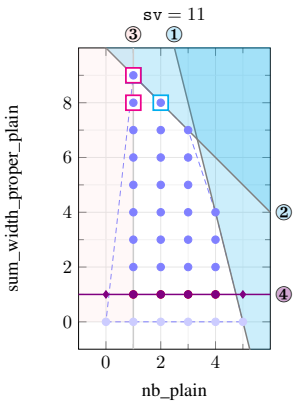
① $4 * x + y \leq 2 * sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 10$: $(\lfloor (sv - 1)/2 \rfloor, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 6)$
- $sv \bmod 2 = 1 \wedge sv \geq 13$: $(\lfloor (sv - 1)/2 \rfloor - 1, 4)$ $(\lfloor (sv - 1)/2 \rfloor - 2, 8)$

② $x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$



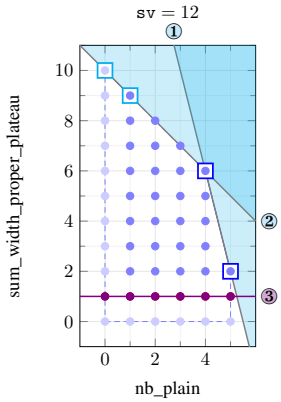


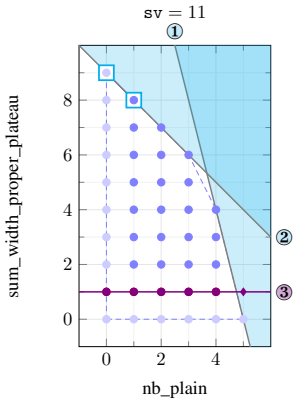
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$

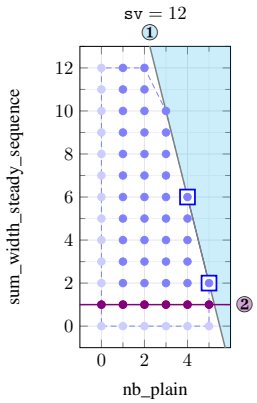


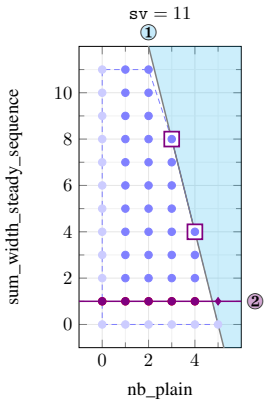


NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 6: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 9: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

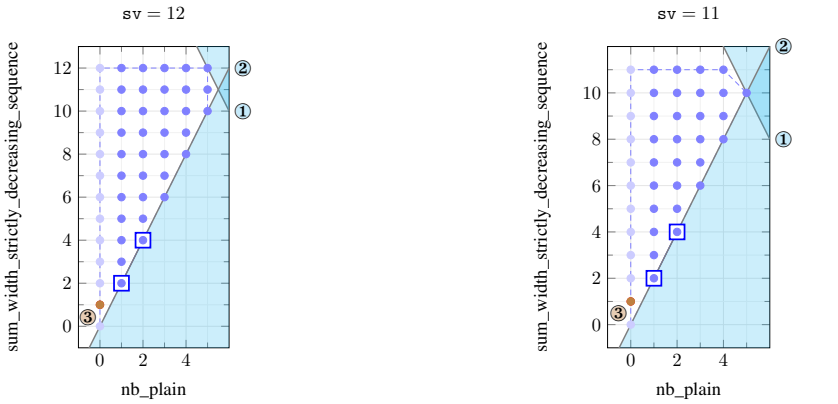
② $y \neq 1$





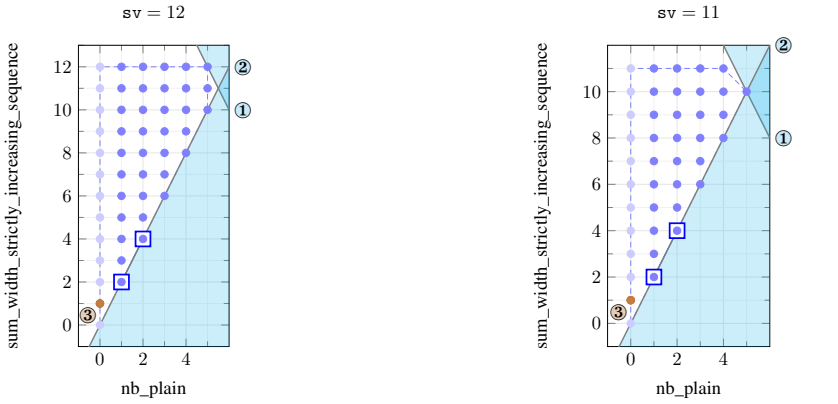
NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$

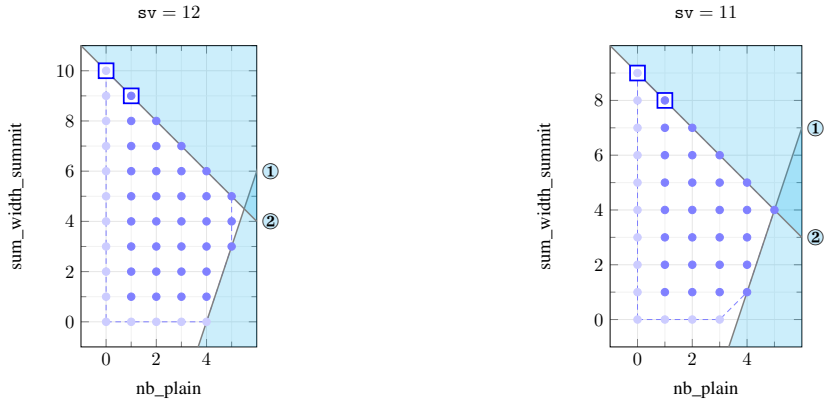


NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

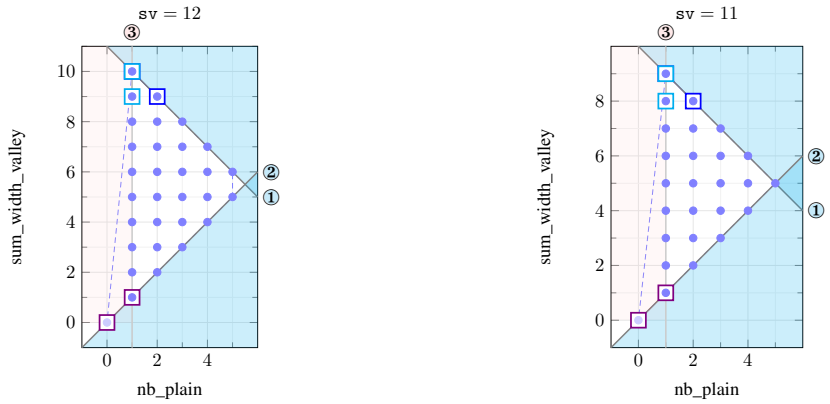
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$

- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$



NB_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- sv ≥ 7: (1, 3) (2, 5)

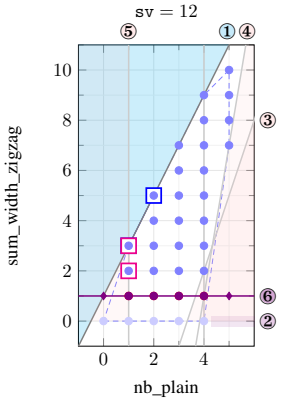
② $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

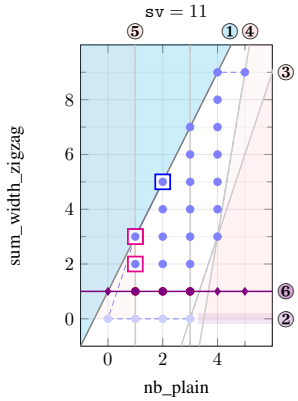
③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: $(\lfloor (sv-1)/2 \rfloor, sv-5)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv - 11)$
- sv mod 2 = 1 ∧ sv ≥ 17: $(\lfloor (sv-1)/2 \rfloor - 1, sv-8)$ $(\lfloor (sv-1)/2 \rfloor - 2, sv - 14)$

⑤ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, 2) (1, 3)

⑥ $y \neq 1$



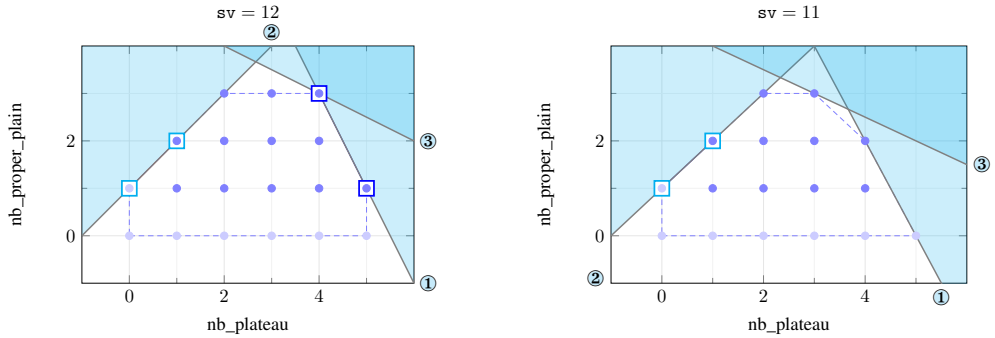


NB_PLATEAU(x, VARIABLES) ∧
NB_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

② $y \leq x + 1$
- sv ≥ 7: (0, 1) (1, 2)

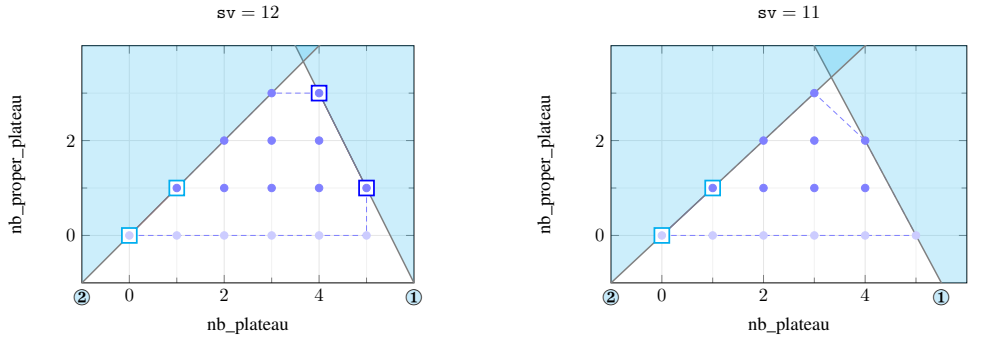
③ $sv > 1 \Rightarrow x + 2 * y \leq sv - 2$



NB_PLATEAU(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 4)$

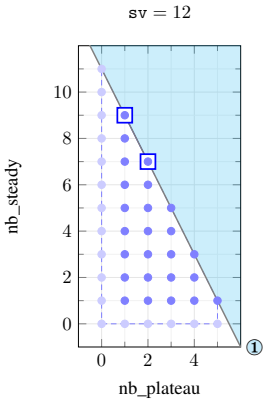
② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

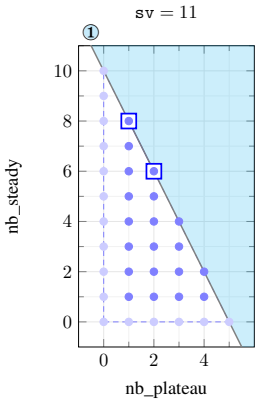


NB_PLATEAU(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 3)$ $(2, sv - 5)$

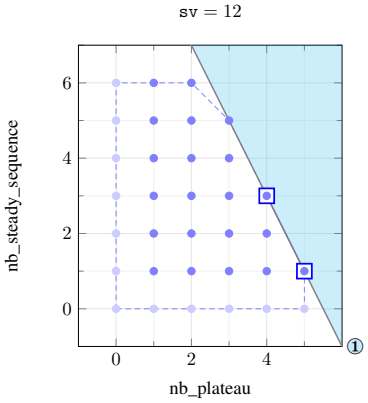


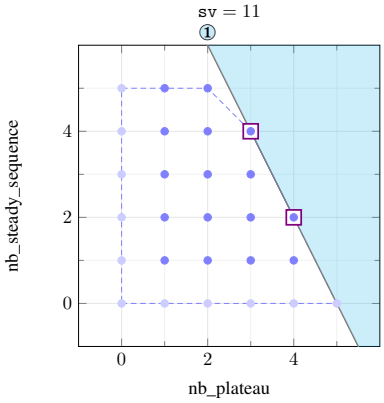


NB_PLATEAU(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

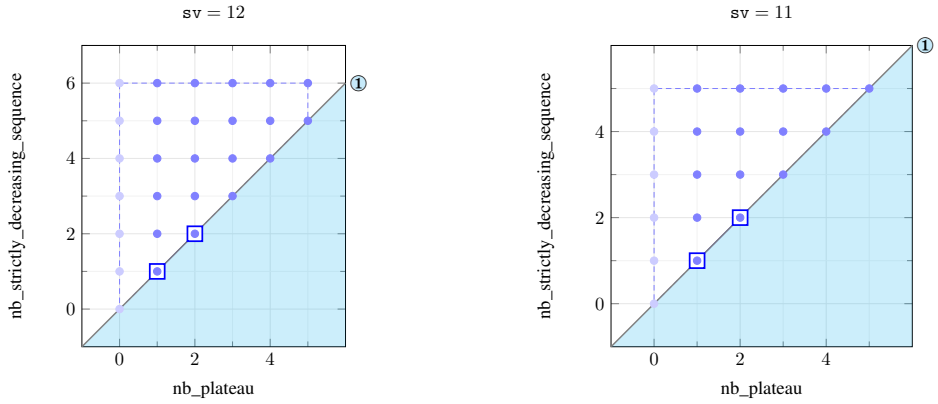
- $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor, 1)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$
- $sv \bmod 2 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 2, 4)$





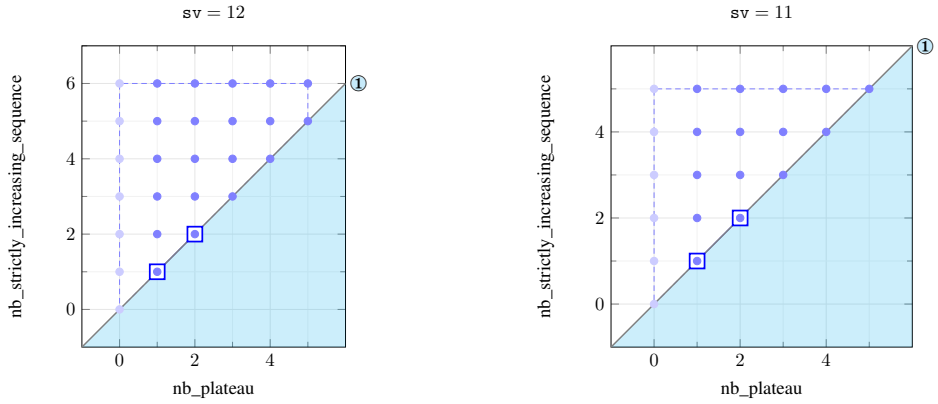
NB_PLATEAU(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① x ≤ y
- sv ≥ 5: (1, 1) (2, 2)



NB_PLATEAU(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

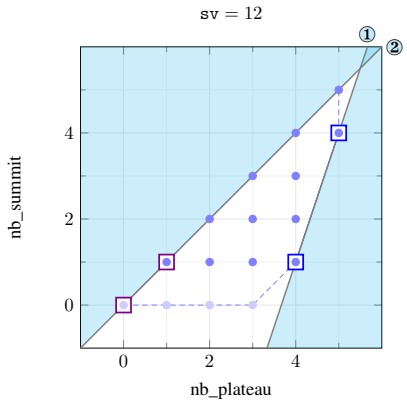
① x ≤ y
- sv ≥ 5: (1, 1) (2, 2)

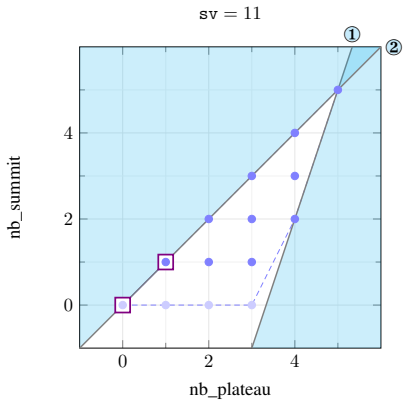


NB_PLATEAU(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor - 4)$

② $y \leq x$
- sv ≥ 3: (0, 0) (1, 1)



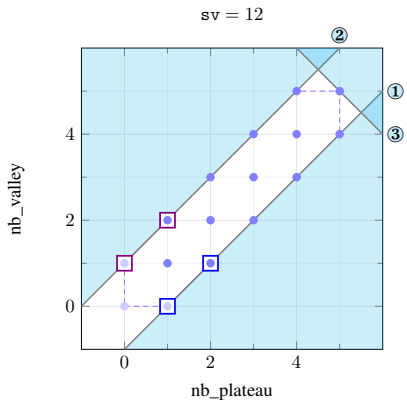


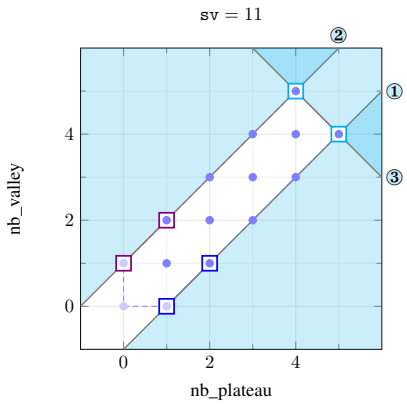
NB_PLATEAU(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $y \leq x + 1$
- sv ≥ 5: (0, 1) (1, 2)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: $(\lfloor (sv-1)/2 \rfloor, \lfloor (sv-1)/2 \rfloor - 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, \lfloor (sv-1)/2 \rfloor)$



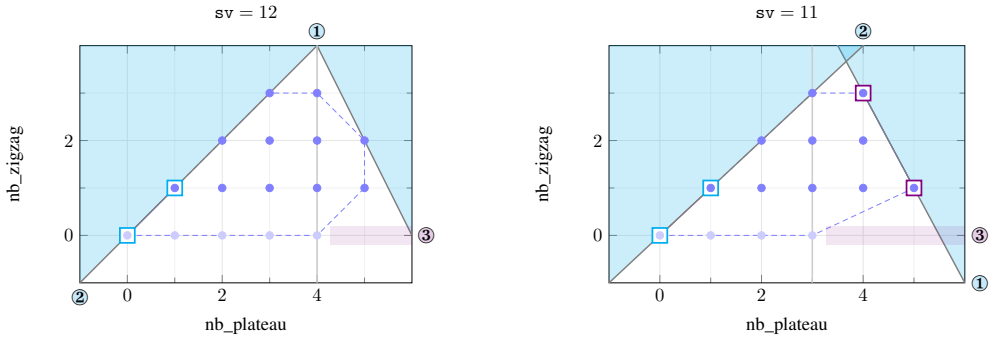


NB_PLATEAU(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 14: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

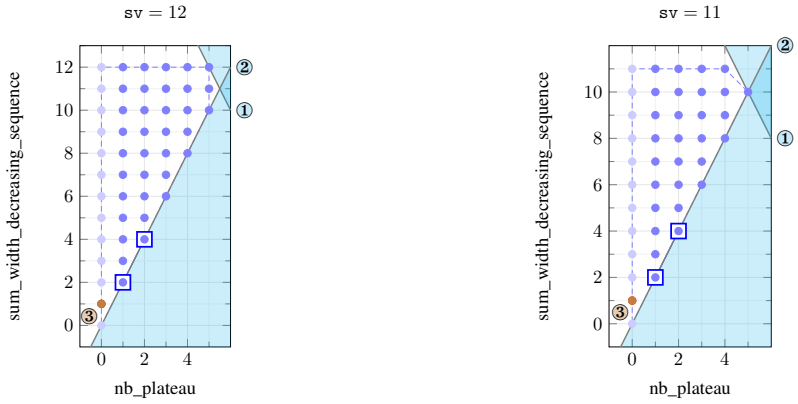


NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- sv ≥ 5: (1, 2) (2, 4)

③ $x \neq 0 \vee y \neq 1$

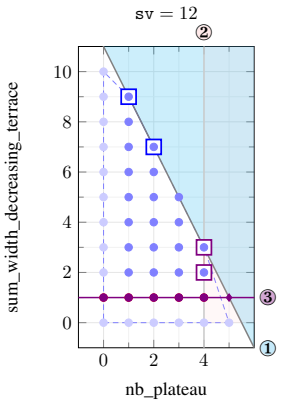


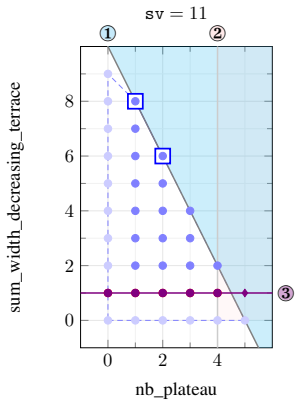
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$

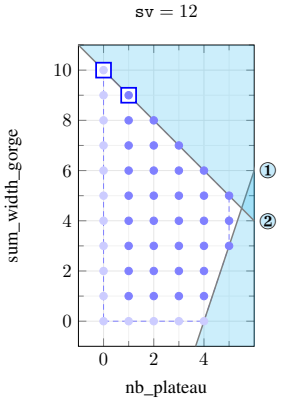


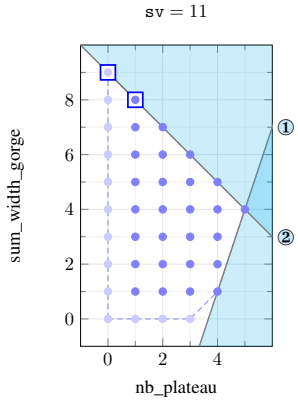


NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



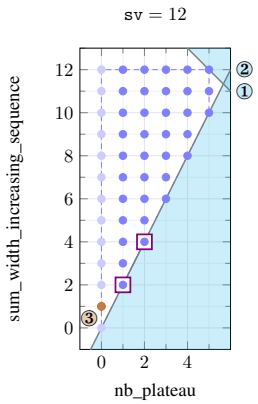


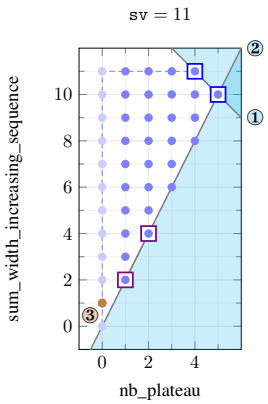
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 3$: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



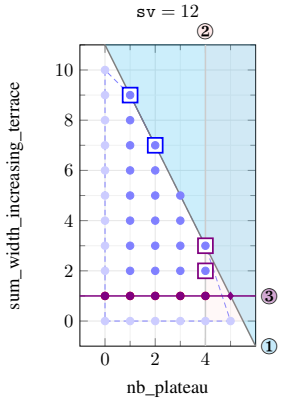


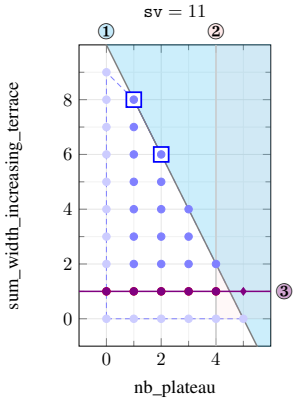
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv-3)$ $(2, sv-5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv-1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv-1)/2 \rfloor - 1, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

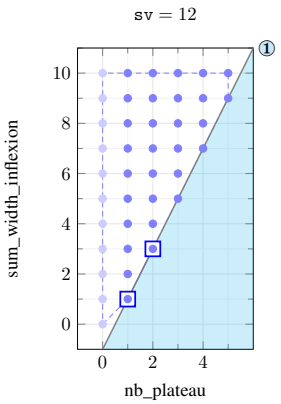
③ $y \neq 1$

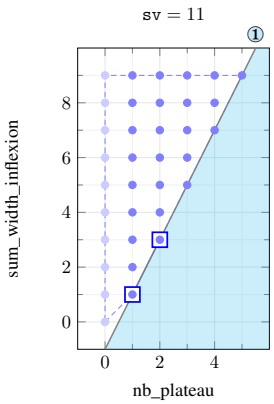




NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- sv ≥ 5: (1, 1) (2, 3)



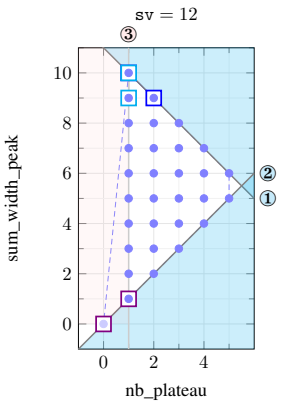


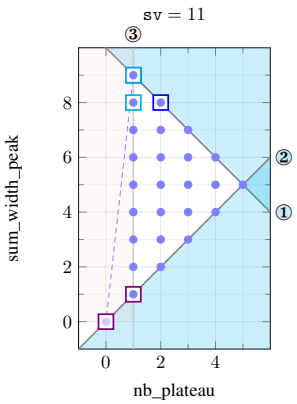
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 2) (2, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 2) (1, sv − 3)

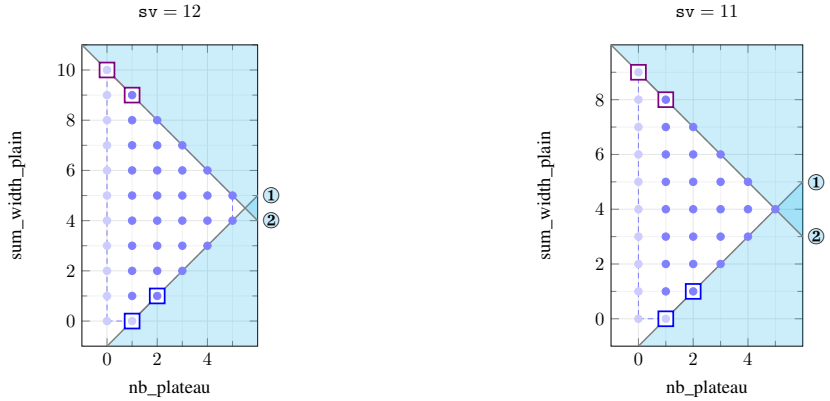




NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 3: (0, sv − 2) (1, sv − 3)

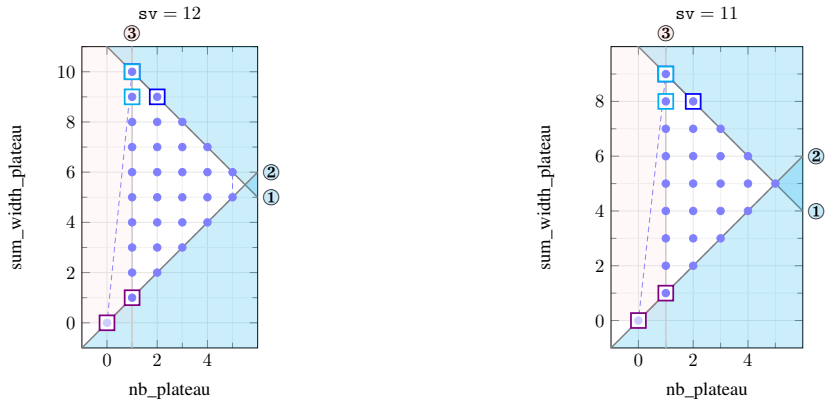


NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- sv ≥ 5: (1, sv − 2) (2, sv − 3)

② $x \leq y$
- sv ≥ 3: (0, 0) (1, 1)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 2) (1, sv − 3)

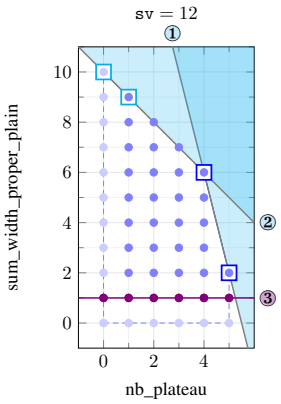


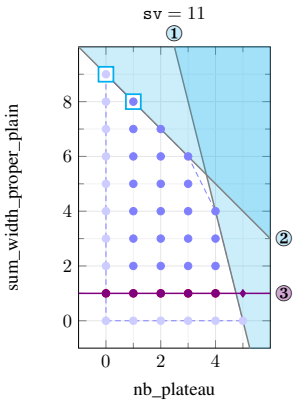
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$





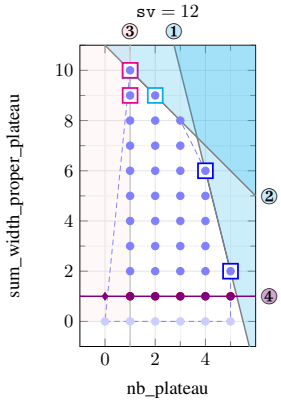
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

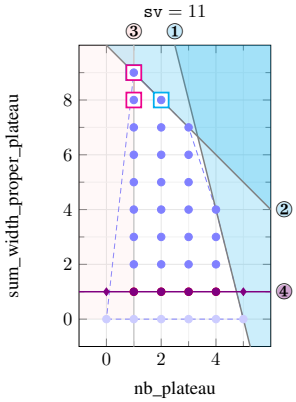
① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $x + y \leq sv - 1$
- sv ≥ 7: $(1, sv-2)$ $(2, sv-3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv-2)$ $(1, sv-3)$

④ $y \neq 1$

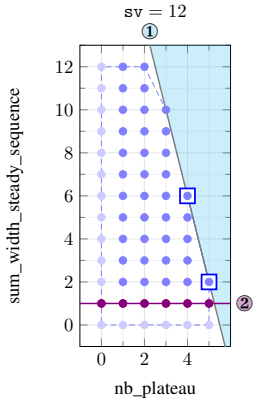


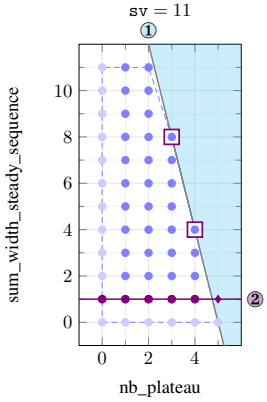


NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 6: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 9: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

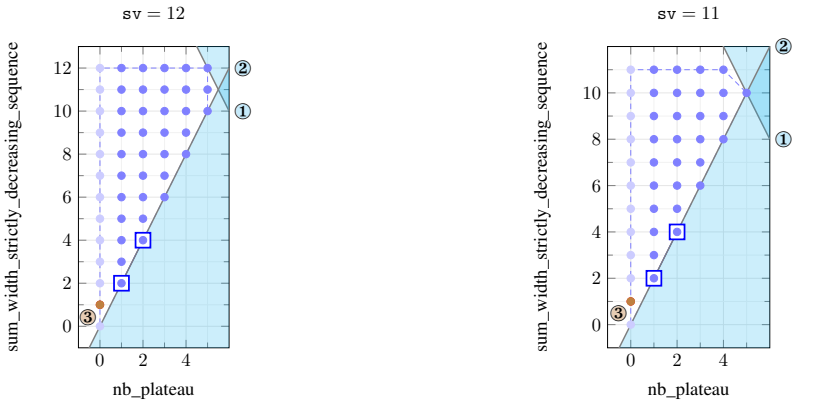
② $y \neq 1$





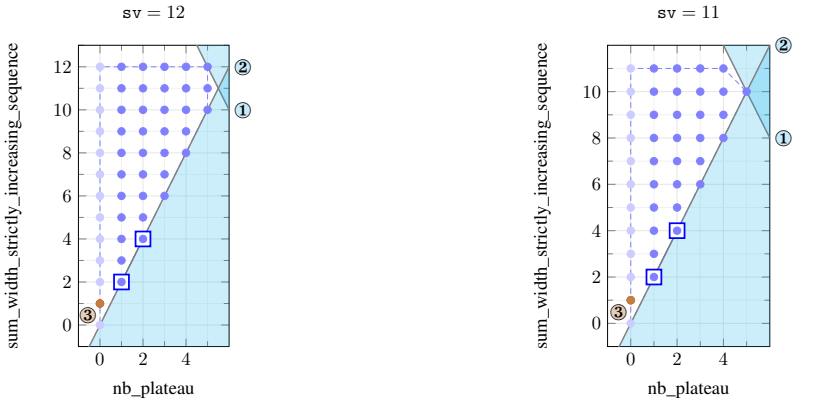
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

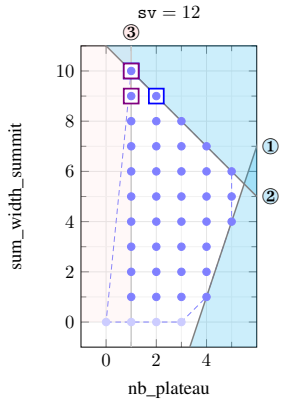
① $3 * x \leq y + sv - 1$

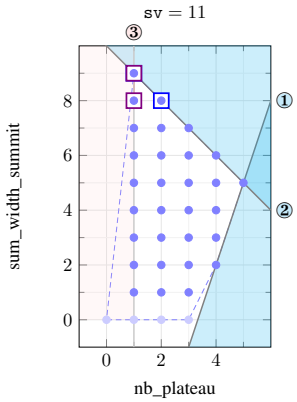
② $x + y \leq sv - 1$

- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$





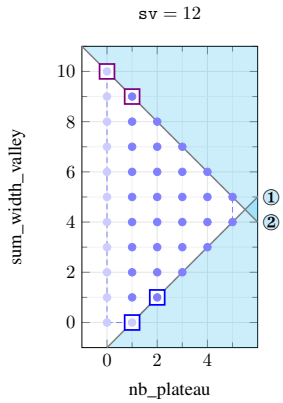
NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

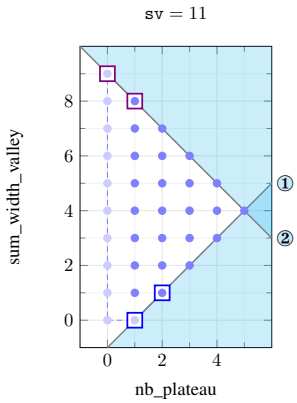
① $x \leq y + 1$

- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$





NB_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- $sv \geq 7$: (1, 3) (2, 5)

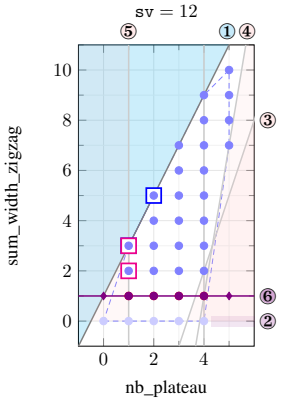
② $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

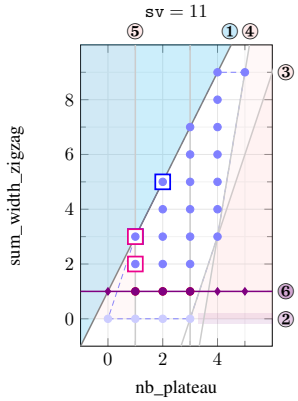
③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 14$: $(\lfloor (sv-1)/2 \rfloor, sv-5)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv-11)$
- $sv \bmod 2 = 1 \wedge sv \geq 17$: $(\lfloor (sv-1)/2 \rfloor - 1, sv-8)$ $(\lfloor (sv-1)/2 \rfloor - 2, sv-14)$

⑤ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: (1, 2) (1, 3)

⑥ $y \neq 1$





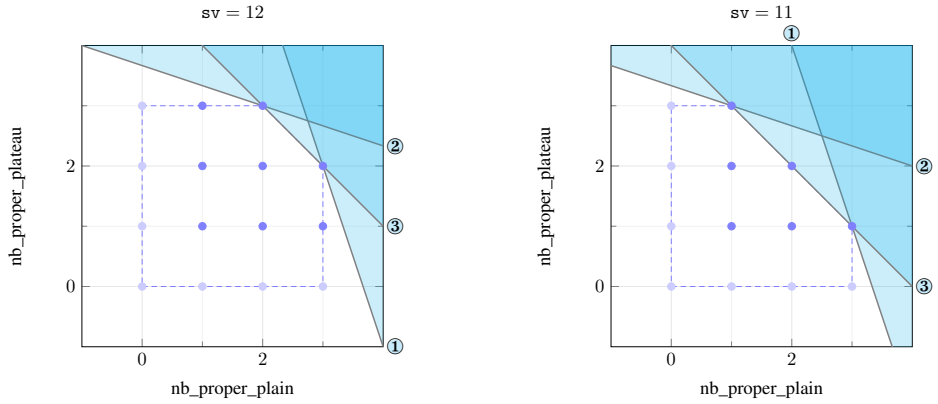
NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + y \leq sv - 1$
- sv mod 3 = 0 ∧ sv ≥ 24: $(\lfloor (sv-1)/3 \rfloor, 2)$ $(\lfloor (sv-1)/3 \rfloor - 1, 5)$
- sv mod 3 = 1 ∧ sv ≥ 28: $(\lfloor (sv-1)/3 \rfloor - 1, 3)$ $(\lfloor (sv-1)/3 \rfloor - 2, 6)$
- sv mod 3 = 2 ∧ sv ≥ 20: $(\lfloor (sv-1)/3 \rfloor, 1)$ $(\lfloor (sv-1)/3 \rfloor - 1, 4)$

② $x + 3 * y \leq sv - 1$
- sv mod 3 = 0 ∧ sv ≥ 24: $(2, \lfloor (sv-1)/3 \rfloor)$ $(5, \lfloor (sv-1)/3 \rfloor - 1)$
- sv mod 3 = 1 ∧ sv ≥ 28: $(3, \lfloor (sv-1)/3 \rfloor - 1)$ $(6, \lfloor (sv-1)/3 \rfloor - 2)$
- sv mod 3 = 2 ∧ sv ≥ 20: $(1, \lfloor (sv-1)/3 \rfloor)$ $(4, \lfloor (sv-1)/3 \rfloor - 1)$

③ $sv > 1 \Rightarrow 2 * x + 2 * y \leq sv - 2 - (sv - 2) \bmod 2$

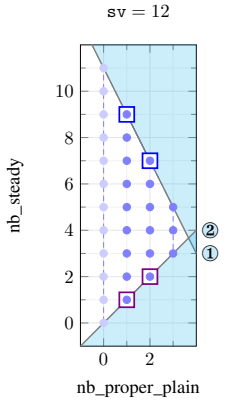
④ $o^{=} \geq x + y$

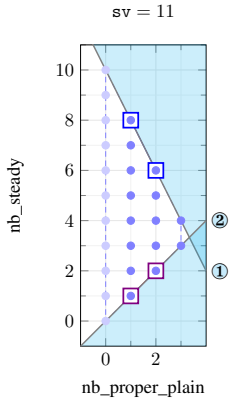


NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $x \leq y$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

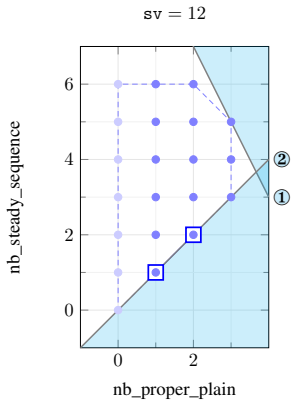


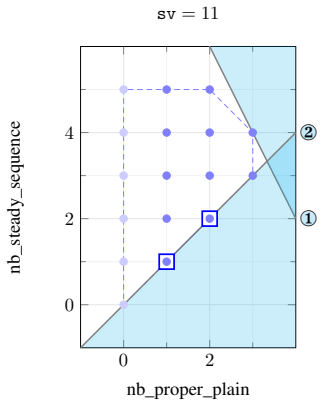


NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

② $x \leq y$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

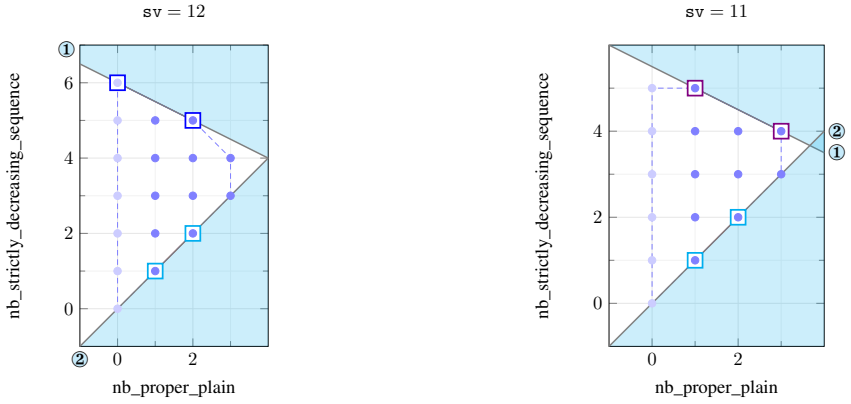




NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(0, \lfloor sv/2 \rfloor)$ $(2, \lfloor sv/2 \rfloor - 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(1, \lfloor sv/2 \rfloor)$ $(3, \lfloor sv/2 \rfloor - 1)$

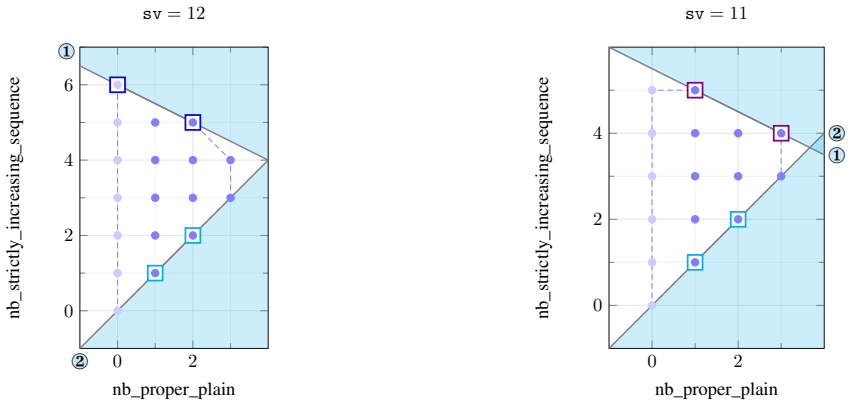
② $x \leq y$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$



NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(0, \lfloor sv/2 \rfloor)$ $(2, \lfloor sv/2 \rfloor - 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(1, \lfloor sv/2 \rfloor)$ $(3, \lfloor sv/2 \rfloor - 1)$

② $x \leq y$
- $sv \geq 7$: $(1, 1)$ $(2, 2)$

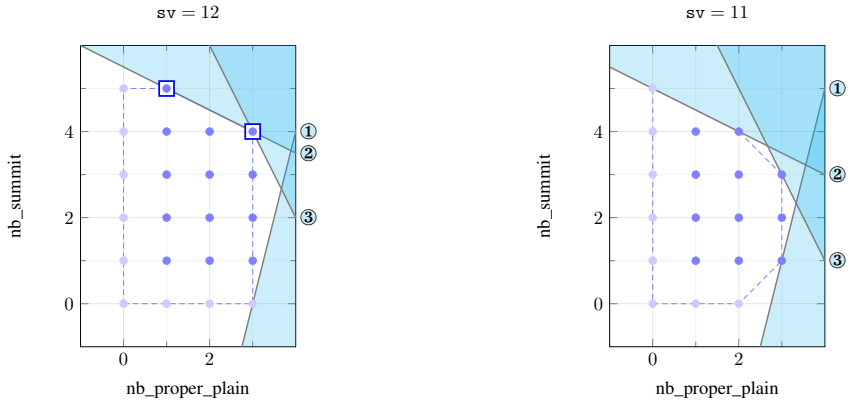


NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $4 * x \leq y + sv$

② $x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 12: (1, ⌊(sv − 1)/2⌋) (3, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 15: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)

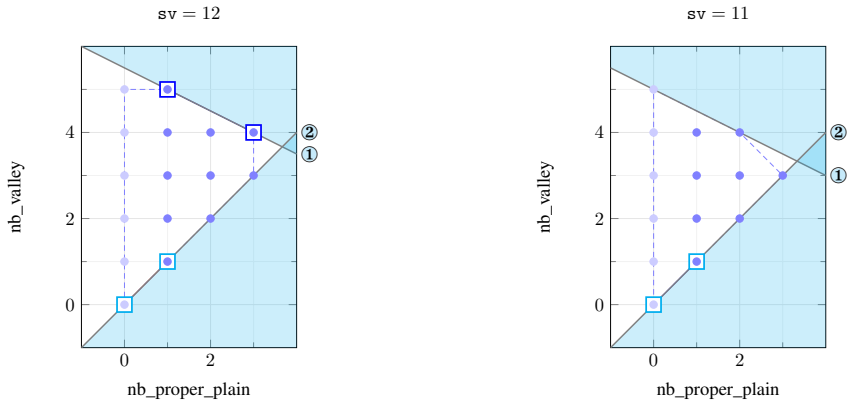
③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$



NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 10: (1, ⌊(sv − 1)/2⌋) (3, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 13: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)

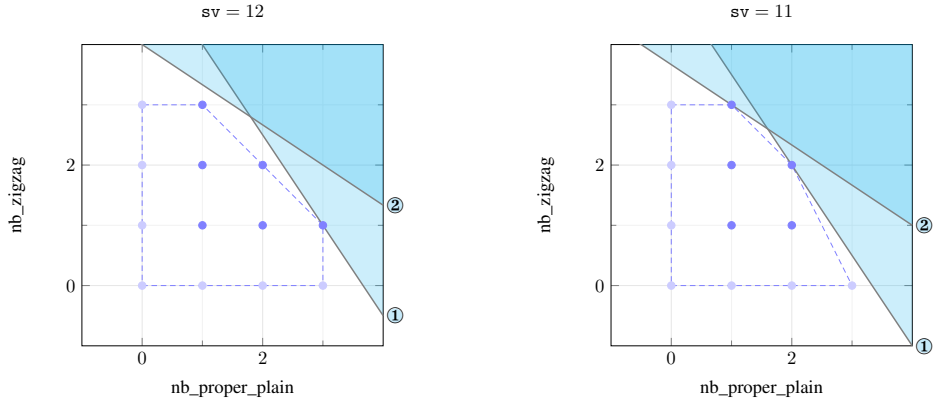
② $x \leq y$
- sv ≥ 4: (0, 0) (1, 1)



NB_PROPER_PLAIN(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $3 * \mathtt{x} + 2 * \mathtt{y} \leq \mathtt{sv} - 1$
- sv mod 3 = 0 ∧ sv ≥ 18: $(\lfloor(\mathtt{sv}-1)/3\rfloor, 1)$ $(\lfloor(\mathtt{sv}-1)/3\rfloor - 2, 4)$
- sv mod 3 = 2 ∧ sv ≥ 23: $(\lfloor(\mathtt{sv}-1)/3\rfloor - 1, 2)$ $(\lfloor(\mathtt{sv}-1)/3\rfloor - 3, 5)$

② $2 * \mathtt{x} + 3 * \mathtt{y} \leq \mathtt{sv}$
- sv mod 3 = 0 ∧ sv ≥ 33: $(3, \lfloor(\mathtt{sv}-1)/3\rfloor - 1)$ $(6, \lfloor(\mathtt{sv}-1)/3\rfloor - 3)$
- sv mod 3 = 1 ∧ sv ≥ 28: $(2, \lfloor(\mathtt{sv}-1)/3\rfloor - 1)$ $(5, \lfloor(\mathtt{sv}-1)/3\rfloor - 3)$
- sv mod 3 = 2 ∧ sv ≥ 23: $(1, \lfloor(\mathtt{sv}-1)/3\rfloor)$ $(4, \lfloor(\mathtt{sv}-1)/3\rfloor - 2)$



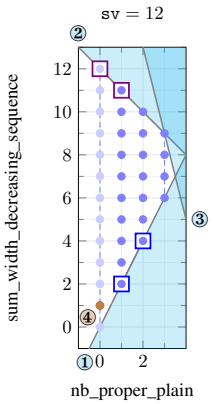
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

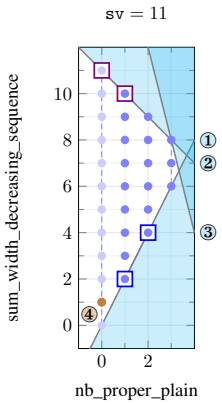
① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 5: (0, sv) (1, sv − 1)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$





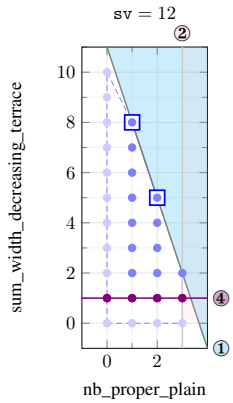
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

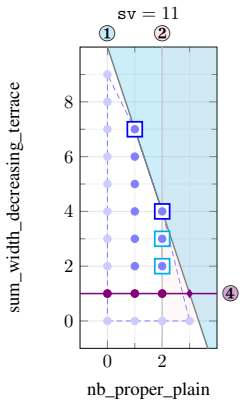
① $3 * x + y \leq sv - 1$
- $sv \geq 10$: $(1, sv - 4)$ $(2, sv - 7)$

② $y > 0 \Rightarrow 3 * x \leq sv - 3 - (sv - 3) \bmod 3$
- $sv \bmod 3 = 1 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$
- $sv \bmod 3 = 2 \wedge sv \geq 5$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

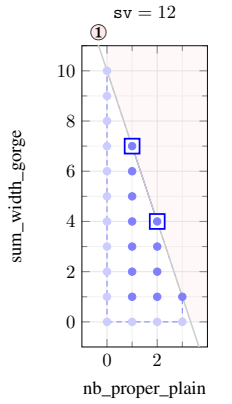
④ $y \neq 1$

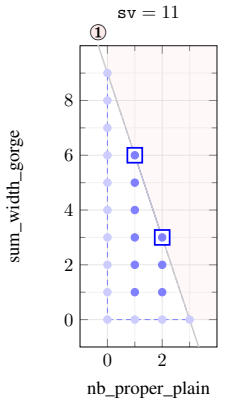




NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 2$
- $sv \geq 8$: $(1, sv - 5)$ $(2, sv - 8)$





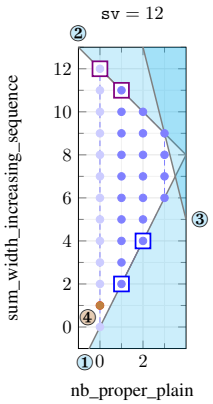
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

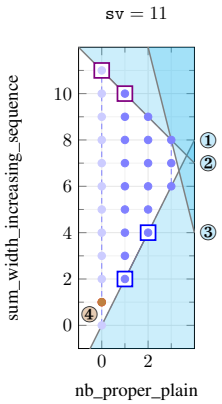
① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 5: (0, sv) (1, sv − 1)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$





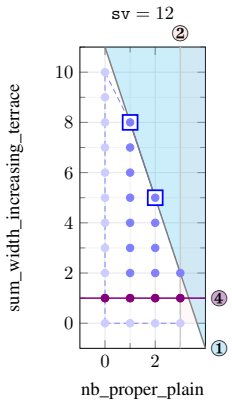
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

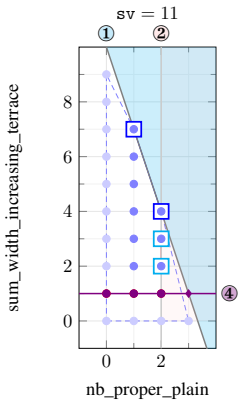
① $3 * x + y \leq sv - 1$
- $sv \geq 9$: $(1, sv - 4)$ $(2, sv - 7)$

② $y > 0 \Rightarrow 3 * x \leq sv - 3 - (sv - 3) \bmod 3$
- $sv \bmod 3 = 1 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$
- $sv \bmod 3 = 2 \wedge sv \geq 5$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

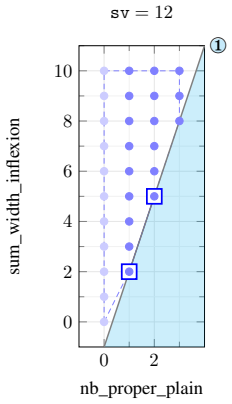
④ $y \neq 1$

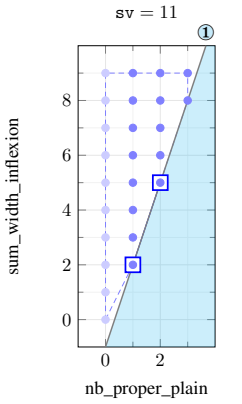




NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + 1$
- $sv \geq 7$: $(1, 2)$ $(2, 5)$

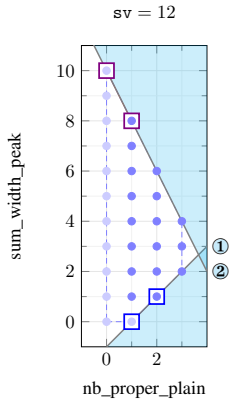


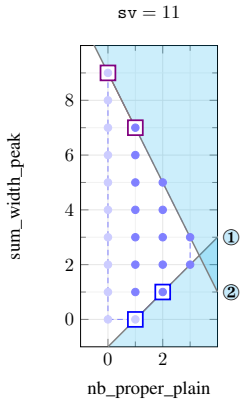


NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 7$: (1, 0) (2, 1)

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 4$: (0, sv − 2) (1, sv − 4)



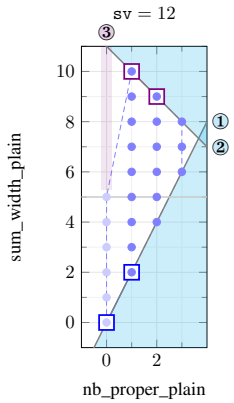


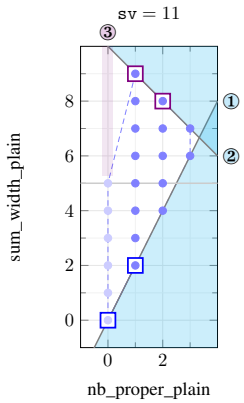
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- $sv \geq 4$: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- $sv \geq 7$: (1, sv − 2) (2, sv − 3)

③ $x = 0 \Rightarrow 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

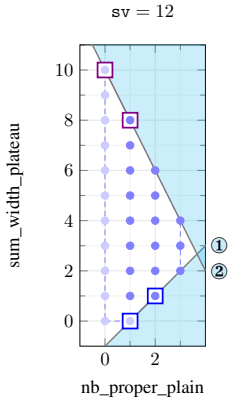


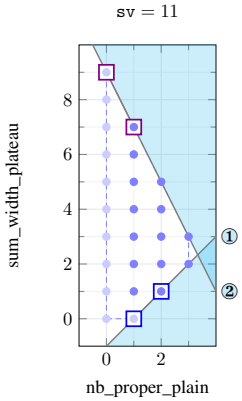


NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 7: (1, 0) (2, 1)

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- sv ≥ 4: (0, sv − 2) (1, sv − 4)



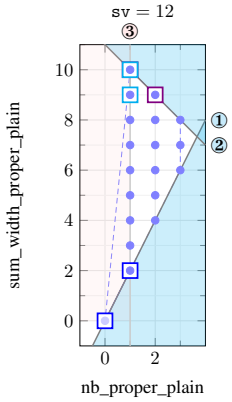


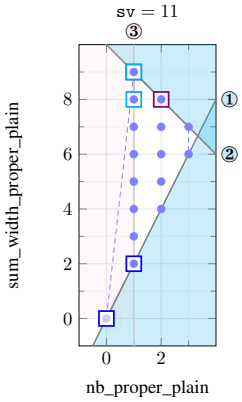
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)





NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x = \lfloor (sv - 1)/3 \rfloor - 1 \Rightarrow y \leq 6 + 2 * (sv - 1) \bmod 3$

② $x = \lfloor (sv - 1)/3 \rfloor \Rightarrow y \leq 2 * (sv - 1) \bmod 3$

③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

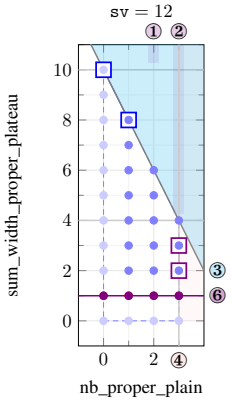
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

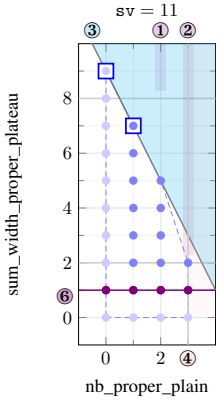
④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑤ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

⑥ $y \neq 1$





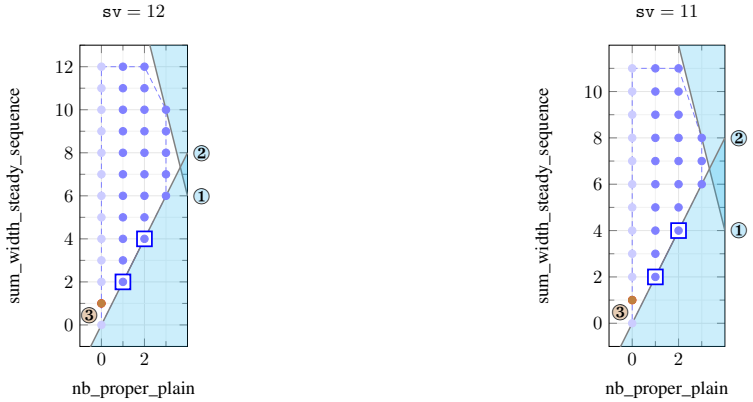
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 7$: (1, 2) (2, 4)

③ $x \neq 0 \vee y \neq 1$



NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$

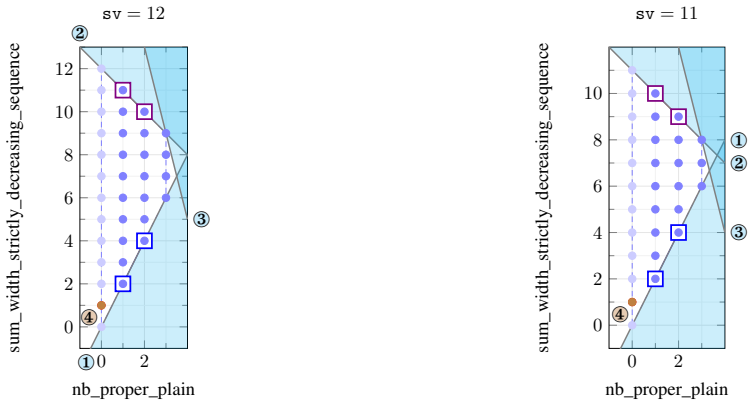
- $sv \geq 7$: (1, 2) (2, 4)

② $x + y \leq sv$

- $sv \geq 8$: (1, sv − 1) (2, sv − 2)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$



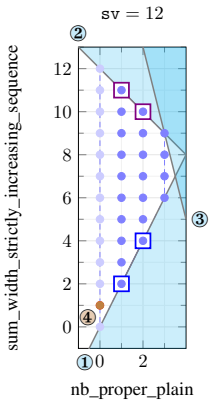
NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

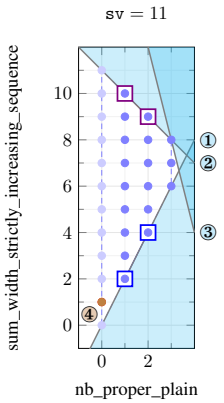
① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 8: (1, sv − 1) (2, sv − 2)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$

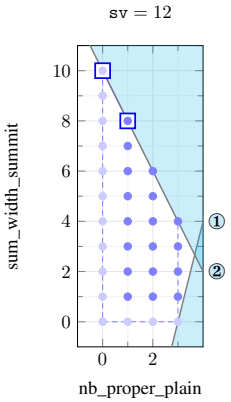


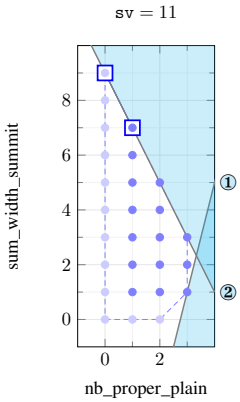


NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $4 * x \leq y + sv$

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- sv ≥ 4: (0, sv − 2) (1, sv − 4)

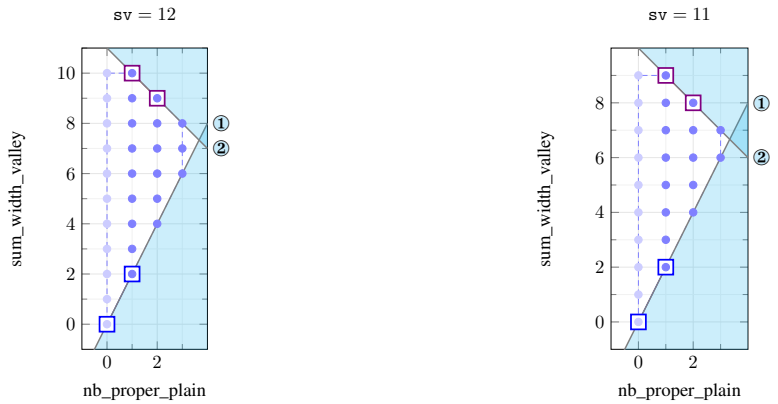




NB_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

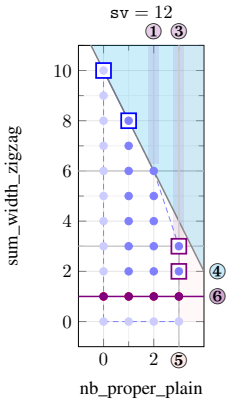


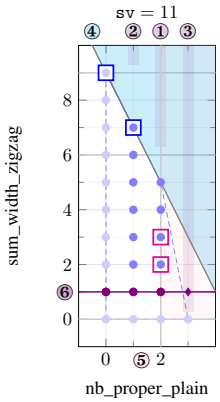
NB_PROPER_PLAIN($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_ZIGZAG($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} = \lfloor(\mathtt{sv}-1)/3\rfloor - 1 \Rightarrow \mathtt{y} \leq 6 - 3 * (\mathtt{sv} \bmod 3 = 1)$
② $\mathtt{x} = \lfloor(\mathtt{sv}-1)/3\rfloor - 2 \Rightarrow \mathtt{y} \leq 9 + 3 * (\mathtt{sv} \bmod 3 = 0)$
③ $\mathtt{x} = \lfloor(\mathtt{sv}-1)/3\rfloor \Rightarrow \mathtt{y} \leq 3 * (\mathtt{sv} \bmod 3 = 0)$
④ $\mathtt{sv} > 1 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 6$: $(0, \mathtt{sv}-2)$ $(1, \mathtt{sv}-4)$

⑤ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 3 * \mathtt{x} \leq \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 3$
- $\mathtt{sv} \bmod 3 = 0 \wedge \mathtt{sv} \geq 9$: $(\lfloor(\mathtt{sv}-1)/3\rfloor, 2)$ $(\lfloor(\mathtt{sv}-1)/3\rfloor, 3)$
- $\mathtt{sv} \bmod 3 = 1 \wedge \mathtt{sv} \geq 7$: $(\lfloor(\mathtt{sv}-1)/3\rfloor - 1, 2)$ $(\lfloor(\mathtt{sv}-1)/3\rfloor - 1, 3)$
- $\mathtt{sv} \bmod 3 = 2 \wedge \mathtt{sv} \geq 5$: $(\lfloor(\mathtt{sv}-1)/3\rfloor - 1, 2)$ $(\lfloor(\mathtt{sv}-1)/3\rfloor - 1, 3)$

⑥ $\mathtt{y} \neq 1$

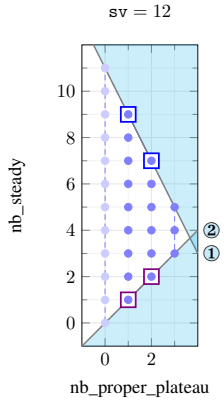


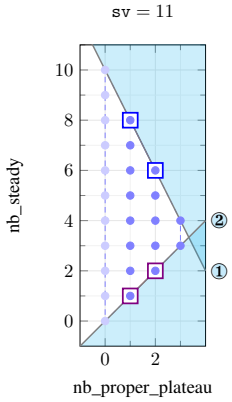


NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_STEADY(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 3) (2, sv − 5)

② $x \leq y$
- sv ≥ 7: (1, 1) (2, 2)

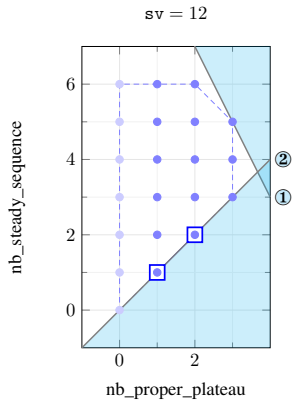


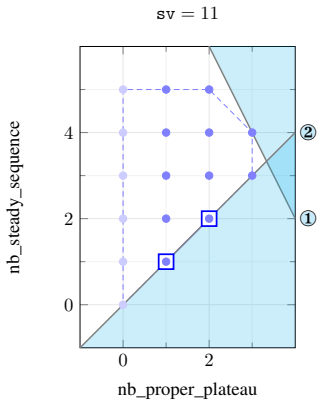


NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$

② $x \leq y$
- sv ≥ 7: (1, 1) (2, 2)

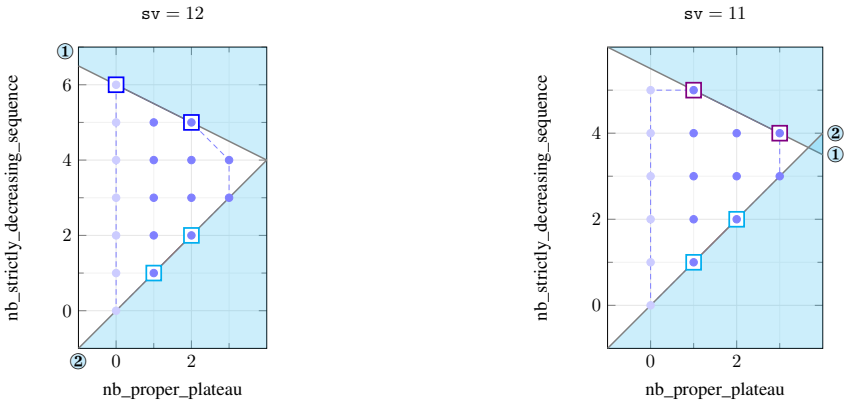




NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: (0, ⌊sv/2⌋) (2, ⌊sv/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 11: (1, ⌊sv/2⌋) (3, ⌊sv/2⌋ − 1)

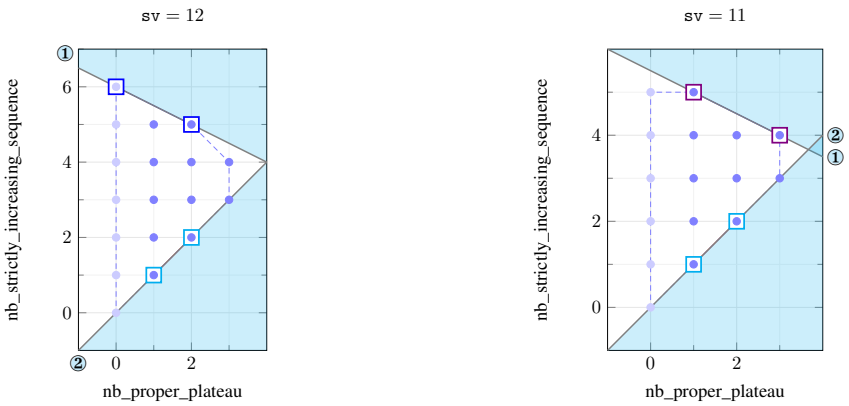
② $x \leq y$
- sv ≥ 7: (1, 1) (2, 2)



NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: (0, ⌊sv/2⌋) (2, ⌊sv/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 11: (1, ⌊sv/2⌋) (3, ⌊sv/2⌋ − 1)

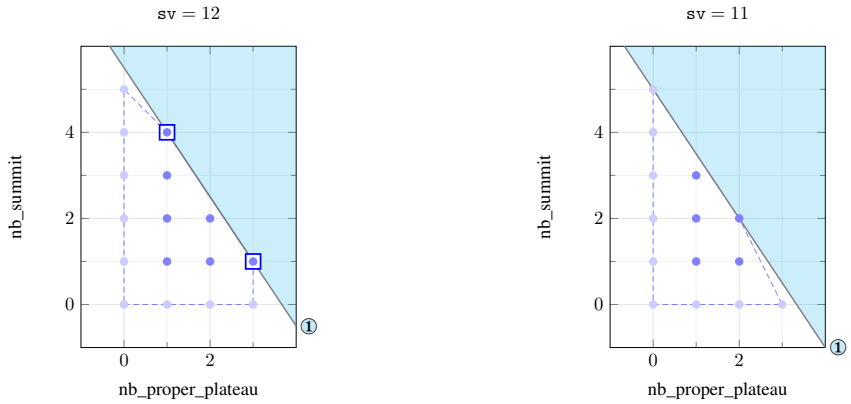
② $x \leq y$
- sv ≥ 7: (1, 1) (2, 2)



NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + 2 * y \leq sv - 1$

- $sv \bmod 2 = 0 \wedge sv \geq 10$: $(1, \lfloor (sv-1)/2 \rfloor - 1)$ $(3, \lfloor (sv-1)/2 \rfloor - 4)$



NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

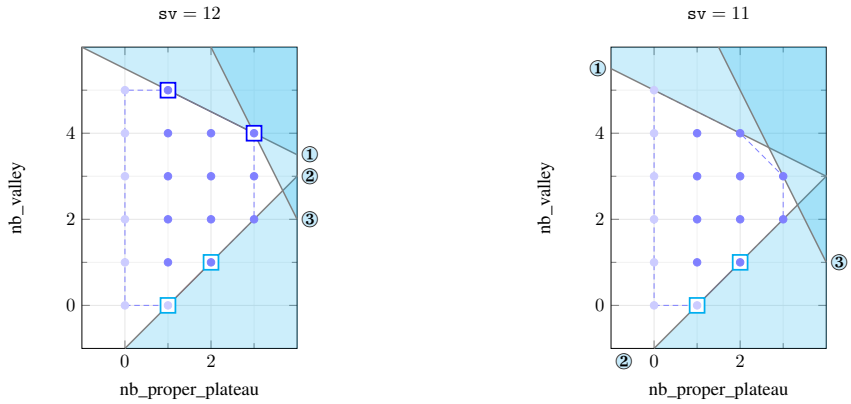
① $x + 2 * y \leq sv - 1$

- $sv \bmod 2 = 0 \wedge sv \geq 12$: $(1, \lfloor (sv-1)/2 \rfloor)$ $(3, \lfloor (sv-1)/2 \rfloor - 1)$
- $sv \bmod 2 = 1 \wedge sv \geq 15$: $(2, \lfloor (sv-1)/2 \rfloor - 1)$ $(4, \lfloor (sv-1)/2 \rfloor - 2)$

② $x \leq y + 1$

- $sv \geq 7$: $(1, 0)$ $(2, 1)$

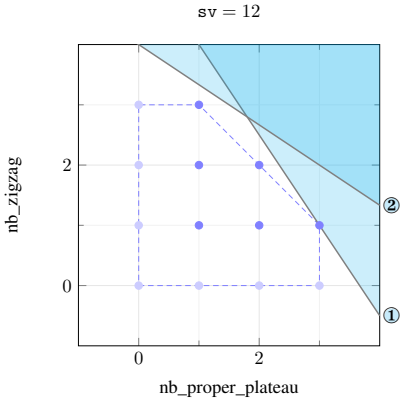
③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

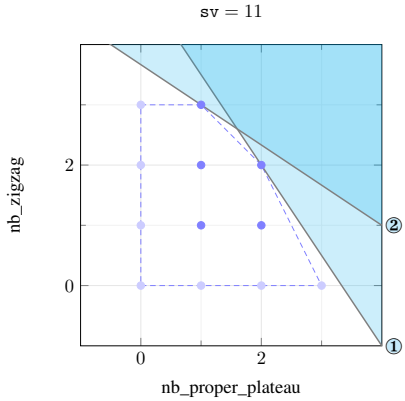


NB_PROPER_PLATEAU(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + 2 * y \leq sv - 1$
- sv mod 3 = 0 ∧ sv ≥ 18: $(\lfloor (sv-1)/3 \rfloor, 1)$ $(\lfloor (sv-1)/3 \rfloor - 2, 4)$
- sv mod 3 = 2 ∧ sv ≥ 23: $(\lfloor (sv-1)/3 \rfloor - 1, 2)$ $(\lfloor (sv-1)/3 \rfloor - 3, 5)$

② $2 * x + 3 * y \leq sv$
- sv mod 3 = 0 ∧ sv ≥ 33: $(3, \lfloor (sv-1)/3 \rfloor - 1)$ $(6, \lfloor (sv-1)/3 \rfloor - 3)$
- sv mod 3 = 1 ∧ sv ≥ 28: $(2, \lfloor (sv-1)/3 \rfloor - 1)$ $(5, \lfloor (sv-1)/3 \rfloor - 3)$
- sv mod 3 = 2 ∧ sv ≥ 23: $(1, \lfloor (sv-1)/3 \rfloor)$ $(4, \lfloor (sv-1)/3 \rfloor - 2)$





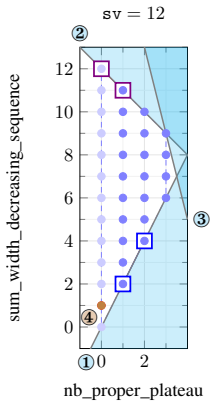
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

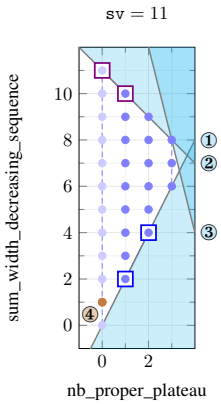
① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 5: (0, sv) (1, sv − 1)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$





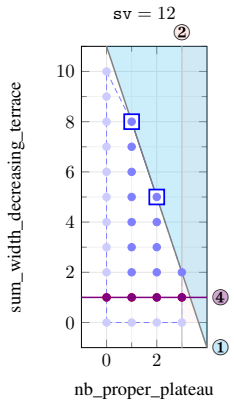
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

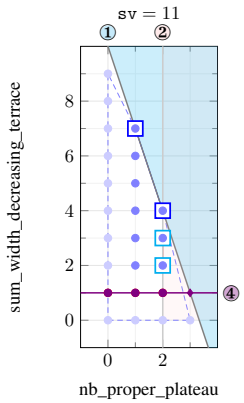
① $3 * x + y \leq sv - 1$
- $sv \geq 9$: $(1, sv - 4)$ $(2, sv - 7)$

② $y > 0 \Rightarrow 3 * x \leq sv - 3 - (sv - 3) \bmod 3$
- $sv \bmod 3 = 1 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$
- $sv \bmod 3 = 2 \wedge sv \geq 5$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

④ $y \neq 1$

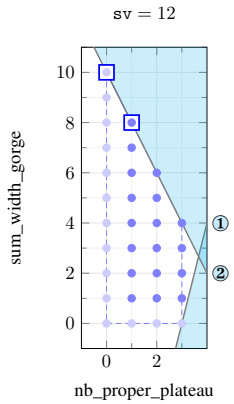


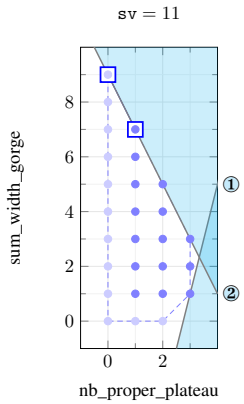


NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x \leq y + sv$

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 4)$





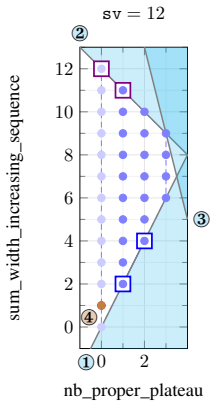
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

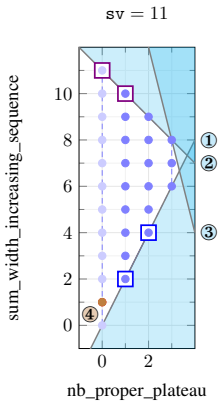
① $2 * x \leq y$
- $sv \geq 7$: (1, 2) (2, 4)

② $x + y \leq sv$
- $sv \geq 5$: (0, sv) (1, sv − 1)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$





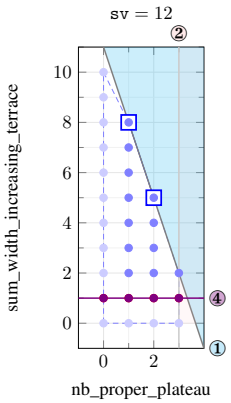
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

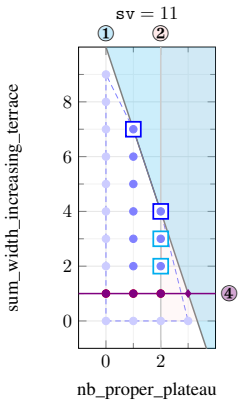
① $3 * x + y \leq sv - 1$
- $sv \geq 10$: $(1, sv - 4)$ $(2, sv - 7)$

② $y > 0 \Rightarrow 3 * x \leq sv - 3 - (sv - 3) \bmod 3$
- $sv \bmod 3 = 1 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$
- $sv \bmod 3 = 2 \wedge sv \geq 5$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

③ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

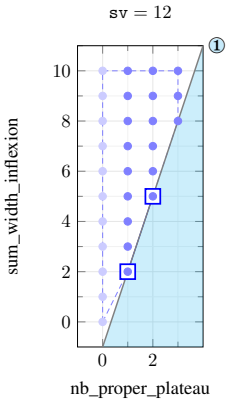
④ $y \neq 1$

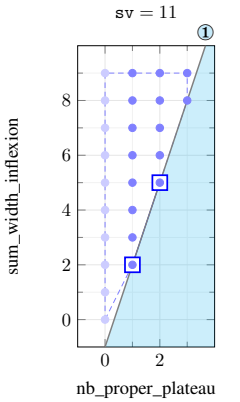




NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + 1$
- $sv \geq 7$: $(1, 2)$ $(2, 5)$

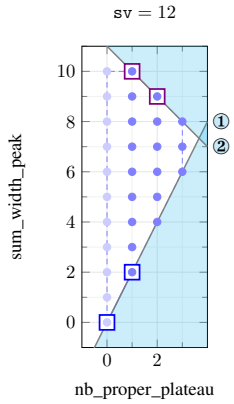


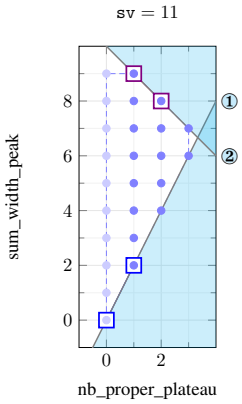


NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

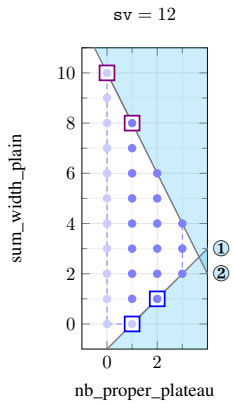


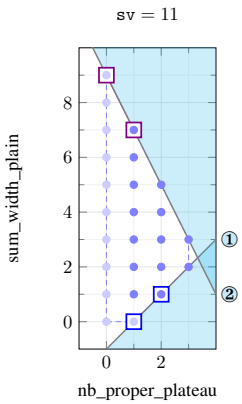


NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 7: (1, 0) (2, 1)

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- sv ≥ 4: (0, sv − 2) (1, sv − 4)



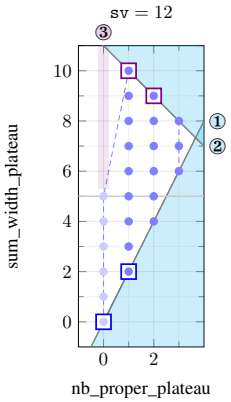


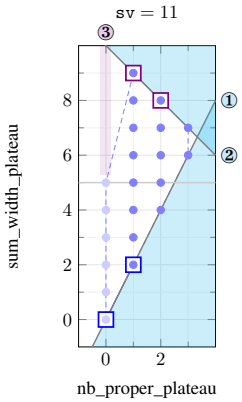
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

③ $x = 0 \Rightarrow 2 * y \leq sv - 1 - (sv - 1) \bmod 2$





NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x = \lfloor (sv - 1)/3 \rfloor - 1 \Rightarrow y \leq 6 + 2 * (sv - 1) \bmod 3$

② $x = \lfloor (sv - 1)/3 \rfloor \Rightarrow y \leq 2 * (sv - 1) \bmod 3$

③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

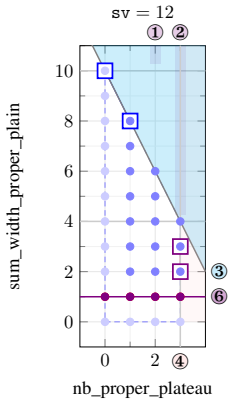
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

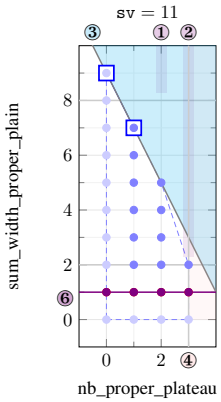
④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑤ $o^{=} \geq x + \lfloor (y + 1)/2 \rfloor$

⑥ $y \neq 1$



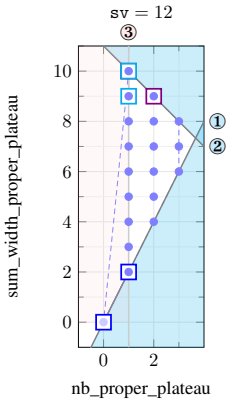


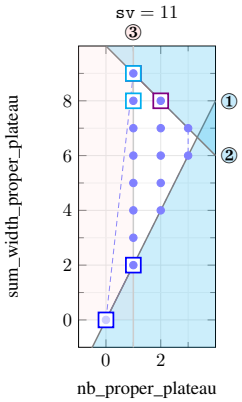
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)



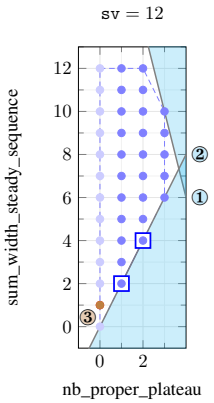


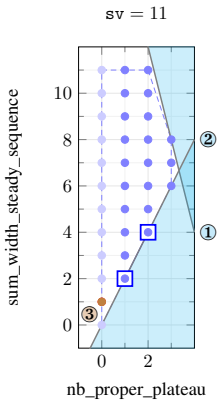
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

③ $x \neq 0 \vee y \neq 1$





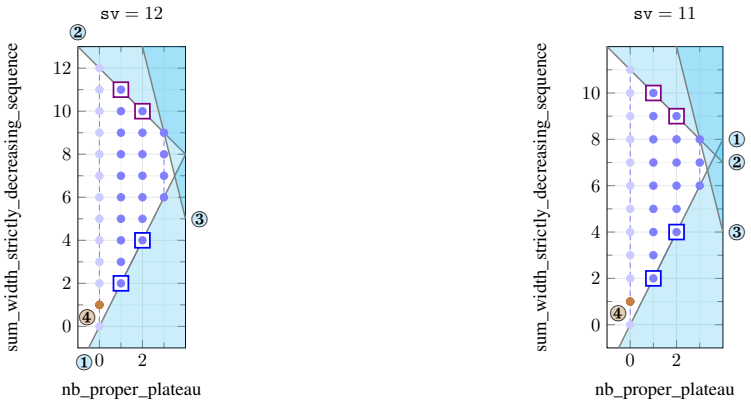
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 8: (1, sv − 1) (2, sv − 2)

③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

④ $x \neq 0 \vee y \neq 1$



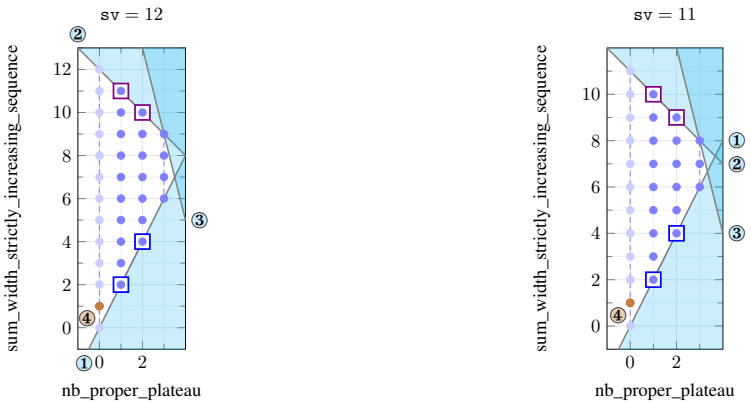
NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $x + y \leq sv$
- sv ≥ 8: (1, sv − 1) (2, sv − 2)

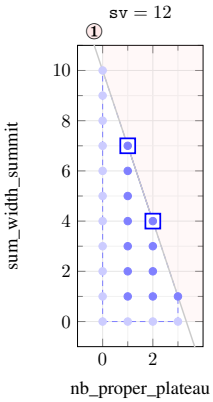
③ $sv > 1 \Rightarrow 4 * x + y \leq 2 * sv - 2 - (sv \bmod 3 = 0)$

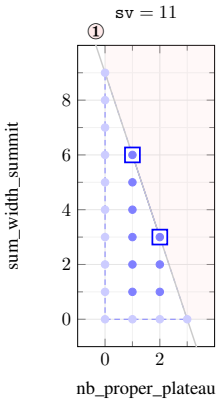
④ $x \neq 0 \vee y \neq 1$



NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 3 * x + y \leq sv - 2$
- $sv \geq 8$: $(1, sv - 5)$ $(2, sv - 8)$

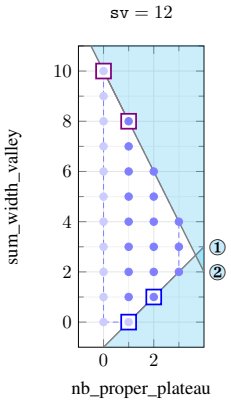


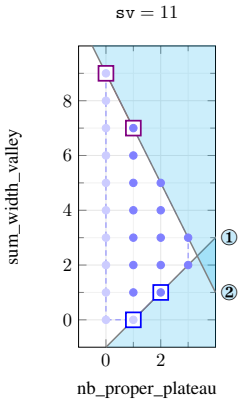


NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 7$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 4)$





NB_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x = \lfloor (sv - 1)/3 \rfloor - 1 \Rightarrow y \leq 6 - 3 * (sv \bmod 3 = 1)$

② $x = \lfloor (sv - 1)/3 \rfloor - 2 \Rightarrow y \leq 9 + 3 * (sv \bmod 3 = 0)$

③ $x = \lfloor (sv - 1)/3 \rfloor \Rightarrow y \leq 3 * (sv \bmod 3 = 0)$

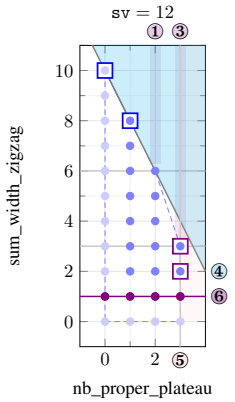
④ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

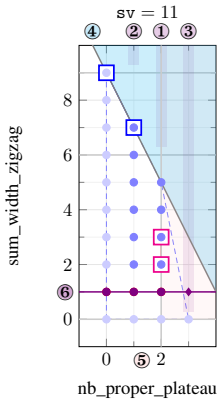
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

⑤ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 3 - (sv - 3) \bmod 3$

- $sv \bmod 3 = 0 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $sv \bmod 3 = 1 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$
- $sv \bmod 3 = 2 \wedge sv \geq 5$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑥ $y \neq 1$





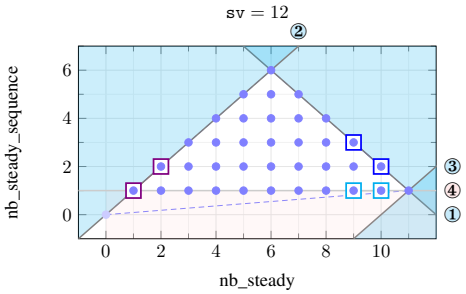
NB_STEADY(x, VARIABLES) ∧
NB_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

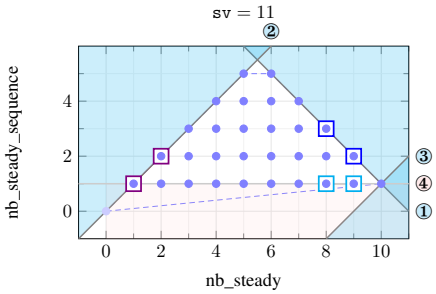
① $x + y \leq sv$
- $sv \geq 6$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $y \leq x$
- $sv \geq 4$: $(1, 1)$ $(2, 2)$

③ $sv > 1 \Rightarrow x \leq y + sv - 2$

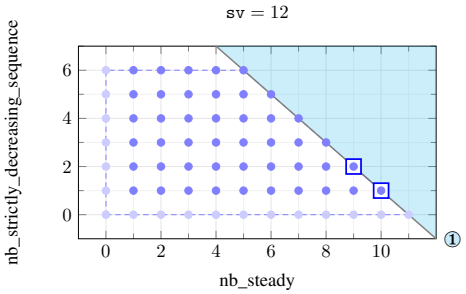
④ $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$

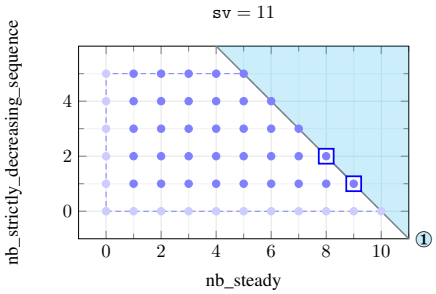




NB_STEADY(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

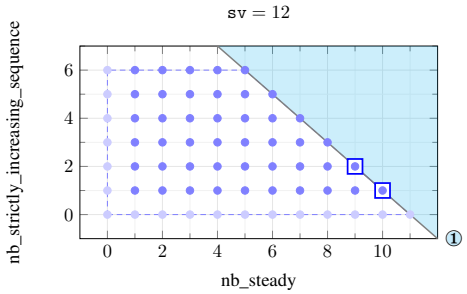


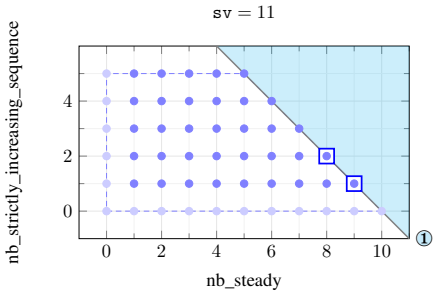


NB_STEADY(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 2)$

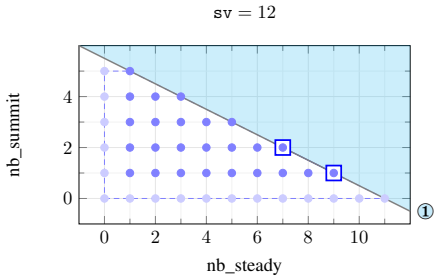


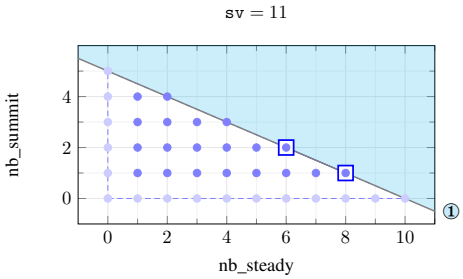


NB_STEADY(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$

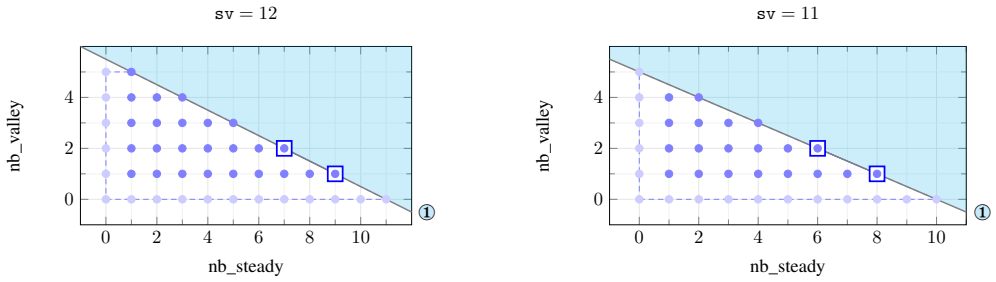
- $sv \geq 5$: $(sv - 3, 1)$ $(sv - 5, 2)$





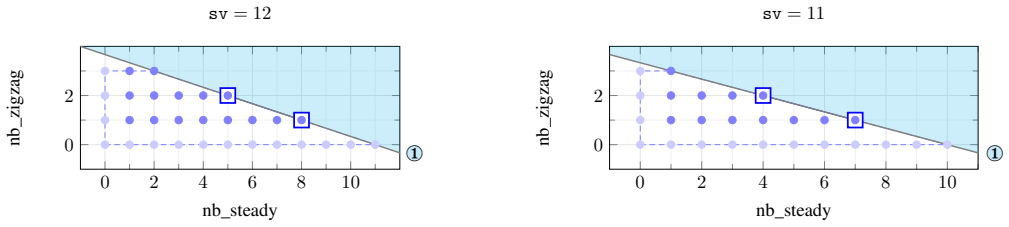
NB_STEADY(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv - 1$
- $sv \geq 5$: $(sv - 3, 1)$ $(sv - 5, 2)$



NB_STEADY(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + 3 * y \leq sv - 1$
- $sv \geq 7$: $(sv - 4, 1)$ $(sv - 7, 2)$



NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x = sv - 2 \Rightarrow y \leq 2$

② $sv > 1 \Rightarrow 2 * x + y \leq 3 * sv - 4$

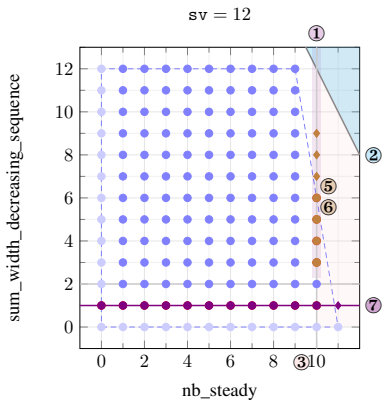
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$

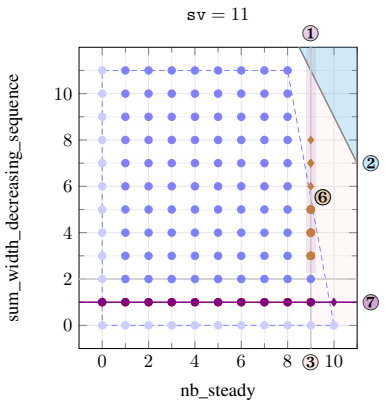
④ $\bigvee \begin{pmatrix} x \neq \max(0, sv - 1) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 4, \\ sv < 7 \end{pmatrix}$

⑤ $x \neq \max(0, sv - 1) - 1 \vee y < 3 \vee y > sv * \min(1, \max(0, sv - 1)) - 3$

⑥ $\bigvee \begin{pmatrix} x \neq \max(0, sv - 1) - 1, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{pmatrix}$

⑦ $y \neq 1$





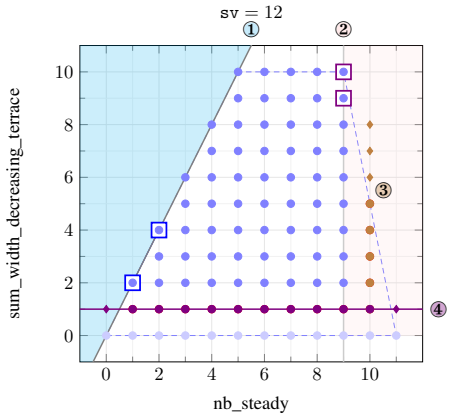
NB_STEADY$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_DECREASING_TERRACE$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

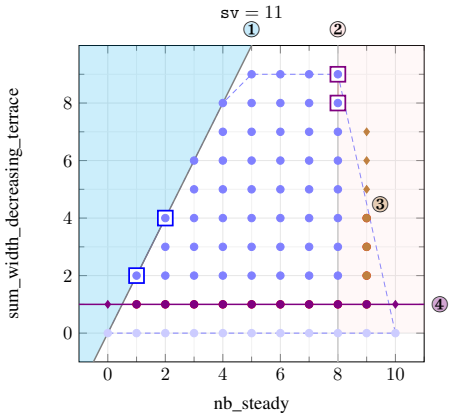
① $\mathtt{y} \leq 2 * \mathtt{x}$
- $\mathtt{sv} \geq 6$: $(1, 2)$ $(2, 4)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 3)$

③ $\bigvee \begin{pmatrix} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \end{pmatrix}$

④ $\mathtt{y} \neq 1$





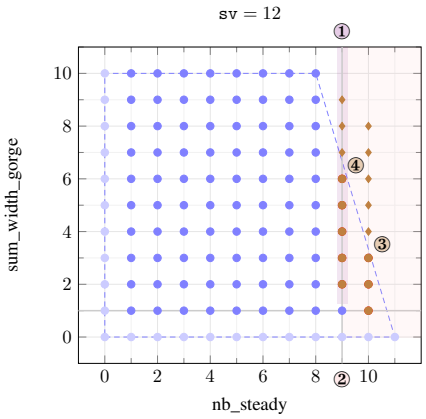
NB_STEADY($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_GORGE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

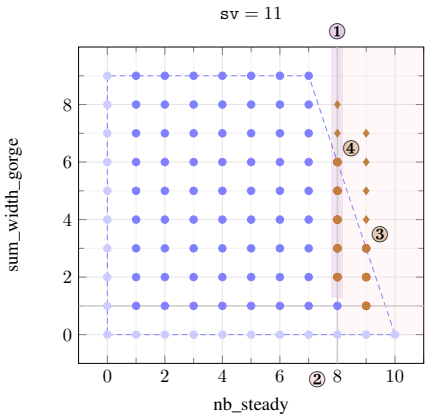
① $\mathtt{x} = \mathtt{sv} - 3 \Rightarrow \mathtt{y} \leq 1$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 2, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1 \end{array} \right)$





NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x = sv - 2 \Rightarrow y \leq 2$

② $sv > 1 \Rightarrow 2 * x + y \leq 3 * sv - 4$

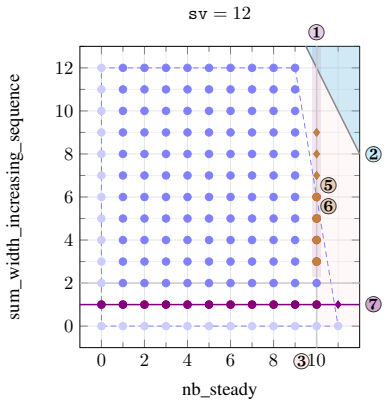
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 2$

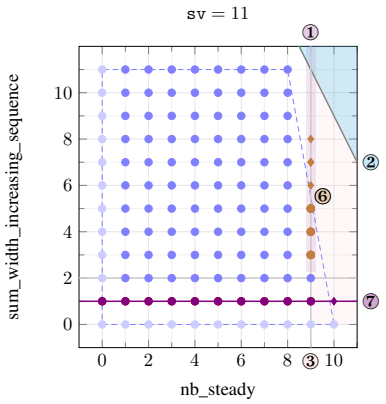
④ $\bigvee \begin{pmatrix} x \neq \max(0, sv - 1) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 4, \\ sv < 7 \end{pmatrix}$

⑤ $x \neq \max(0, sv - 1) - 1 \vee y < 3 \vee y > sv * \min(1, \max(0, sv - 1)) - 3$

⑥ $\bigvee \begin{pmatrix} x \neq \max(0, sv - 1) - 1, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{pmatrix}$

⑦ $y \neq 1$





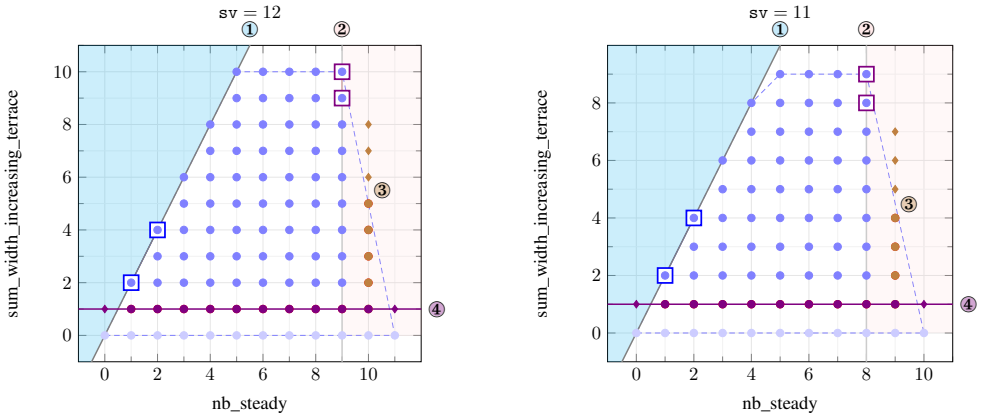
NB_STEADY($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INCREASING_TERRACE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{y} \leq 2 * \mathtt{x}$
- $\mathtt{sv} \geq 6$: $(1, 2)$ $(2, 4)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 3)$

③ $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \end{array} \right)$$

④ $\mathtt{y} \neq 1$

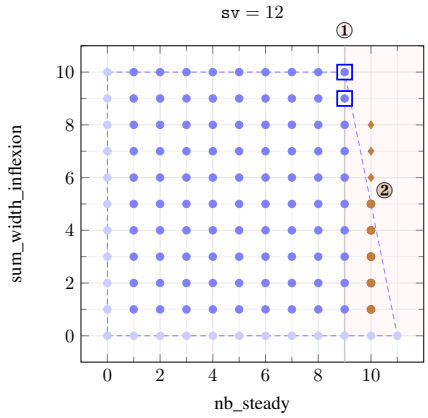


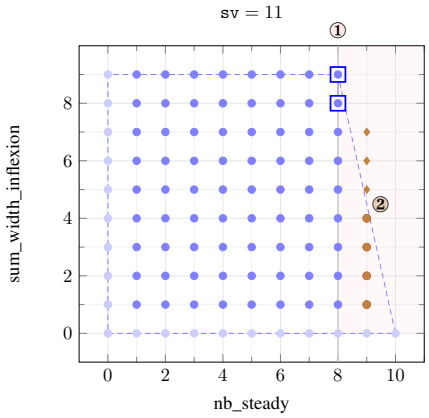
NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

□ $sv \geq 3$: $(sv-3, sv-2)$ $(sv-3, sv-3)$

② $\bigvee \begin{pmatrix} x \neq \max(0, sv-1) - 1, \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 2 \end{pmatrix}$



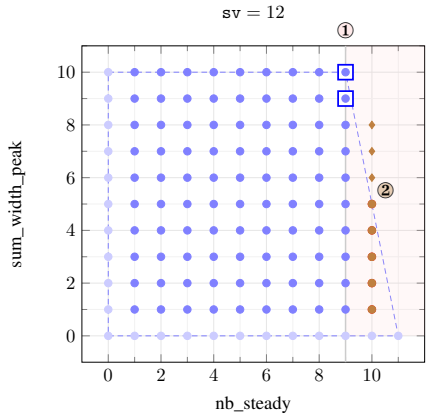


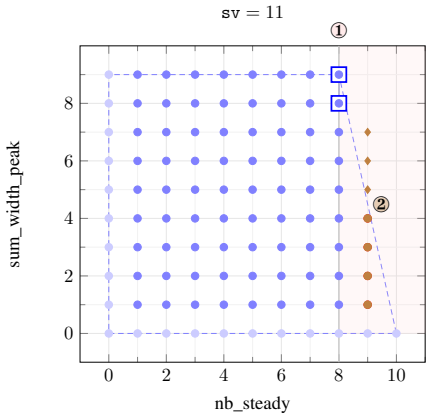
NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

□ $sv \geq 3$: $(sv-3, sv-2)$ $(sv-3, sv-3)$

② $\bigvee \begin{pmatrix} x \neq \max(0, sv-1) - 1, \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 2 \end{pmatrix}$



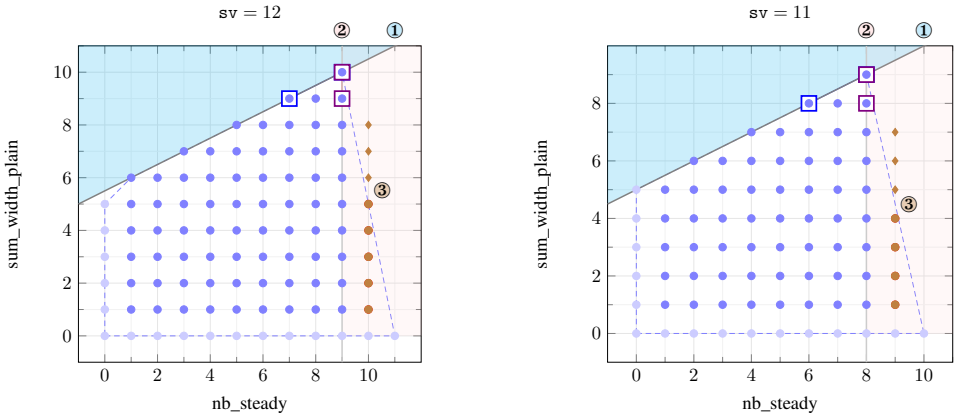


NB_STEADY$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_PLAIN$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $2 * \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 1$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 5, \mathtt{sv} - 3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 3$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 3)$

③ $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2 \end{array} \right)$$

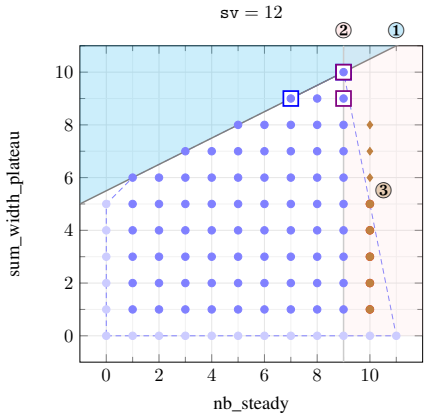


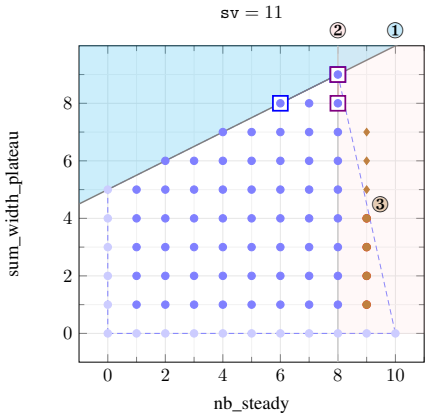
NB_STEADY$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_PLATEAU$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $2 * \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 1$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-5, \mathtt{sv}-3)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 3$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-3, \mathtt{sv}-3)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv}-1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2 \end{array} \right)$





NB_STEADY($\mathtt{x}, \mathtt{VARIABLES}$) $\wedge$
SUM_WIDTH_PROPER_PLAIN($\mathtt{y}, \mathtt{VARIABLES}$) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $2 * \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 1$
- $\mathtt{sv} \geq 7$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 5, \mathtt{sv} - 3)$

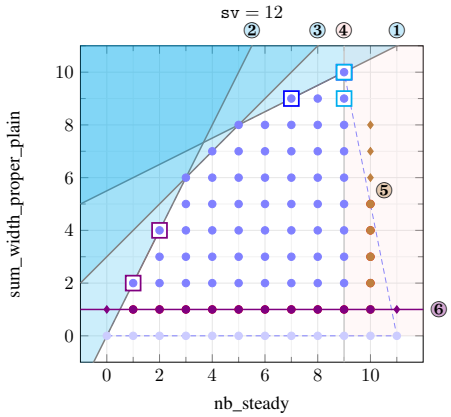
② $\mathtt{y} \leq 2 * \mathtt{x}$
- $\mathtt{sv} \geq 7$: $(1, 2)$ $(2, 4)$

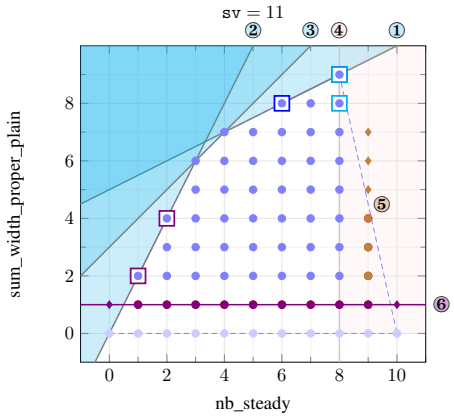
③ $3 * \mathtt{y} \leq 3 * \mathtt{x} + \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 3$

④ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 3, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 3)$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv} - 1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \end{pmatrix}$

⑥ $\mathtt{y} \neq 1$





NB_STEADY$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_PROPER_PLATEAU$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $2 * \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 1$
- $\mathtt{sv} \geq 7$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-5, \mathtt{sv}-3)$

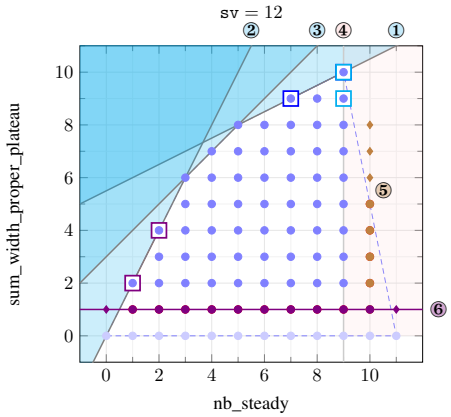
② $\mathtt{y} \leq 2 * \mathtt{x}$
- $\mathtt{sv} \geq 7$: $(1, 2)$ $(2, 4)$

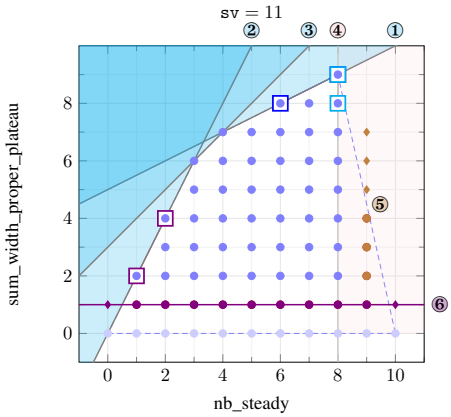
③ $3 * \mathtt{y} \leq 3 * \mathtt{x} + \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 3$

④ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 5$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-3, \mathtt{sv}-3)$

⑤ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv}-1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2 \end{array} \right)$

⑥ $\mathtt{y} \neq 1$





NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

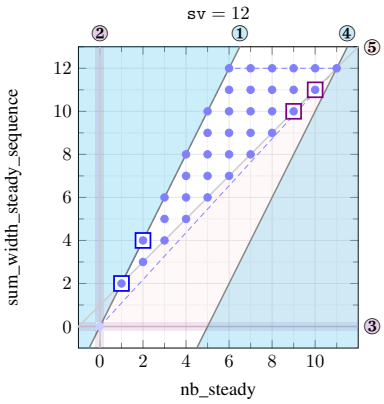
① $y \leq 2 * x$
- $sv \geq 4$: $(1, 2)$ $(2, 4)$

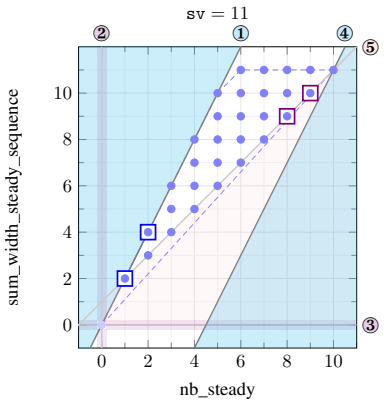
② $x = 0 \Rightarrow y = 0$

③ $y = 0 \Rightarrow x = 0$

④ $sv > 1 \Rightarrow 2 * x \leq y + sv - 2$

⑤ $x > 0 \wedge y > 0 \Rightarrow x \leq y - 1$
- $sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv - 2)$

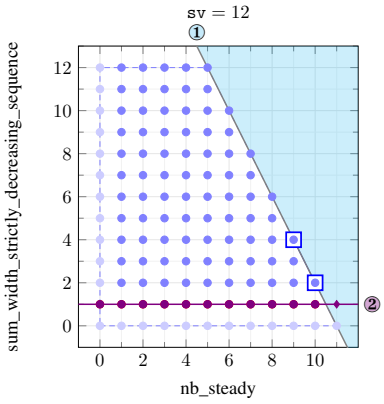


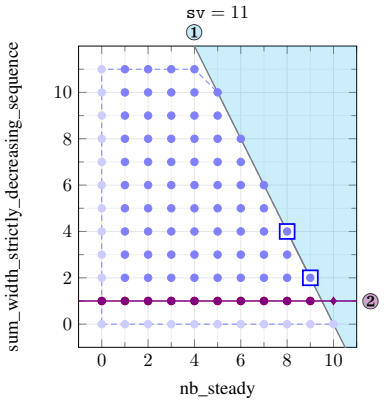


NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$

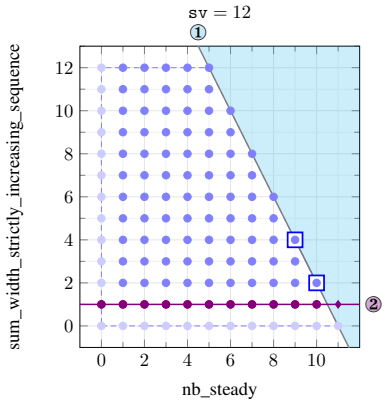


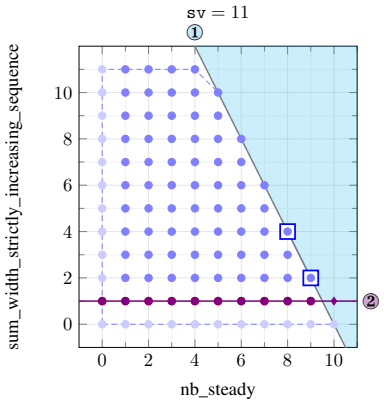


NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $y \neq 1$





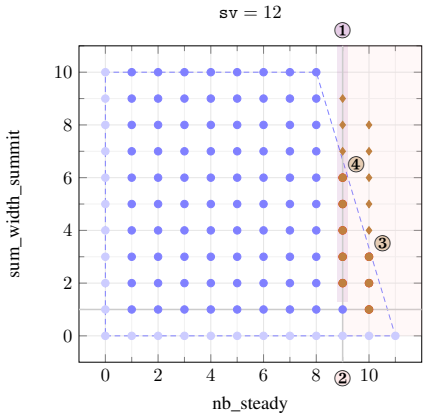
NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

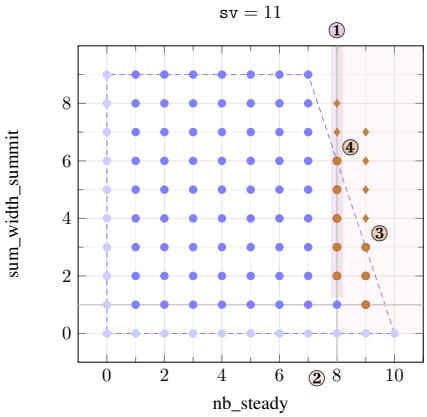
① $x = sv - 3 \Rightarrow y \leq 1$

② $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

③ $\bigvee \left(\begin{array}{l} x \neq \max(0, sv - 1) - 1, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} x \neq \max(0, sv - 1) - 2, \\ y < 2, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 1 \end{array} \right)$



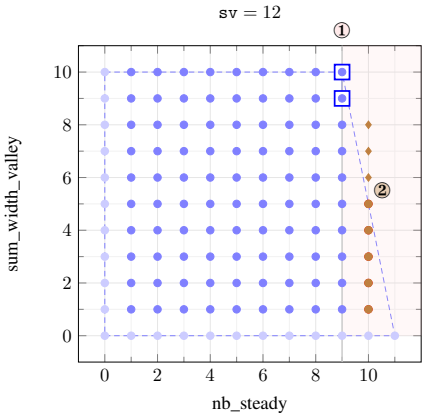


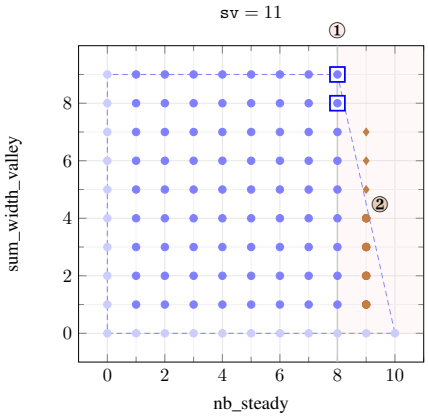
NB_STEADY($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_VALLEY($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

- $\mathtt{sv} \geq 3$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-3, \mathtt{sv}-3)$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \mathtt{sv}-1) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2 \end{array} \right)$



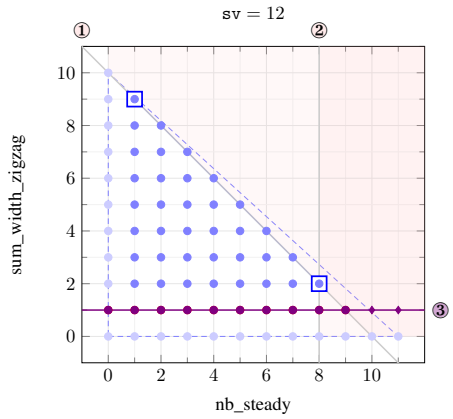


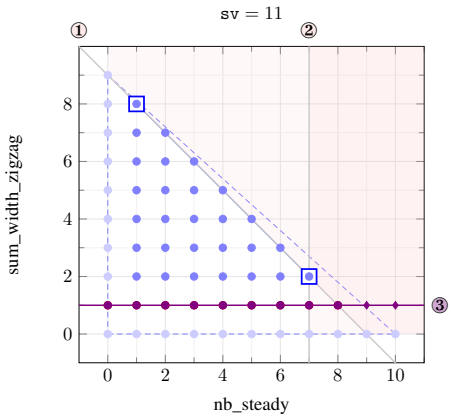
NB_STEADY(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 4, 2)$ $(1, sv - 3)$

② $y > 0 \Rightarrow x \leq sv - 4$

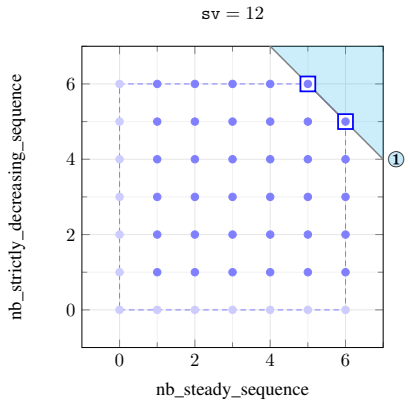
③ $y \neq 1$

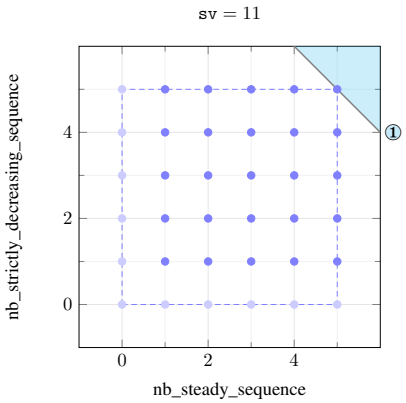




NB_STEADY_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

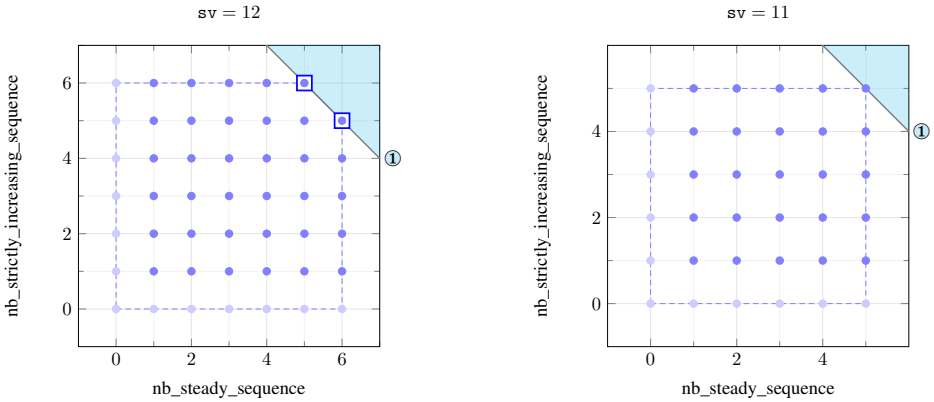
① $x + y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor sv/2 \rfloor - 1, \lfloor sv/2 \rfloor)$





NB_STEADY_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

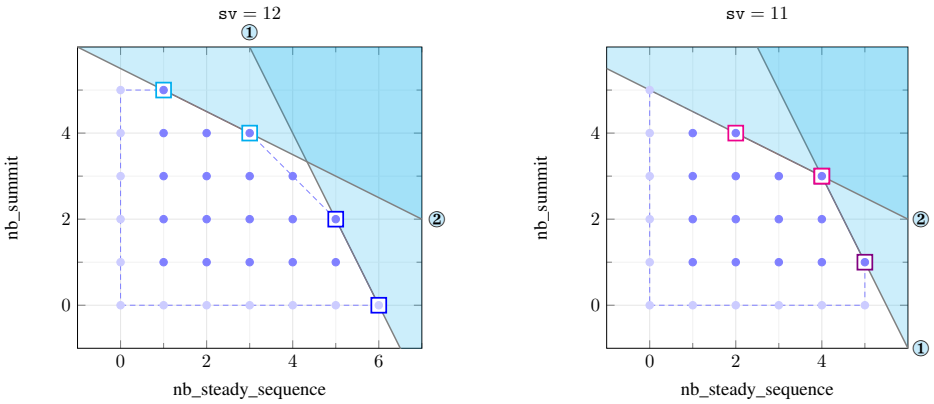
① $x + y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 2: (⌊sv/2⌋, ⌊sv/2⌋ − 1) (⌊sv/2⌋ − 1, ⌊sv/2⌋)



NB_STEADY_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

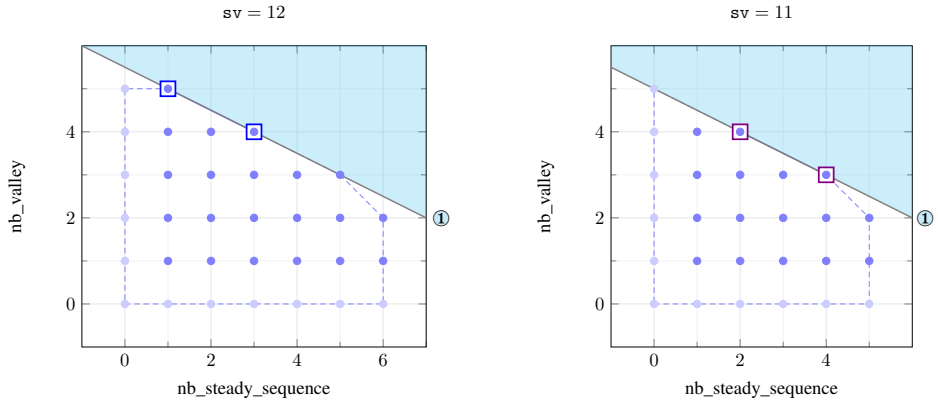
① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 8: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 2)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊sv/2⌋, 1) (⌊sv/2⌋ − 1, 3)

② $x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 8: (1, ⌊(sv − 1)/2⌋) (3, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 11: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)



NB_STEADY_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

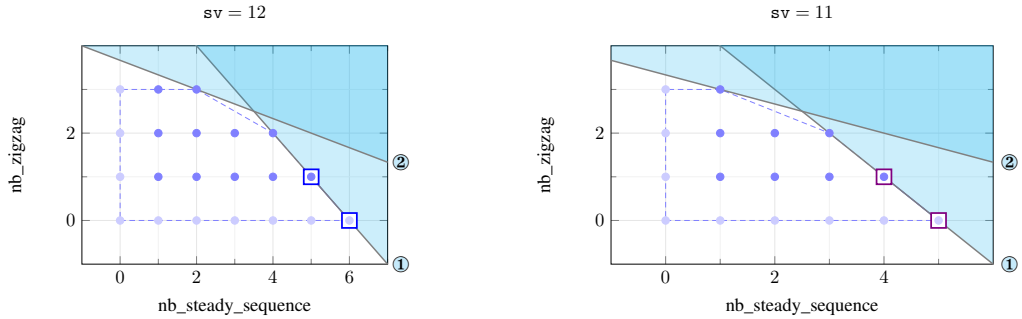
① $x + 2 * y \leq sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 6: (1, ⌊(sv − 1)/2⌋) (3, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 9: (2, ⌊(sv − 1)/2⌋ − 1) (4, ⌊(sv − 1)/2⌋ − 2)



NB_STEADY_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq sv - sv \bmod 2$
- sv mod 2 = 0 ∧ sv ≥ 6: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 1)
- sv mod 2 = 1 ∧ sv ≥ 5: (⌊sv/2⌋, 0) (⌊sv/2⌋ − 1, 1)

② $x + 3 * y \leq sv - 1$
- sv mod 3 = 0 ∧ sv ≥ 18: (2, ⌊(sv − 1)/3⌋) (5, ⌊(sv − 1)/3⌋ − 1)
- sv mod 3 = 1 ∧ sv ≥ 10: (0, ⌊(sv − 1)/3⌋) (3, ⌊(sv − 1)/3⌋ − 1)
- sv mod 3 = 2 ∧ sv ≥ 14: (1, ⌊(sv − 1)/3⌋) (4, ⌊(sv − 1)/3⌋ − 1)



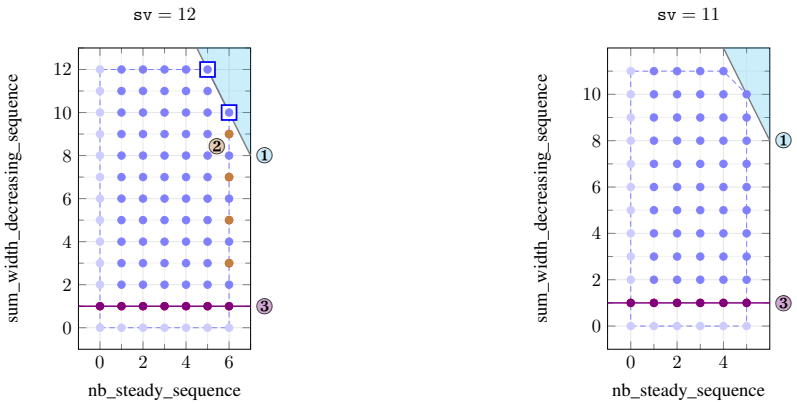
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \begin{pmatrix} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{pmatrix}$

③ $\mathtt{y} \neq 1$



NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$

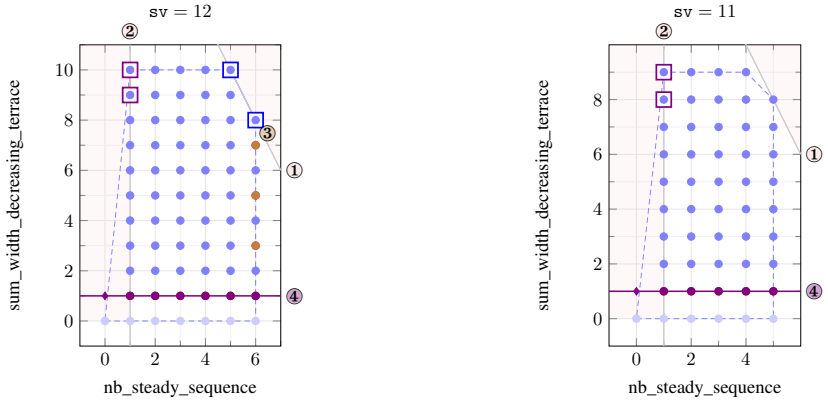
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, sv - 4)$ $(\lfloor sv/2 \rfloor - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

③ $$\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 3)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$$

④ $y \neq 1$



NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_GORGE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

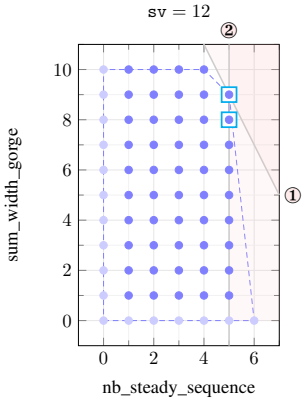
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 5$

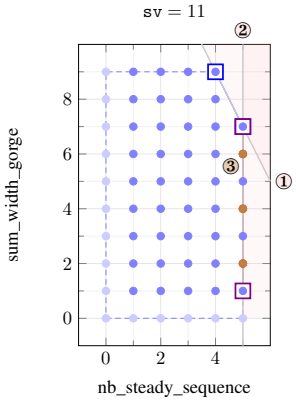
- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 5$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 2$

- $(\mathtt{sv} - 1) \bmod 2 = 0 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor, 1)$
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 3)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 4)$

③ $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 3, \\ 0 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$$





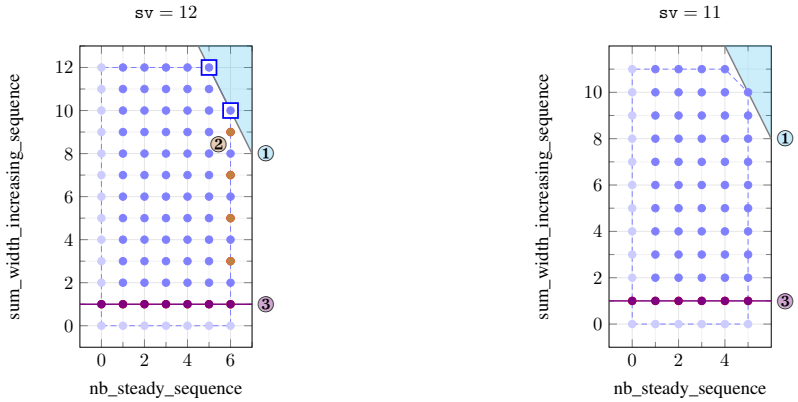
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$$

③ $\mathtt{y} \neq 1$



NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INCREASING_TERRACE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 4$

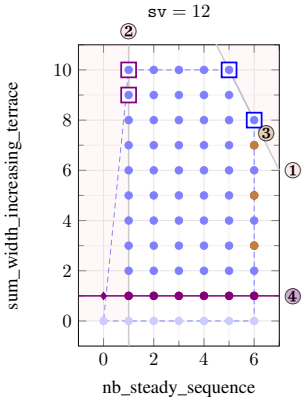
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 2)$

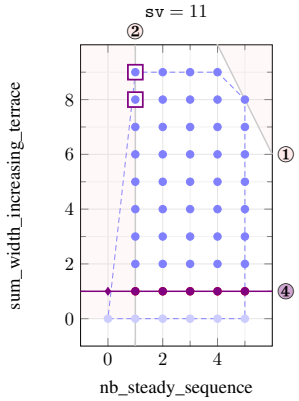
② $\mathtt{y} > 0 \Rightarrow \mathtt{x} \geq 1$

- $\mathtt{sv} \geq 5$: $(1, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 3)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

④ $\mathtt{y} \neq 1$





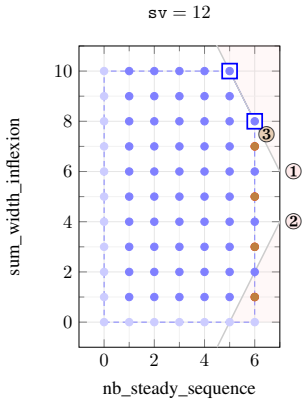
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

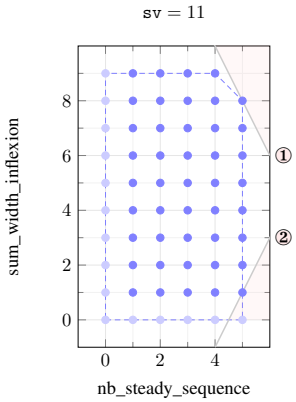
① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$

□ $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, sv - 4)$ $(\lfloor sv/2 \rfloor - 1, sv - 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$

③ $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \mathtt{min}(1, \mathtt{max}(0, sv - 2)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$





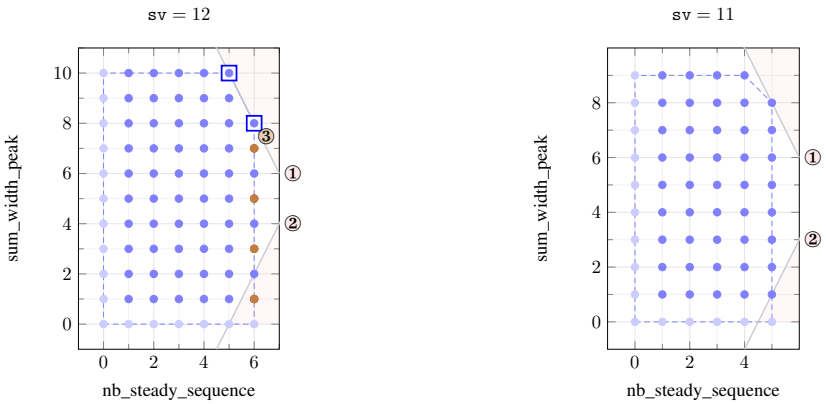
NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_PEAK($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 4$

- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 2$

③ $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$$



NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

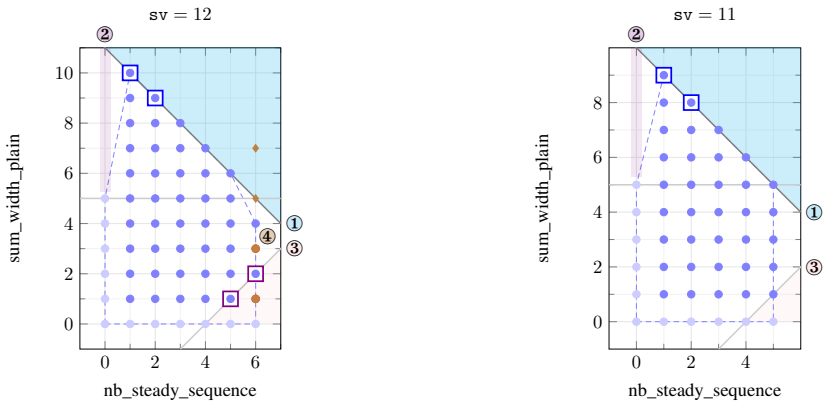
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x = 0 \Rightarrow 2 * y \leq sv - 1 - (sv - 1) \bmod 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq 2 * y + sv - 3 - (sv - 3) \bmod 2$

- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 1)$

④ $$\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$$



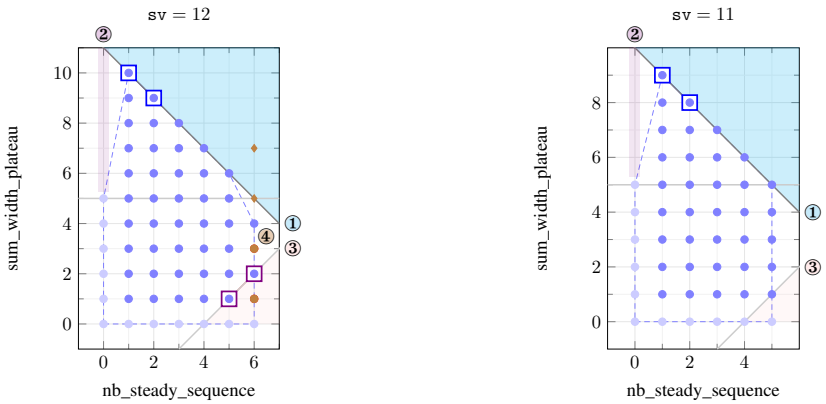
NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_PLATEAU($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 1$
- $\mathtt{sv} \geq 5$: $(1, \mathtt{sv} - 2)$ $(2, \mathtt{sv} - 3)$

② $\mathtt{x} = 0 \Rightarrow 2 * \mathtt{y} \leq \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 2$

③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq 2 * \mathtt{y} + \mathtt{sv} - 3 - (\mathtt{sv} - 3) \bmod 2$
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor, 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, 1)$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$



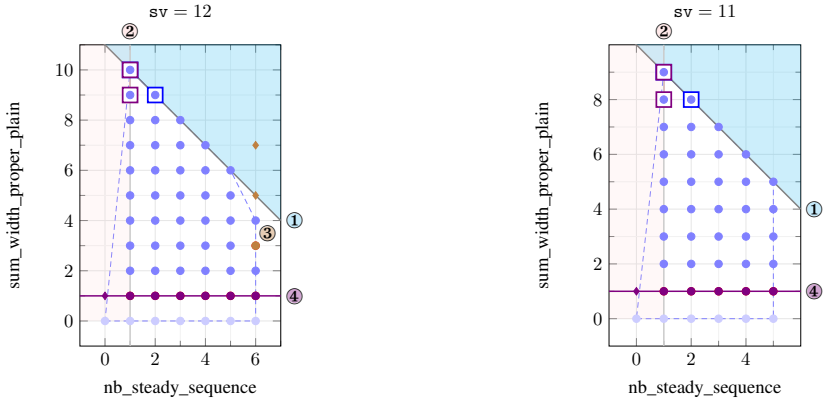
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

③ $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \mathtt{min}(1, \mathtt{max}(0, sv - 3)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

④ $y \neq 1$



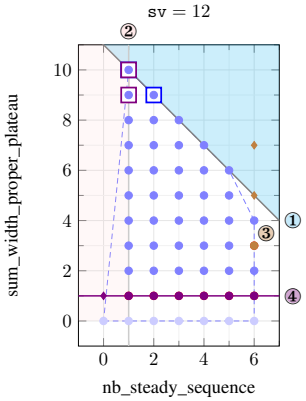
NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_PROPER_PLATEAU($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

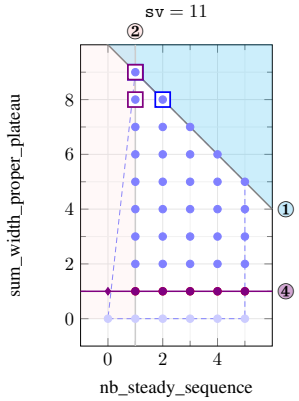
① $\mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 1$
- $\mathtt{sv} \geq 5$: $(1, \mathtt{sv} - 2)$ $(2, \mathtt{sv} - 3)$

② $\mathtt{y} > 0 \Rightarrow \mathtt{x} \geq 1$
- $\mathtt{sv} \geq 5$: $(1, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 3)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

④ $\mathtt{y} \neq 1$



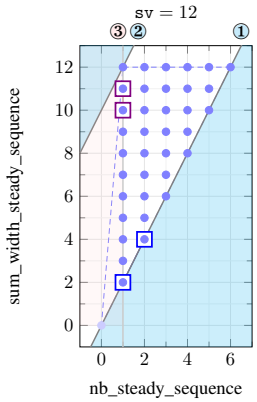


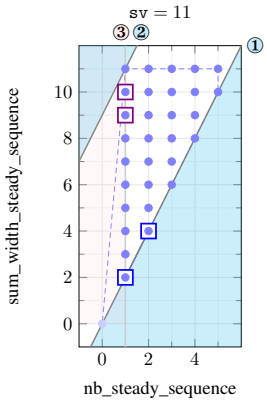
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 1) (1, sv − 2)





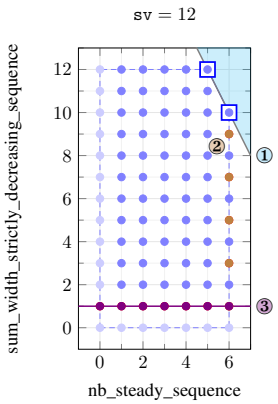
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

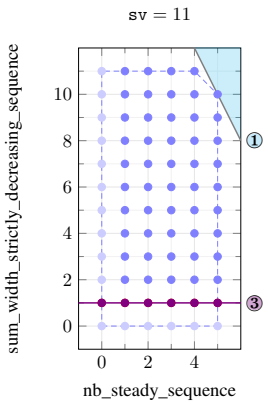
① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \begin{pmatrix} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{pmatrix}$

③ $\mathtt{y} \neq 1$





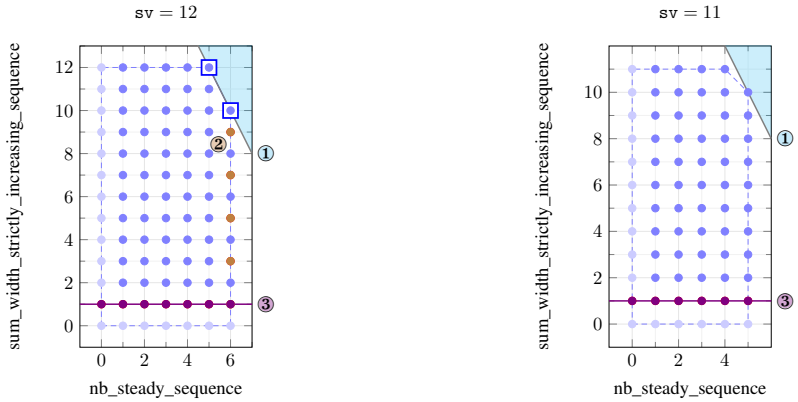
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

③ $\mathtt{y} \neq 1$



NB_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_SUMMIT($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

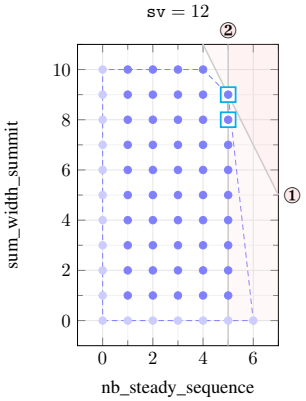
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 5$

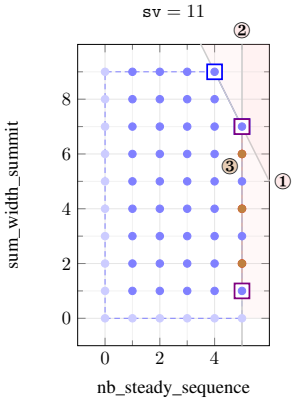
- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 5$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 2$

- $(\mathtt{sv} - 1) \bmod 2 = 0 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 4)$ $(\lfloor \mathtt{sv}/2 \rfloor, 1)$
- $(\mathtt{sv} - 1) \bmod 2 = 1 \wedge \mathtt{sv} \geq 6$: $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 3)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv} - 4)$

③ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 3, \\ 0 = \mathtt{sv} \bmod 2, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$





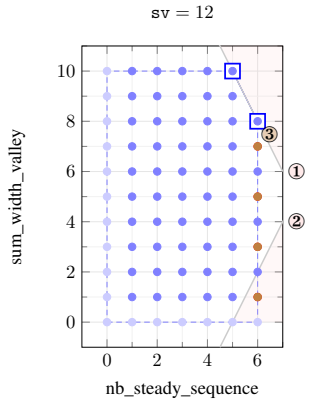
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

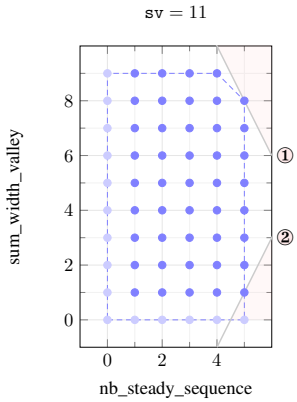
① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$

□ $sv \bmod 2 = 0 \wedge sv \geq 6$: $(\lfloor sv/2 \rfloor, sv - 4)$ $(\lfloor sv/2 \rfloor - 1, sv - 2)$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$

③ $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \mathtt{min}(1, \mathtt{max}(0, sv - 2)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$





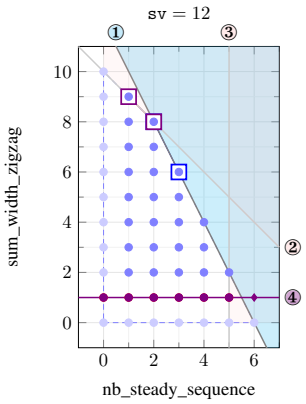
NB_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

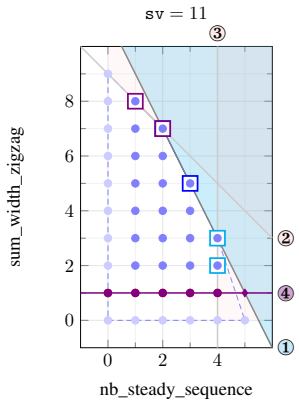
① $2 * x + y \leq sv$
- $sv \geq 8$: $(2, sv - 4)$ $(3, sv - 6)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(1, sv - 3)$ $(2, sv - 4)$

③ $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

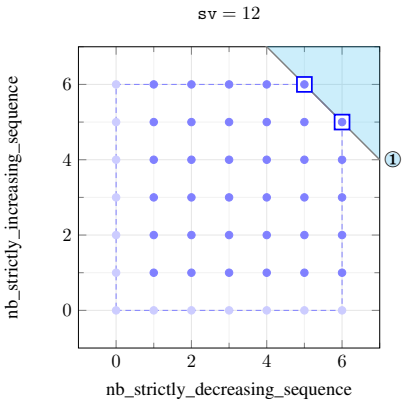
④ $y \neq 1$

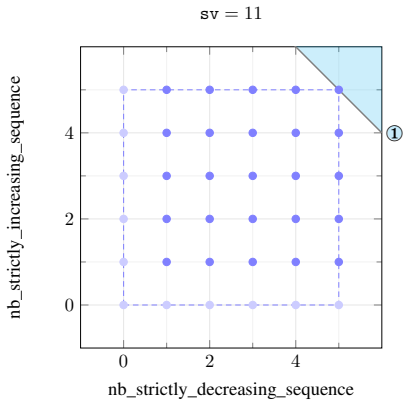




NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, \lfloor sv/2 \rfloor - 1)$ $(\lfloor sv/2 \rfloor - 1, \lfloor sv/2 \rfloor)$

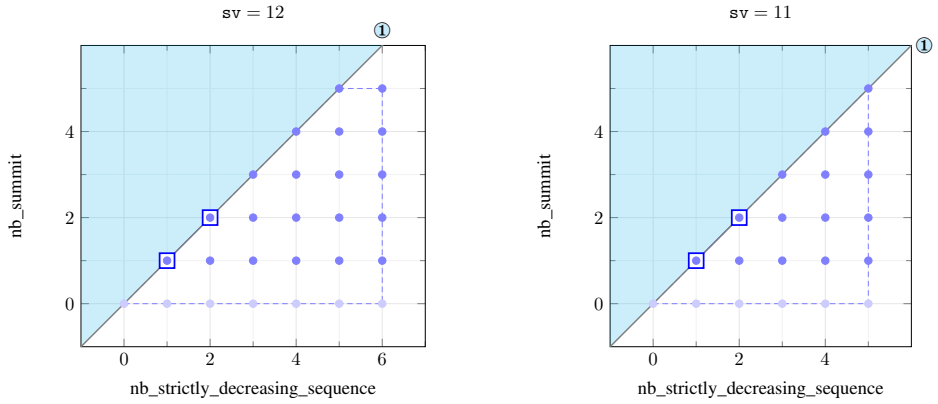




NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

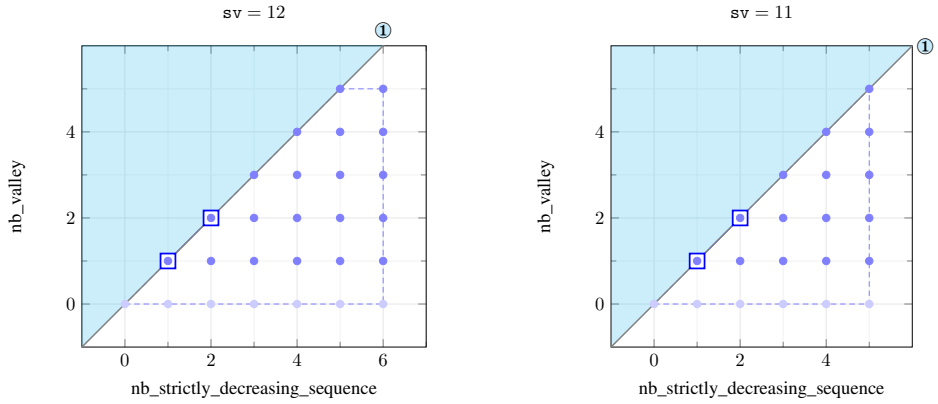
- sv ≥ 5: (1, 1) (2, 2)



NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

- sv ≥ 5: (1, 1) (2, 2)

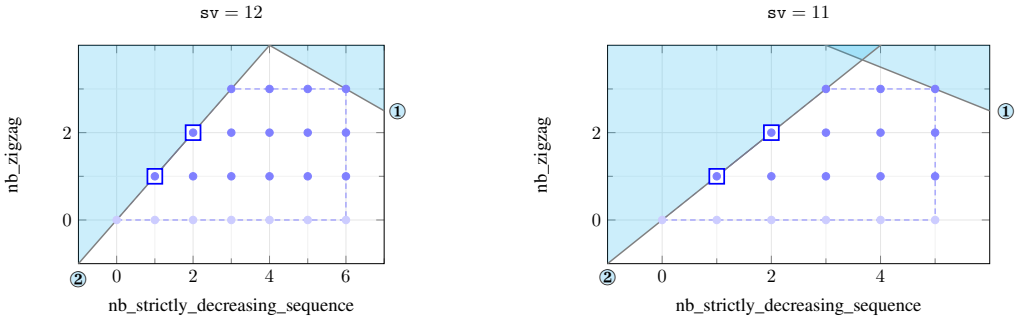


NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$

② $y \leq x$

- $sv \geq 7$: (1, 1) (2, 2)



NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

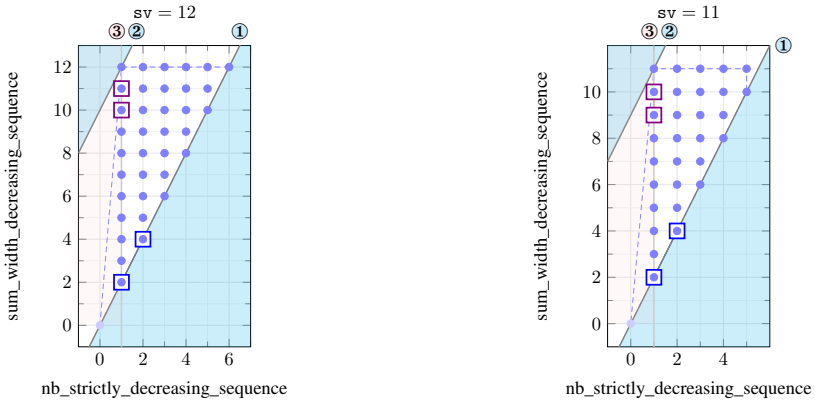
① $2 * x \leq y$

- $sv \geq 4$: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: (1, sv − 1) (1, sv − 2)



NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x = 0 \Rightarrow y = 0$

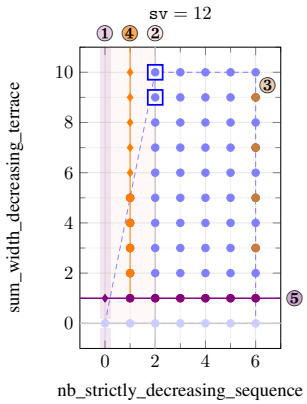
② $x > 0 \wedge y > 0 \Rightarrow x \geq 2$

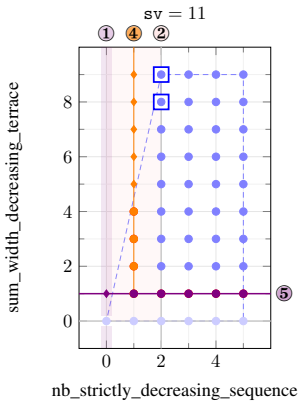
□ $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

③ $$\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \mathtt{min}(1, \mathtt{max}(0, sv - 3)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$$

④ $x \neq 1 \vee y < 1$

⑤ $y \neq 1$



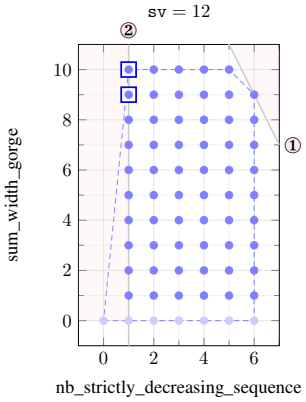


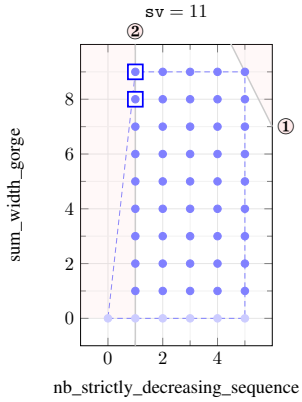
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$





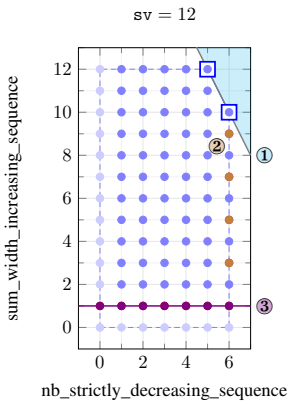
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

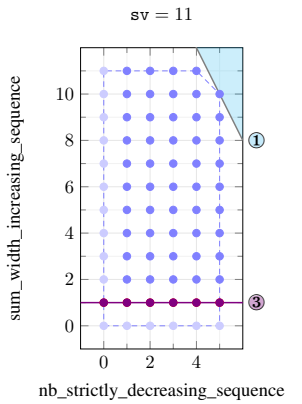
① $2 * x + y \leq 2 * sv - 2$

- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

③ $y \neq 1$





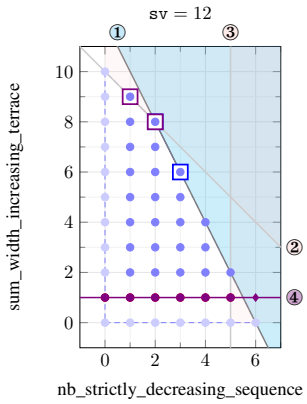
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

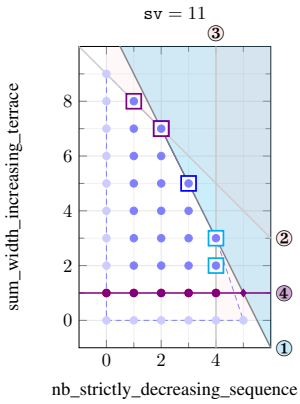
① $2 * x + y \leq sv$
- $sv \geq 8$: $(2, sv - 4)$ $(3, sv - 6)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(1, sv - 3)$ $(2, sv - 4)$

③ $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

④ $y \neq 1$





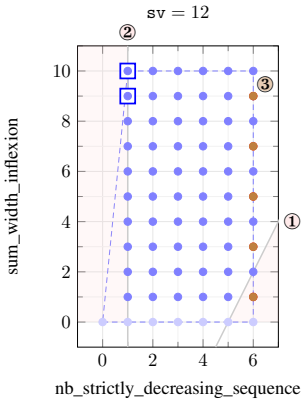
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

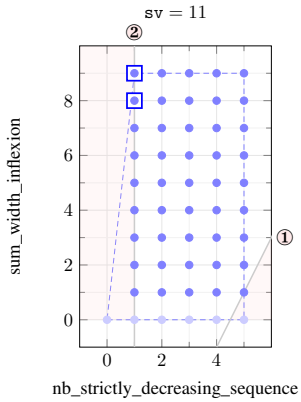
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

③ $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$



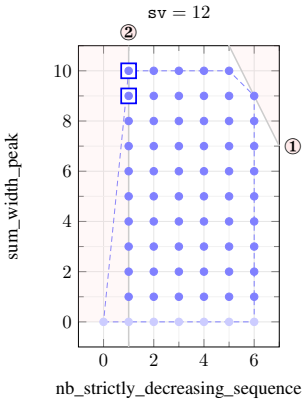


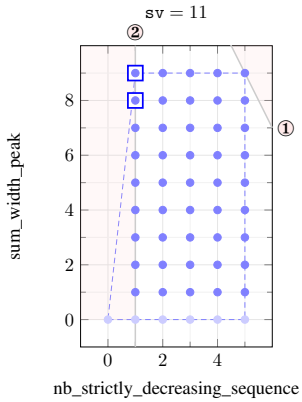
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

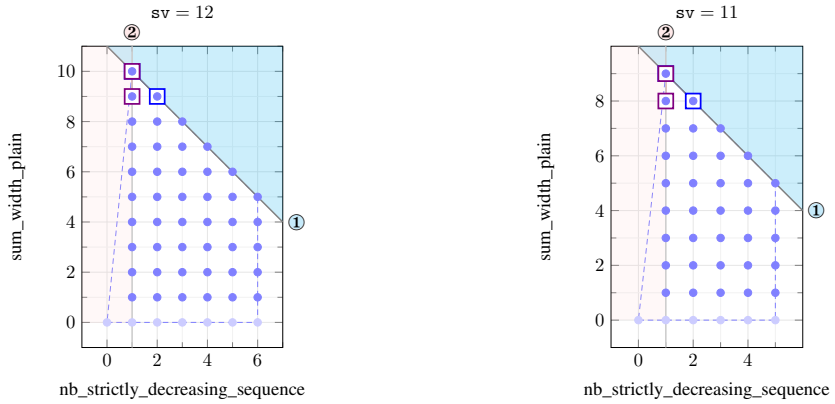




NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

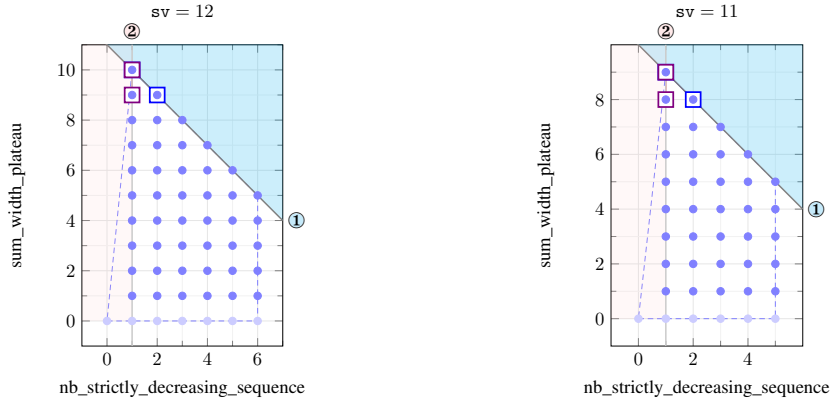
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



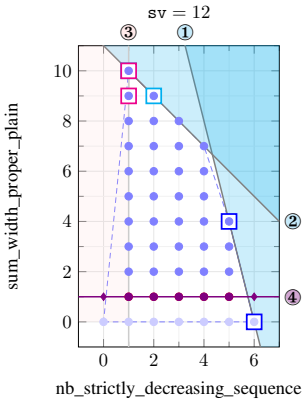
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

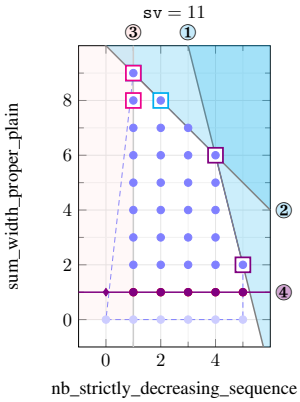
① $4 * x + y \leq 2 * sv$
- sv mod 2 = 0 ∧ sv ≥ 8: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- sv ≥ 5: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





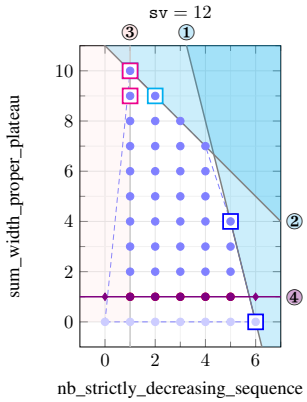
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

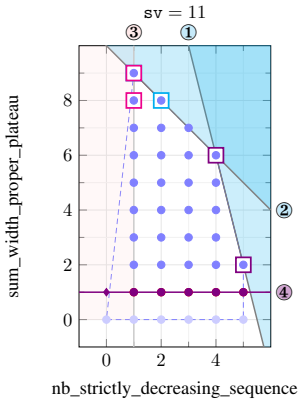
① $4 * x + y \leq 2 * sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





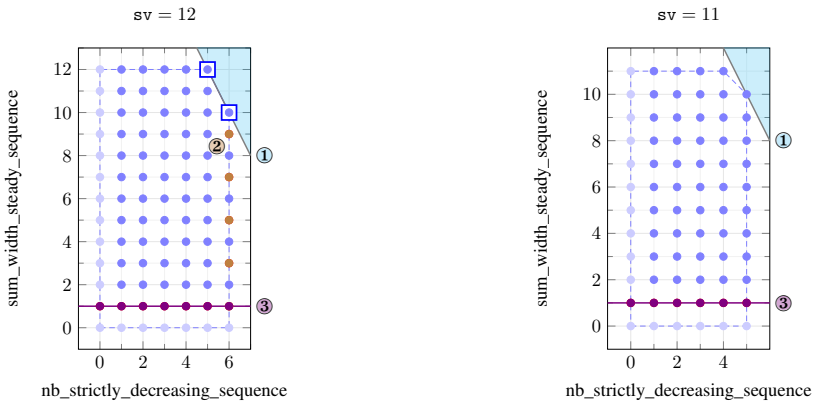
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \begin{pmatrix} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{pmatrix}$

③ $\mathtt{y} \neq 1$

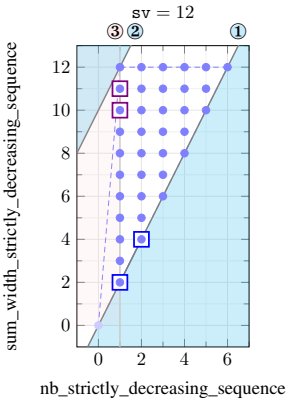


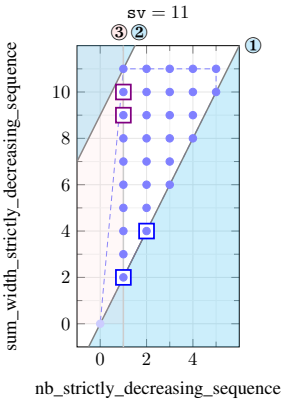
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 1) (1, sv − 2)





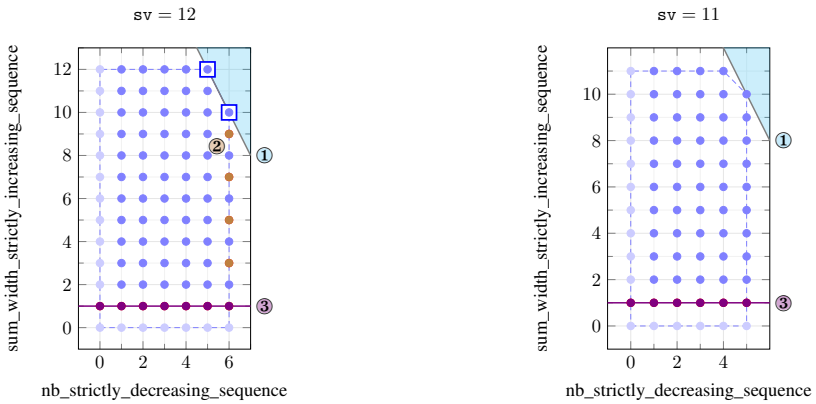
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $\bigvee \begin{pmatrix} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{pmatrix}$

③ $y \neq 1$

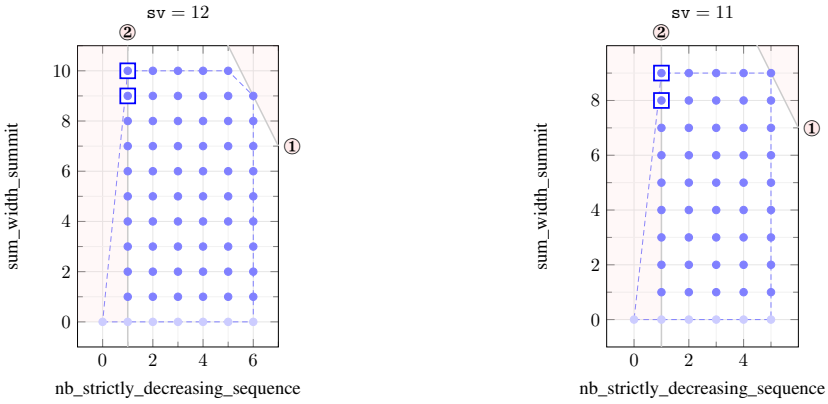


NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



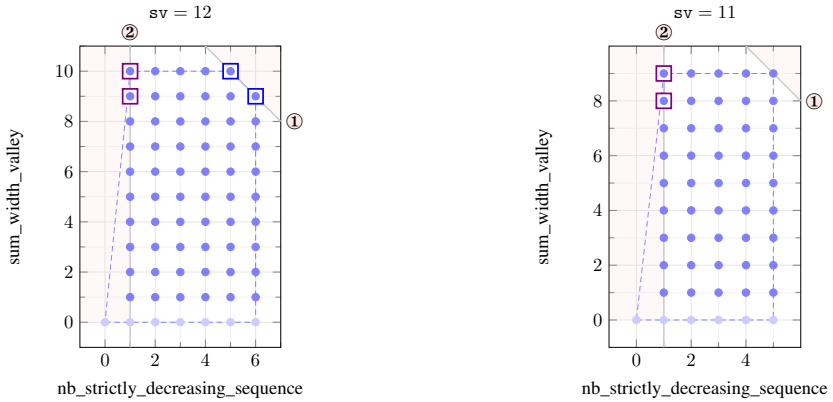
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + 2 * y \leq 3 * sv - 5 - (3 * sv - 5) \bmod 2$

- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, sv - 3)$ $(\lfloor sv/2 \rfloor - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



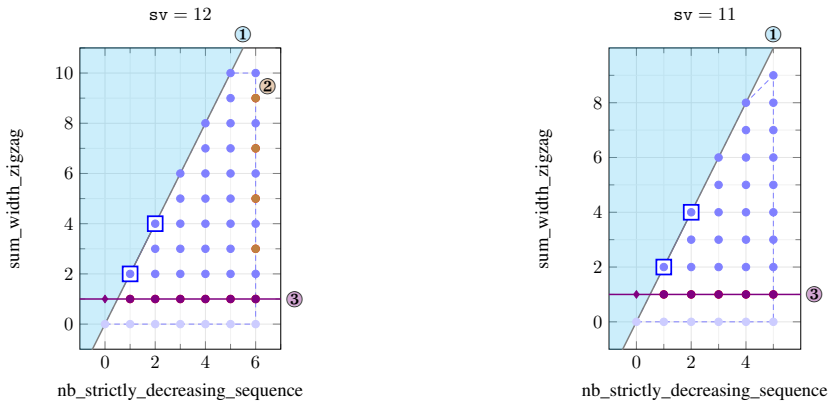
NB_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x$

- $sv \geq 6$: $(1, 2)$ $(2, 4)$

② $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

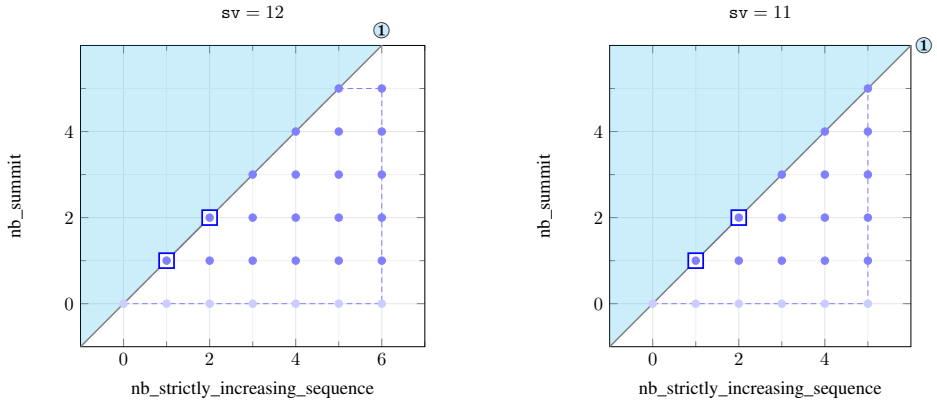
③ $y \neq 1$



NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① y ≤ x

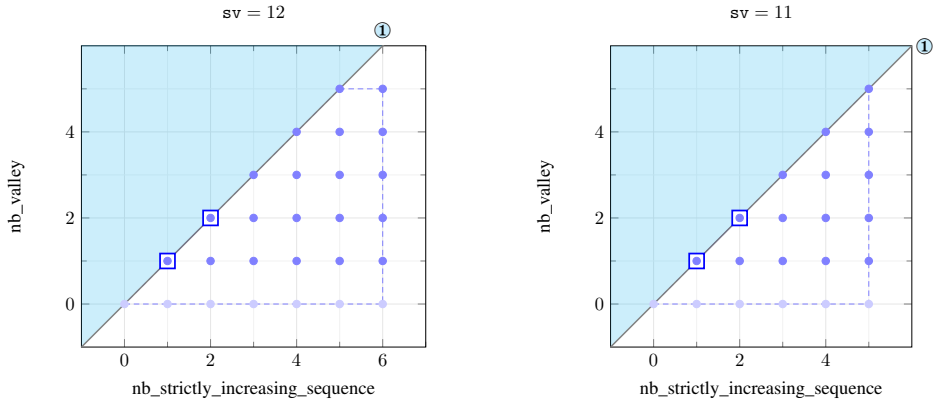
□ sv ≥ 5: (1, 1) (2, 2)



NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① y ≤ x

□ sv ≥ 5: (1, 1) (2, 2)

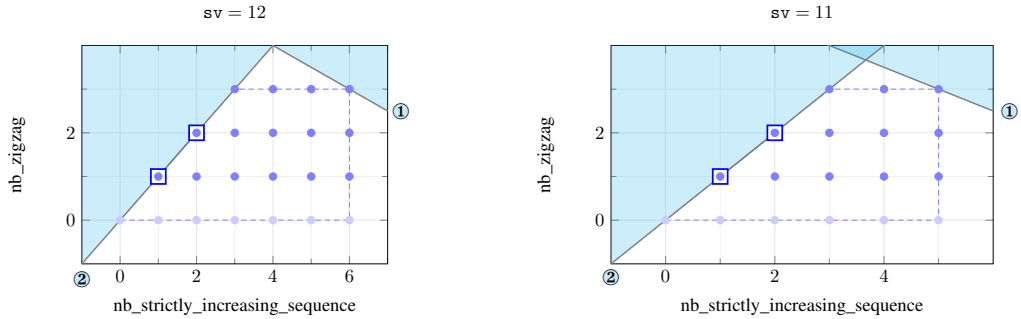


NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq sv$

② $y \leq x$

- $sv \geq 7$: $(1, 1)$ $(2, 2)$



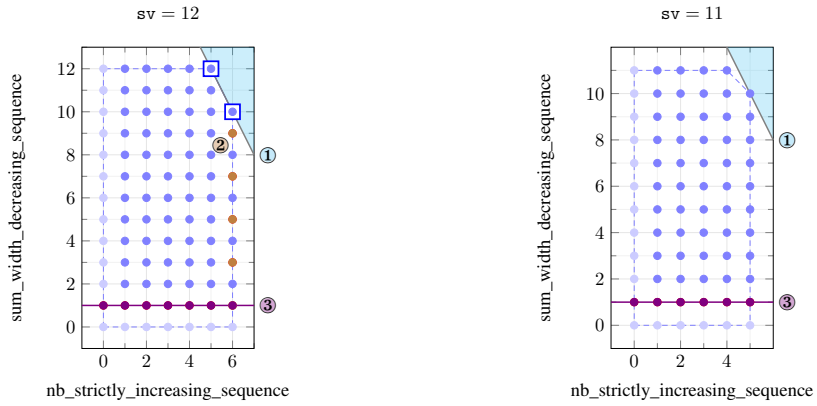
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

- $sv \bmod 2 = 0 \wedge sv \geq 2$: $(\lfloor sv/2 \rfloor, sv - 2)$ $(\lfloor sv/2 \rfloor - 1, sv)$

② $\bigvee \begin{pmatrix} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{pmatrix}$

③ $y \neq 1$



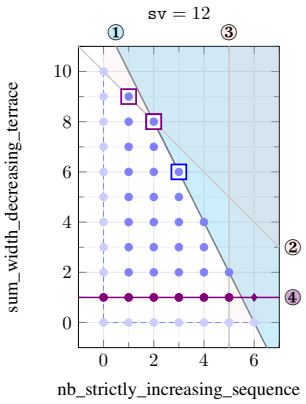
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

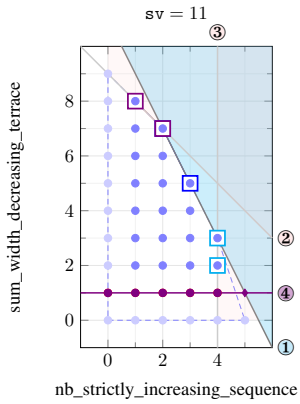
① $2 * x + y \leq sv$
- $sv \geq 8$: $(2, sv - 4)$ $(3, sv - 6)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(1, sv - 3)$ $(2, sv - 4)$

③ $y > 0 \Rightarrow 2 * x \leq sv - 2 - (sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 7$: $(\lfloor sv/2 \rfloor - 1, 2)$ $(\lfloor sv/2 \rfloor - 1, 3)$

④ $y \neq 1$

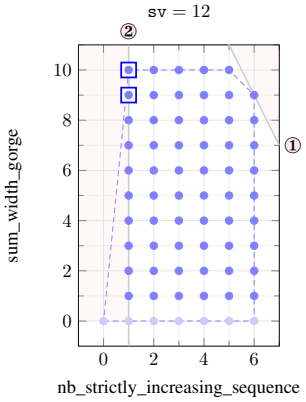


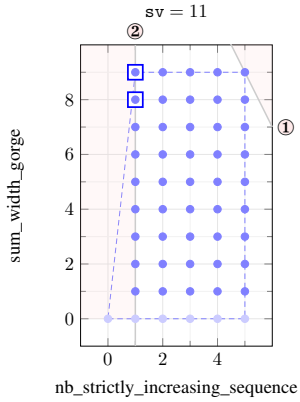


NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



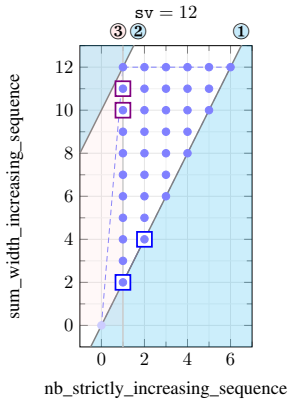


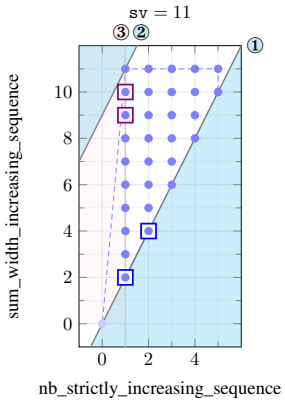
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 1) (1, sv − 2)





NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x = 0 \Rightarrow y = 0$

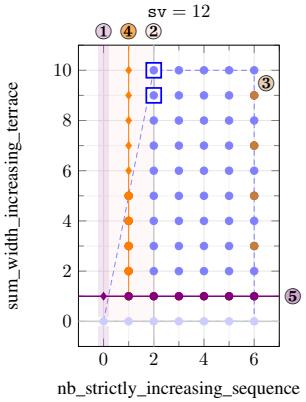
② $x > 0 \wedge y > 0 \Rightarrow x \geq 2$

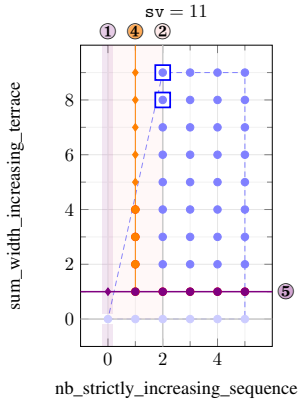
□ $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

③ $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

④ $x \neq 1 \vee y < 1$

⑤ $y \neq 1$





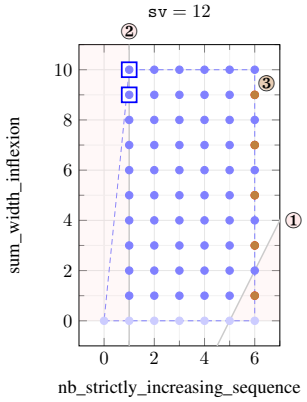
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

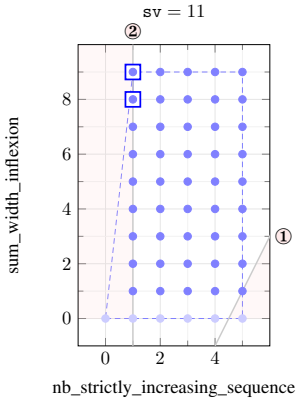
① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$

② $y > 0 \Rightarrow x \geq 1$

□ $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

③ $\bigvee \begin{pmatrix} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{pmatrix}$





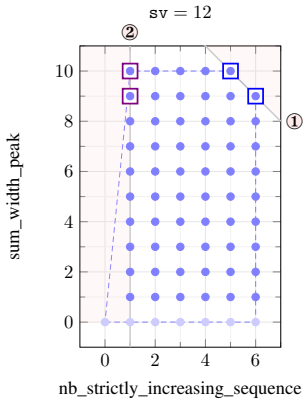
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

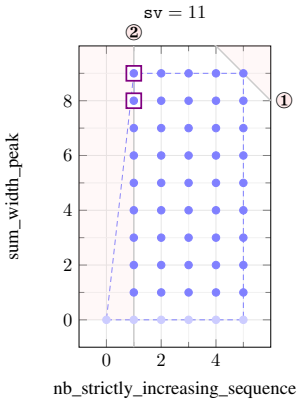
① $x > 0 \wedge y > 0 \Rightarrow 2 * x + 2 * y \leq 3 * sv - 5 - (3 * sv - 5) \bmod 2$

□ $(sv - 1) \bmod 2 = 1 \wedge sv \geq 4$: $(\lfloor sv/2 \rfloor, sv - 3)$ $(\lfloor sv/2 \rfloor - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 1$

□ $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$

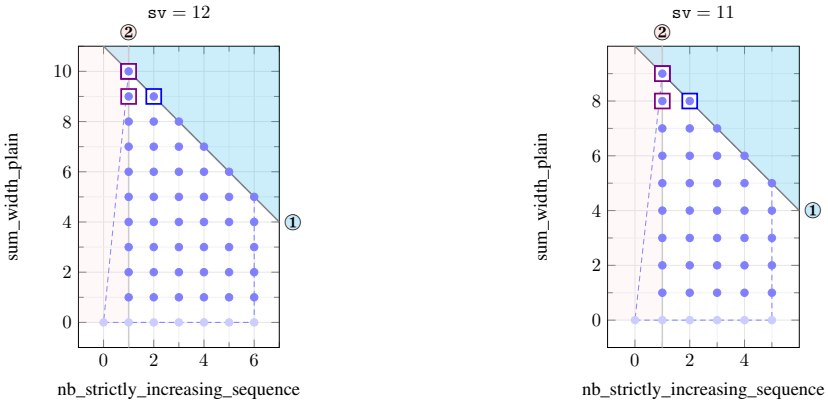




NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

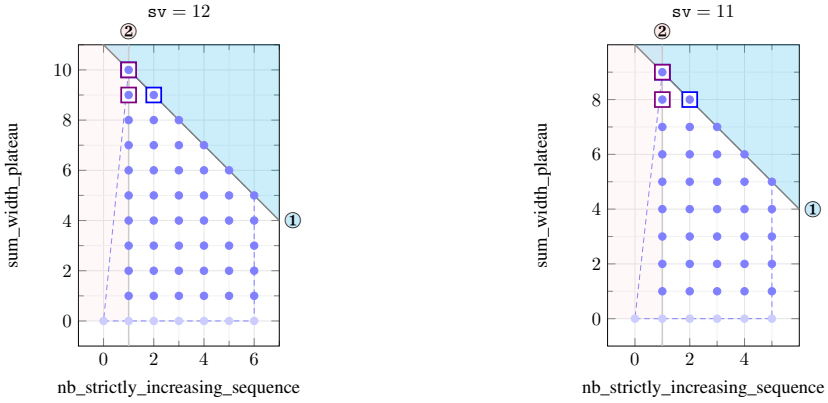
② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 4$: $(1, sv - 2)$ $(2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



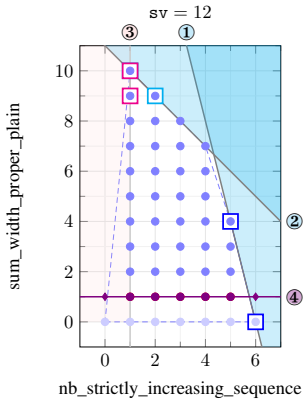
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

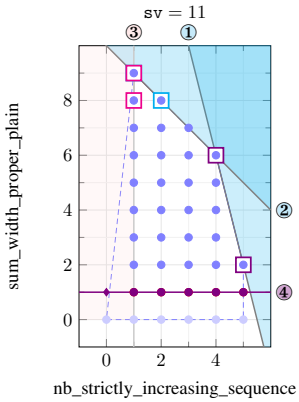
① $4 * x + y \leq 2 * sv$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





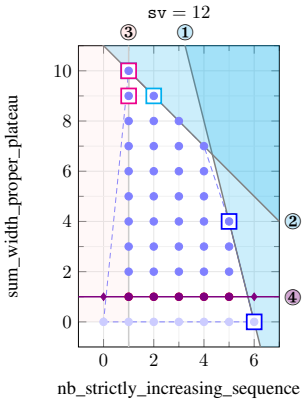
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

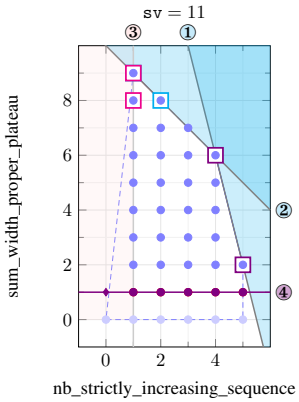
① $4 * x + y \leq 2 * sv$
- sv mod $2 = 0 \wedge sv \geq 8$: $(\lfloor sv/2 \rfloor, 0)$ $(\lfloor sv/2 \rfloor - 1, 4)$
- sv mod $2 = 1 \wedge sv \geq 11$: $(\lfloor sv/2 \rfloor, 2)$ $(\lfloor sv/2 \rfloor - 1, 6)$

② $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

④ $y \neq 1$





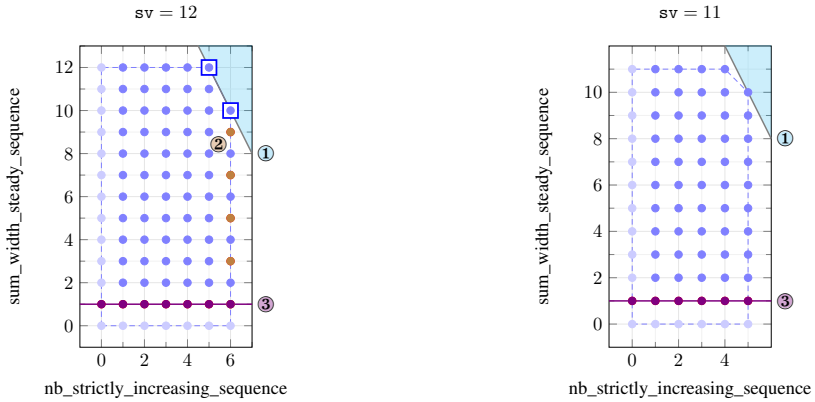
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

③ $\mathtt{y} \neq 1$



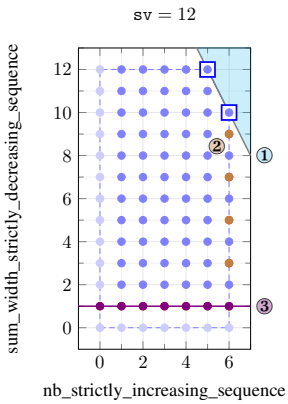
NB_STRICTLY_INCREASING_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

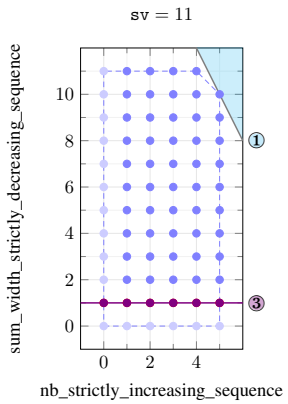
① $2 * \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\lfloor \mathtt{sv}/2 \rfloor, \mathtt{sv} - 2)$ $(\lfloor \mathtt{sv}/2 \rfloor - 1, \mathtt{sv})$

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \lfloor \mathtt{sv}/2 \rfloor, \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 3, \\ 1 = \mathtt{sv} \bmod 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

③ $\mathtt{y} \neq 1$



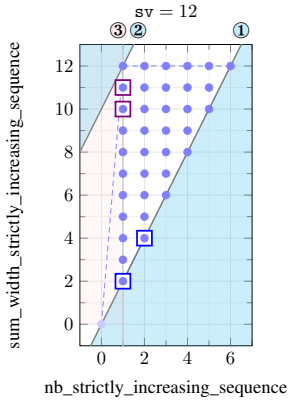


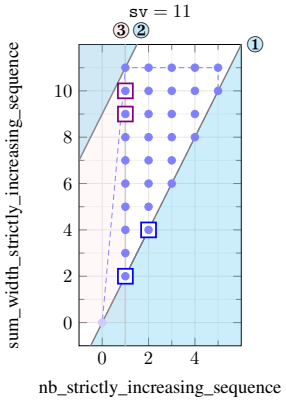
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (1, 2) (2, 4)

② $sv > 1 \Rightarrow y \leq 2 * x + sv - 2$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 4: (1, sv − 1) (1, sv − 2)

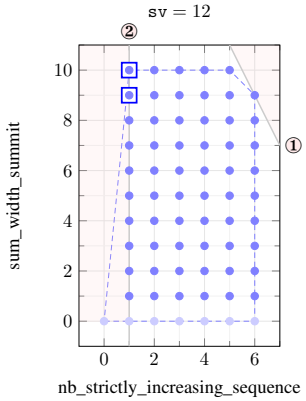


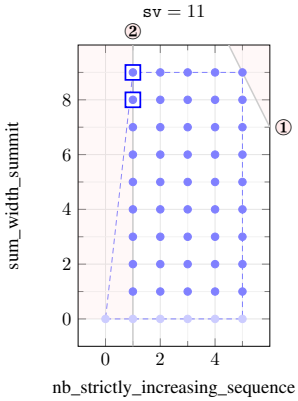


NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$
- sv ≥ 3: (1, sv − 2) (1, sv − 3)



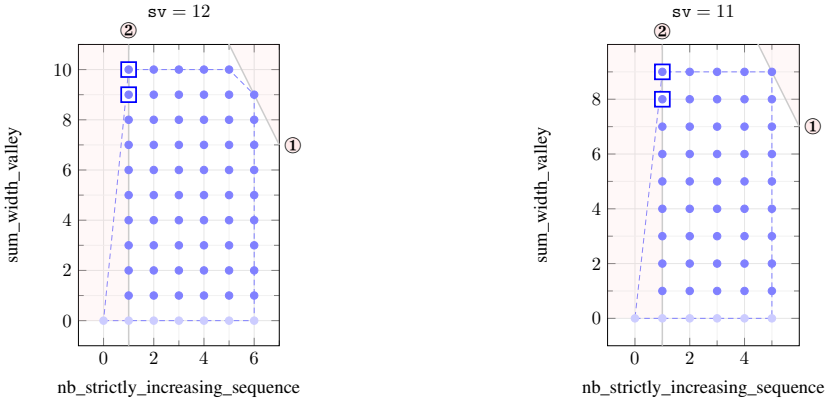


NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 3$

② $y > 0 \Rightarrow x \geq 1$

- $sv \geq 3$: $(1, sv - 2)$ $(1, sv - 3)$



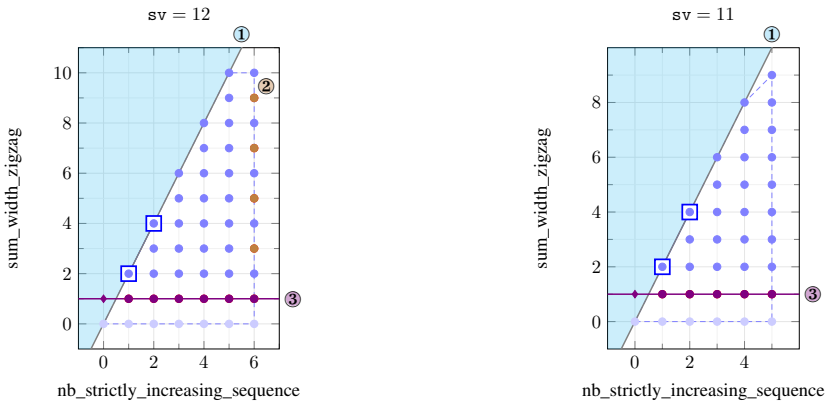
NB_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x$

- $sv \geq 6$: $(1, 2)$ $(2, 4)$

② $\bigvee \left(\begin{array}{l} x \neq \lfloor sv/2 \rfloor, \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 1 = sv \bmod 2, \\ 0 = y \bmod 2 \end{array} \right)$

③ $y \neq 1$

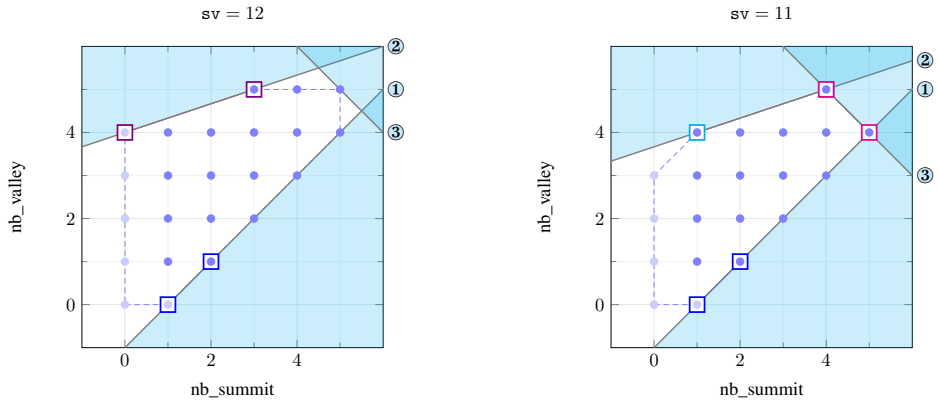


NB_SUMMIT(x, VARIABLES) ∧
NB_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- sv ≥ 5: (1, 0) (2, 1)

② $3 * y \leq x + sv$
- sv mod 2 = 0 ∧ sv ≥ 12: (⌊(sv − 1)/2⌋ − 2, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 5, ⌊(sv − 1)/2⌋ − 1)
- sv mod 2 = 1 ∧ sv ≥ 9: (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋) (⌊(sv − 1)/2⌋ − 4, ⌊(sv − 1)/2⌋ − 1)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv mod 2 = 1 ∧ sv ≥ 3: (⌊(sv − 1)/2⌋, ⌊(sv − 1)/2⌋ − 1) (⌊(sv − 1)/2⌋ − 1, ⌊(sv − 1)/2⌋)

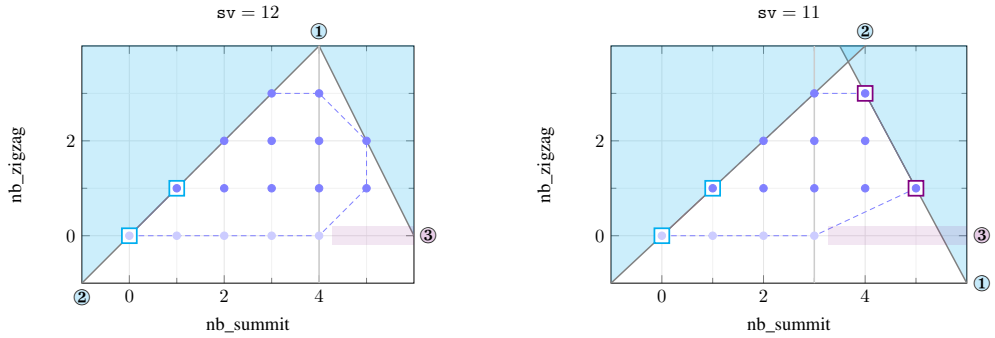


NB_SUMMIT(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, 2) (⌊(sv − 1)/2⌋ − 1, 4)
- sv mod 2 = 1 ∧ sv ≥ 11: (⌊(sv − 1)/2⌋, 1) (⌊(sv − 1)/2⌋ − 1, 3)

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$



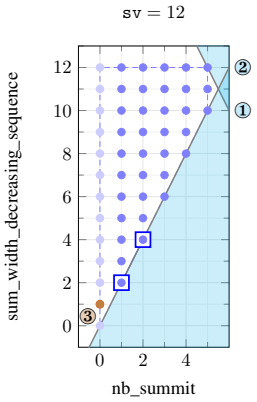
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

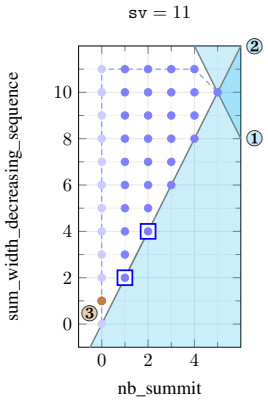
① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$





NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

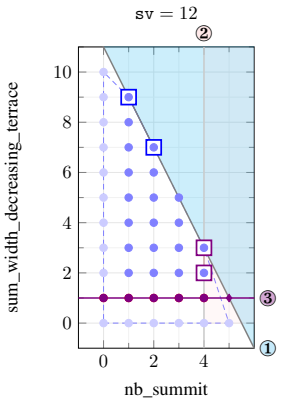
① $2 * x + y \leq sv - 1$

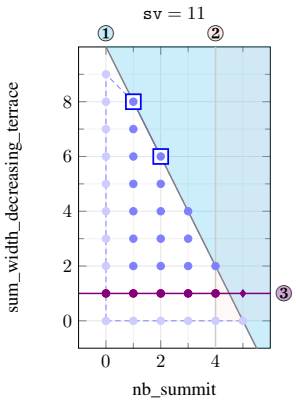
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$



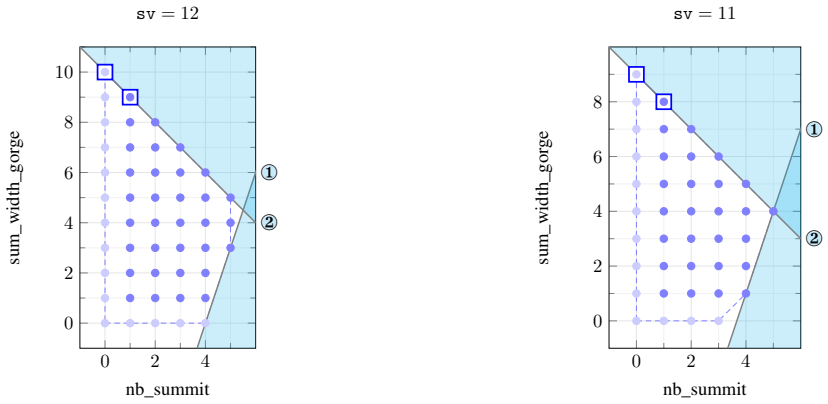


NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



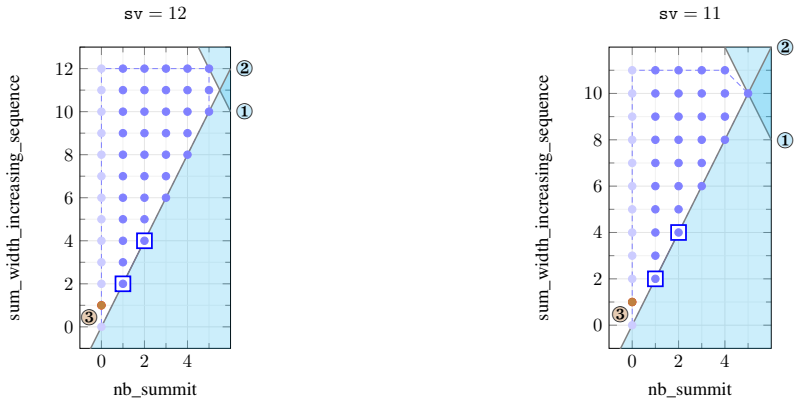
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$

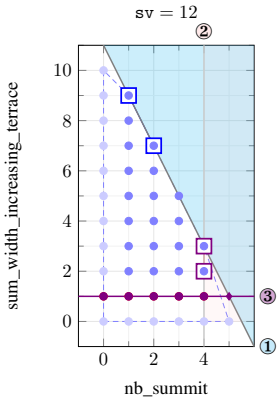


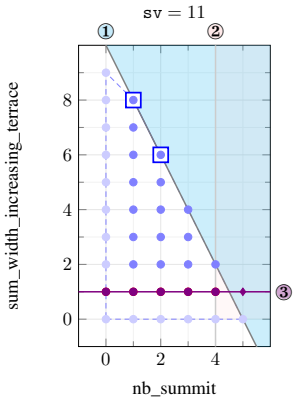
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

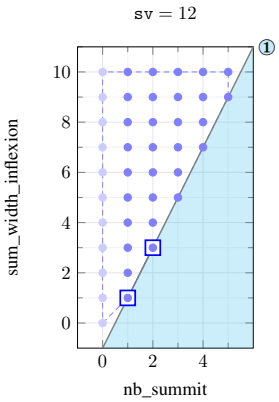
③ $y \neq 1$

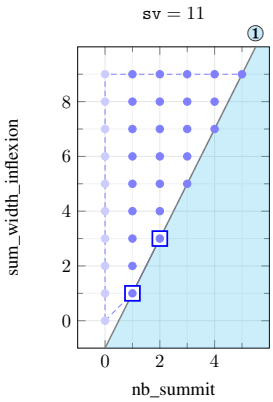




NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$
- $sv \geq 5$: $(1, 1)$ $(2, 3)$



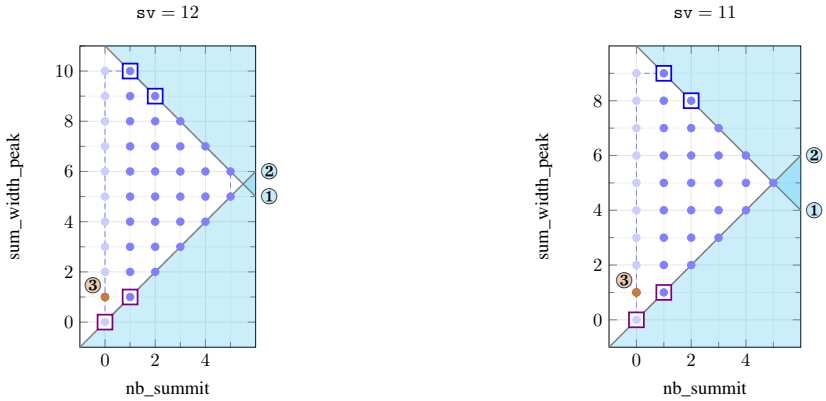


NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

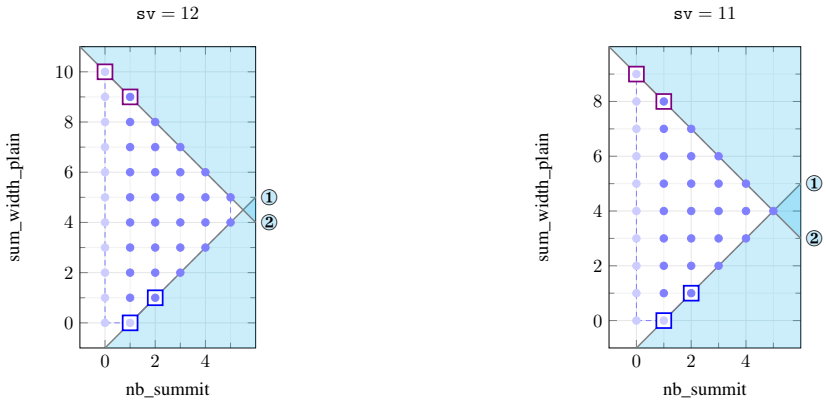
③ $x \neq 0 \vee y \neq 1$



NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



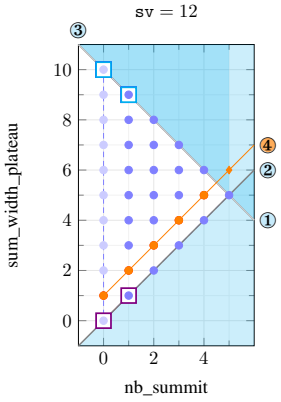
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

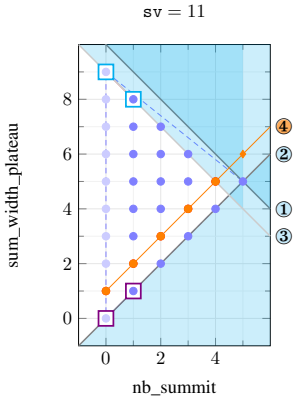
① $x + y \leq sv - 1 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(0, sv - 2)$ $(1, sv - 3)$

② $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $x < \lfloor (sv - 1)/2 \rfloor \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 3)$

④ $y \neq x + 1$



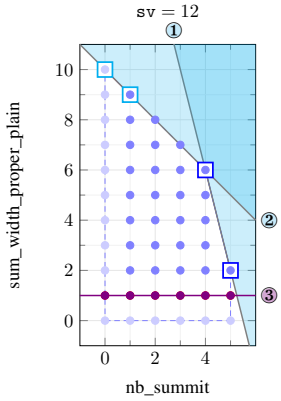


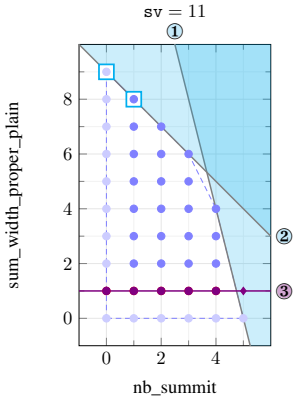
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$



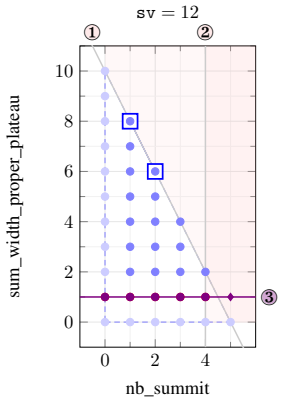


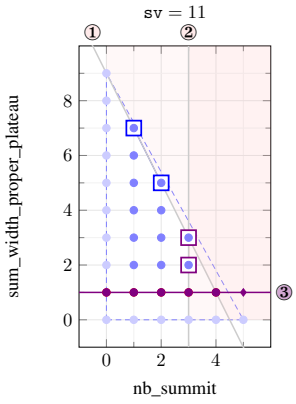
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq sv - 2$
- sv ≥ 8: $(1, sv-4)$ $(2, sv-6)$

② $y > 0 \Rightarrow 2 * x \leq sv - 4 - (sv - 4) \bmod 2$
- sv mod 2 = 1 ∧ sv ≥ 5: $(\lfloor (sv-1)/2 \rfloor - 2, 2)$ $(\lfloor (sv-1)/2 \rfloor - 2, 3)$

③ $y \neq 1$



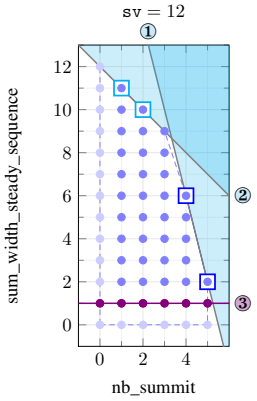


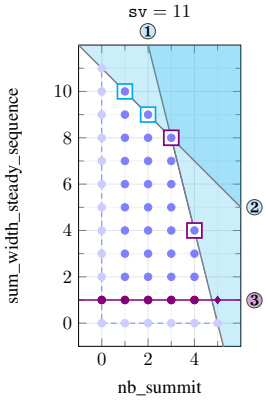
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 8$: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- $sv \bmod 2 = 1 \wedge sv \geq 11$: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $x + y \leq sv$
- $sv \geq 8$: $(1, sv-1)$ $(2, sv-2)$

③ $y \neq 1$



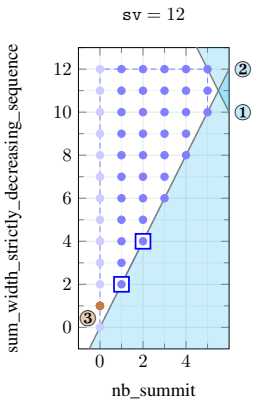


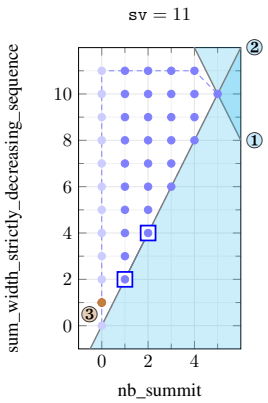
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



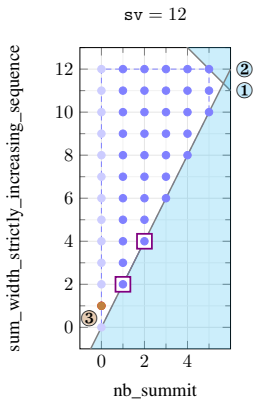


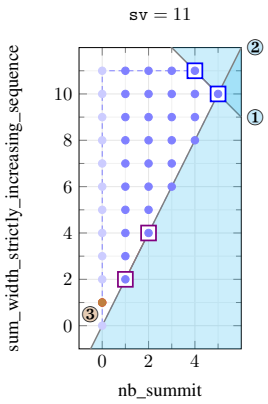
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- $sv \bmod 2 = 1 \wedge sv \geq 3$: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$



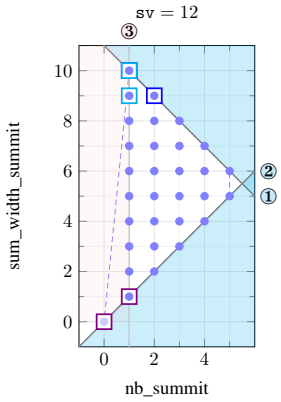


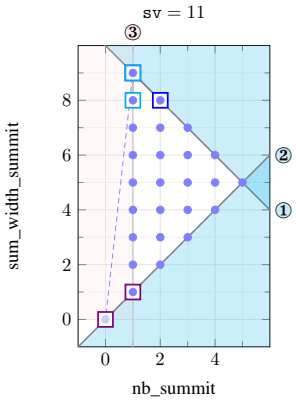
NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv-2)$ $(2, sv-3)$

② $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv-2)$ $(1, sv-3)$

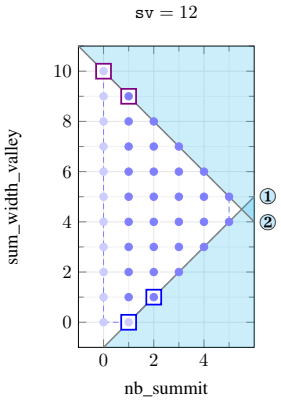


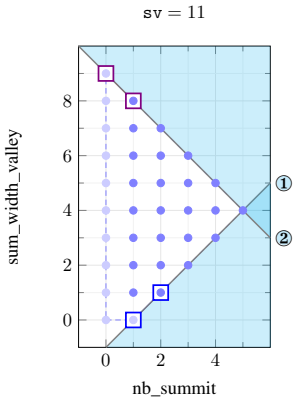


NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 5$: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: (0, sv − 2) (1, sv − 3)





NB_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 3 * x$
- sv ≥ 5: (0, 0) (1, 3)

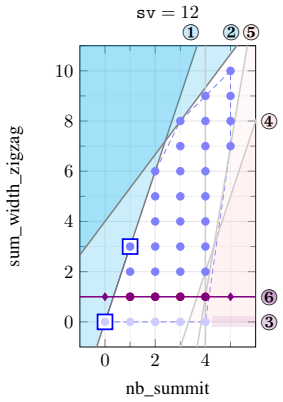
② $3 * y \leq 4 * x + sv$

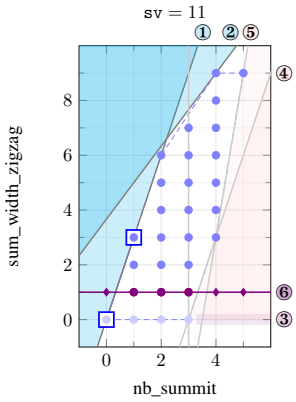
③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

⑤ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, sv − 5) (⌊(sv − 1)/2⌋ − 1, sv − 11)
- sv mod 2 = 1 ∧ sv ≥ 17: (⌊(sv − 1)/2⌋ − 1, sv − 8) (⌊(sv − 1)/2⌋ − 2, sv − 14)

⑥ $y \neq 1$



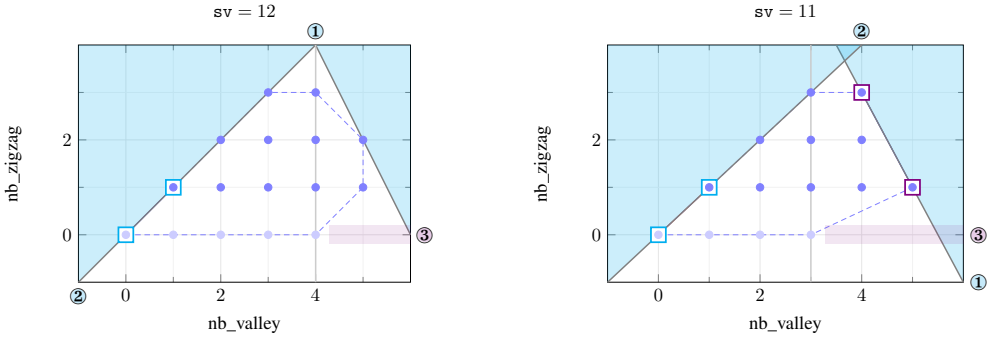


NB_VALLEY(x, VARIABLES) ∧
NB_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- sv mod 2 = 0 ∧ sv ≥ 14: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 4)$
- sv mod 2 = 1 ∧ sv ≥ 11: $(\lfloor (sv-1)/2 \rfloor, 1)$ $(\lfloor (sv-1)/2 \rfloor - 1, 3)$

② $y \leq x$
- sv ≥ 4: (0, 0) (1, 1)

③ $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

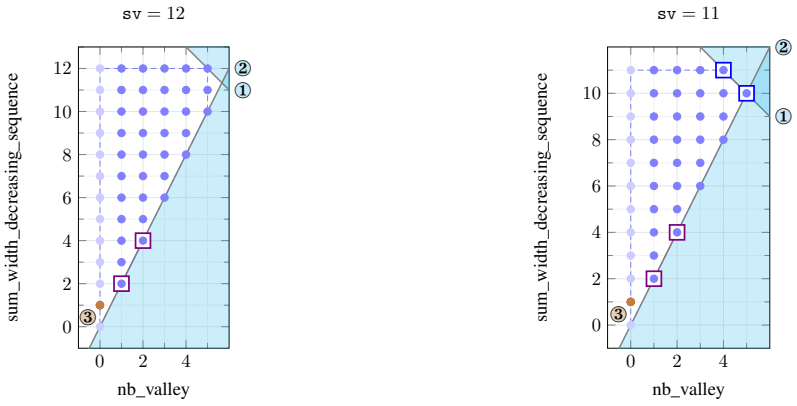


NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + 2 * y \leq 3 * sv - 2 - (3 * sv - 2) \bmod 2$
- sv mod 2 = 1 ∧ sv ≥ 3: $(\lfloor (sv-1)/2 \rfloor, sv-1)$ $(\lfloor (sv-1)/2 \rfloor - 1, sv)$

② $2 * x \leq y$
- sv ≥ 5: (1, 2) (2, 4)

③ $x \neq 0 \vee y \neq 1$

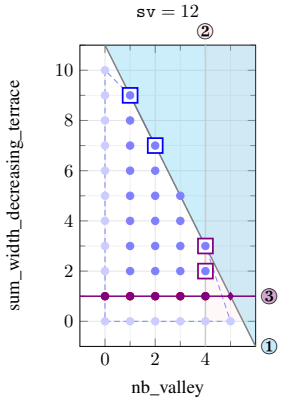


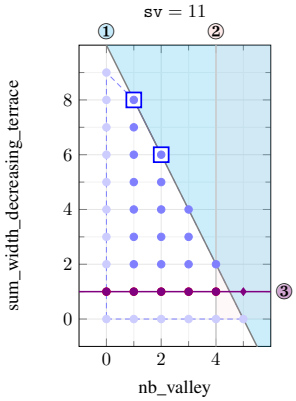
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv - 1$
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$



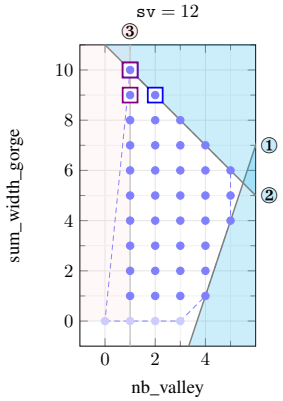


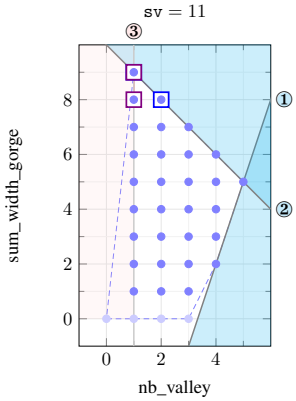
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv - 1$

② $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$





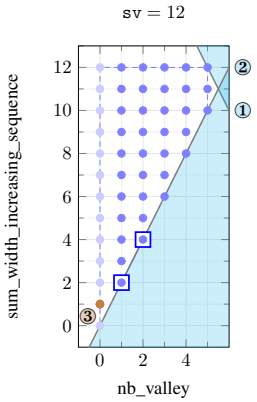
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

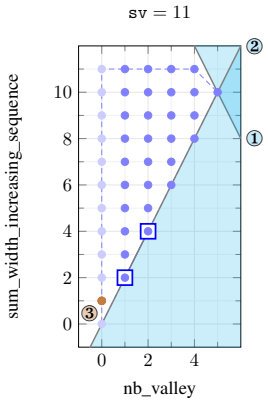
① $2 * x + y \leq 2 * sv - 2$

② $2 * x \leq y$

- $sv \geq 5$: $(1, 2)$ $(2, 4)$

③ $x \neq 0 \vee y \neq 1$





NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

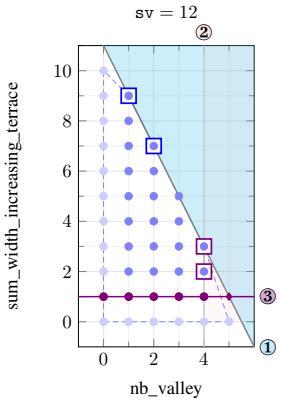
① $2 * x + y \leq sv - 1$

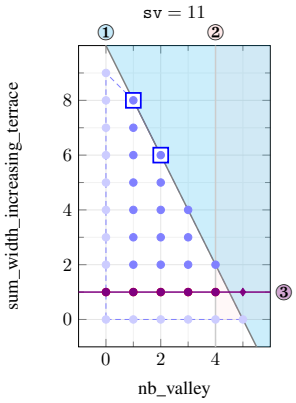
- $sv \geq 7$: $(1, sv - 3)$ $(2, sv - 5)$

② $y > 0 \Rightarrow 2 * x \leq sv - 3 - (sv - 3) \bmod 2$

- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/2 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/2 \rfloor - 1, 3)$

③ $y \neq 1$

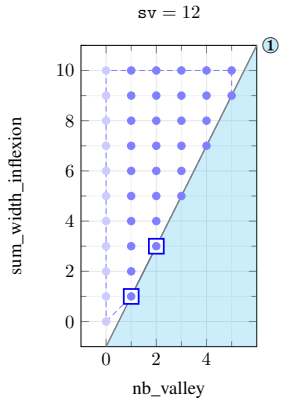


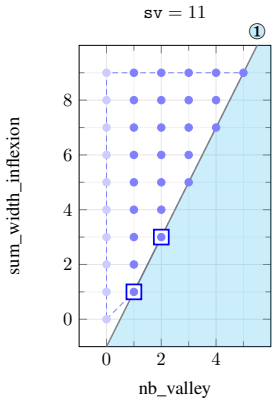


NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y + 1$

- sv ≥ 5: (1, 1) (2, 3)





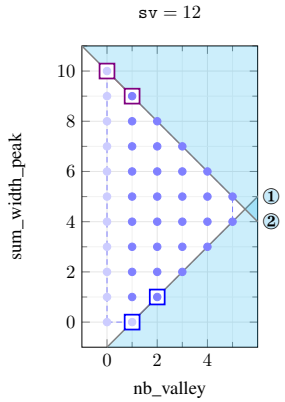
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

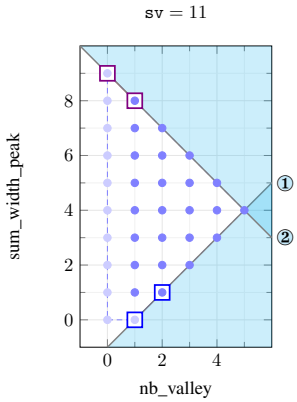
① $x \leq y + 1$

- sv ≥ 5: (1, 0) (2, 1)

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- sv ≥ 3: (0, sv − 2) (1, sv − 3)



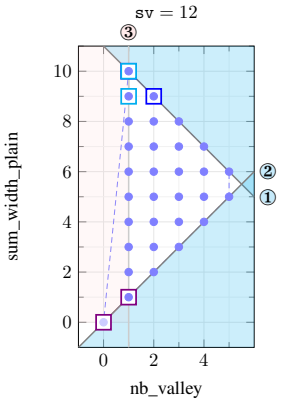


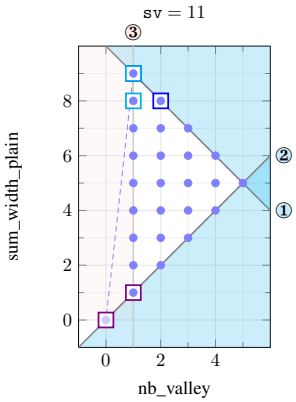
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$
- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$

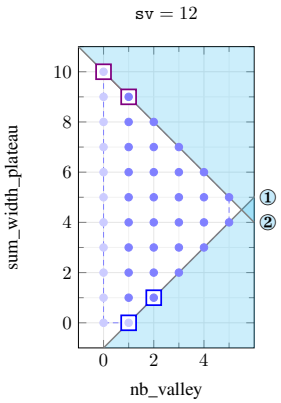


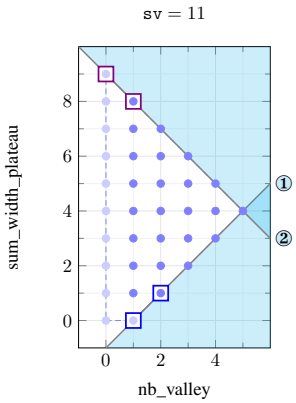


NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y + 1$
- $sv \geq 5$: $(1, 0)$ $(2, 1)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$





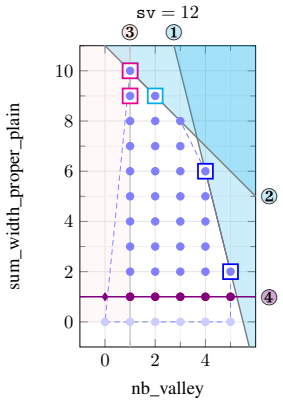
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

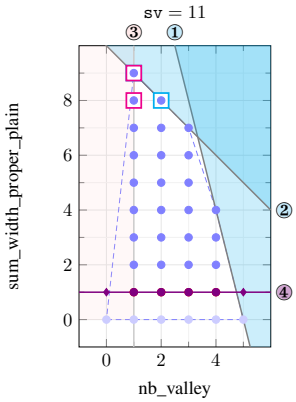
① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 10: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 13: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $x + y \leq sv - 1$
- sv ≥ 7: $(1, sv-2)$ $(2, sv-3)$

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: $(1, sv-2)$ $(1, sv-3)$

④ $y \neq 1$



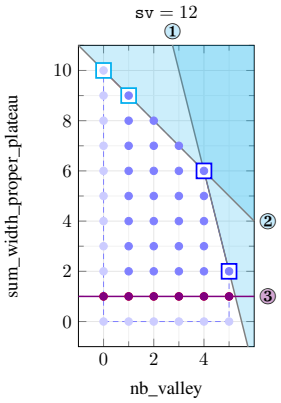


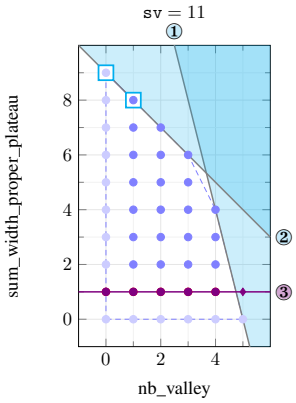
NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 12: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 15: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

② $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 5: $(0, sv-2)$ $(1, sv-3)$

③ $y \neq 1$

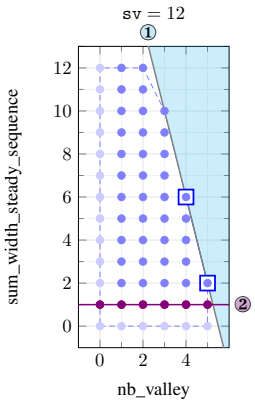


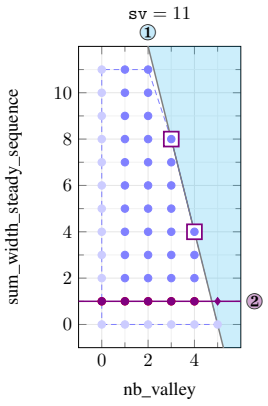


NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $4 * x + y \leq 2 * sv - 2$
- sv mod 2 = 0 ∧ sv ≥ 6: $(\lfloor (sv-1)/2 \rfloor, 2)$ $(\lfloor (sv-1)/2 \rfloor - 1, 6)$
- sv mod 2 = 1 ∧ sv ≥ 9: $(\lfloor (sv-1)/2 \rfloor - 1, 4)$ $(\lfloor (sv-1)/2 \rfloor - 2, 8)$

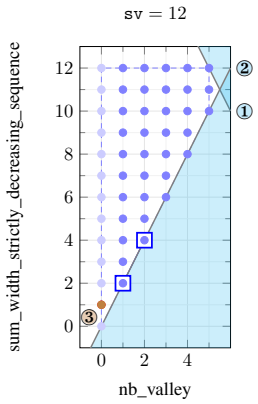
② $y \neq 1$

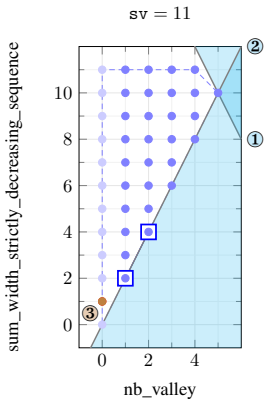




NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

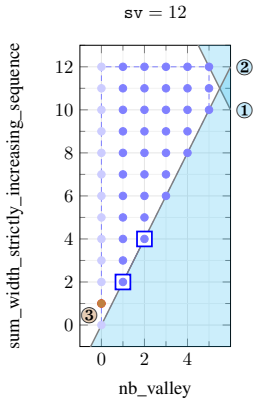
① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$

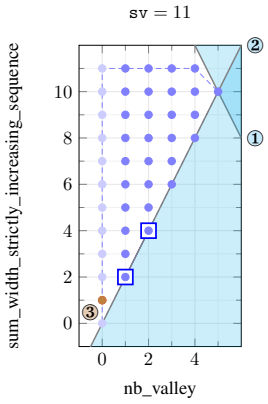




NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
② $2 * x \leq y$
□ $sv \geq 5$: (1, 2) (2, 4)
③ $x \neq 0 \vee y \neq 1$



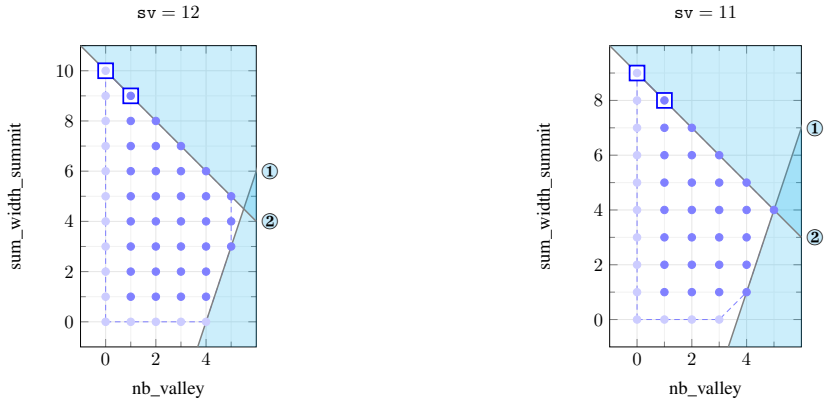


NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $3 * x \leq y + sv$

② $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(0, sv - 2)$ $(1, sv - 3)$



NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$

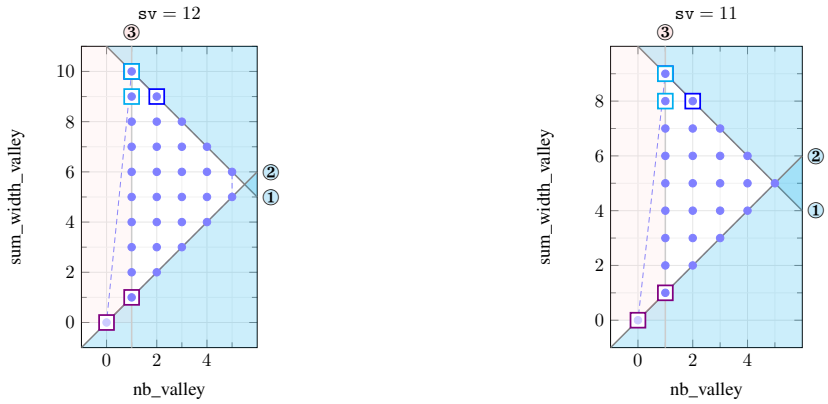
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $x \leq y$

- $sv \geq 3$: $(0, 0)$ $(1, 1)$

③ $y > 0 \Rightarrow x \geq 1$

- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$



NB_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- sv ≥ 7: (1, 3) (2, 5)

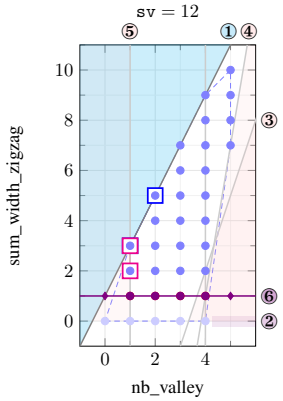
② $y = 0 \Rightarrow 3 * x \leq sv - sv \bmod 3$

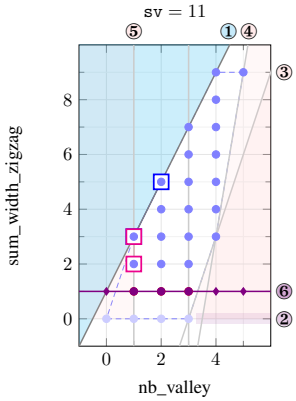
③ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq y + sv - 2$

④ $y > 0 \Rightarrow 6 * x \leq y + 2 * sv - 1$
- sv mod 2 = 0 ∧ sv ≥ 14: (⌊(sv − 1)/2⌋, sv − 5) (⌊(sv − 1)/2⌋ − 1, sv − 11)
- sv mod 2 = 1 ∧ sv ≥ 17: (⌊(sv − 1)/2⌋ − 1, sv − 8) (⌊(sv − 1)/2⌋ − 2, sv − 14)

⑤ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, 2) (1, 3)

⑥ $y \neq 1$





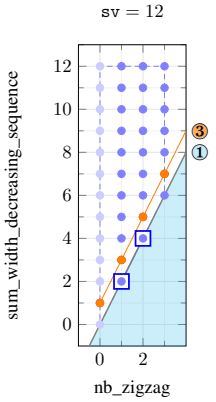
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

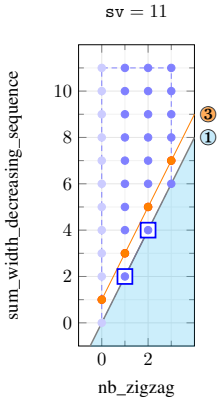
① $2 * \mathtt{x} \leq \mathtt{y}$

- $\mathtt{sv} \geq 7$: (1, 2) (2, 4)

② $$\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor (\mathtt{sv}-1)/3 \rfloor) - 0, \\ \mathtt{y} < 2 * \mathtt{max}(0, \lfloor (\mathtt{sv}-1)/3 \rfloor) + 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ 1 \neq \mathtt{sv} \bmod 3 \end{array} \right)$$

③ $\mathtt{y} \neq 2 * \mathtt{x} + 1$





NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x = \lfloor (sv - 1)/3 \rfloor - 1 \wedge sv > 3 \Rightarrow y \leq 6 + 2 * (sv - 1) \bmod 3$

② $x = \lfloor (sv - 1)/3 \rfloor \wedge sv > 3 \Rightarrow y \leq 2 * (sv - 1) \bmod 3$

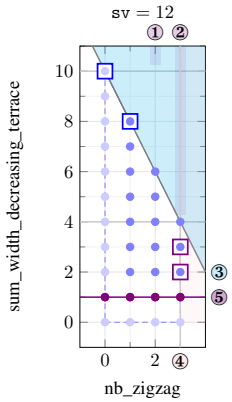
③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

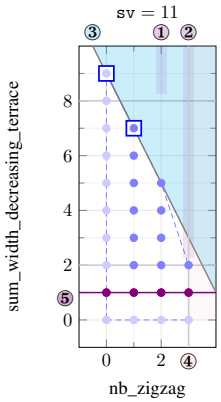
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑤ $y \neq 1$





NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

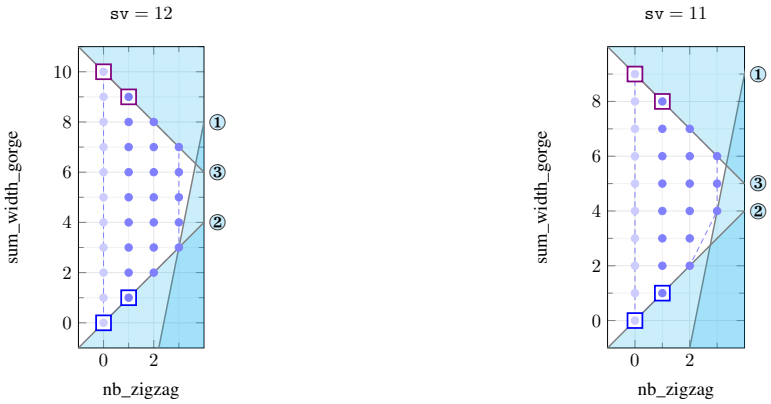
① $5 * x \leq y + sv$

② $x \leq y$

- $sv \geq 4$: $(0, 0)$ $(1, 1)$

③ $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 3)$



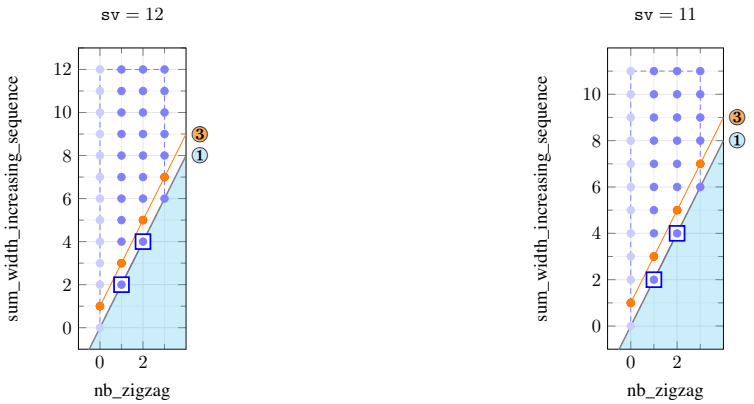
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$

- $sv \geq 7$: $(1, 2)$ $(2, 4)$

② $\bigvee \begin{pmatrix} x \neq \max(0, \lfloor (sv - 1)/3 \rfloor) - 0, \\ y < 2 * \max(0, \lfloor (sv - 1)/3 \rfloor) + 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 \neq sv \bmod 3 \end{pmatrix}$

③ $y \neq 2 * x + 1$



NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x = \lfloor (sv - 1)/3 \rfloor - 1 \wedge sv > 3 \Rightarrow y \leq 6 + 2 * (sv - 1) \bmod 3$

② $x = \lfloor (sv - 1)/3 \rfloor \wedge sv > 3 \Rightarrow y \leq 2 * (sv - 1) \bmod 3$

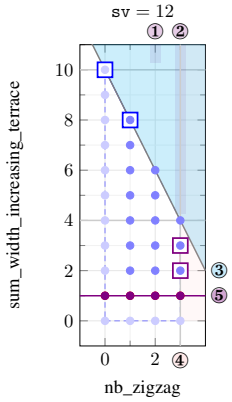
③ $sv > 1 \Rightarrow 2 * x + y \leq sv - 2$

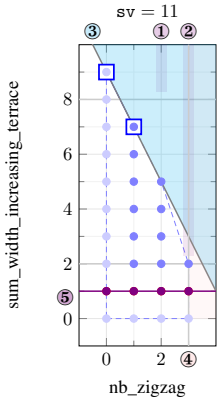
- $sv \geq 6$: $(0, sv - 2)$ $(1, sv - 4)$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$

- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑤ $y \neq 1$





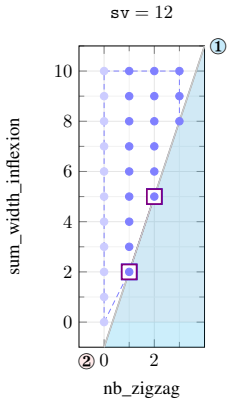
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

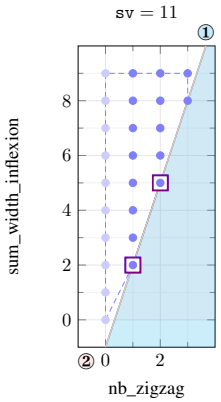
① $3 * x \leq y + 1$

- $sv \geq 7$: $(1, 2)$ $(2, 5)$

② $y > 0 \Rightarrow 3 * x \leq y + 1$

- $sv \geq 7$: $(1, 2)$ $(2, 5)$





NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $5 * x \leq y + sv$

② $2 * x \leq y + 2$

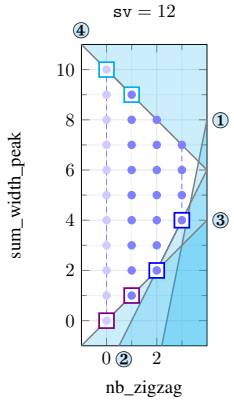
- $sv \geq 11$: (2, 2) (3, 4)

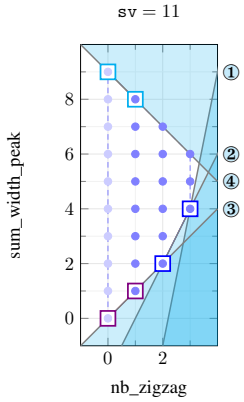
③ $x \leq y$

- $sv \geq 4$: (0, 0) (1, 1)

④ $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 4$: (0, sv − 2) (1, sv − 3)



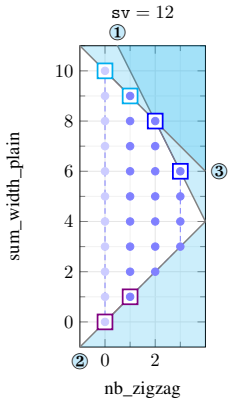


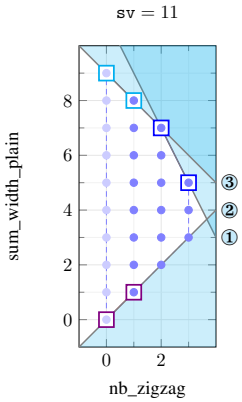
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \geq 11$: $(2, sv - 4)$ $(3, sv - 6)$

② $x \leq y$
- $sv \geq 4$: $(0, 0)$ $(1, 1)$

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 3)$



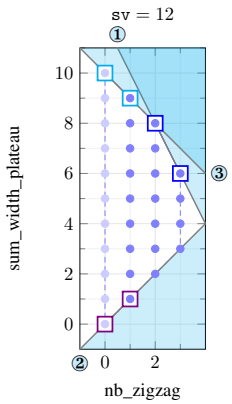


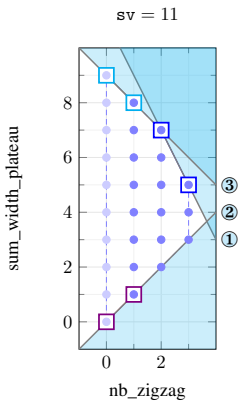
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq sv$
- $sv \geq 11$: $(2, sv - 4)$ $(3, sv - 6)$

② $x \leq y$
- $sv \geq 4$: $(0, 0)$ $(1, 1)$

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(0, sv - 2)$ $(1, sv - 3)$





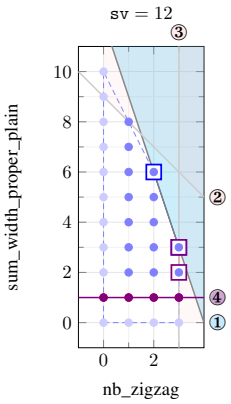
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

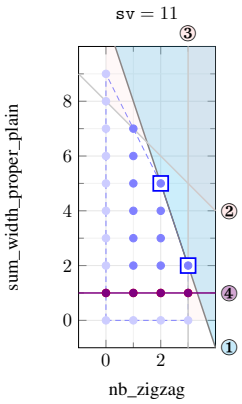
① $3 * x + y \leq sv$
- $sv \geq 11$: $(2, sv - 6)$ $(3, sv - 9)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$

③ $y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

④ $y \neq 1$





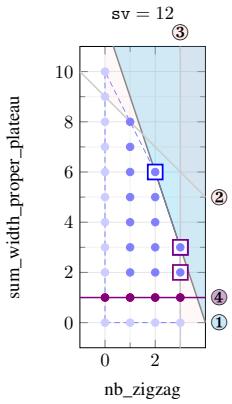
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

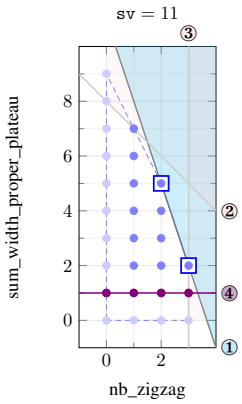
① $3 * x + y \leq sv$
- $sv \geq 11$: $(2, sv - 6)$ $(3, sv - 9)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq sv - 3$

③ $y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 9$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 7$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

④ $y \neq 1$





NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

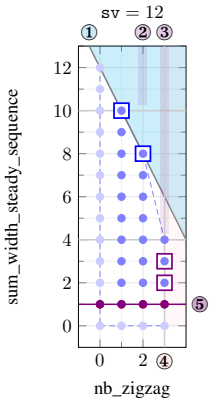
① $2 * x + y \leq sv$
- $sv \geq 10$: $(1, sv - 2)$ $(2, sv - 4)$

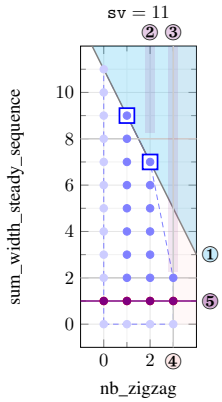
② $x = \lfloor (sv - 1)/3 \rfloor - 1 \wedge sv > 3 \Rightarrow y \leq 6 + 2 * (sv - 1) \bmod 3$

③ $x = \lfloor (sv - 1)/3 \rfloor \wedge sv > 3 \Rightarrow y \leq 2 * (sv - 1) \bmod 3$

④ $x > 0 \wedge y > 0 \Rightarrow 3 * x \leq sv - 2 - (sv - 2) \bmod 3$
- $(sv - 2) \bmod 3 = 1 \wedge sv \geq 6$: $(\lfloor (sv - 1)/3 \rfloor, 2)$ $(\lfloor (sv - 1)/3 \rfloor, 3)$
- $(sv - 2) \bmod 3 = 2 \wedge sv \geq 4$: $(\lfloor (sv - 1)/3 \rfloor - 1, 2)$ $(\lfloor (sv - 1)/3 \rfloor - 1, 3)$

⑤ $y \neq 1$





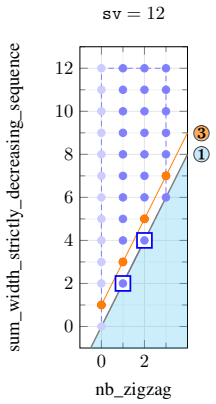
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

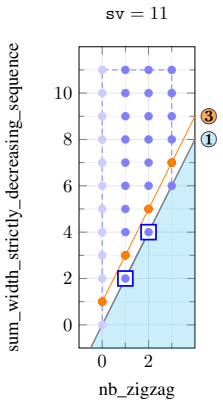
① $2 * \mathtt{x} \leq \mathtt{y}$

□ $\mathtt{sv} \geq 7$: (1, 2) (2, 4)

② $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{max}(0, \lfloor (\mathtt{sv} - 1)/3 \rfloor) - 0, \\ \mathtt{y} < 2 * \mathtt{max}(0, \lfloor (\mathtt{sv} - 1)/3 \rfloor) + 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 1, \\ 1 \neq \mathtt{sv} \bmod 3 \end{array} \right)$

③ $\mathtt{y} \neq 2 * \mathtt{x} + 1$



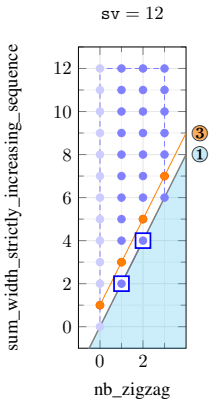


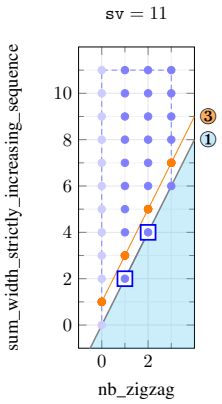
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 7: (1, 2) (2, 4)

② $\bigvee \begin{pmatrix} x \neq \max(0, \lfloor (sv-1)/3 \rfloor) - 0, \\ y < 2 * \max(0, \lfloor (sv-1)/3 \rfloor) + 1, \\ y > sv * \min(1, \max(0, sv-1)) - 1, \\ 1 \neq sv \bmod 3 \end{pmatrix}$

③ $y \neq 2 * x + 1$



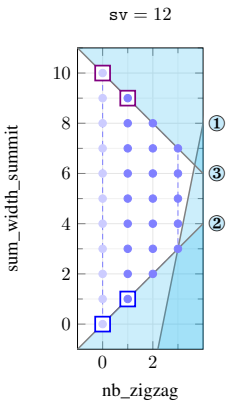


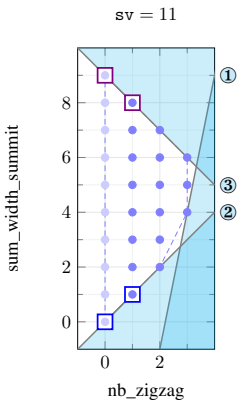
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $5 * x \leq y + sv$

② $x \leq y$
- sv ≥ 4: (0, 0) (1, 1)

③ $sv > 1 \Rightarrow x + y \leq sv - 2$
- sv ≥ 4: (0, sv − 2) (1, sv − 3)





NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $5 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv}$

② $2 * \mathtt{x} \leq \mathtt{y} + 2$

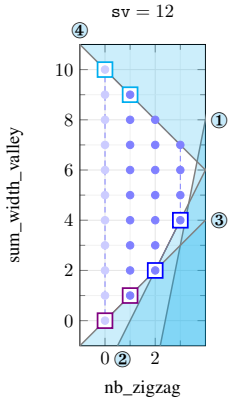
- $\mathtt{sv} \geq 11$: (2, 2) (3, 4)

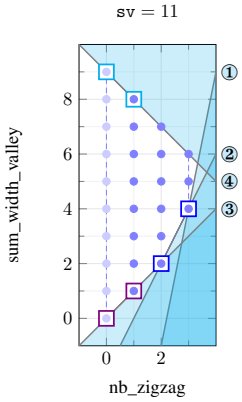
③ $\mathtt{x} \leq \mathtt{y}$

- $\mathtt{sv} \geq 4$: (0, 0) (1, 1)

④ $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$

- $\mathtt{sv} \geq 4$: $(0, \mathtt{sv} - 2)$ $(1, \mathtt{sv} - 3)$



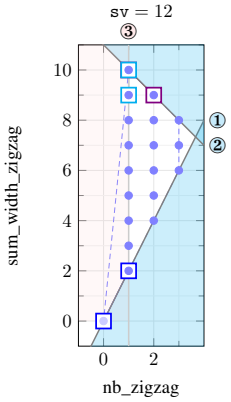


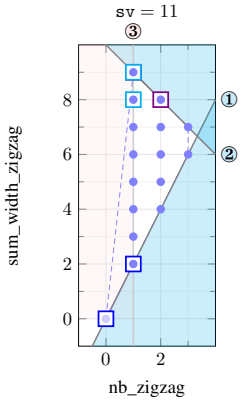
NB_ZIGZAG(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $2 * x \leq y$
- sv ≥ 4: (0, 0) (1, 2)

② $x + y \leq sv - 1$
- sv ≥ 7: (1, sv − 2) (2, sv − 3)

③ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, sv − 2) (1, sv − 3)





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_DECREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow y \leq x - 2$

□ $sv \geq 5$: $(sv, sv - 2)$ $(sv - 1, sv - 3)$

② $y > 0 \Rightarrow x \geq 4$

③ $x \neq 5 \vee y < 4$

④ $\bigvee \begin{pmatrix} x \neq sv * \min(1, \max(0, sv - 1)) - 4, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 3 \end{pmatrix}$

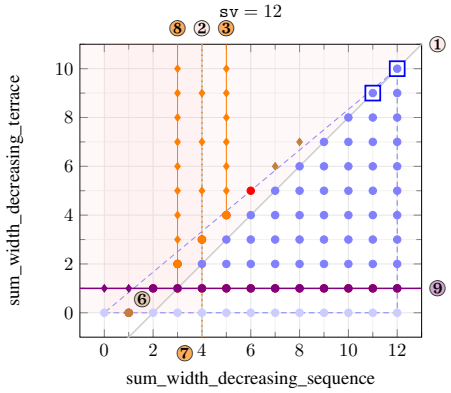
⑤ $\bigvee \begin{pmatrix} x \neq sv * \min(1, \max(0, sv - 1)) - 5, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 4 \end{pmatrix}$

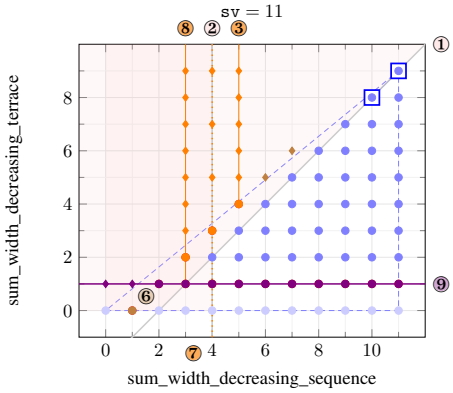
⑥ $x \neq 1 \vee y \neq 0$

⑦ $x \neq 4 \vee 0 = y \bmod 2$

⑧ $x \neq 3 \vee y < 1$

⑨ $y \neq 1$





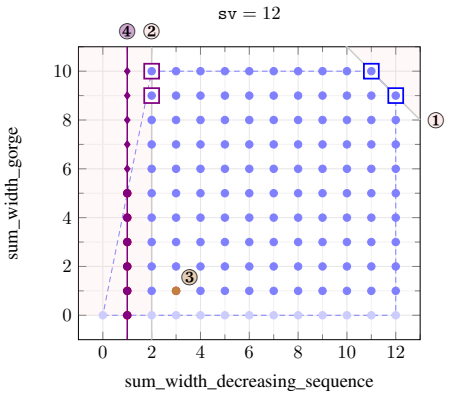
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_GORGE(y, VARIABLES) with sv = |VARIABLES|

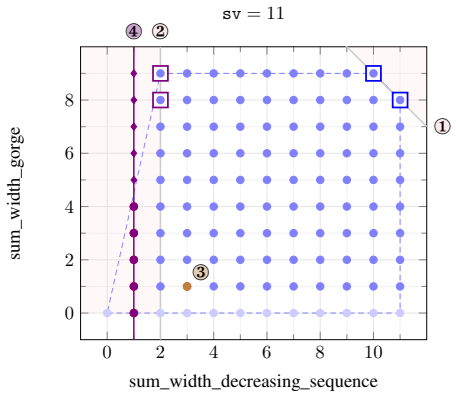
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq 2 * sv - 2$

□ $sv \bmod 2 = 0 \wedge sv \geq 2$: (sv, sv − 2) (sv − 2, sv)

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $x = sv - 2 \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 1$

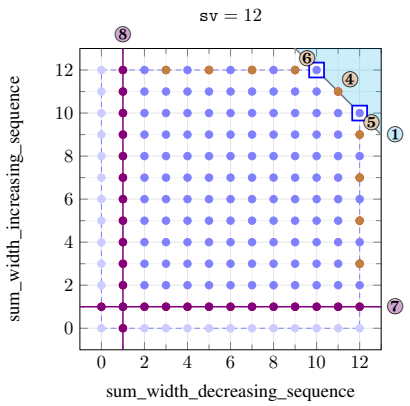
④ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2 \end{array} \right)$

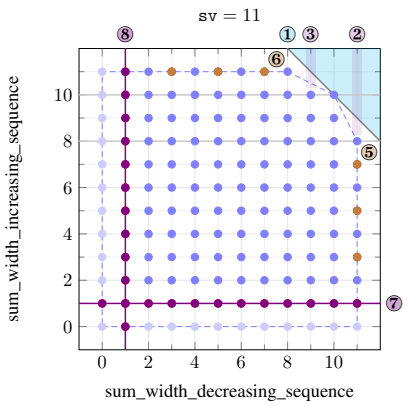
⑤ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)), \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{array} \right)$

⑥ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = x \bmod 2 \end{array} \right)$

⑦ $y \neq 1$

⑧ $x \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

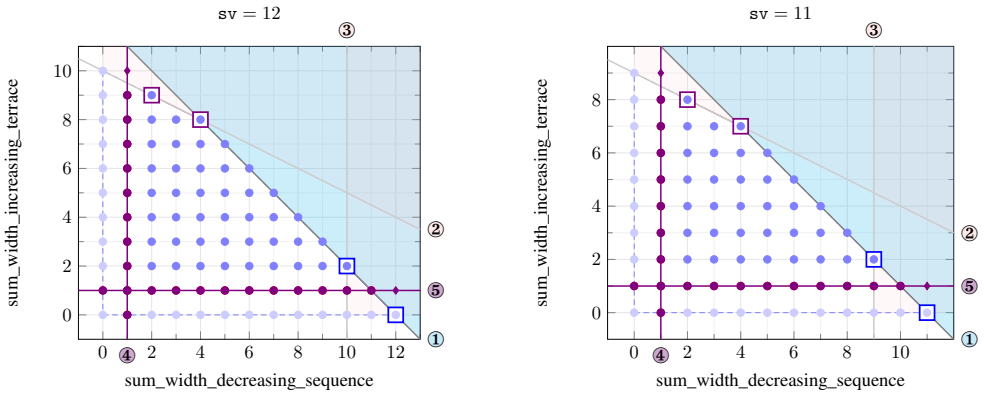
① $x + y \leq sv$
- $sv \geq 6$: $(sv, 0)$ $(sv - 2, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 2 * sv - 4$
- $sv \geq 6$: $(2, sv - 3)$ $(4, sv - 4)$

③ $y > 0 \Rightarrow x \leq sv - 2$

④ $x \neq 1$

⑤ $y \neq 1$



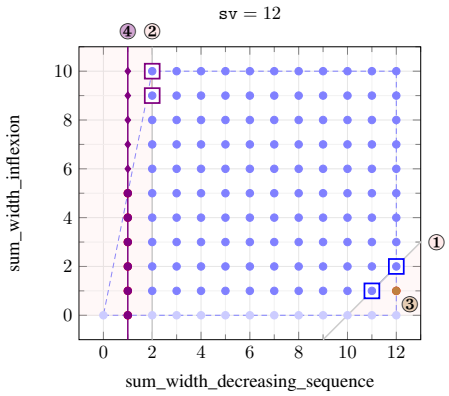
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

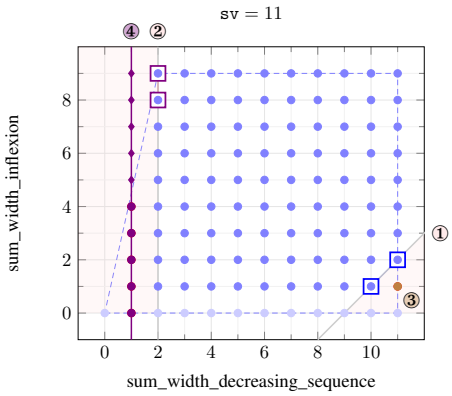
① $x > 0 \wedge y > 0 \Rightarrow x \leq y + sv - 2$
- $sv \geq 6$: $(sv, 2)$ $(sv - 1, 1)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq sv * \min(1, \max(0, sv - 1)) \vee y \neq 1$

④ $x \neq 1$





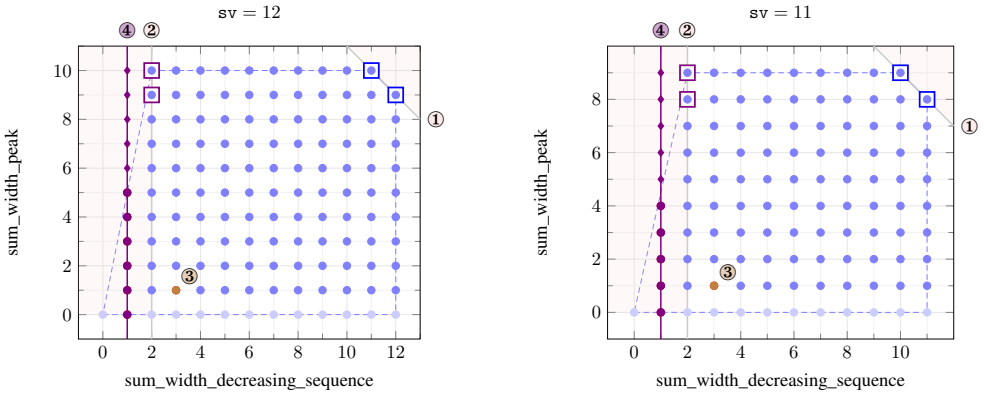
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 6$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$

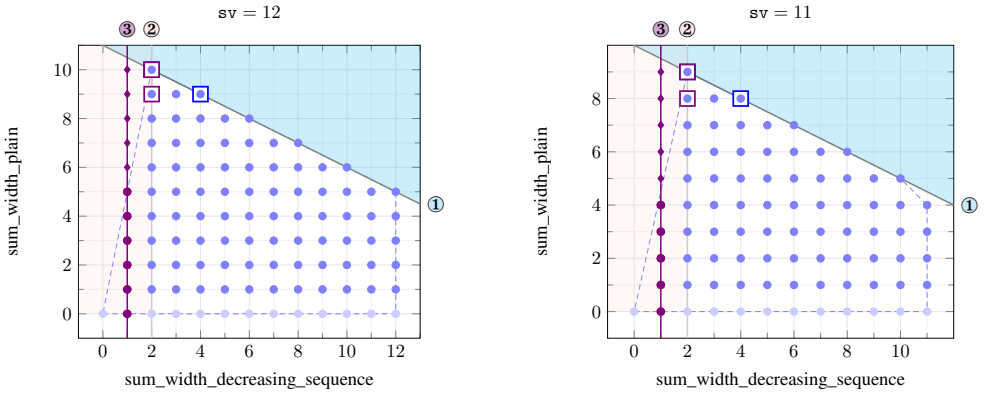


SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 4$: $(2, sv - 2)$ $(4, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 1$

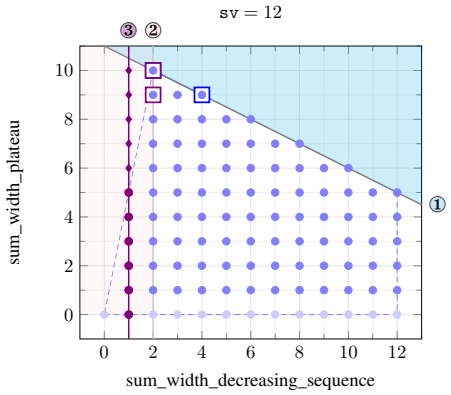


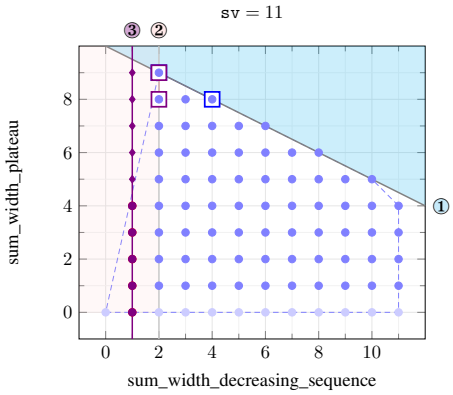
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 4$: $(2, sv - 2)$ $(4, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + 3 * y \leq 4 * sv - 2 - (4 * sv - 2) \bmod 3$

② $2 * x + y \leq 2 * sv$

- $sv \geq 5$: $(sv, 0)$ $(sv - 1, 2)$

③ $x + 2 * y \leq 2 * sv - 2$

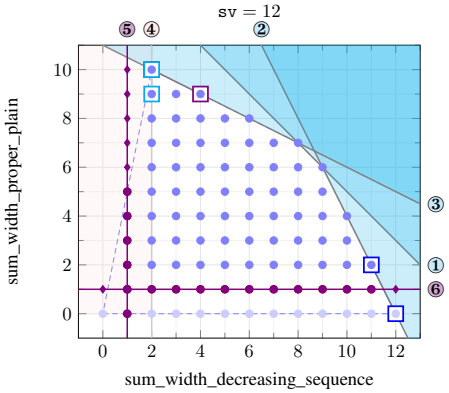
- $sv \geq 5$: $(2, sv - 2)$ $(4, sv - 3)$

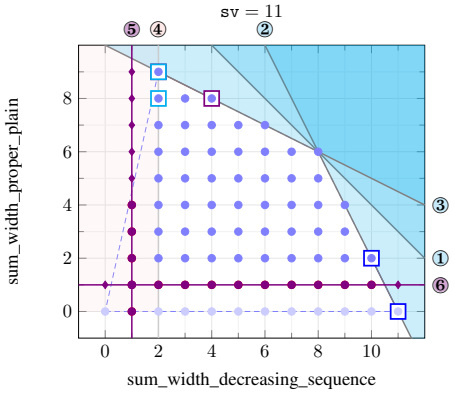
④ $y > 0 \Rightarrow x \geq 2$

- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

⑤ $x \neq 1$

⑥ $y \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

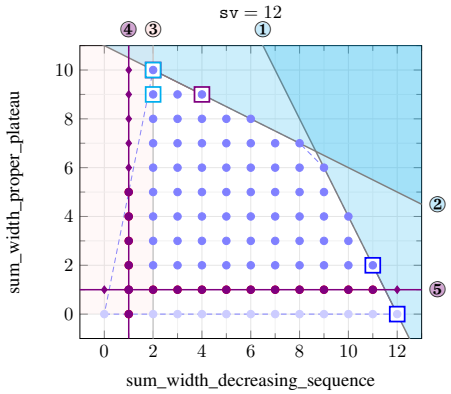
① $2 * x + y \leq 2 * sv$
- $sv \geq 5$: $(sv, 0)$ $(sv - 1, 2)$

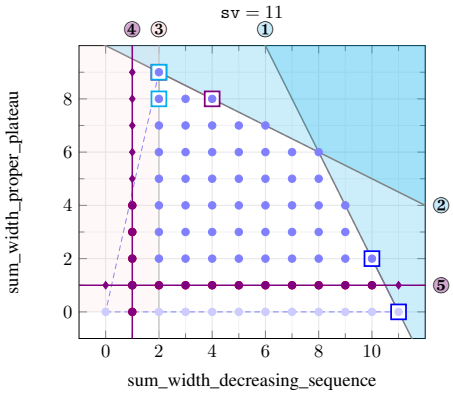
② $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 5$: $(2, sv - 2)$ $(4, sv - 3)$

③ $y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

④ $x \neq 1$

⑤ $y \neq 1$





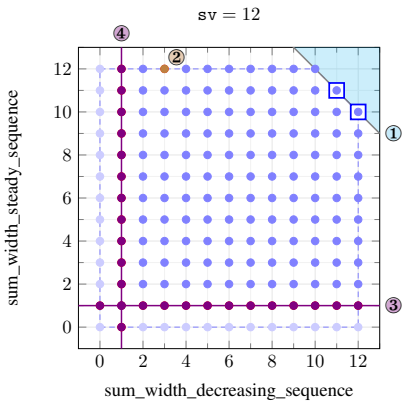
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

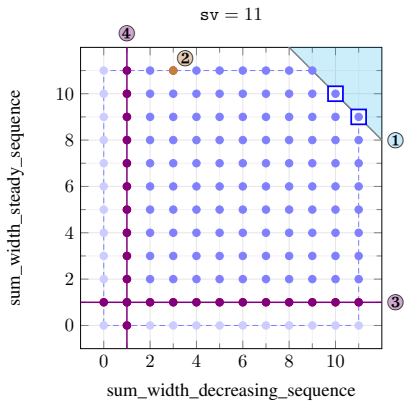
① $x + y \leq 2 * sv - 2$
- $sv \geq 5$: $(sv, sv - 2)$ $(sv - 1, sv - 1)$

② $x \neq 3 \vee y \neq sv * \min(1, \max(0, sv - 1))$

③ $y \neq 1$

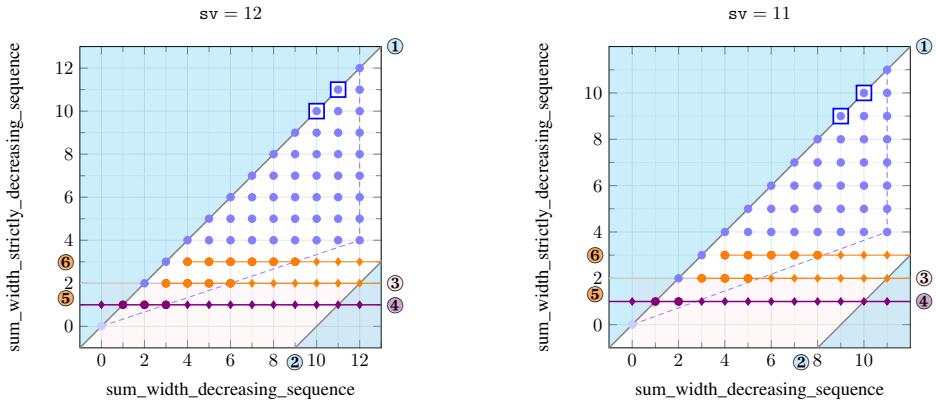
④ $x \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
□ $sv \geq 4$: $(sv-1, sv-1)$ $(sv-2, sv-2)$
② $sv > 1 \Rightarrow x \leq y + sv - 2$
③ $x > 0 \Rightarrow y \geq 2$
④ $y \neq 1$
⑤ $x < 3 \vee y \neq 2$
⑥ $x < 4 \vee y \neq 3$



SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq 2 * sv - 2$

□ $sv \bmod 2 = 0 \wedge sv \geq 2$: $(sv, sv - 2)$ $(sv - 2, sv)$

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $x = sv - 2 \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 1$

④ $y = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow x \leq sv - 3$

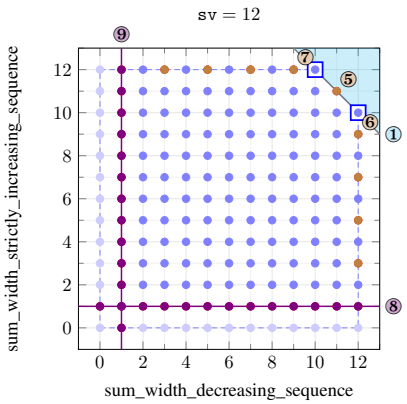
⑤ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2 \end{array} \right)$

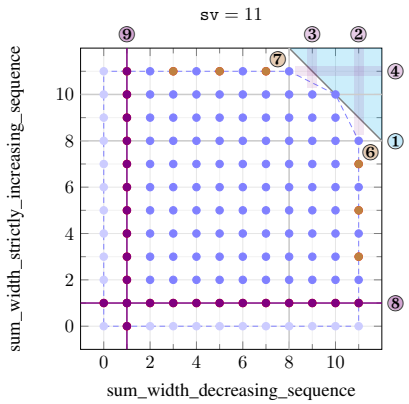
⑥ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)), \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{array} \right)$

⑦ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = x \bmod 2 \end{array} \right)$

⑧ $y \neq 1$

⑨ $x \neq 1$





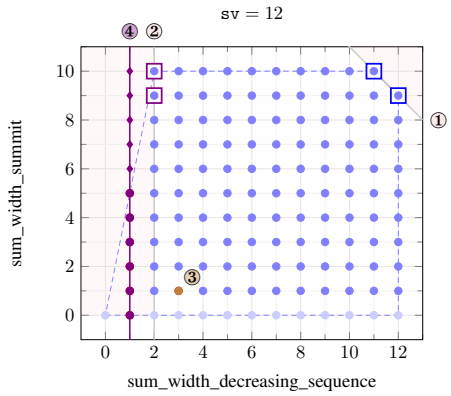
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

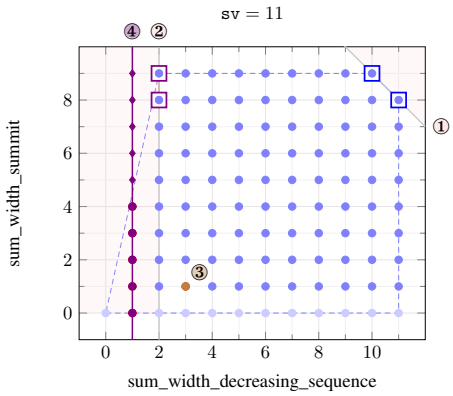
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 6$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





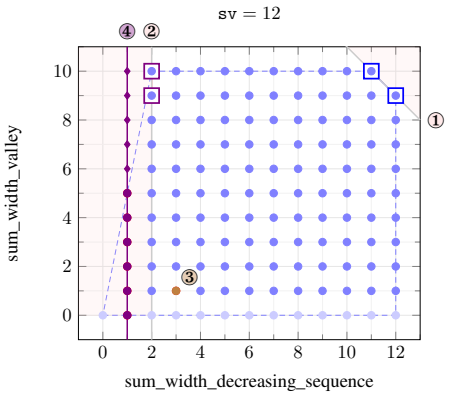
SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

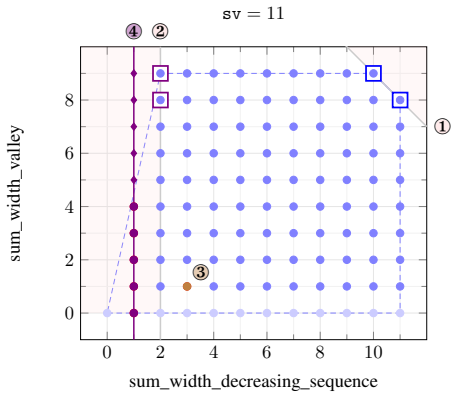
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 4$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





SUM_WIDTH_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- □ $sv \bmod 2 = 0 \wedge sv \geq 3$: $(sv-2, sv-2)$ $(0,0)$
- □ $sv \bmod 2 = 1 \wedge sv \geq 6$: $(sv-3, sv-3)$ $(2,2)$

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 2$

④ $$\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-3)), \\ x < 3, \\ x > sv * \min(1, \max(0, sv-1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = x \bmod 2 \end{array} \right)$$

⑤ $x \neq 5 \vee y < 4$

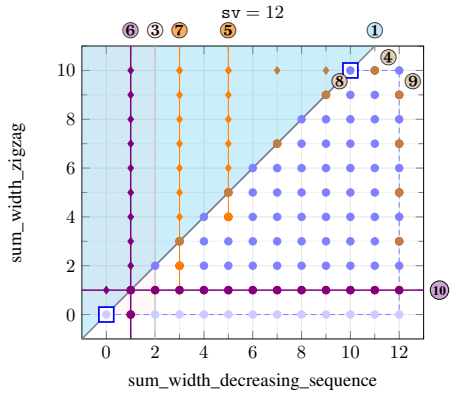
⑥ $x \neq 1$

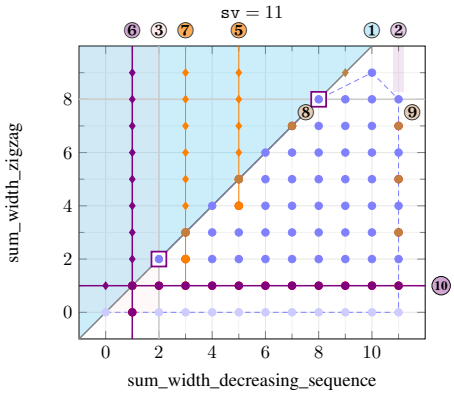
⑦ $x \neq 3 \vee y < 1$

⑧ $y \neq x \vee 0 = x \bmod 2$

⑨ $$\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv-1)), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ 0 = y \bmod 2 \end{array} \right)$$

⑩ $y \neq 1$





SUM_WIDTH_DECREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_GORGE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

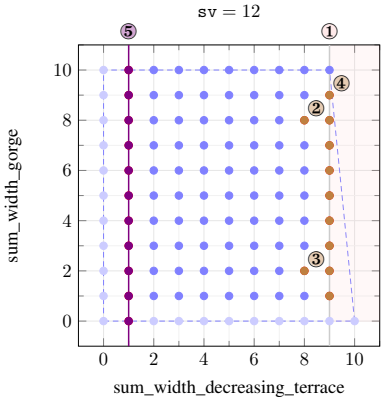
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

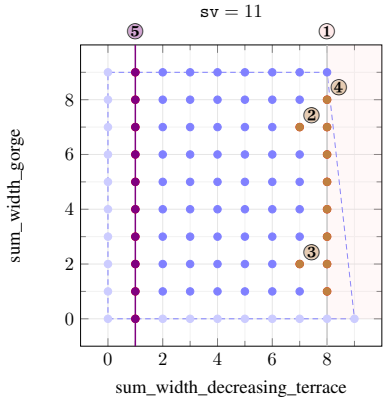
② $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ \mathtt{sv} < 5 \end{pmatrix}$

③ $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2 \vee \mathtt{y} \neq 2 \vee \mathtt{sv} < 5$

④ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1 \end{pmatrix}$

⑤ $\mathtt{x} \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

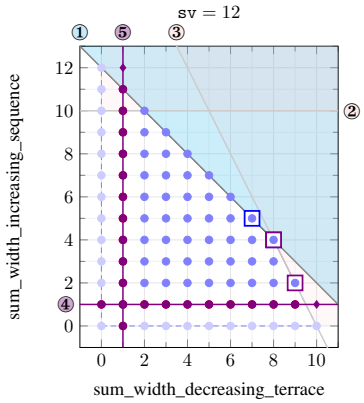
① $x + y \leq sv$
- $sv \geq 7$: $(sv - 4, 4)$ $(sv - 5, 5)$

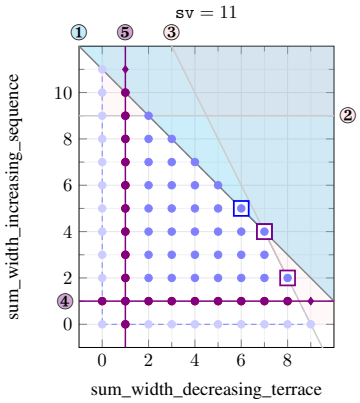
② $x > 0 \Rightarrow y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 6$: $(sv - 3, 2)$ $(sv - 4, 4)$

④ $y \neq 1$

⑤ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE($\mathtt{x}, \mathtt{VARIABLES}$) $\wedge$
SUM_WIDTH_INCREASING_TERRACE($\mathtt{y}, \mathtt{VARIABLES}$) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 4$: $(\mathtt{sv}-2, 0)$ $(0, \mathtt{sv}-2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 8$: $(\mathtt{sv}-5, 2)$ $(2, \mathtt{sv}-5)$

③ $\mathtt{o}^{=} \geq \lfloor (\mathtt{x}+1)/2 \rfloor + \lfloor (\mathtt{y}+1)/2 \rfloor$

④ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3 \end{pmatrix}$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2 \end{pmatrix}$

⑥ $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4 \vee \mathtt{y} \neq 4 \vee \mathtt{sv} < 7$

⑦ $\mathtt{x} \neq 4 \vee \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4 \vee \mathtt{sv} < 7$

⑧ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ 0 = \mathtt{y} \bmod 2 \end{pmatrix}$

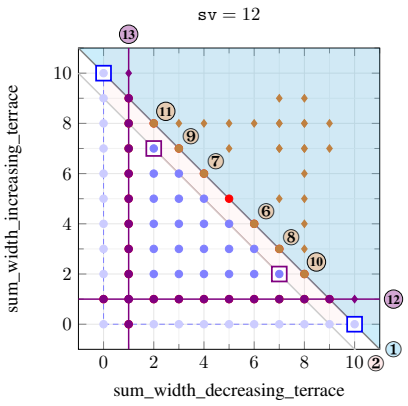
⑨ $\bigvee \begin{pmatrix} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3, \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ 0 = \mathtt{x} \bmod 2 \end{pmatrix}$

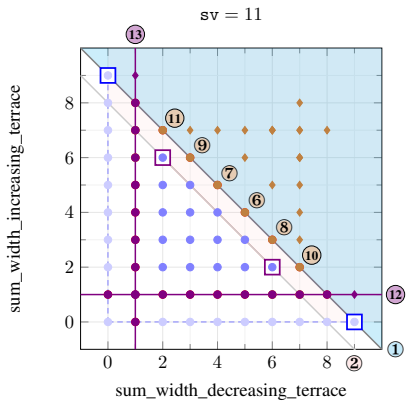
⑩ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1 \end{pmatrix}$

⑪ $\bigvee \begin{pmatrix} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1 \end{pmatrix}$

⑫ $\mathtt{y} \neq 1$

⑬ $\mathtt{x} \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

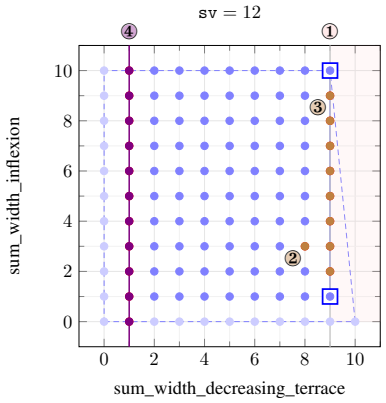
① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

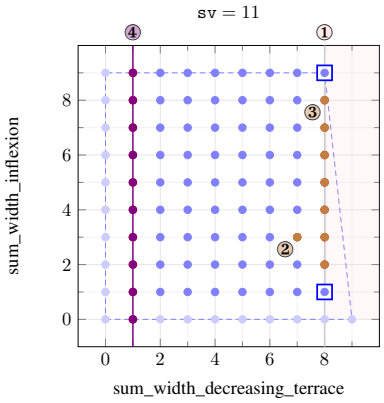
□ $sv \geq 5$: $(sv - 3, sv - 2)$ $(sv - 3, 1)$

② $x \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 2 \vee y \neq 3 \vee sv < 7$

③ $\bigvee \begin{pmatrix} x \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ y < 2, \\ y > (sv - 2) * \min(1, \max(0, sv - 2)) - 1 \end{pmatrix}$

④ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_PEAK$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + 2 * \mathtt{sv} - 9$

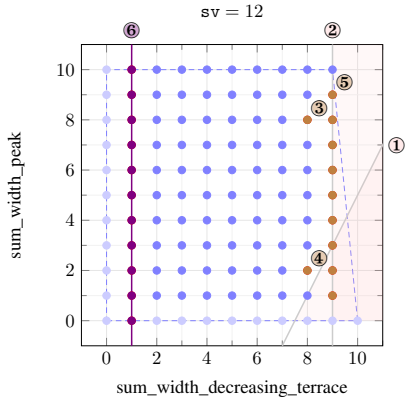
② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

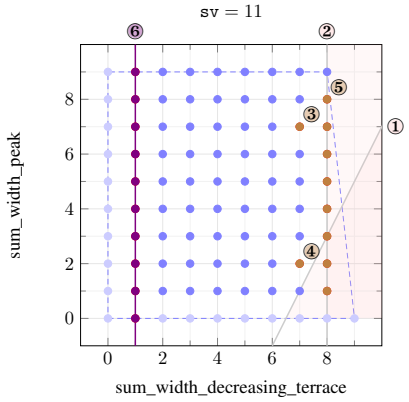
③ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ \mathtt{sv} < 5 \end{pmatrix}$

④ $\mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \vee \mathtt{y} \neq 2 \vee \mathtt{sv} < 5$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1 \end{pmatrix}$

⑥ $\mathtt{x} \neq 1$



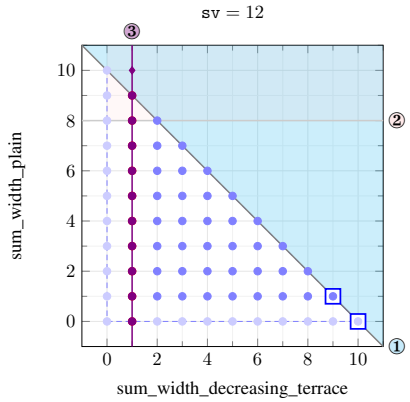


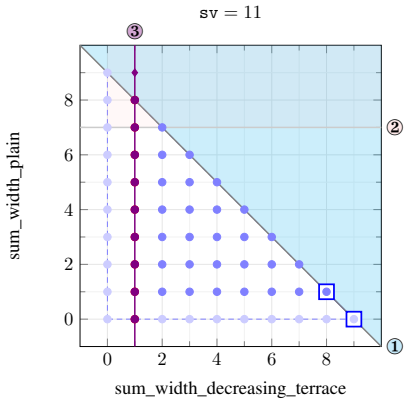
SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $x \neq 1$



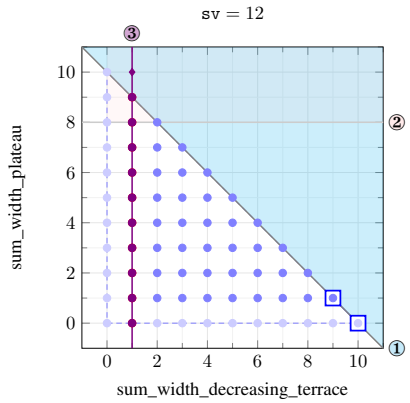


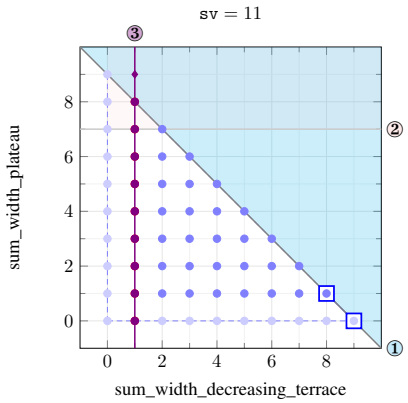
SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 7$: $(sv - 2, 0)$ $(sv - 4, 2)$

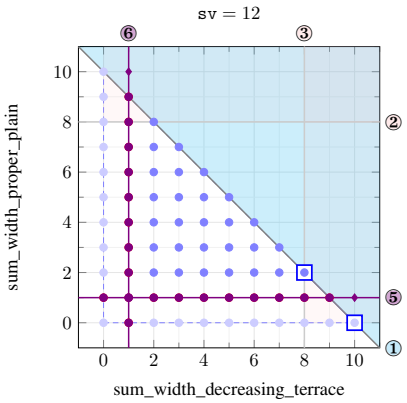
② $x > 0 \Rightarrow y \leq sv - 4$

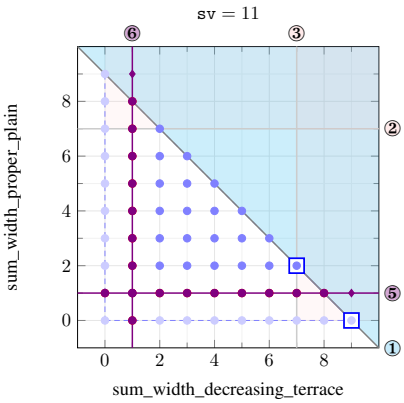
③ $y > 0 \Rightarrow x \leq sv - 4$

④ $o^{=} \geq \lfloor (x+1)/2 \rfloor + \lfloor (y+1)/2 \rfloor$

⑤ $y \neq 1$

⑥ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 6$: $(\mathtt{sv}-2, 0)$ $(\mathtt{sv}-4, 2)$

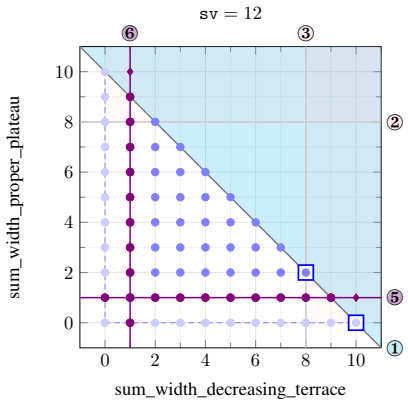
② $\mathtt{x} > 0 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 4$

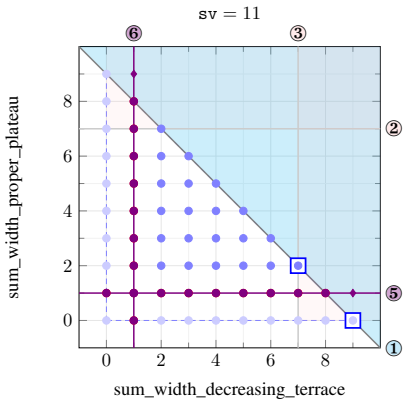
③ $\mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 4$

④ $\mathtt{o}^{=} \geq \lfloor(\mathtt{x}+1)/2\rfloor + \lfloor(\mathtt{y}+1)/2\rfloor$

⑤ $\mathtt{y} \neq 1$

⑥ $\mathtt{x} \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 5: (sv − 2, sv − 2) (sv − 3, sv − 3)

② $x > 0 \Rightarrow y \geq 2$

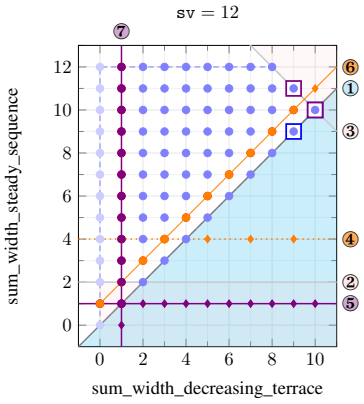
③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- sv ≥ 5: (sv − 2, sv − 2) (sv − 3, sv − 1)

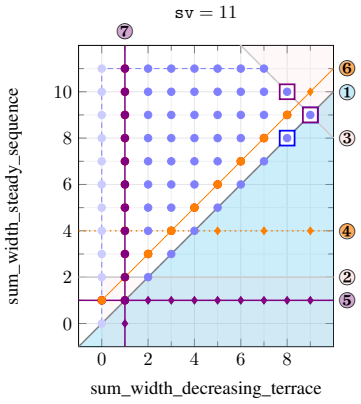
④ $y \neq 4 \vee 0 = x \bmod 2$

⑤ $y \neq 1$

⑥ $y \neq x + 1$

⑦ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x = \max(0, sv - 2) \wedge sv \bmod 2 = 1 \wedge sv > 3 \Rightarrow y \leq sv - 1$

② $y = 0 \Rightarrow x = 0$

③ $x > 0 \wedge y > 0 \Rightarrow y \geq 4$

□ $sv \geq 5$: $(sv - 2, 4)$ $(sv - 3, 4)$

④ $\bigvee \begin{pmatrix} x \neq (sv - 2) * \min(1, \max(0, sv - 3)), \\ y < 3, \\ y > sv * \min(1, \max(0, sv - 1)) - 1, \\ 0 = y \bmod 2 \end{pmatrix}$

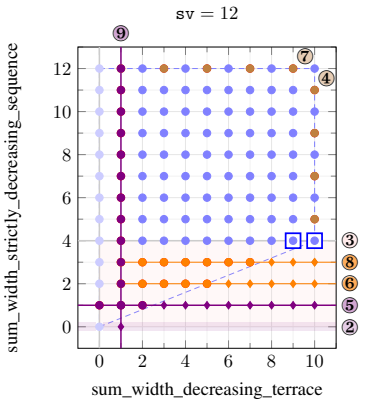
⑤ $y \neq 1$

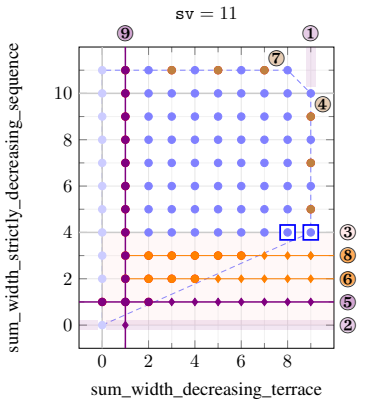
⑥ $x < 1 \vee y \neq 2$

⑦ $\bigvee \begin{pmatrix} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 0 = x \bmod 2 \end{pmatrix}$

⑧ $x < 1 \vee y \neq 3$

⑨ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

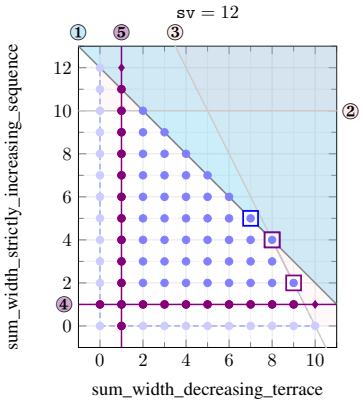
① $x + y \leq sv$
- $sv \geq 7$: $(sv - 4, 4)$ $(sv - 5, 5)$

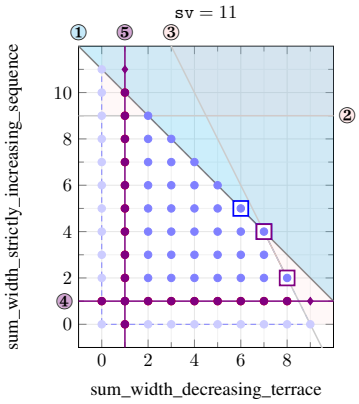
② $x > 0 \Rightarrow y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 6$: $(sv - 3, 2)$ $(sv - 4, 4)$

④ $y \neq 1$

⑤ $x \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + 2 * sv - 9$

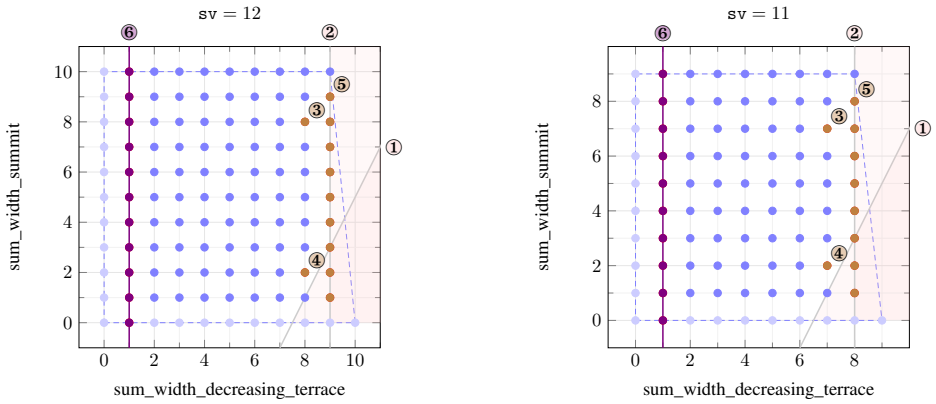
② $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

③ $\bigvee \begin{pmatrix} x \neq (sv-2) * \min(1, \max(0, sv-3)) - 2, \\ y \neq (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ sv < 5 \end{pmatrix}$

④ $x \neq (sv-2) * \min(1, \max(0, sv-3)) - 2 \vee y \neq 2 \vee sv < 5$

⑤ $\bigvee \begin{pmatrix} x \neq (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1 \end{pmatrix}$

⑥ $x \neq 1$



SUM_WIDTH_DECREASING_TERRACE$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_VALLEY$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

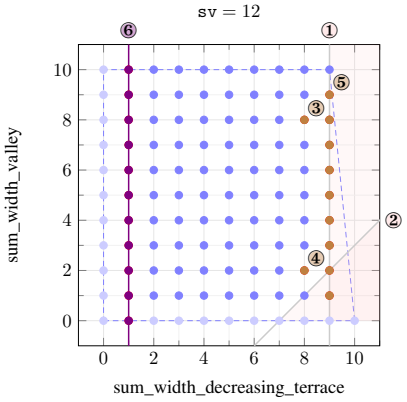
② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 5$

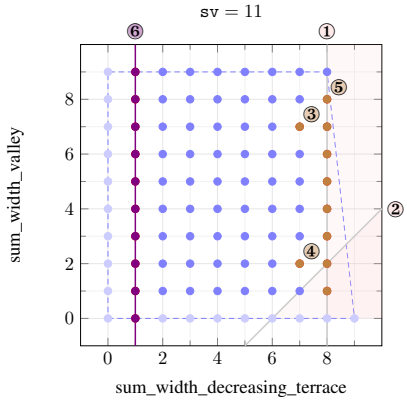
③ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ \mathtt{sv} < 5 \end{pmatrix}$

④ $\mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \vee \mathtt{y} \neq 2 \vee \mathtt{sv} < 5$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1 \end{pmatrix}$

⑥ $\mathtt{x} \neq 1$





SUM_WIDTH_DECREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

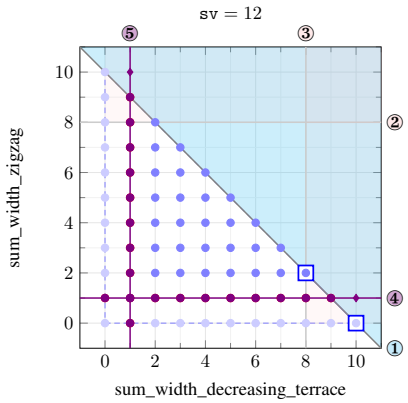
① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 6$: $(\mathtt{sv} - 2, 0)$ $(\mathtt{sv} - 4, 2)$

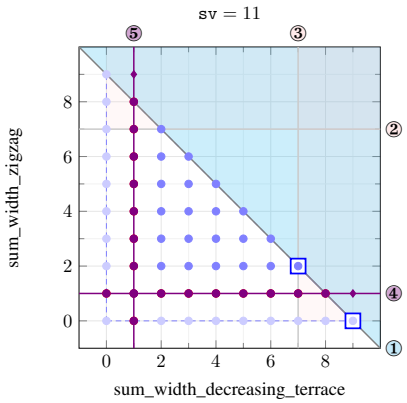
② $\mathtt{x} > 0 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 4$

③ $\mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 4$

④ $\mathtt{y} \neq 1$

⑤ $\mathtt{x} \neq 1$





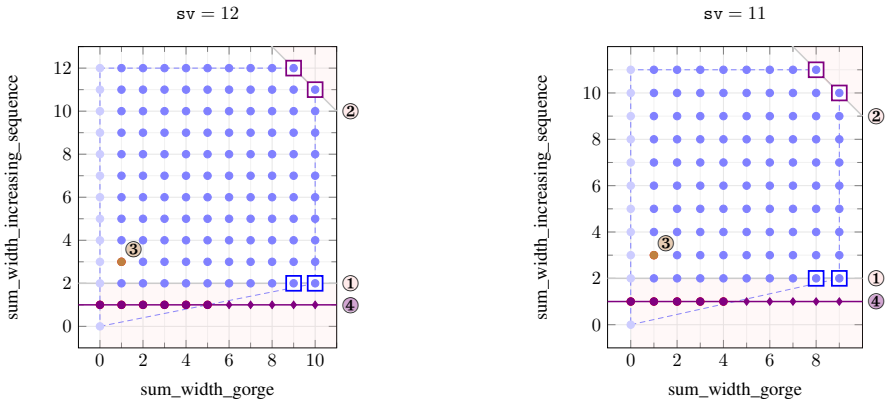
SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \Rightarrow y \geq 2$
- sv ≥ 3: (sv − 2, 2) (sv − 3, 2)

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- sv ≥ 6: (sv − 2, sv − 1) (sv − 3, sv)

③ $x \neq 1 \vee y \neq 3$

④ $y \neq 1$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 3$

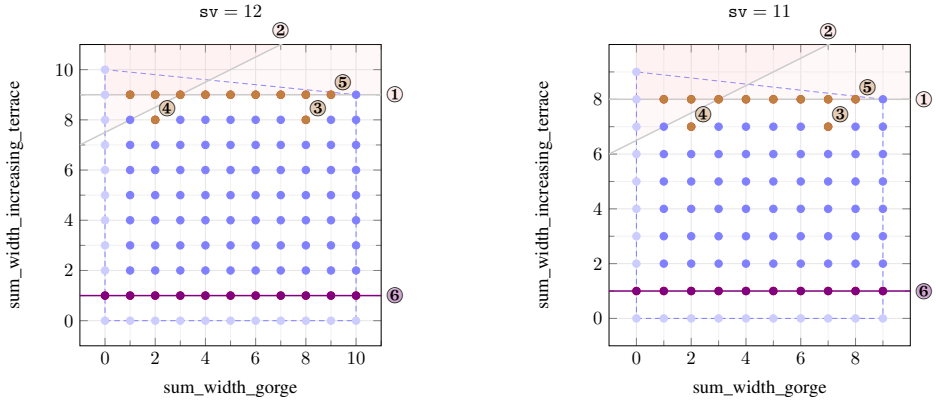
② $x > 0 \wedge y > 0 \Rightarrow 2 * y \leq x + 2 * sv - 9$

③ $\bigvee \left(\begin{array}{l} x \neq (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ y \neq (sv-2) * \min(1, \max(0, sv-3)) - 2, \\ sv < 5 \end{array} \right)$

④ $x \neq 2 \vee y \neq (sv-2) * \min(1, \max(0, sv-3)) - 2 \vee sv < 5$

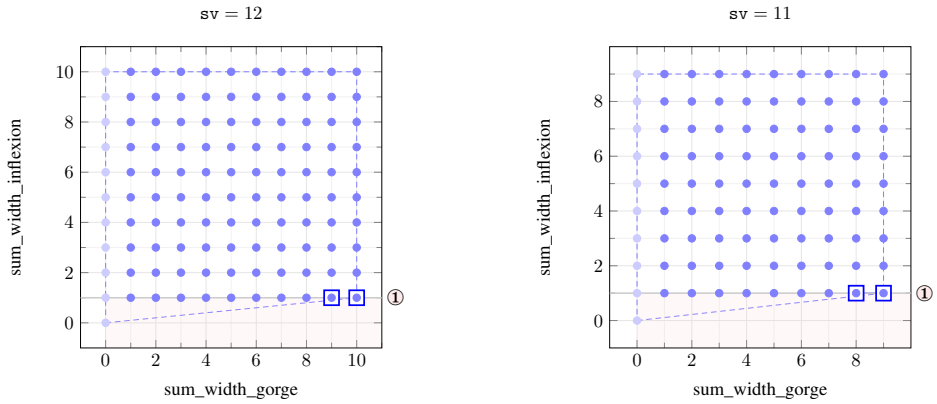
⑤ $\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ x < 1, \\ x > (sv-2) * \min(1, \max(0, sv-2)) - 1 \end{array} \right)$

⑥ $y \neq 1$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

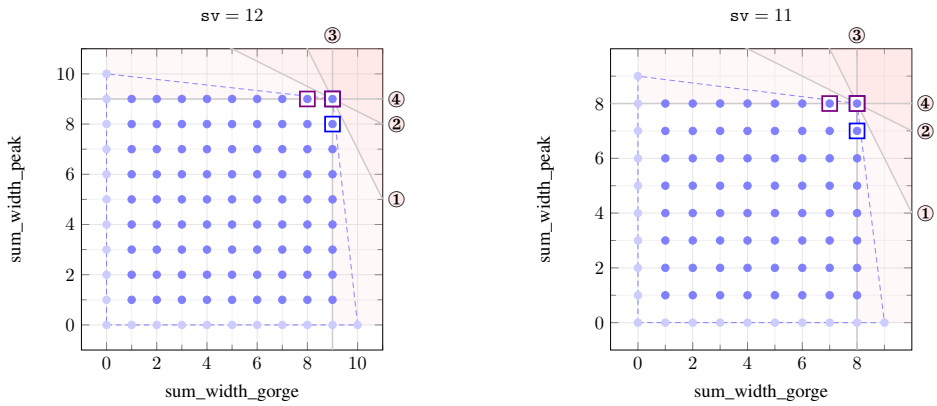
① $x > 0 \Rightarrow y \geq 1$
- sv ≥ 5: (sv − 2, 1) (sv − 3, 1)



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 3 * sv - 9$
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 3 * sv - 9$
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$
- sv ≥ 4: (sv − 3, sv − 3) (sv − 3, sv − 4)

④ $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 3$
- sv ≥ 4: (sv − 3, sv − 3) (sv − 4, sv − 3)



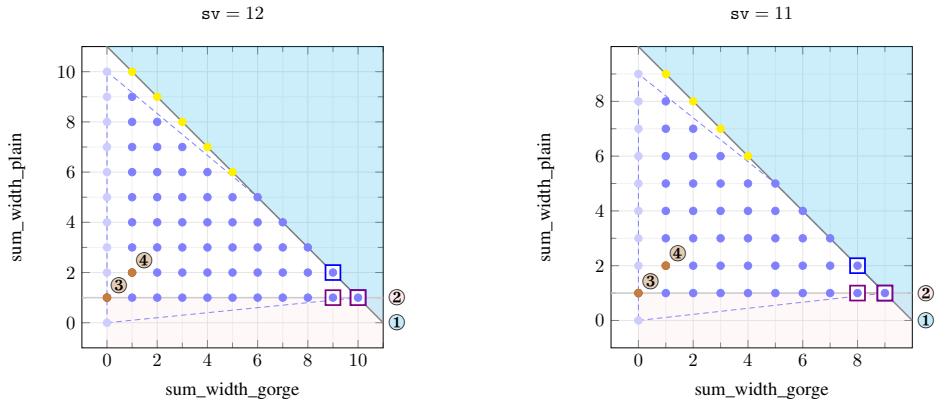
SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(sv - 2, 1)$ $(sv - 3, 2)$

② $x > 0 \Rightarrow y \geq 1$
- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$

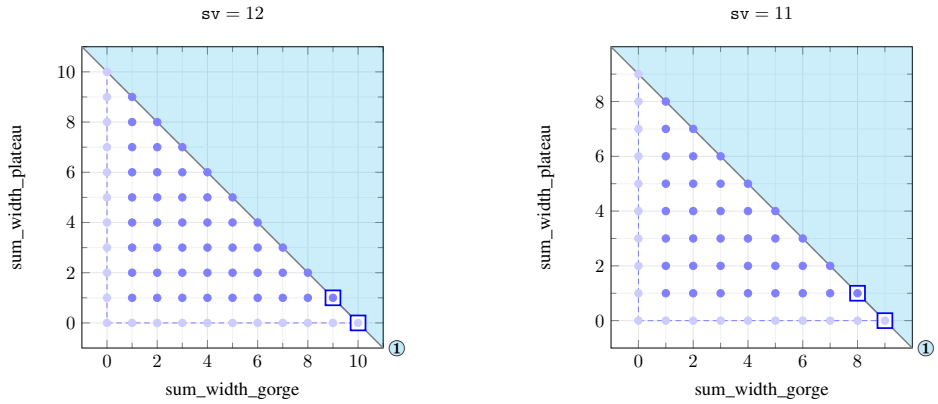
③ $x \neq 0 \vee y \neq 1$

④ $x \neq 1 \vee y \neq 2$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(sv - 2, 0)$ $(sv - 3, 1)$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 3$: $(\mathtt{sv}-2, 0)$ $(0, \mathtt{sv}-2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 7$: $(\mathtt{sv}-5, 2)$ $(1, \mathtt{sv}-4)$

③ $\mathtt{x} \neq 4 \vee \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4 \vee \mathtt{sv} < 7$

④ $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 4 \vee \mathtt{y} < 4 \vee \mathtt{sv} < 7$

⑤ $\vee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3, \\ \mathtt{x} < 3, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

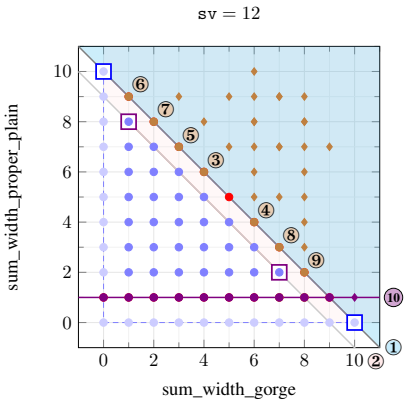
⑥ $\vee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

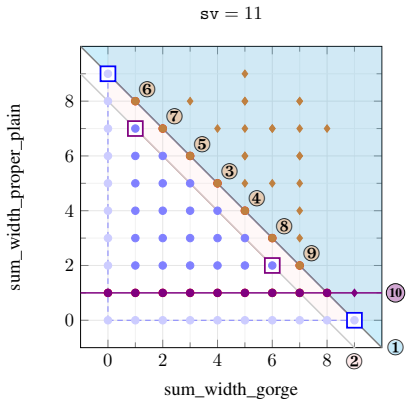
⑦ $\vee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{x} < 2, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1, \\ 1 = \mathtt{x} \bmod 2 \end{array} \right)$

⑧ $\vee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 3, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑨ $\vee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1 \end{array} \right)$

⑩ $\mathtt{y} \neq 1$



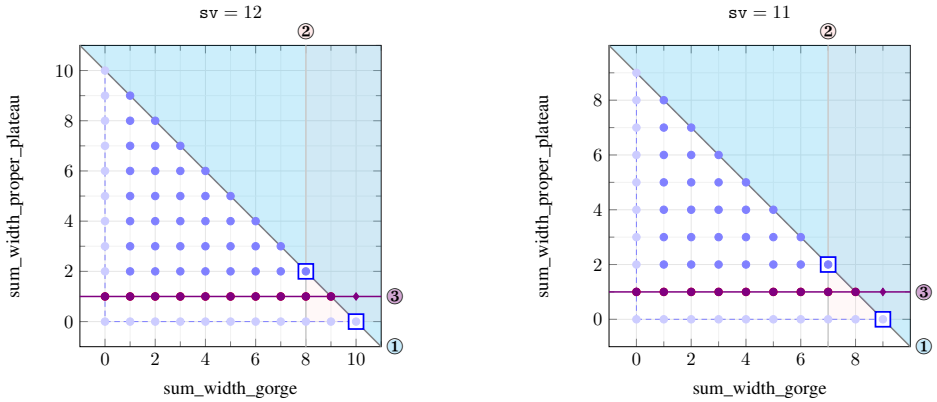


SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

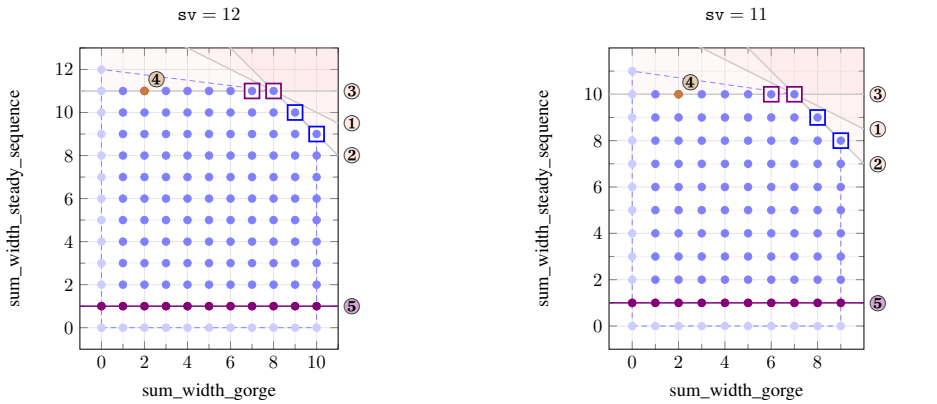
① $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 3 * sv - 6$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 5$
- $sv \geq 6$: $(sv - 2, sv - 3)$ $(sv - 3, sv - 2)$

③ $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 1$
- $sv \geq 8$: $(sv - 4, sv - 1)$ $(sv - 5, sv - 1)$

④ $x \neq 2 \vee y \neq sv * \min(1, \max(0, sv - 1)) - 1$

⑤ $y \neq 1$



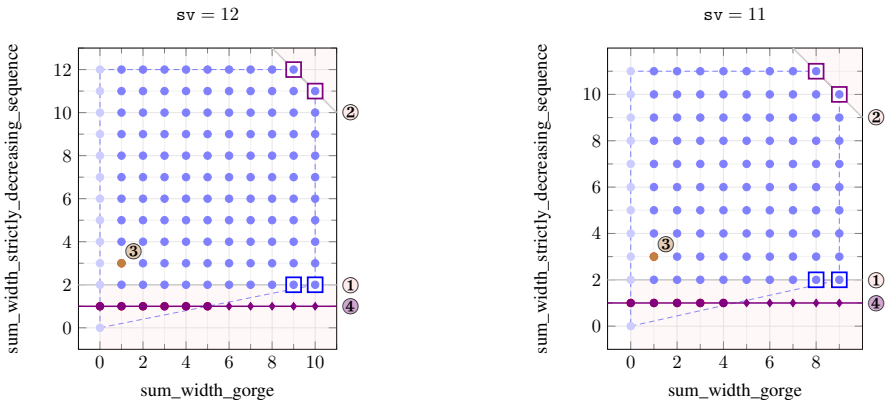
SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv - 2, sv - 1)$ $(sv - 3, sv)$

③ $x \neq 1 \vee y \neq 3$

④ $y \neq 1$



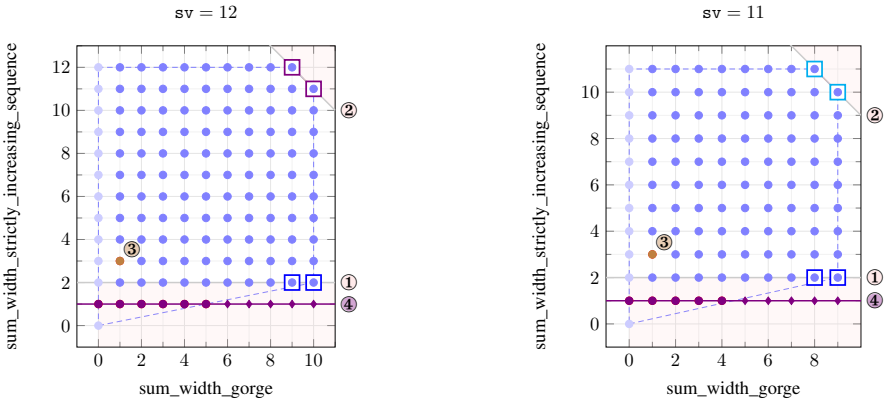
SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv-2, 2)$ $(sv-3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(sv-2, sv-1)$ $(sv-3, sv)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(sv-2, sv-1)$ $(sv-3, sv)$

③ $x \neq 1 \vee y \neq 3$

④ $y \neq 1$



SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 3 * sv - 9$

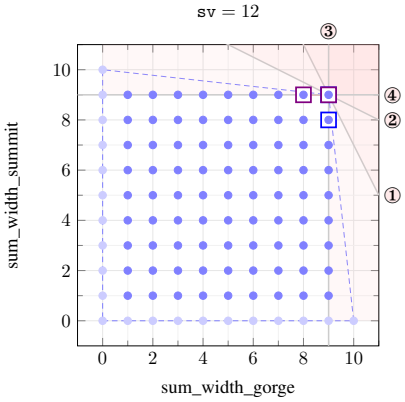
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 3 * sv - 9$

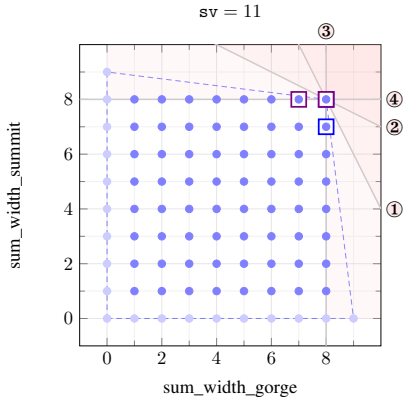
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 3, sv - 4)$

④ $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 3$

- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 4, sv - 3)$





SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$

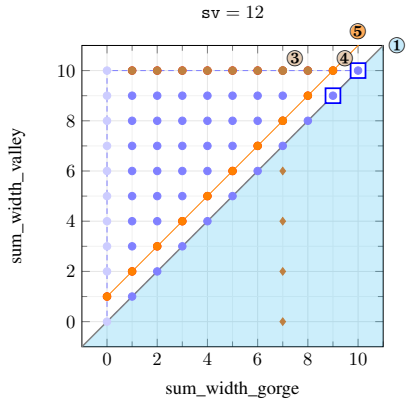
□ $sv \geq 3$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

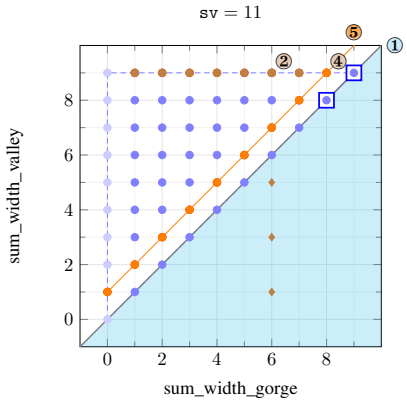
② $\bigvee \left(\begin{array}{l} x \neq (sv-2) * \min(1, \max(0, sv-2)) - 3, \\ 0 = y \bmod 2, \\ 0 = sv \bmod 2, \\ sv < 6 \end{array} \right)$

③ $\bigvee \left(\begin{array}{l} x \neq (sv-2) * \min(1, \max(0, sv-2)) - 3, \\ 1 = y \bmod 2, \\ 1 = sv \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-2)), \\ x < 1, \\ x > (sv-2) * \min(1, \max(0, sv-2)) - 1 \end{array} \right)$

⑤ $y \neq x + 1$





SUM_WIDTH_GORGE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

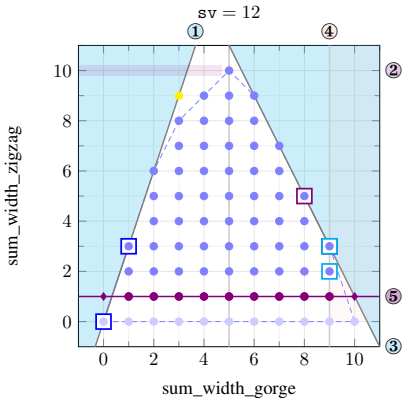
① $y \leq 3 * x$
- sv ≥ 5: (0, 0) (1, 3)

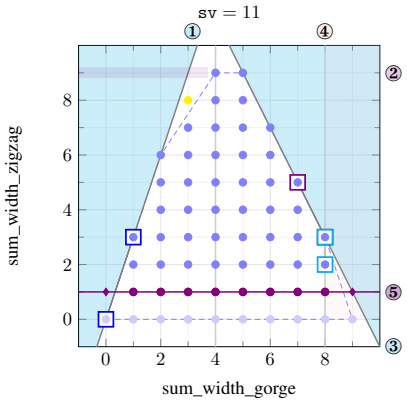
② $y = \max(0, sv - 2) \wedge sv > 1 \Rightarrow 2 * x \geq sv - 2 - sv \bmod 2$

③ $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3$
- sv ≥ 7: (sv − 3, 3) (sv − 4, 5)

④ $y > 0 \Rightarrow x \leq sv - 3$
- sv ≥ 5: (sv − 3, 2) (sv − 3, 3)

⑤ $y \neq 1$





SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INCREASING_TERRACE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow y \leq x - 2$

□ $sv \geq 5$: $(sv, sv - 2)$ $(sv - 1, sv - 3)$

② $y > 0 \Rightarrow x \geq 4$

③ $x \neq 5 \vee y < 4$

④ $\bigvee \begin{pmatrix} x \neq sv * \min(1, \max(0, sv - 1)) - 4, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 3 \end{pmatrix}$

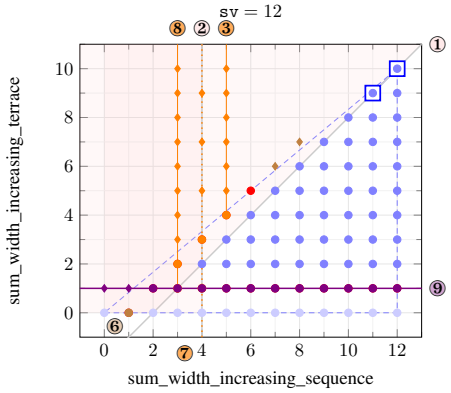
⑤ $\bigvee \begin{pmatrix} x \neq sv * \min(1, \max(0, sv - 1)) - 5, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 4 \end{pmatrix}$

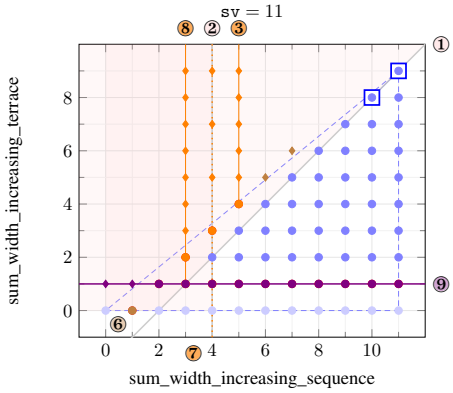
⑥ $x \neq 1 \vee y \neq 0$

⑦ $x \neq 4 \vee 0 = y \bmod 2$

⑧ $x \neq 3 \vee y < 1$

⑨ $y \neq 1$





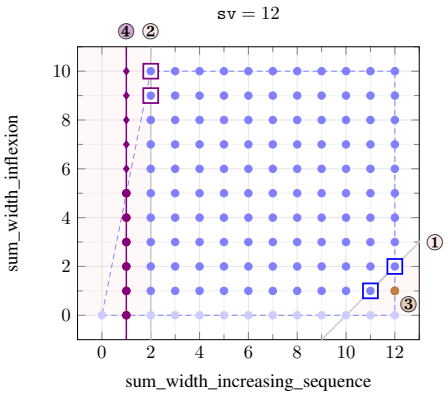
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_INFLEXION(y, VARIABLES) with sv = |VARIABLES|

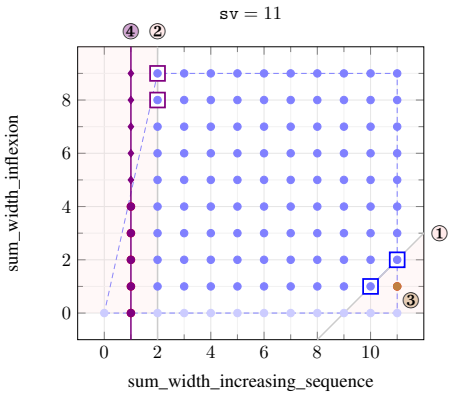
① $x > 0 \wedge y > 0 \Rightarrow x \leq y + sv - 2$
- $sv \geq 6$: $(sv, 2)$ $(sv - 1, 1)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq sv * \min(1, \max(0, sv - 1)) \vee y \neq 1$

④ $x \neq 1$





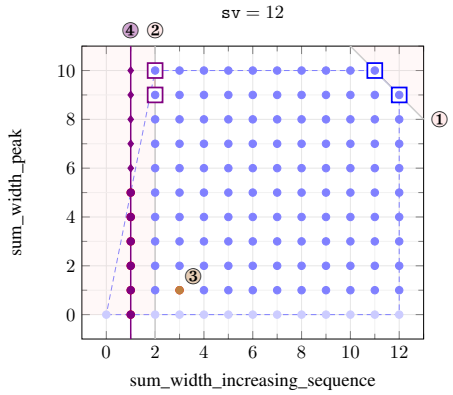
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

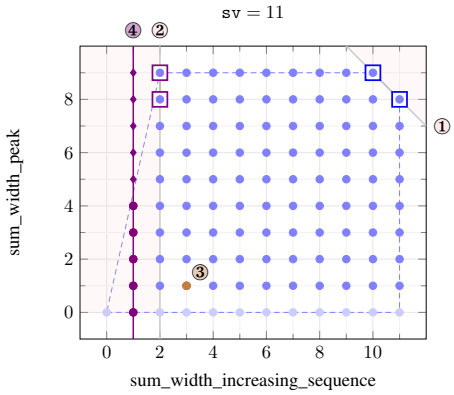
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 4$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$



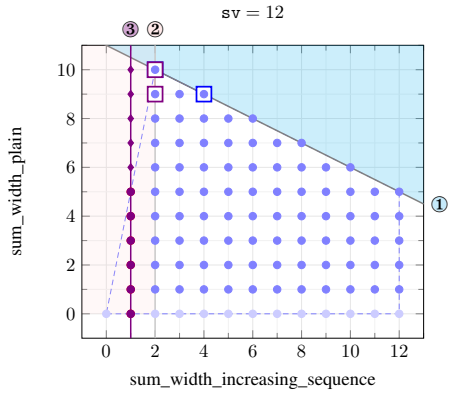


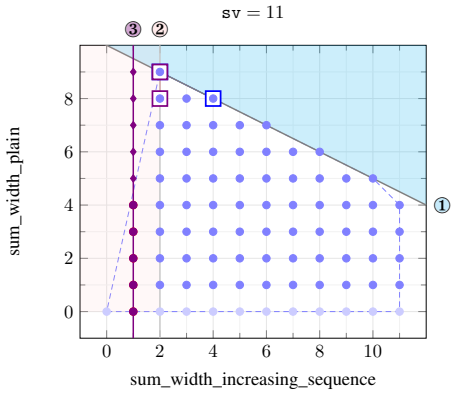
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 4$: $(2, sv - 2)$ $(4, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 1$



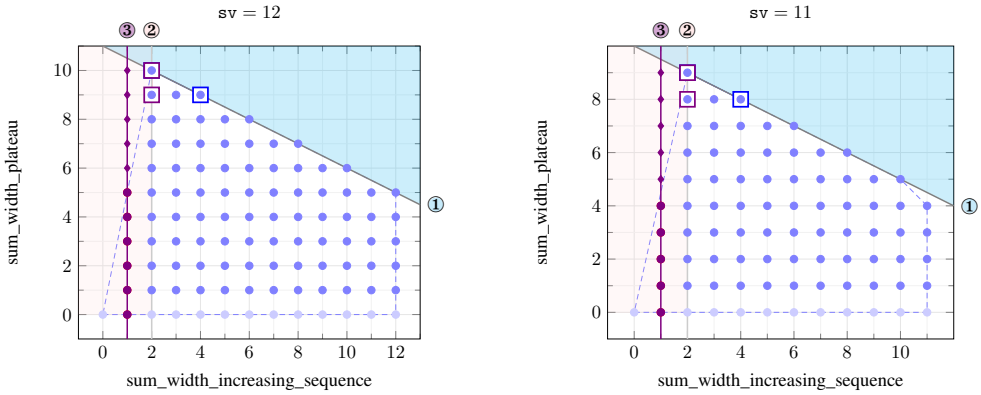


SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 4$: $(2, sv - 2)$ $(4, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 1$



SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

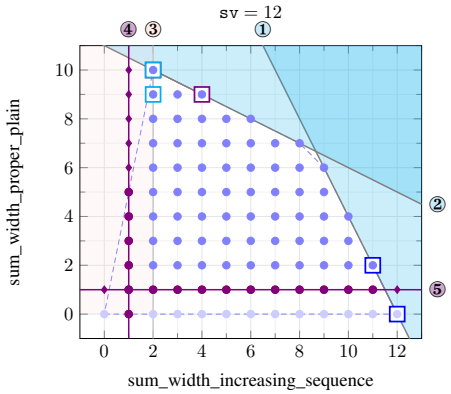
① $2 * x + y \leq 2 * sv$
- $sv \geq 5$: $(sv, 0)$ $(sv - 1, 2)$

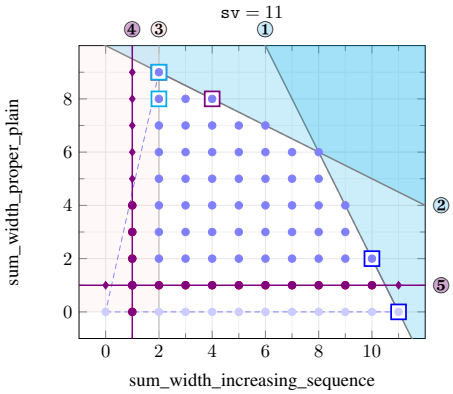
② $x + 2 * y \leq 2 * sv - 2$
- $sv \geq 5$: $(2, sv - 2)$ $(4, sv - 3)$

③ $y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

④ $x \neq 1$

⑤ $y \neq 1$





SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + 3 * y \leq 4 * sv - 2 - (4 * sv - 2) \bmod 3$

② $2 * x + y \leq 2 * sv$

- $sv \geq 5$: $(sv, 0)$ $(sv - 1, 2)$

③ $x + 2 * y \leq 2 * sv - 2$

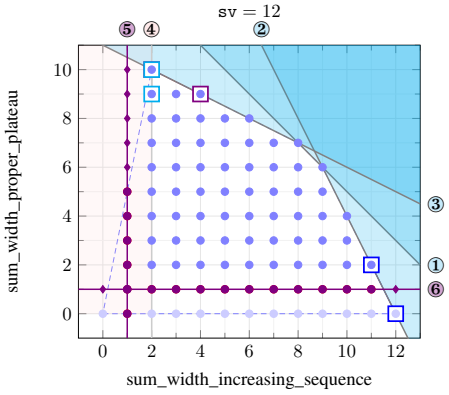
- $sv \geq 5$: $(2, sv - 2)$ $(4, sv - 3)$

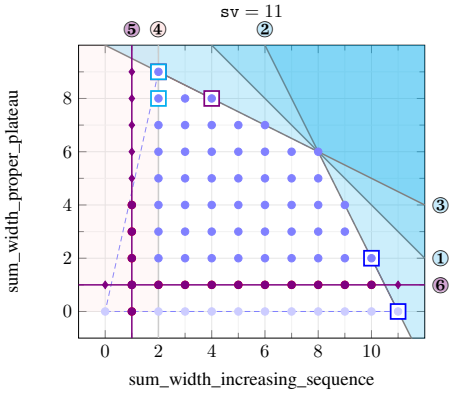
④ $y > 0 \Rightarrow x \geq 2$

- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

⑤ $x \neq 1$

⑥ $y \neq 1$





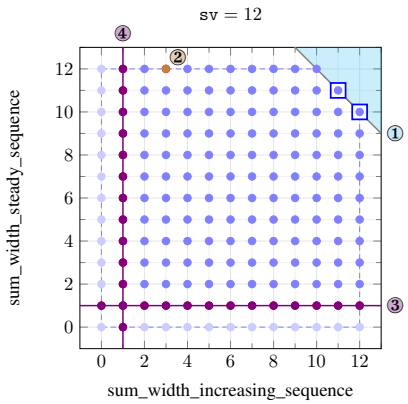
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

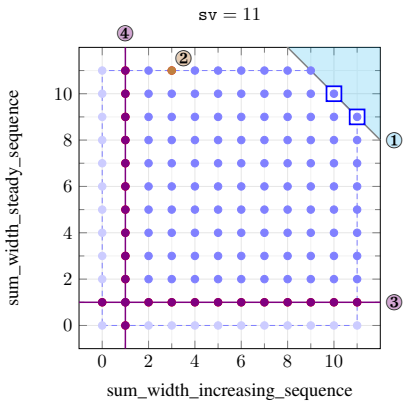
① $x + y \leq 2 * sv - 2$
- $sv \geq 5$: $(sv, sv - 2)$ $(sv - 1, sv - 1)$

② $x \neq 3 \vee y \neq sv * \min(1, \max(0, sv - 1))$

③ $y \neq 1$

④ $x \neq 1$





SUM_WIDTH_INCREASING_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\mathtt{sv}, \mathtt{sv}-2)$ $(\mathtt{sv}-2, \mathtt{sv})$

② $\mathtt{x} = \mathtt{sv} \wedge \mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} > 1 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 3$

③ $\mathtt{y} = \mathtt{sv} \wedge \mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} > 1 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

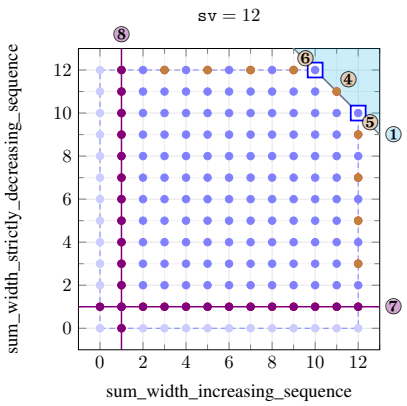
④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ 1 = \mathtt{sv} \bmod 2 \end{array} \right)$

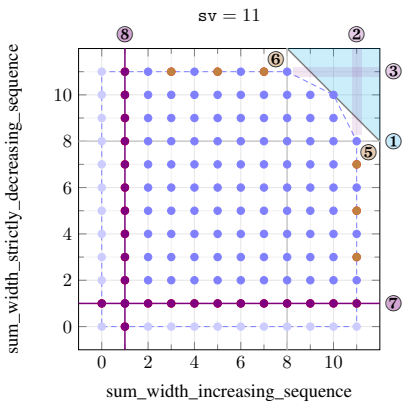
⑤ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)), \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 3, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑥ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)), \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 3, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑦ $\mathtt{y} \neq 1$

⑧ $\mathtt{x} \neq 1$





SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 4$: $(sv-1, sv-1)$ $(sv-2, sv-2)$

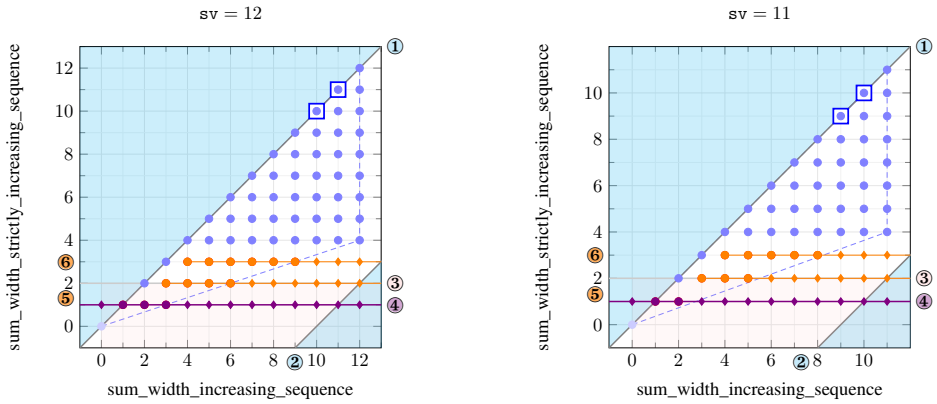
② $sv > 1 \Rightarrow x \leq y + sv - 2$

③ $x > 0 \Rightarrow y \geq 2$

④ $y \neq 1$

⑤ $x < 3 \vee y \neq 2$

⑥ $x < 4 \vee y \neq 3$



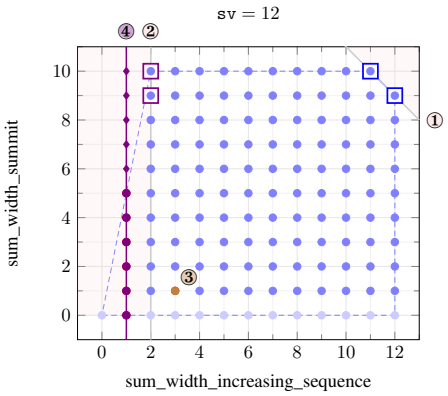
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

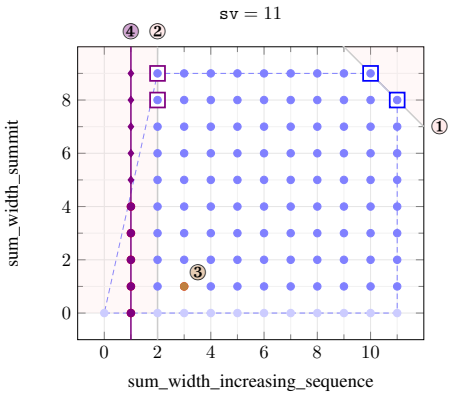
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





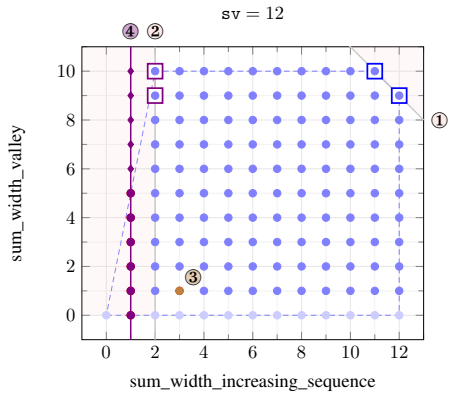
SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

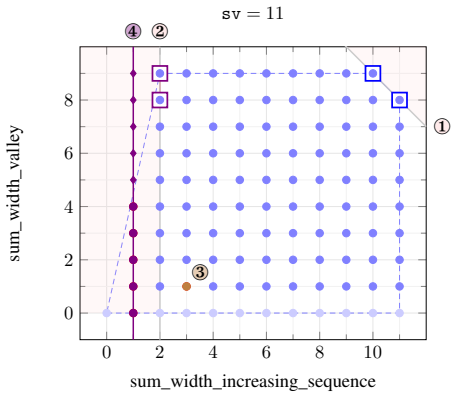
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 6$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





SUM_WIDTH_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- sv mod 2 = 0 ∧ sv ≥ 3: (sv − 2, sv − 2) (0, 0)
- sv mod 2 = 1 ∧ sv ≥ 6: (sv − 3, sv − 3) (2, 2)

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 2$

④ $\bigvee \begin{pmatrix} y \neq (sv-2) * \min(1, \max(0, sv-3)), \\ x < 3, \\ x > sv * \min(1, \max(0, sv-1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = x \bmod 2 \end{pmatrix}$

⑤ $x \neq 5 \vee y < 4$

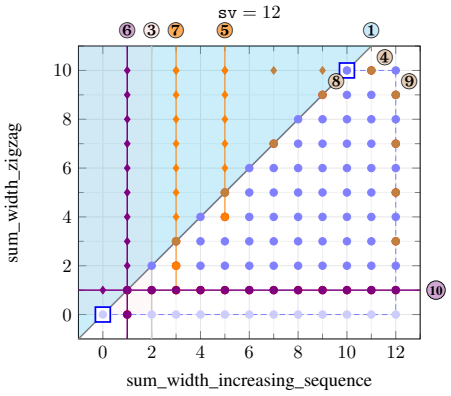
⑥ $x \neq 1$

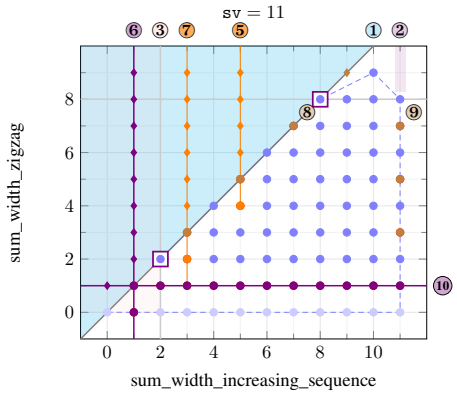
⑦ $x \neq 3 \vee y < 1$

⑧ $y \neq x \vee 0 = x \bmod 2$

⑨ $\bigvee \begin{pmatrix} x \neq sv * \min(1, \max(0, sv-1)), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ 0 = y \bmod 2 \end{pmatrix}$

⑩ $y \neq 1$





SUM_WIDTH_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_INFLEXION($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

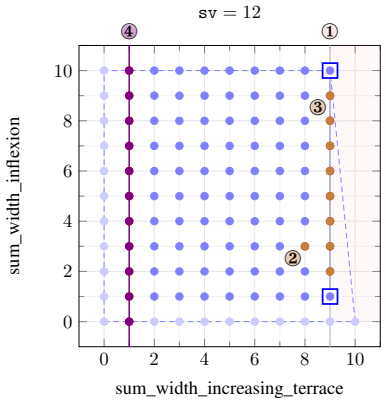
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

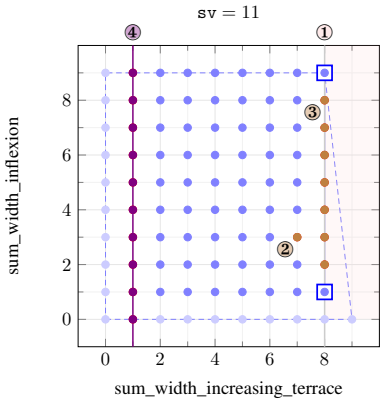
□ $\mathtt{sv} \geq 5$: $(\mathtt{sv}-3, \mathtt{sv}-2)$ $(\mathtt{sv}-3, 1)$

② $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2 \vee \mathtt{y} \neq 3 \vee \mathtt{sv} < 7$

③ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1 \end{pmatrix}$

④ $\mathtt{x} \neq 1$





SUM_WIDTH_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_PEAK($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

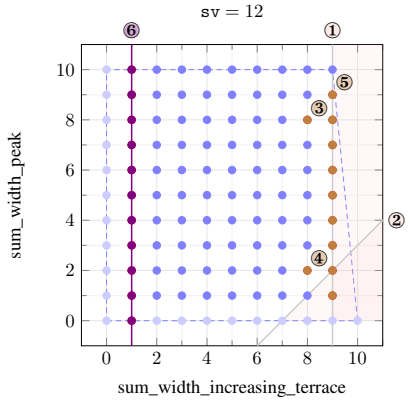
② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 5$

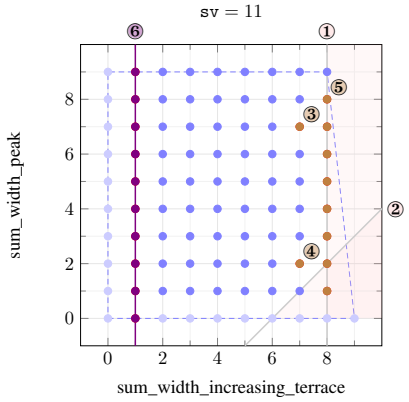
③ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ \mathtt{sv} < 5 \end{pmatrix}$

④ $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2 \vee \mathtt{y} \neq 2 \vee \mathtt{sv} < 5$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1 \end{pmatrix}$

⑥ $\mathtt{x} \neq 1$



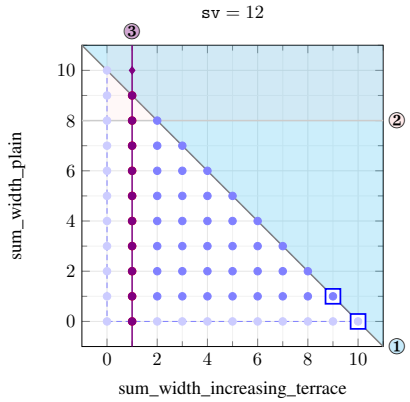


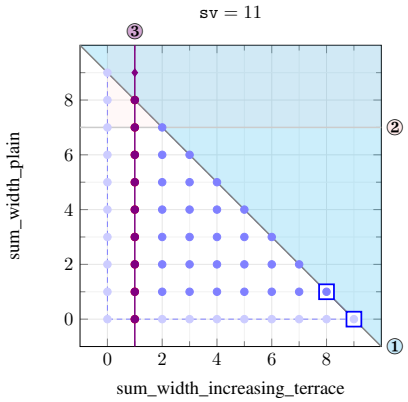
SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $x \neq 1$



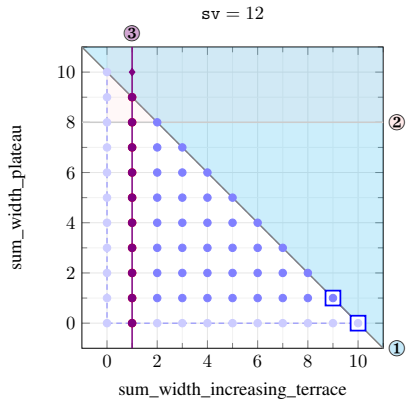


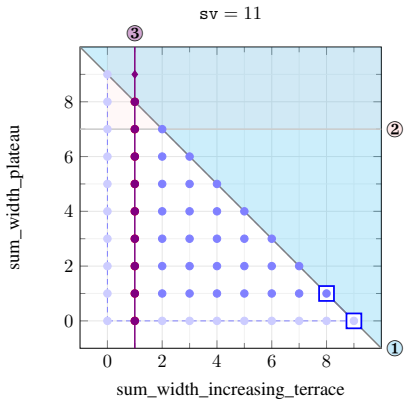
SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

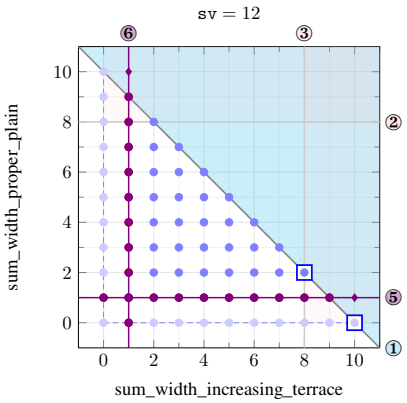
③ $x \neq 1$

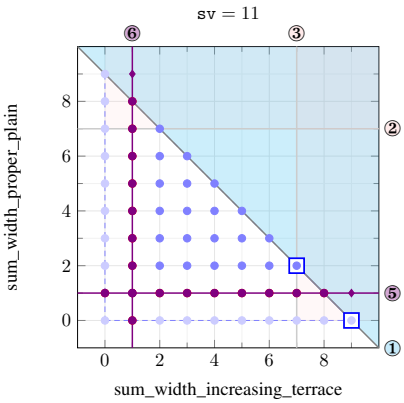




SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
□ $sv \geq 6$: $(sv - 2, 0)$ $(sv - 4, 2)$
② $x > 0 \Rightarrow y \leq sv - 4$
③ $y > 0 \Rightarrow x \leq sv - 4$
④ $o^{=} \geq \lfloor (x+1)/2 \rfloor + \lfloor (y+1)/2 \rfloor$
⑤ $y \neq 1$
⑥ $x \neq 1$





SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

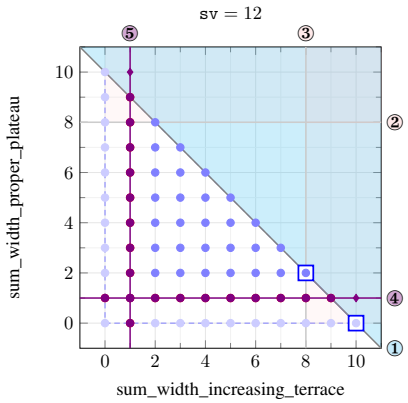
① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 7$: $(\mathtt{sv}-2, 0)$ $(\mathtt{sv}-4, 2)$

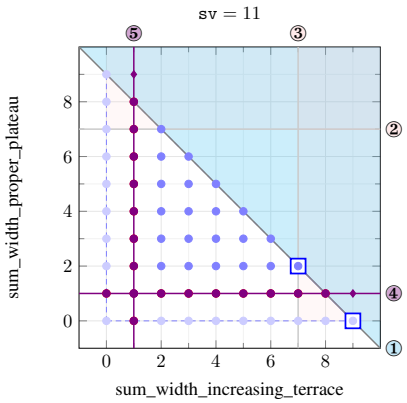
② $\mathtt{x} > 0 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 4$

③ $\mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 4$

④ $\mathtt{y} \neq 1$

⑤ $\mathtt{x} \neq 1$





SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- sv ≥ 5: (sv − 2, sv − 2) (sv − 3, sv − 3)

② $x > 0 \Rightarrow y \geq 2$

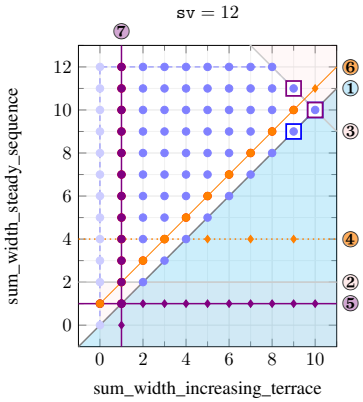
③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- sv ≥ 5: (sv − 2, sv − 2) (sv − 3, sv − 1)

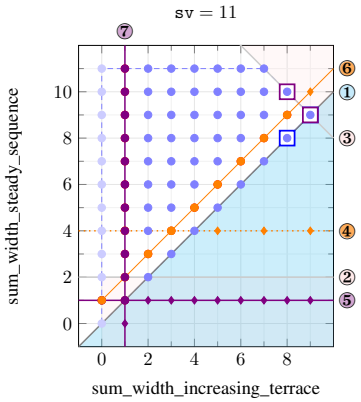
④ $y \neq 4 \vee 0 = x \bmod 2$

⑤ $y \neq 1$

⑥ $y \neq x + 1$

⑦ $x \neq 1$





SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

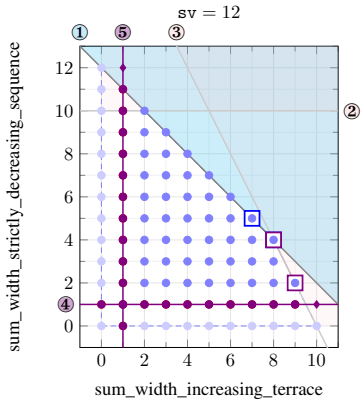
① $x + y \leq sv$
- $sv \geq 7$: $(sv - 4, 4)$ $(sv - 5, 5)$

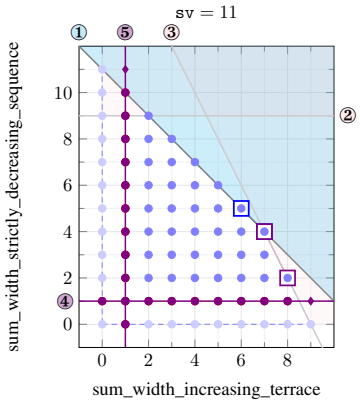
② $x > 0 \Rightarrow y \leq sv - 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 6$: $(sv - 3, 2)$ $(sv - 4, 4)$

④ $y \neq 1$

⑤ $x \neq 1$





SUM_WIDTH_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} = \mathtt{max}(0, \mathtt{sv} - 2) \wedge \mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} > 3 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 1$

② $\mathtt{y} = 0 \Rightarrow \mathtt{x} = 0$

③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{y} \geq 4$

□ $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 2, 4)$ $(\mathtt{sv} - 3, 4)$

④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)), \\ \mathtt{y} < 3, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 1, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

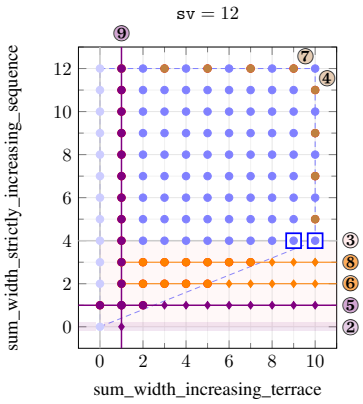
⑤ $\mathtt{y} \neq 1$

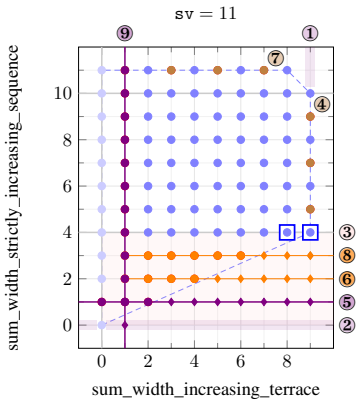
⑥ $\mathtt{x} < 1 \vee \mathtt{y} \neq 2$

⑦ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)), \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 1, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑧ $\mathtt{x} < 1 \vee \mathtt{y} \neq 3$

⑨ $\mathtt{x} \neq 1$





SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

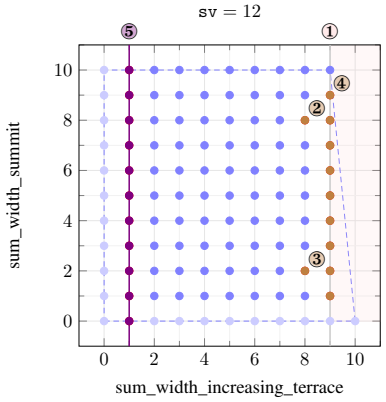
① $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

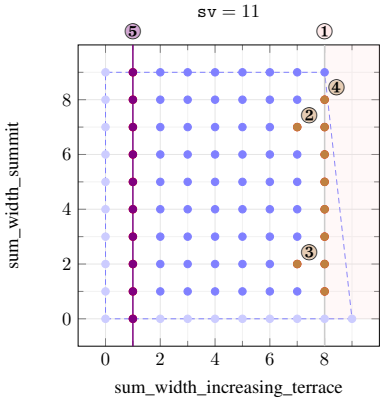
② $\bigvee \begin{pmatrix} x \neq (sv-2) * \min(1, \max(0, sv-3)) - 2, \\ y \neq (sv-2) * \min(1, \max(0, sv-2)) - 2, \\ sv < 5 \end{pmatrix}$

③ $x \neq (sv-2) * \min(1, \max(0, sv-3)) - 2 \vee y \neq 2 \vee sv < 5$

④ $\bigvee \begin{pmatrix} x \neq (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1 \end{pmatrix}$

⑤ $x \neq 1$





SUM_WIDTH_INCREASING_TERRACE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_VALLEY($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + 2 * \mathtt{sv} - 9$

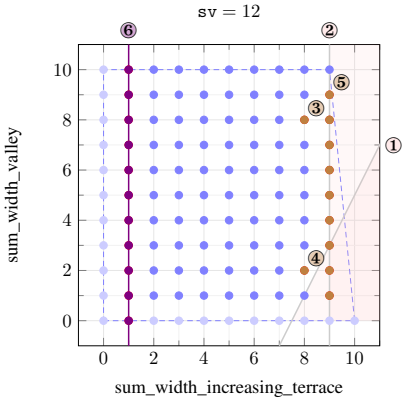
② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 3$

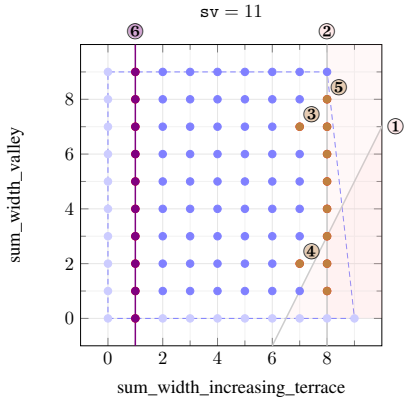
③ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2, \\ \mathtt{y} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 2, \\ \mathtt{sv} < 5 \end{pmatrix}$

④ $\mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 2 \vee \mathtt{y} \neq 2 \vee \mathtt{sv} < 5$

⑤ $\bigvee \begin{pmatrix} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1 \end{pmatrix}$

⑥ $\mathtt{x} \neq 1$





SUM_WIDTH_INCREASING_TERRACE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

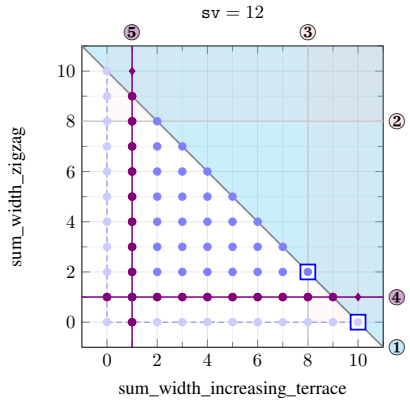
① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 2, 0)$ $(sv - 4, 2)$

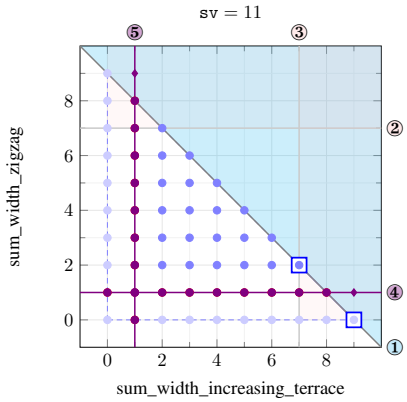
② $x > 0 \Rightarrow y \leq sv - 4$

③ $y > 0 \Rightarrow x \leq sv - 4$

④ $y \neq 1$

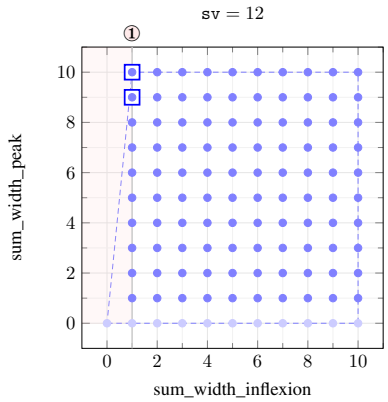
⑤ $x \neq 1$

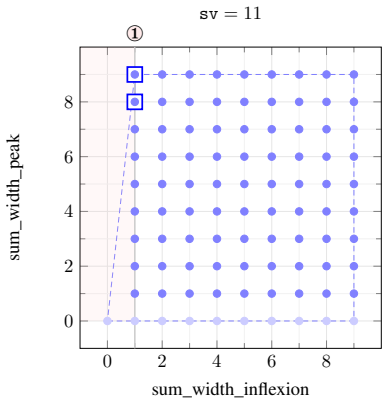




SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PEAK(y, VARIABLES) with sv = |VARIABLES|

① $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

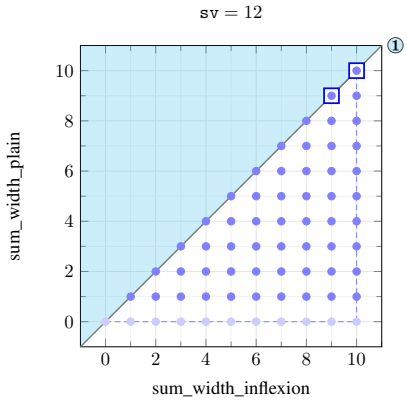


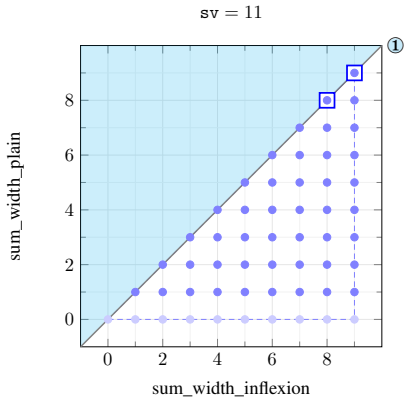


SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

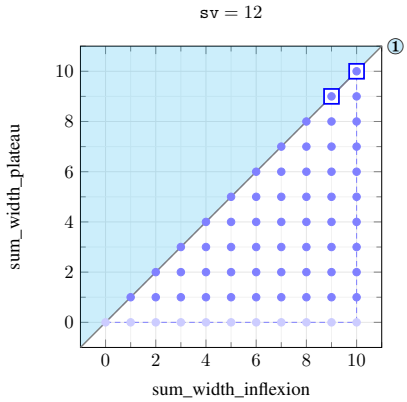


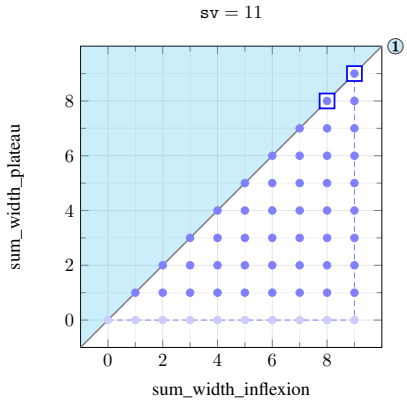


SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$



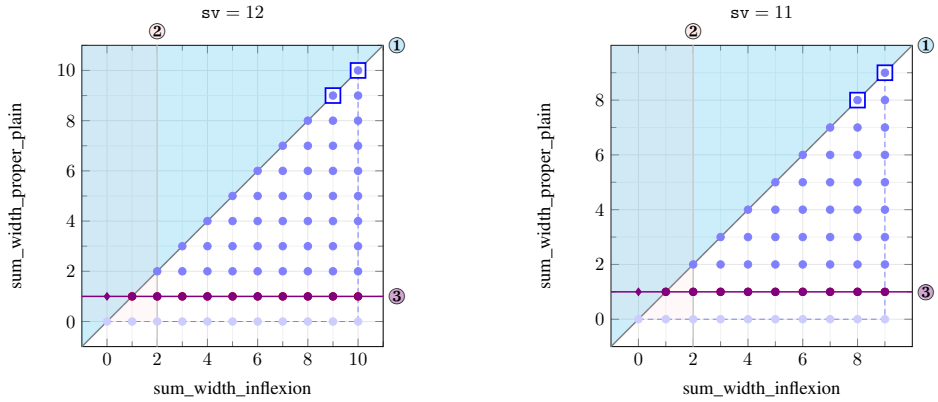


SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

② $y > 0 \Rightarrow x \geq 2$

③ $y \neq 1$

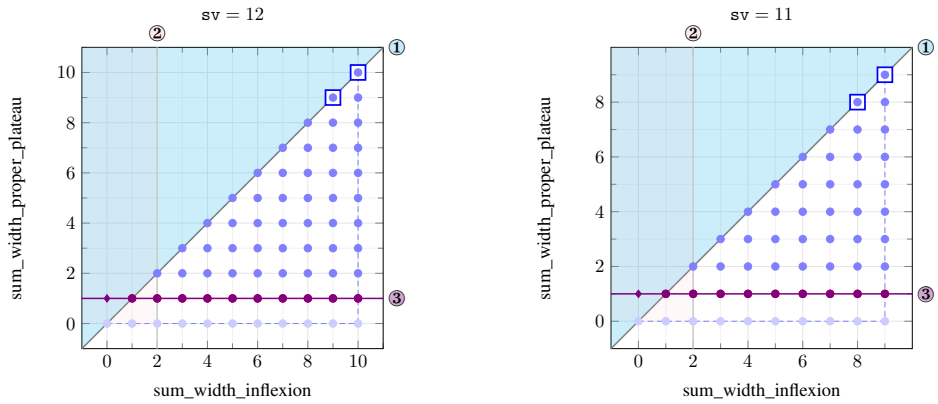


SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

② $y > 0 \Rightarrow x \geq 2$

③ $y \neq 1$



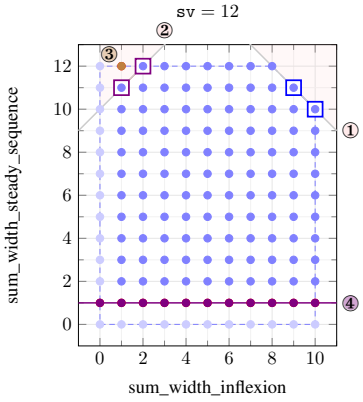
SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

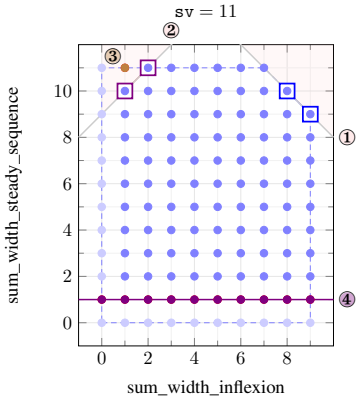
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 1)$

② $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- $sv \geq 6$: $(1, sv - 1)$ $(2, sv)$

③ $x \neq 1 \vee y \neq sv * \min(1, \max(0, sv - 1))$

④ $y \neq 1$





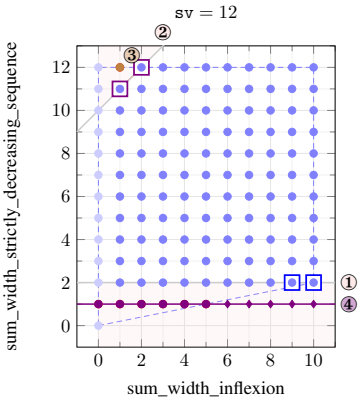
SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

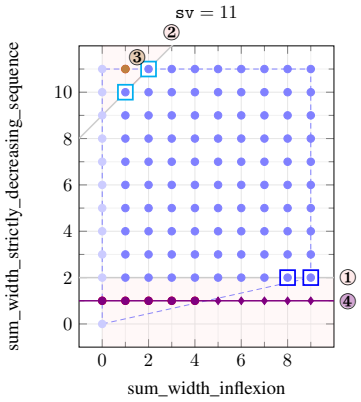
① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(1, sv - 1)$ $(2, sv)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(1, sv - 1)$ $(2, sv)$

③ $x \neq 1 \vee y \neq sv * \min(1, \max(0, sv - 1))$

④ $y \neq 1$





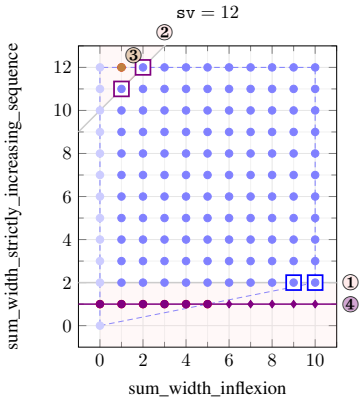
SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

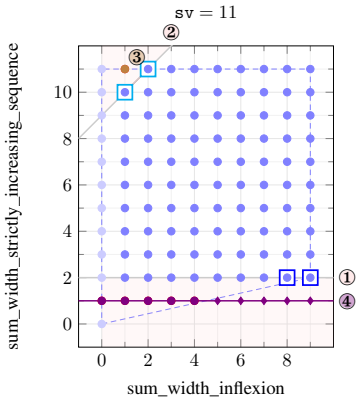
① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(1, sv - 1)$ $(2, sv)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(1, sv - 1)$ $(2, sv)$

③ $x \neq 1 \vee y \neq sv * \mathtt{min}(1, \mathtt{max}(0, sv - 1))$

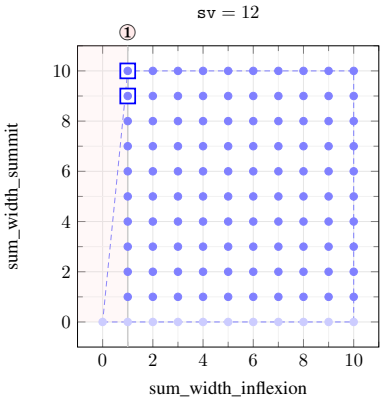
④ $y \neq 1$

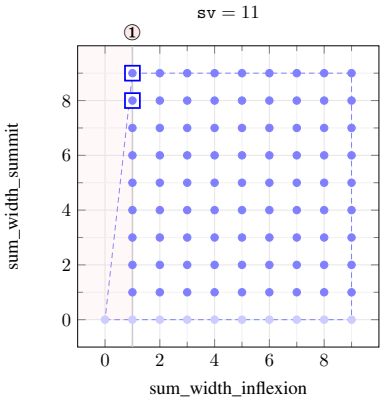




SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$

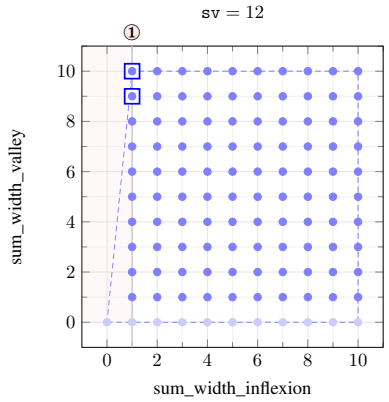


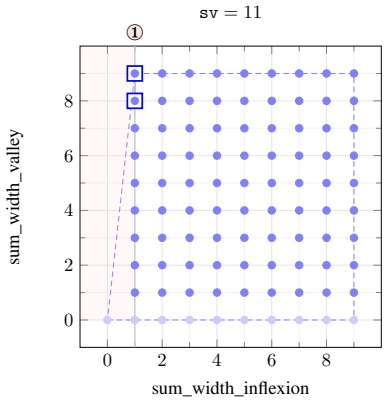


SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $y > 0 \Rightarrow x \geq 1$

- $sv \geq 5$: $(1, sv - 2)$ $(1, sv - 3)$





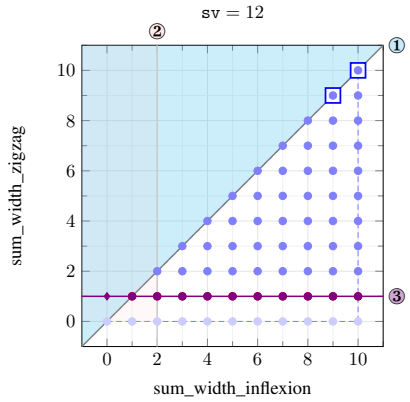
SUM_WIDTH_INFLEXION(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

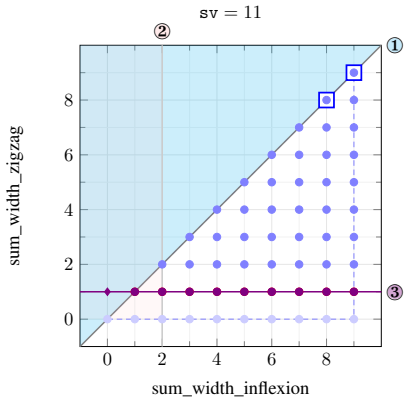
① $y \leq x$

- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$

③ $y \neq 1$

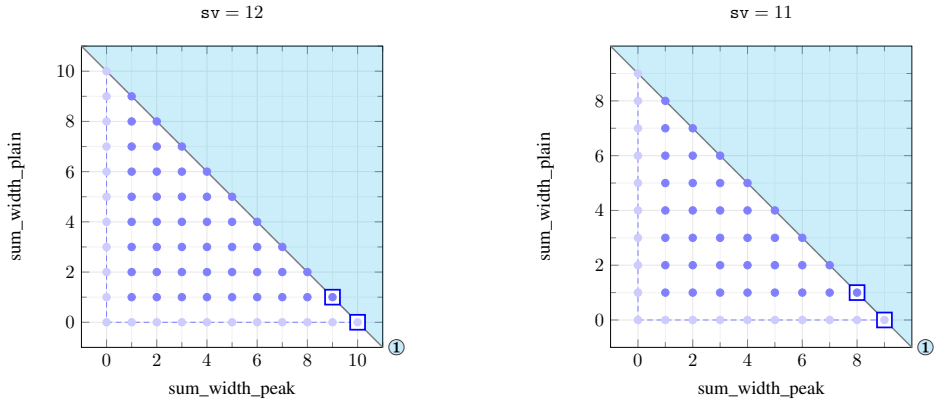




SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$

- $sv \geq 3$: $(sv - 2, 0)$ $(sv - 3, 1)$



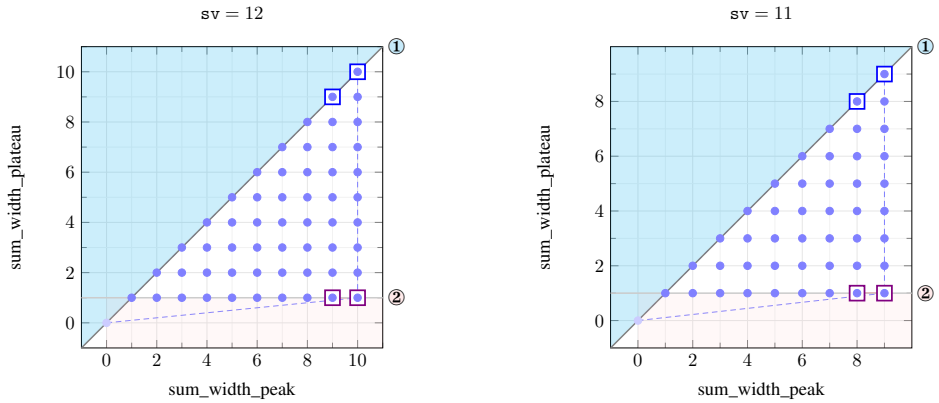
SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $x > 0 \Rightarrow y \geq 1$

- $sv \geq 4$: $(sv - 2, 1)$ $(sv - 3, 1)$

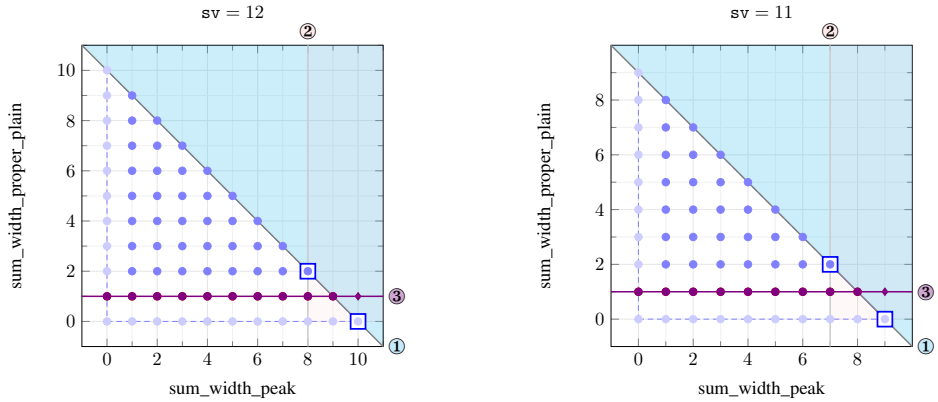


SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$

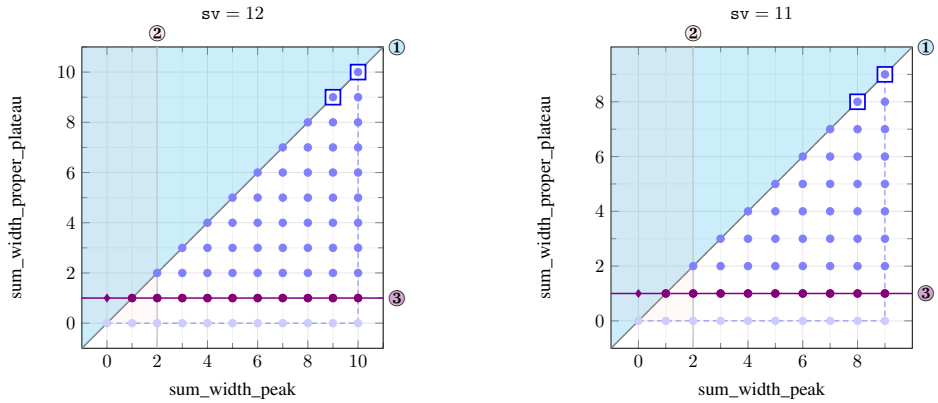


SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $y > 0 \Rightarrow x \geq 2$

③ $y \neq 1$



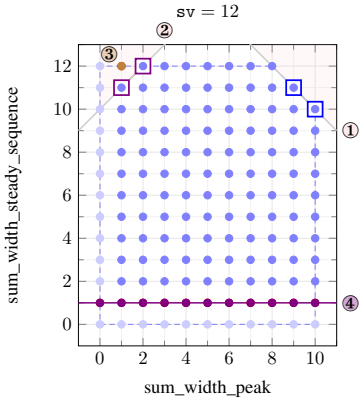
SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

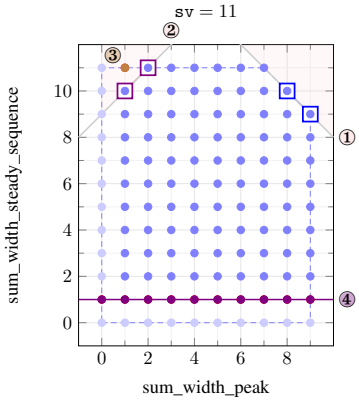
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 1)$

② $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$
- $sv \geq 6$: $(1, sv - 1)$ $(2, sv)$

③ $x \neq 1 \vee y \neq sv * \min(1, \max(0, sv - 1))$

④ $y \neq 1$





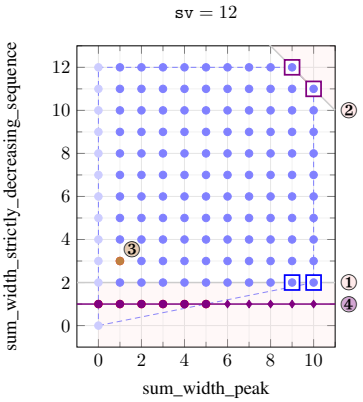
SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

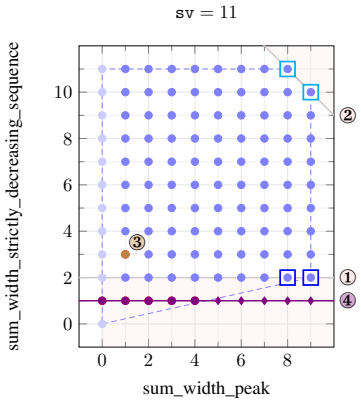
① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(sv - 2, sv - 1)$ $(sv - 3, sv)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(sv - 2, sv - 1)$ $(sv - 3, sv)$

③ $x \neq 1 \vee y \neq 3$

④ $y \neq 1$





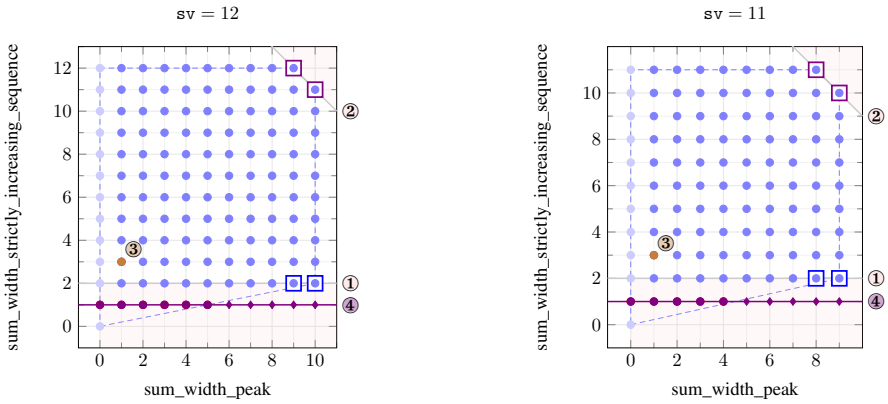
SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \Rightarrow y \geq 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 2)$

② $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv - 2, sv - 1)$ $(sv - 3, sv)$

③ $x \neq 1 \vee y \neq 3$

④ $y \neq 1$



SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$

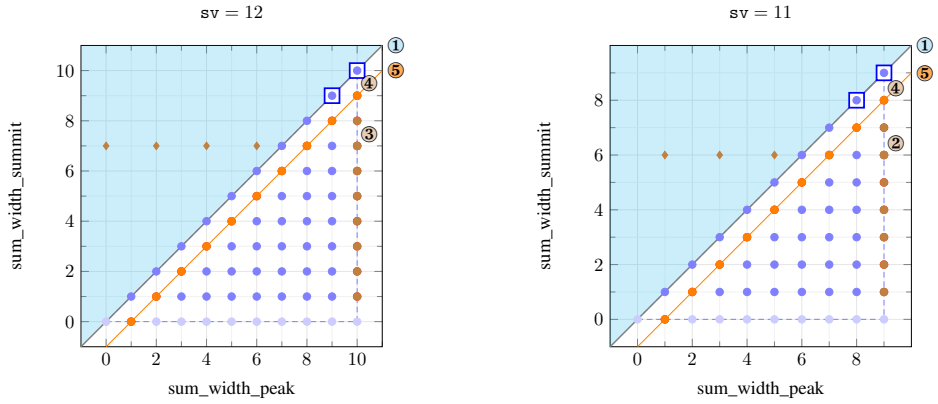
□ $sv \geq 3$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

② $\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-2)) - 3, \\ 0 = x \bmod 2, \\ 0 = sv \bmod 2, \\ sv < 6 \end{array} \right)$

③ $\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-2)) - 3, \\ 1 = x \bmod 2, \\ 1 = sv \bmod 2 \end{array} \right)$

④ $\bigvee \left(\begin{array}{l} x \neq (sv-2) * \min(1, \max(0, sv-2)), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-2)) - 1 \end{array} \right)$

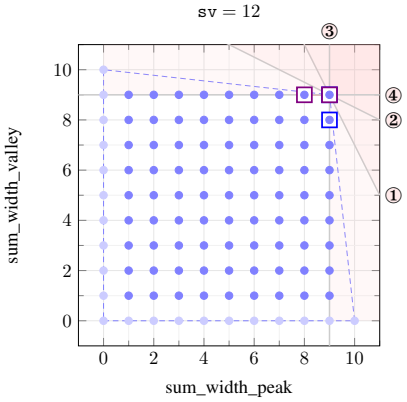
⑤ $y \neq x - 1$

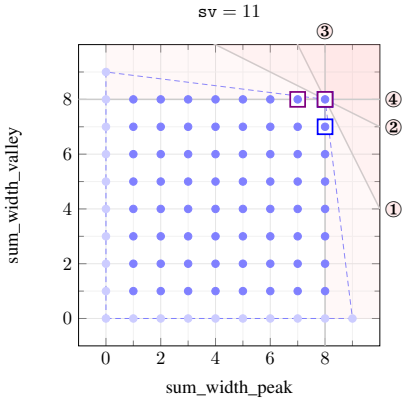


SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 3 * sv - 9$
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 3 * sv - 9$
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$
- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 3, sv - 4)$

④ $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 3$
- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 4, sv - 3)$





SUM_WIDTH_PEAK(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

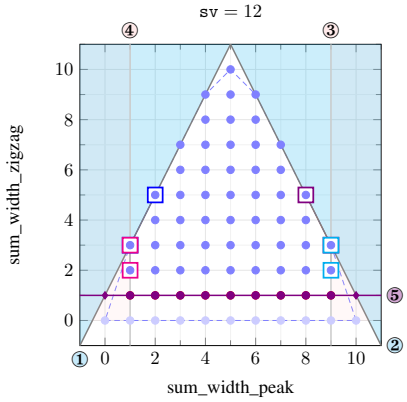
① $y \leq 2 * x + 1$
- $sv \geq 7$: $(1, 3)$ $(2, 5)$

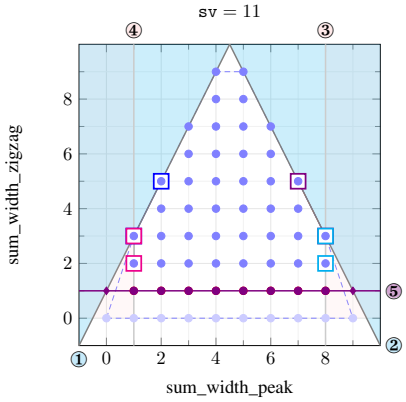
② $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3$
- $sv \geq 7$: $(sv - 3, 3)$ $(sv - 4, 5)$

③ $y > 0 \Rightarrow x \leq sv - 3$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 3, 3)$

④ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, 2)$ $(1, 3)$

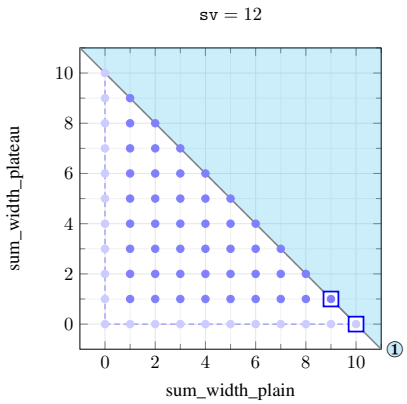
⑤ $y \neq 1$

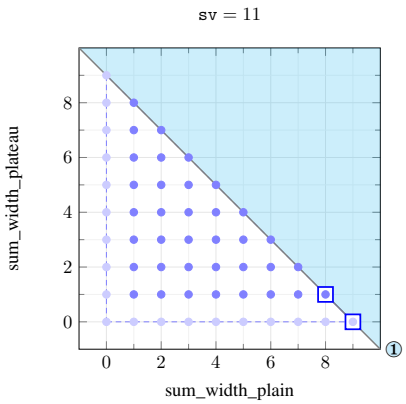




SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(sv - 2, 0)$ $(sv - 3, 1)$





SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

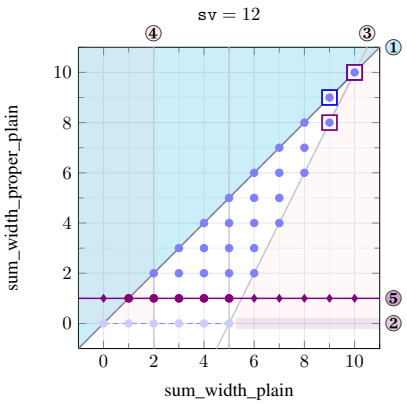
① $y \leq x$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

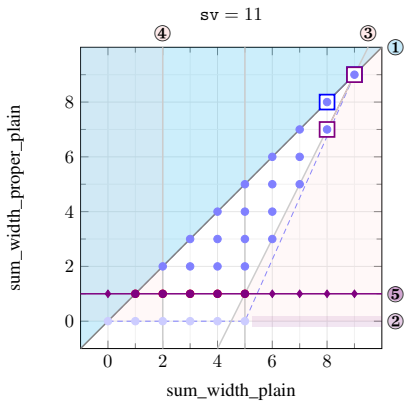
② $y = 0 \Rightarrow 2 * x \leq sv - 1 - (sv - 1) \bmod 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$
- $sv \geq 6$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 4)$

④ $y > 0 \Rightarrow x \geq 2$

⑤ $y \neq 1$



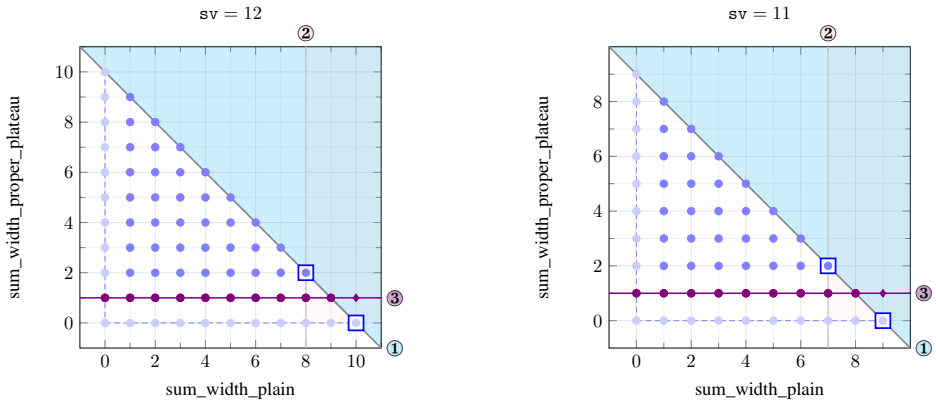


SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$



SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $y = 0 \Rightarrow 2 * x \leq sv - 1 - (sv - 1) \bmod 2$

② $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$

- $sv \geq 6$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 4)$

③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$

- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 1)$

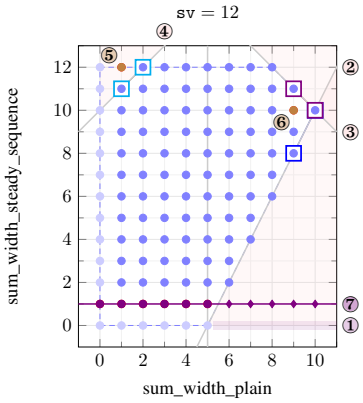
④ $x > 0 \wedge y > 0 \Rightarrow y \leq x + sv - 2$

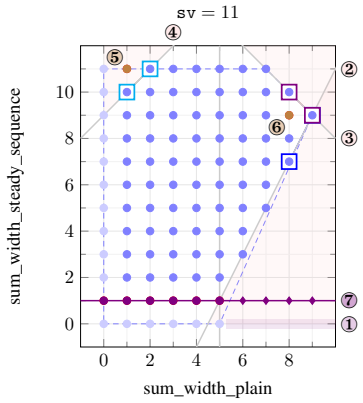
- $sv \geq 6$: $(1, sv - 1)$ $(2, sv)$

⑤ $x \neq 1 \vee y \neq sv * \min(1, \max(0, sv - 1))$

⑥ $\bigvee \left(\begin{array}{l} x \neq (sv - 2) * \min(1, \max(0, sv - 2)) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 2, \\ sv < 5 \end{array} \right)$

⑦ $y \neq 1$



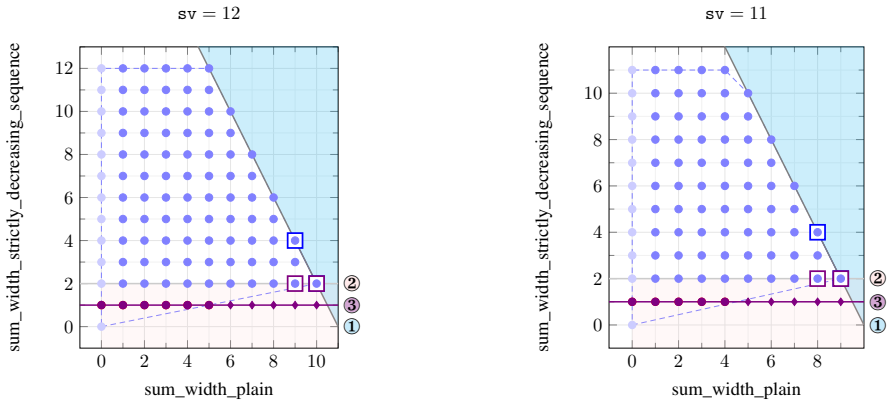


SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

③ $y \neq 1$

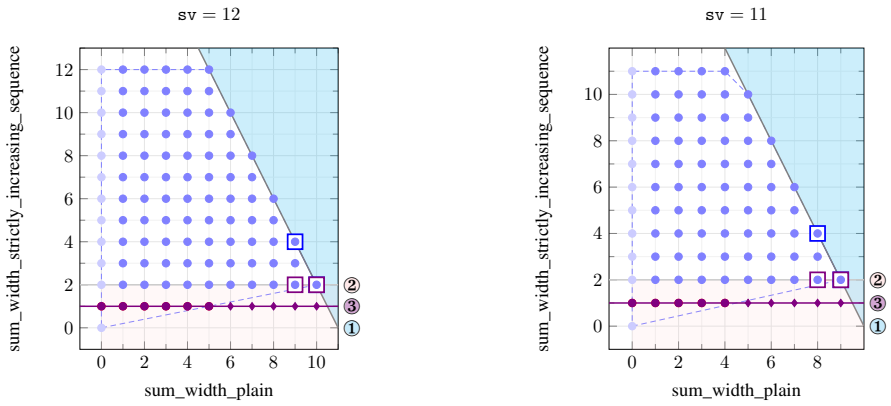


SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

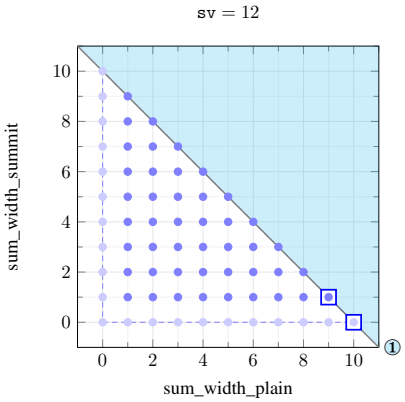
② $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

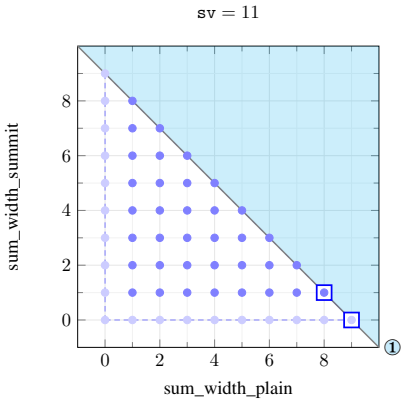
③ $y \neq 1$



SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(sv - 2, 0)$ $(sv - 3, 1)$

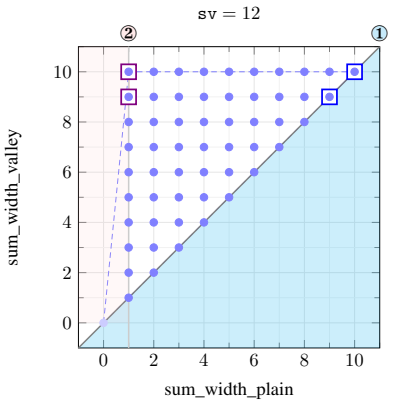


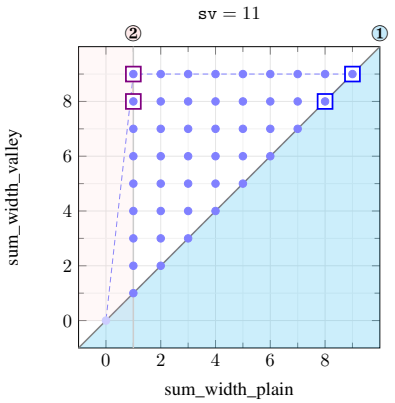


SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 3$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$





SUM_WIDTH_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- sv ≥ 7: (1, 3) (2, 5)

② $y < \max(0, sv - 2) \wedge sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- sv ≥ 6: (sv − 2, 0) (sv − 3, 2)

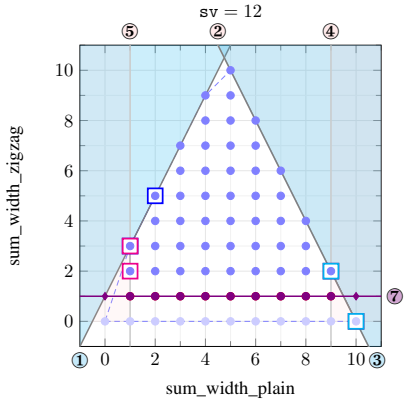
③ $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3 - (sv - 1) \bmod 2$
- (sv − 1) mod 2 = 1 ∧ sv ≥ 6: (sv − 2, 0) (sv − 3, 2)

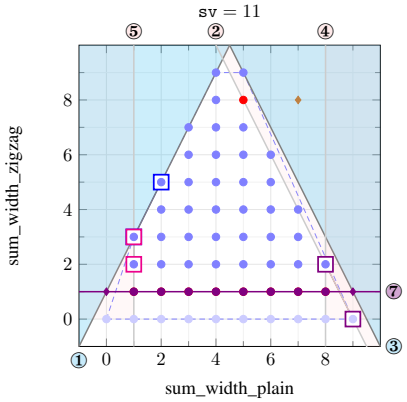
④ $y > 0 \Rightarrow x \leq sv - 3$

⑤ $y > 0 \Rightarrow x \geq 1$
- sv ≥ 5: (1, 2) (1, 3)

⑥ $\bigvee \left(\begin{array}{l} x \neq (sv - 2) * \min(1, \max(0, sv - 2)) - 2, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 0 = sv \bmod 2, \\ sv < 6 \end{array} \right)$

⑦ $y \neq 1$



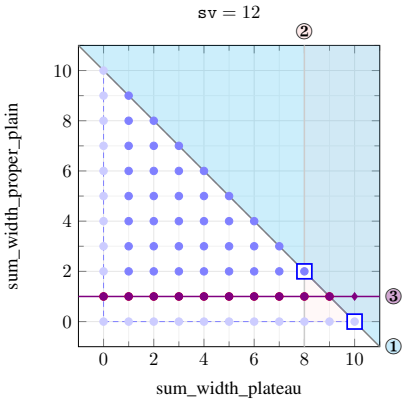


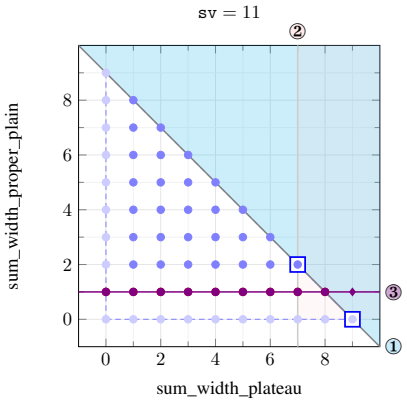
SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLAIN(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 4$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $y > 0 \Rightarrow x \leq sv - 4$

③ $y \neq 1$





SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

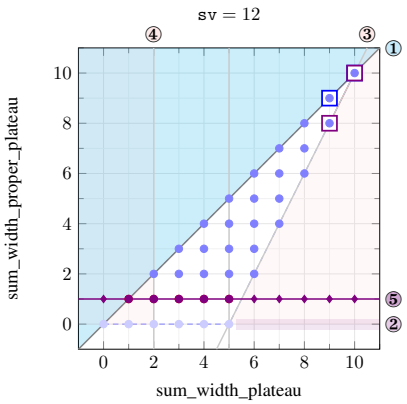
① $y \leq x$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

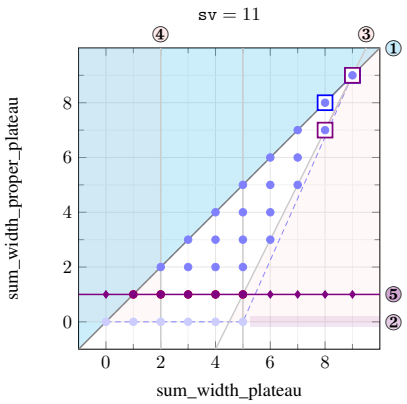
② $y = 0 \Rightarrow 2 * x \leq sv - 1 - (sv - 1) \bmod 2$

③ $x > 0 \wedge y > 0 \Rightarrow 2 * x \leq y + sv - 2$
- $sv \geq 6$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 4)$

④ $y > 0 \Rightarrow x \geq 2$

⑤ $y \neq 1$





SUM_WIDTH_PLATEAU$(\mathtt{x}, \mathtt{VARIABLES}) \wedge$
SUM_WIDTH_STEADY_SEQUENCE$(\mathtt{y}, \mathtt{VARIABLES})$ with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{y} = 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{sv} - 1 - (\mathtt{sv} - 1) \bmod 2$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow 2 * \mathtt{x} \leq \mathtt{y} + \mathtt{sv} - 2$

- $\mathtt{sv} \geq 6$: $(\mathtt{sv} - 2, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 4)$

③ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 4$

- $\mathtt{sv} \geq 5$: $(\mathtt{sv} - 2, \mathtt{sv} - 2)$ $(\mathtt{sv} - 3, \mathtt{sv} - 1)$

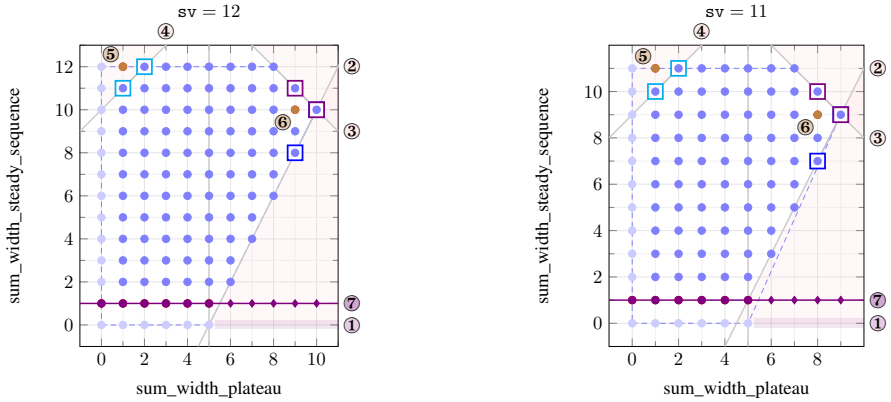
④ $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{y} \leq \mathtt{x} + \mathtt{sv} - 2$

- $\mathtt{sv} \geq 6$: $(1, \mathtt{sv} - 1)$ $(2, \mathtt{sv})$

⑤ $\mathtt{x} \neq 1 \vee \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1))$

⑥ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv} - 2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 2)) - 1, \\ \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv} - 1)) - 2, \\ \mathtt{sv} < 5 \end{array} \right)$

⑦ $\mathtt{y} \neq 1$

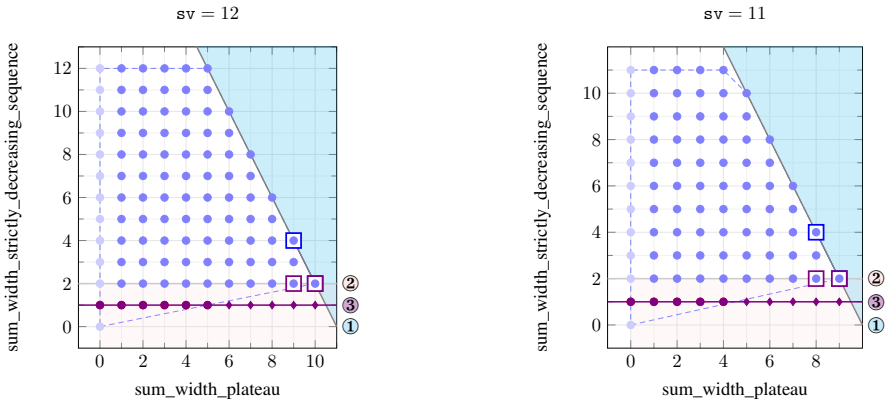


SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

③ $y \neq 1$

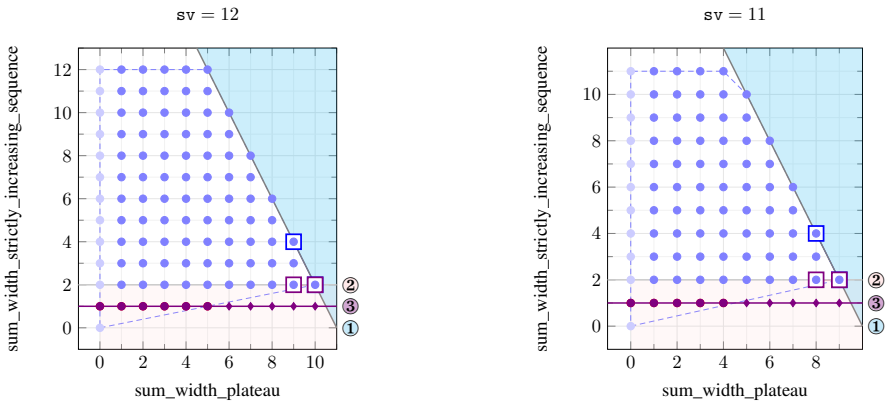


SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 4$: $(sv - 2, 2)$ $(sv - 3, 4)$

② $x > 0 \Rightarrow y \geq 2$
- $sv \geq 3$: $(sv - 2, 2)$ $(sv - 3, 2)$

③ $y \neq 1$



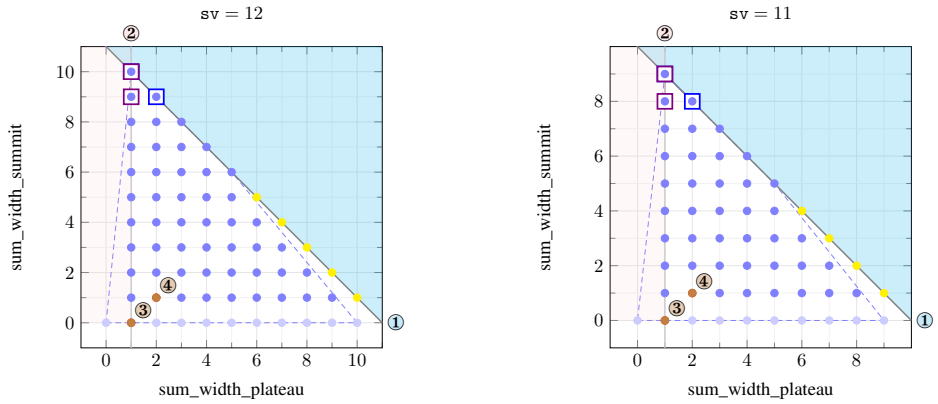
SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq sv - 1$
- $sv \geq 5$: $(1, sv - 2)$ $(2, sv - 3)$

② $y > 0 \Rightarrow x \geq 1$
- $sv \geq 4$: $(1, sv - 2)$ $(1, sv - 3)$

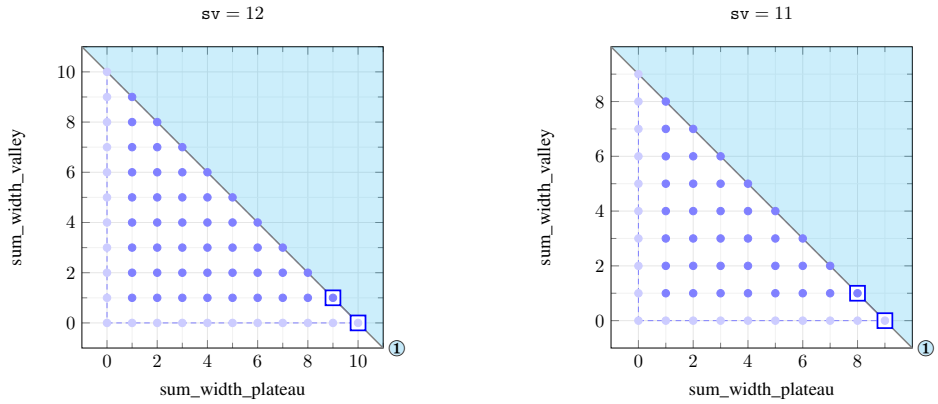
③ $x \neq 1 \vee y \neq 0$

④ $x \neq 2 \vee y \neq 1$



SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 3$: $(sv - 2, 0)$ $(sv - 3, 1)$



SUM_WIDTH_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq 2 * x + 1$
- $sv \geq 7$: (1, 3) (2, 5)

② $y < \max(0, sv - 2) \wedge sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 4$
- $sv \geq 6$: (sv − 2, 0) (sv − 3, 2)

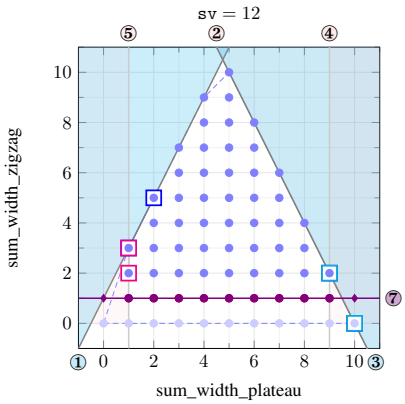
③ $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3 - (sv - 1) \bmod 2$
- $(sv - 1) \bmod 2 = 1 \wedge sv \geq 6$: (sv − 2, 0) (sv − 3, 2)

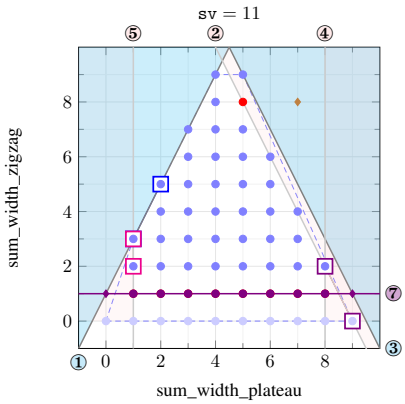
④ $y > 0 \Rightarrow x \leq sv - 3$

⑤ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: (1, 2) (1, 3)

⑥ $\bigvee \begin{pmatrix} x \neq (sv - 2) * \min(1, \max(0, sv - 2)) - 2, \\ y \neq (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 0 = sv \bmod 2, \\ sv < 6 \end{pmatrix}$

⑦ $y \neq 1$





SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_PROPER_PLATEAU(y, VARIABLES) with sv = |VARIABLES|

① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 6$: $(\mathtt{sv}-2, 0)$ $(\mathtt{sv}-4, 2)$

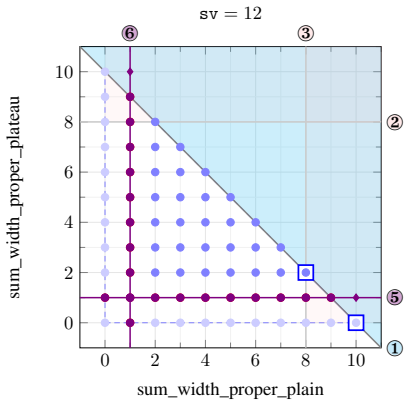
② $\mathtt{x} > 0 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 4$

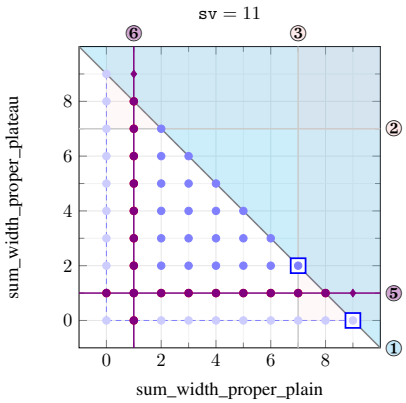
③ $\mathtt{y} > 0 \Rightarrow \mathtt{x} \leq \mathtt{sv} - 4$

④ $\mathtt{o}^{=} \geq \lfloor(\mathtt{x}+1)/2\rfloor + \lfloor(\mathtt{y}+1)/2\rfloor$

⑤ $\mathtt{y} \neq 1$

⑥ $\mathtt{x} \neq 1$





SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

② $x > 0 \Rightarrow y \geq 2$

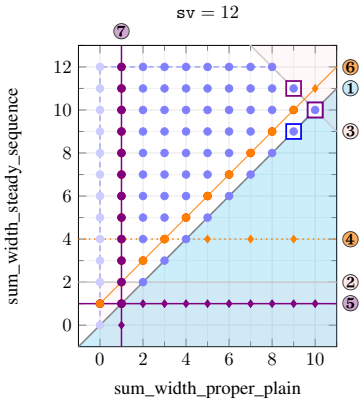
③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-1)$

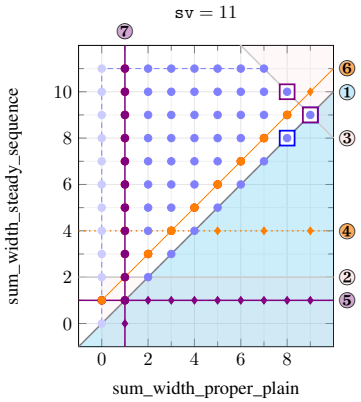
④ $y \neq 4 \vee 0 = x \bmod 2$

⑤ $y \neq 1$

⑥ $y \neq x + 1$

⑦ $x \neq 1$





SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $3 * x + 3 * y \leq 4 * sv - 2 - (4 * sv - 2) \bmod 3$

② $2 * x + y \leq 2 * sv - 2$

- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 4)$

③ $x + 2 * y \leq 2 * sv$

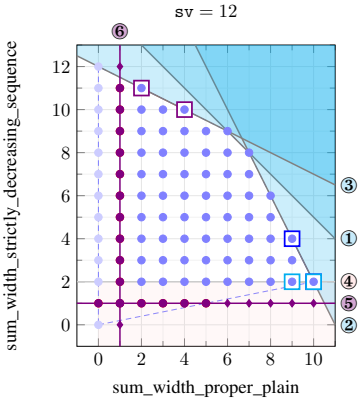
- $sv \geq 8$: $(2, sv - 1)$ $(4, sv - 2)$

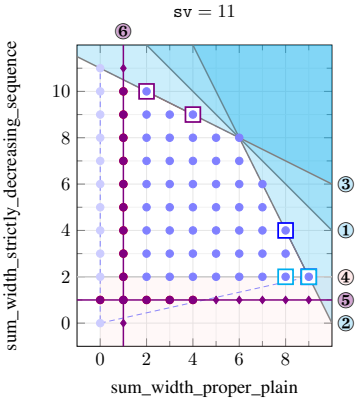
④ $x > 0 \Rightarrow y \geq 2$

- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 2)$

⑤ $y \neq 1$

⑥ $x \neq 1$





SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

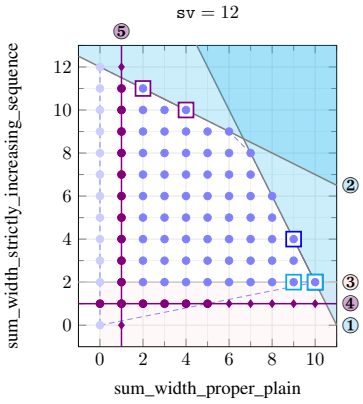
① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 4)$

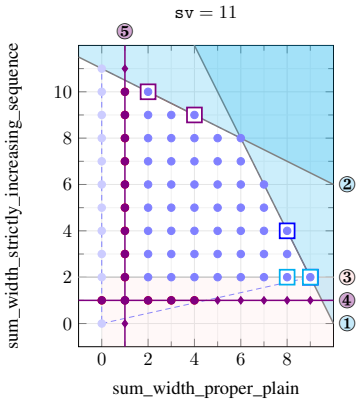
② $x + 2 * y \leq 2 * sv$
- $sv \geq 8$: $(2, sv - 1)$ $(4, sv - 2)$

③ $x > 0 \Rightarrow y \geq 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 2)$

④ $y \neq 1$

⑤ $x \neq 1$



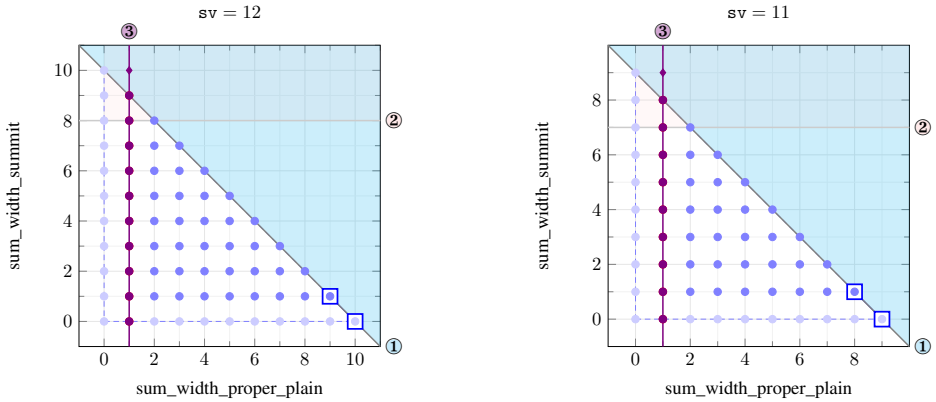


SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $x \neq 1$

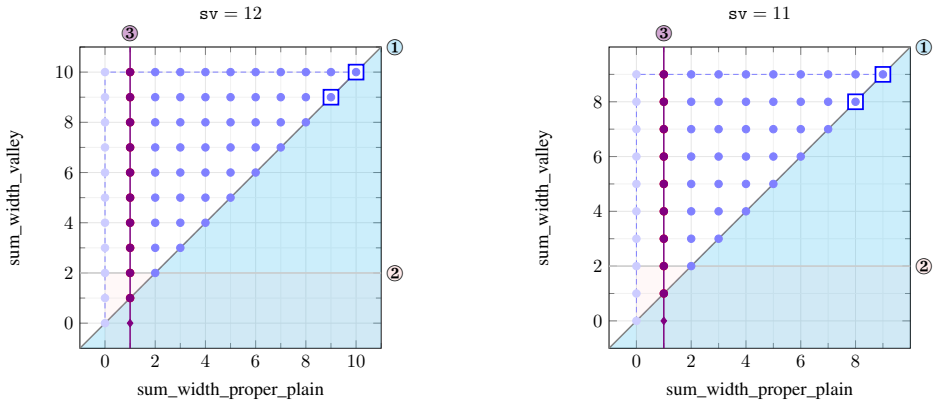


SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- $sv \geq 5$: $(sv - 2, sv - 2)$ $(sv - 3, sv - 3)$

② $x > 0 \Rightarrow y \geq 2$

③ $x \neq 1$



SUM_WIDTH_PROPER_PLAIN(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

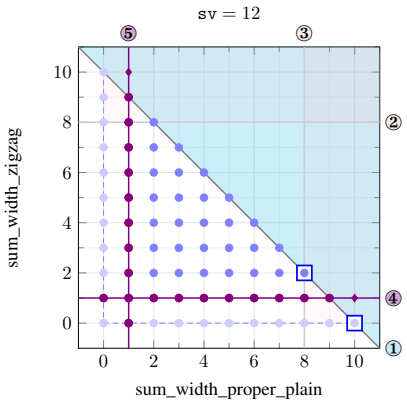
① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 2, 0)$ $(sv - 4, 2)$

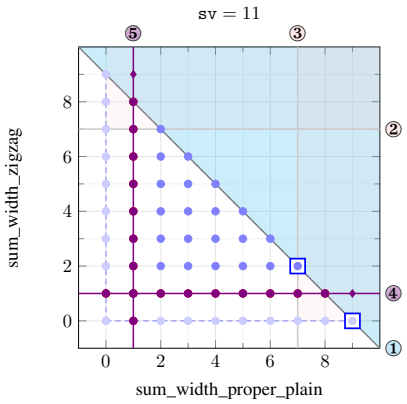
② $x > 0 \Rightarrow y \leq sv - 4$

③ $y > 0 \Rightarrow x \leq sv - 4$

④ $y \neq 1$

⑤ $x \neq 1$





SUM_WIDTH_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STEADY_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x \leq y$
- □ $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-3)$

② $x > 0 \Rightarrow y \geq 2$

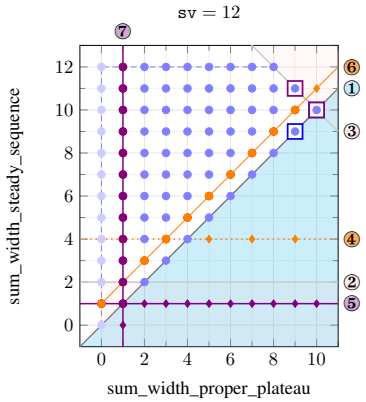
③ $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- □ $sv \geq 5$: $(sv-2, sv-2)$ $(sv-3, sv-1)$

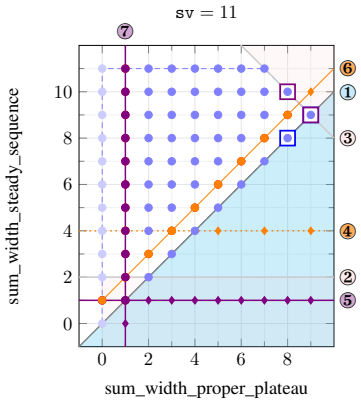
④ $y \neq 4 \vee 0 = x \bmod 2$

⑤ $y \neq 1$

⑥ $y \neq x + 1$

⑦ $x \neq 1$





SUM_WIDTH_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

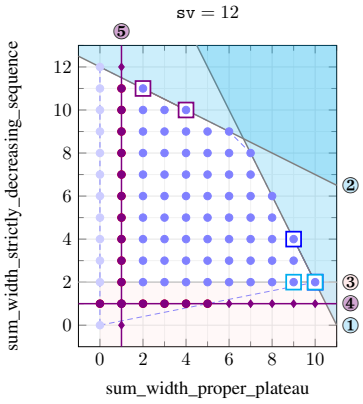
① $2 * x + y \leq 2 * sv - 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 4)$

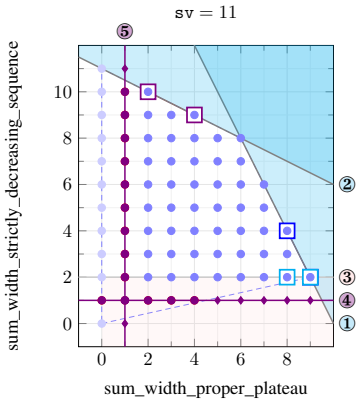
② $x + 2 * y \leq 2 * sv$
- $sv \geq 8$: $(2, sv - 1)$ $(4, sv - 2)$

③ $x > 0 \Rightarrow y \geq 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 2)$

④ $y \neq 1$

⑤ $x \neq 1$





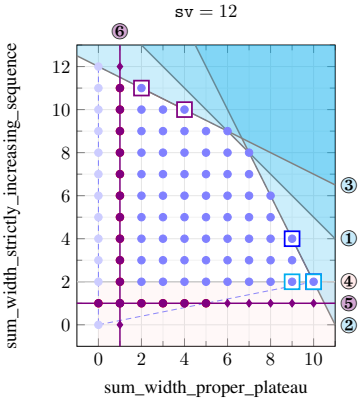
SUM_WIDTH_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

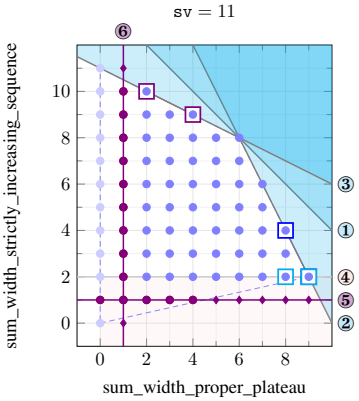
① $3 * x + 3 * y \leq 4 * sv - 2 - (4 * sv - 2) \bmod 3$
② $2 * x + y \leq 2 * sv - 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 4)$

③ $x + 2 * y \leq 2 * sv$
- $sv \geq 8$: $(2, sv - 1)$ $(4, sv - 2)$

④ $x > 0 \Rightarrow y \geq 2$
- $sv \geq 5$: $(sv - 2, 2)$ $(sv - 3, 2)$

⑤ $y \neq 1$
⑥ $x \neq 1$





SUM_WIDTH_PROPER_PLATEAU($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_SUMMIT($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{sv} > 1 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 2$
- $\mathtt{sv} \geq 3$: $(\mathtt{sv}-2, 0)$ $(0, \mathtt{sv}-2)$

② $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq \mathtt{sv} - 3$
- $\mathtt{sv} \geq 7$: $(\mathtt{sv}-4, 1)$ $(\mathtt{sv}-5, 2)$

③ $\mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 4 \vee \mathtt{y} \neq 4 \vee \mathtt{sv} < 7$

④ $\mathtt{x} < 4 \vee \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 4 \vee \mathtt{sv} < 7$

⑤ $\vee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 3, \\ \mathtt{y} < 3, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

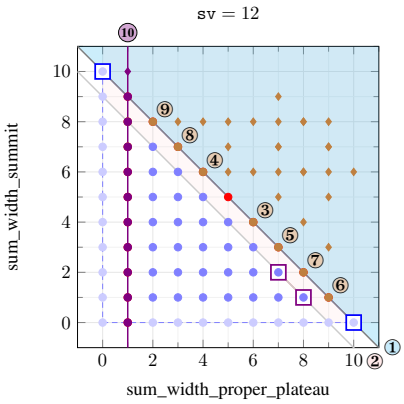
⑥ $\vee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ \mathtt{y} < 1, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

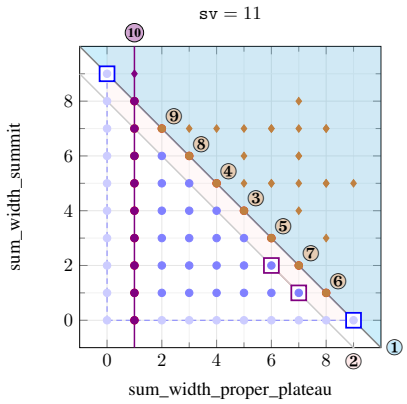
⑦ $\vee \left(\begin{array}{l} \mathtt{x} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 2, \\ \mathtt{y} < 2, \\ \mathtt{y} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 1, \\ 1 = \mathtt{y} \bmod 2 \end{array} \right)$

⑧ $\vee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 3, \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑨ $\vee \left(\begin{array}{l} \mathtt{y} \neq (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-2)) - 2, \\ \mathtt{x} < 1, \\ \mathtt{x} > (\mathtt{sv}-2) * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-3)) - 1 \end{array} \right)$

⑩ $\mathtt{x} \neq 1$



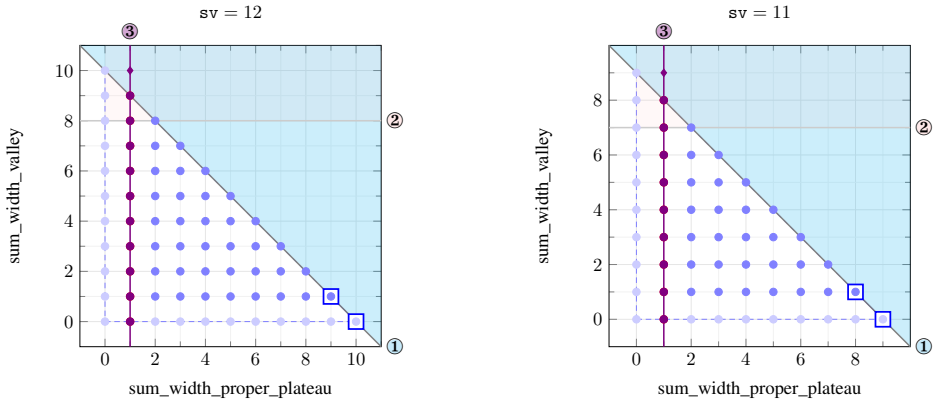


SUM_WIDTH_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 5$: $(sv - 2, 0)$ $(sv - 3, 1)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $x \neq 1$



SUM_WIDTH_PROPER_PLATEAU(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

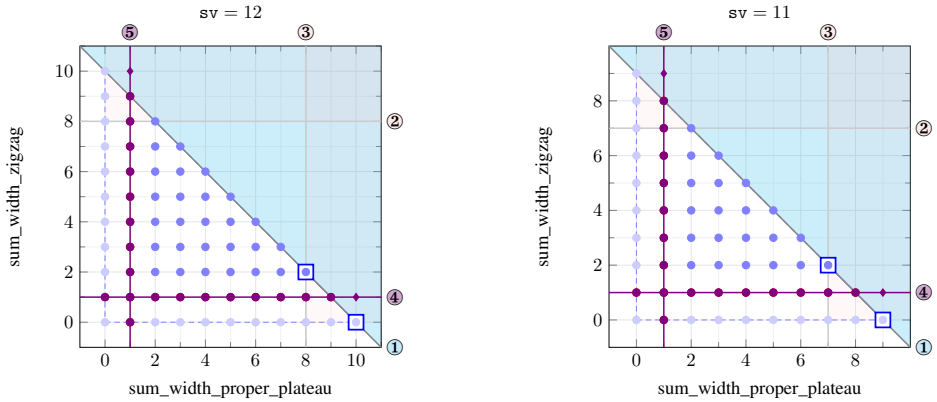
① $sv > 1 \Rightarrow x + y \leq sv - 2$
- $sv \geq 6$: $(sv - 2, 0)$ $(sv - 4, 2)$

② $x > 0 \Rightarrow y \leq sv - 4$

③ $y > 0 \Rightarrow x \leq sv - 4$

④ $y \neq 1$

⑤ $x \neq 1$



SUM_WIDTH_STEADY_SEQUENCE($\mathtt{x}$, VARIABLES) $\wedge$
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE($\mathtt{y}$, VARIABLES) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

① $\mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 2$

□ $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 2$: $(\mathtt{sv}, \mathtt{sv}-2)$ $(\mathtt{sv}-2, \mathtt{sv})$

② $\mathtt{x} = \mathtt{sv} \wedge \mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} > 1 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 3$

③ $\mathtt{x} = \mathtt{sv} - 2 \wedge \mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} > 1 \Rightarrow \mathtt{y} \leq \mathtt{sv} - 1$

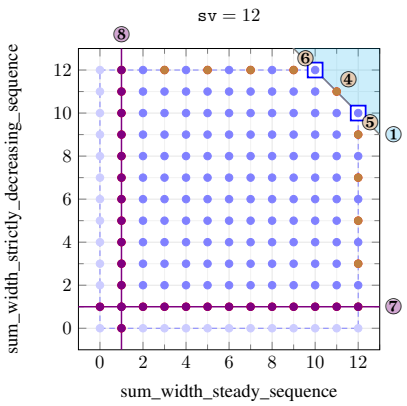
④ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 1, \\ 1 = \mathtt{sv} \bmod 2 \end{array} \right)$

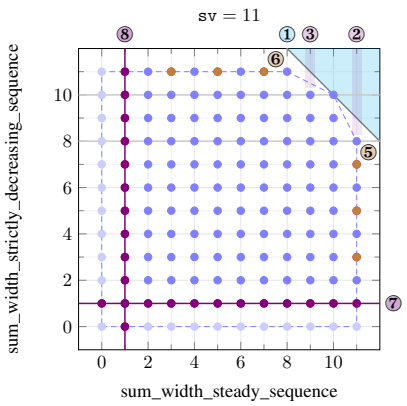
⑤ $\bigvee \left(\begin{array}{l} \mathtt{x} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)), \\ \mathtt{y} < 1, \\ \mathtt{y} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 3, \\ 0 = \mathtt{y} \bmod 2 \end{array} \right)$

⑥ $\bigvee \left(\begin{array}{l} \mathtt{y} \neq \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)), \\ \mathtt{x} < 1, \\ \mathtt{x} > \mathtt{sv} * \mathtt{min}(1, \mathtt{max}(0, \mathtt{sv}-1)) - 3, \\ 0 = \mathtt{x} \bmod 2 \end{array} \right)$

⑦ $\mathtt{y} \neq 1$

⑧ $\mathtt{x} \neq 1$





SUM_WIDTH_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq 2 * sv - 2$

□ $sv \bmod 2 = 0 \wedge sv \geq 2$: $(sv, sv-2)$ $(sv-2, sv)$

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $x = sv - 2 \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 1$

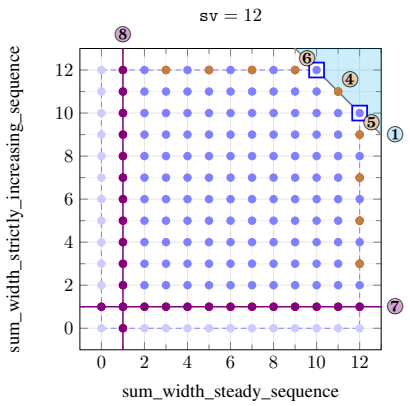
④ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv-1)) - 1, \\ y \neq sv * \min(1, \max(0, sv-1)) - 1, \\ 1 = sv \bmod 2 \end{array} \right)$

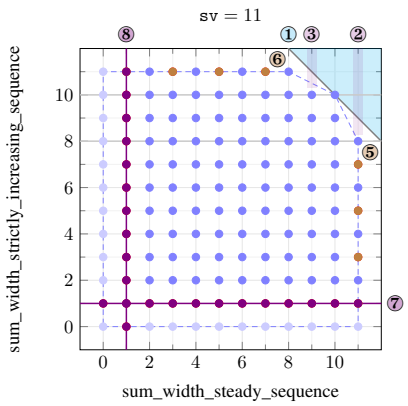
⑤ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv-1)), \\ y < 1, \\ y > sv * \min(1, \max(0, sv-1)) - 3, \\ 0 = y \bmod 2 \end{array} \right)$

⑥ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv-1)), \\ x < 1, \\ x > sv * \min(1, \max(0, sv-1)) - 3, \\ 0 = x \bmod 2 \end{array} \right)$

⑦ $y \neq 1$

⑧ $x \neq 1$





SUM_WIDTH_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \land y > 0 \Rightarrow 2 * x + y \leq 3 * sv - 6$

② $x > 0 \land y > 0 \Rightarrow x + y \leq 2 * sv - 5$

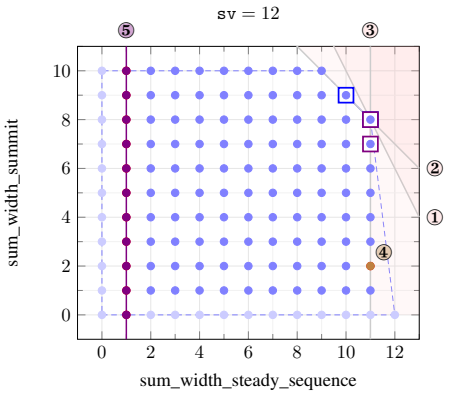
- $sv \geq 7$: $(sv - 1, sv - 4)$ $(sv - 2, sv - 3)$

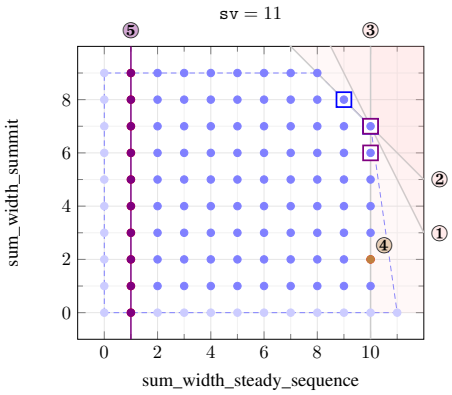
③ $x > 0 \land y > 0 \Rightarrow x \leq sv - 1$

- $sv \geq 8$: $(sv - 1, sv - 4)$ $(sv - 1, sv - 5)$

④ $x \neq sv * \min(1, \max(0, sv - 1)) - 1 \lor y \neq 2$

⑤ $x \neq 1$





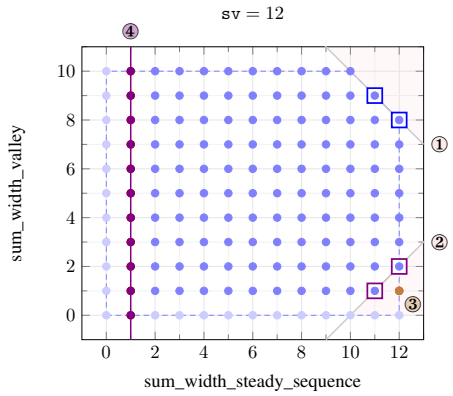
SUM_WIDTH_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

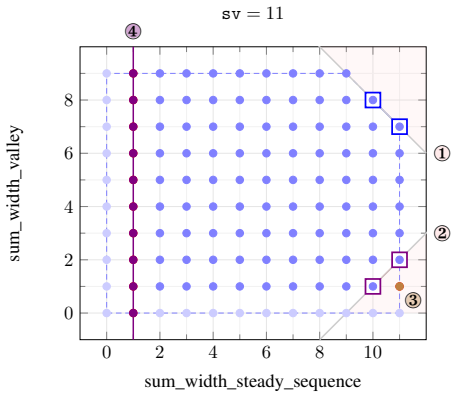
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 4$
- $sv \geq 6$: $(sv, sv - 4)$ $(sv - 1, sv - 3)$

② $x > 0 \wedge y > 0 \Rightarrow x \leq y + sv - 2$
- $sv \geq 6$: $(sv, 2)$ $(sv - 1, 1)$

③ $x \neq sv * \min(1, \max(0, sv - 1)) \vee y \neq 1$

④ $x \neq 1$





SUM_WIDTH_STEADY_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

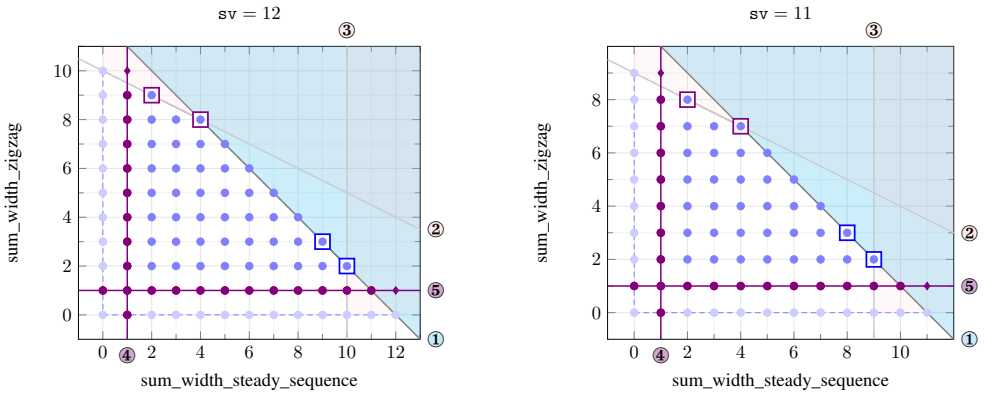
① $x + y \leq sv$
- $sv \geq 7$: $(sv - 2, 2)$ $(sv - 3, 3)$

② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 2 * sv - 4$
- $sv \geq 6$: $(2, sv - 3)$ $(4, sv - 4)$

③ $y > 0 \Rightarrow x \leq sv - 2$

④ $x \neq 1$

⑤ $y \neq 1$



SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(y, VARIABLES) with sv = |VARIABLES|

① $x + y \leq 2 * sv - 2$

□ $sv \bmod 2 = 0 \wedge sv \geq 2$: (sv, sv − 2) (sv − 2, sv)

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $x = sv - 2 \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 1$

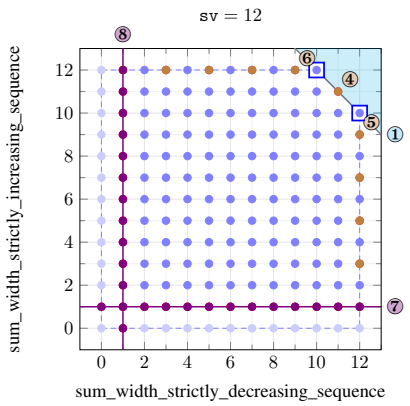
④ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ y \neq sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2 \end{array} \right)$

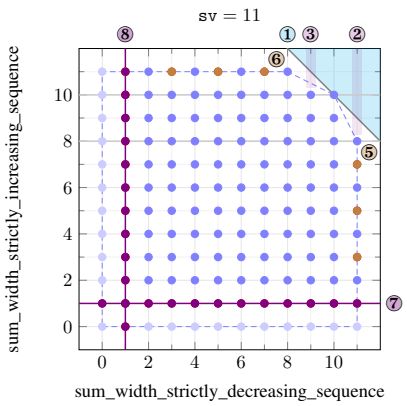
⑤ $\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)), \\ y < 1, \\ y > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = y \bmod 2 \end{array} \right)$

⑥ $\bigvee \left(\begin{array}{l} y \neq sv * \min(1, \max(0, sv - 1)), \\ x < 1, \\ x > sv * \min(1, \max(0, sv - 1)) - 3, \\ 0 = x \bmod 2 \end{array} \right)$

⑦ $y \neq 1$

⑧ $x \neq 1$





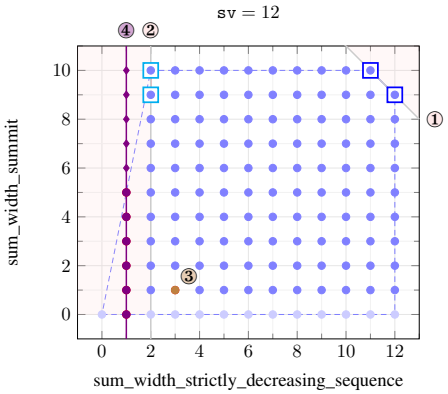
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

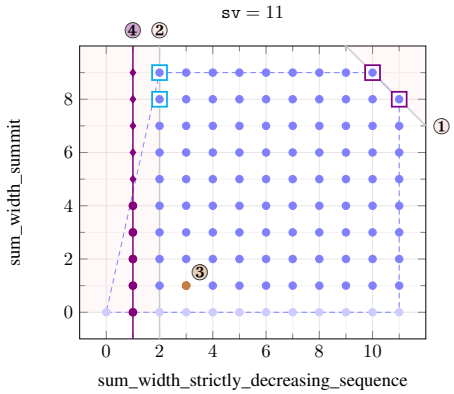
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \bmod 2 = 0 \wedge sv \geq 4$: $(sv, sv-3)$ $(sv-1, sv-2)$
- $sv \bmod 2 = 1 \wedge sv \geq 5$: $(sv, sv-3)$ $(sv-1, sv-2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 3$: $(2, sv-2)$ $(2, sv-3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





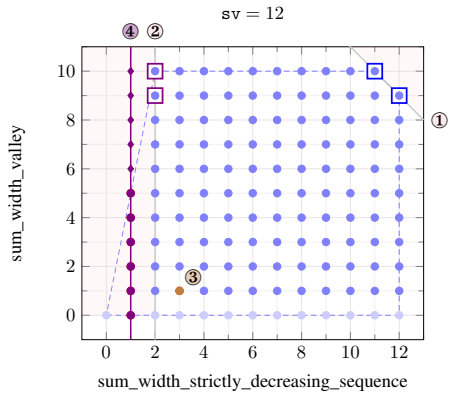
SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

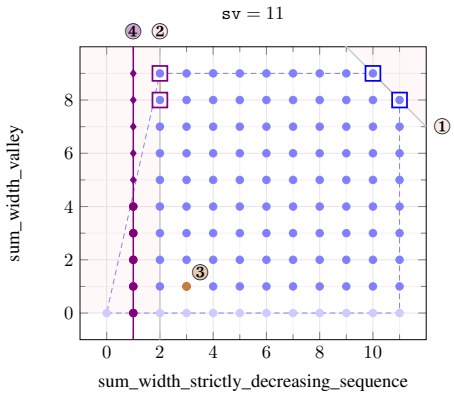
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 4$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





SUM_WIDTH_STRICTLY_DECREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- sv mod 2 = 0 ∧ sv ≥ 3: (sv − 2, sv − 2) (0, 0)
- sv mod 2 = 1 ∧ sv ≥ 6: (sv − 3, sv − 3) (2, 2)

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 2$

④ $$\bigvee \left(\begin{array}{l} y \neq (sv - 2) * \min(1, \max(0, sv - 3)), \\ x < 3, \\ x > sv * \min(1, \max(0, sv - 1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = x \bmod 2 \end{array} \right)$$

⑤ $x \neq 5 \vee y < 4$

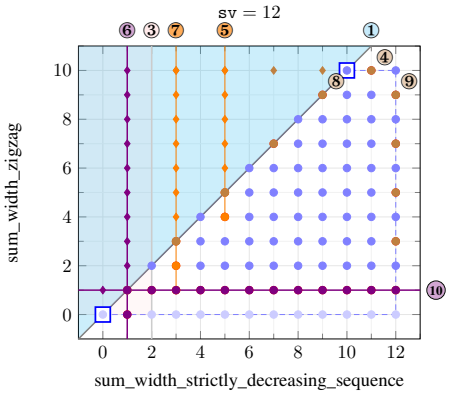
⑥ $x \neq 1$

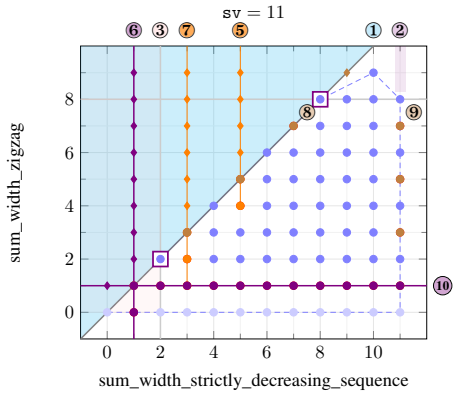
⑦ $x \neq 3 \vee y < 1$

⑧ $y \neq x \vee 0 = x \bmod 2$

⑨ $$\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv - 1)), \\ y < 1, \\ y > (sv - 2) * \min(1, \max(0, sv - 3)) - 1, \\ 0 = y \bmod 2 \end{array} \right)$$

⑩ $y \neq 1$





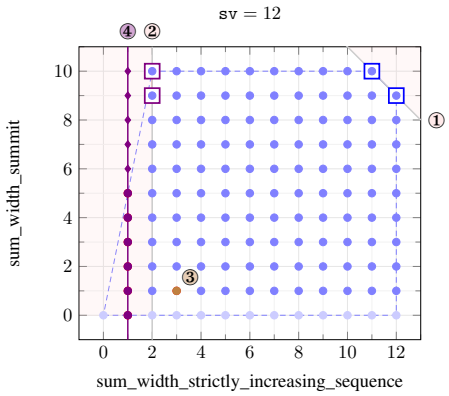
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_SUMMIT(y, VARIABLES) with sv = |VARIABLES|

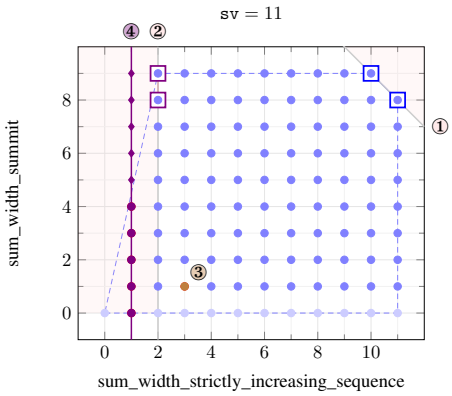
① $x > 0 \wedge y > 0 \Rightarrow x + y \leq 2 * sv - 3$
- $sv \geq 3$: $(sv, sv - 3)$ $(sv - 1, sv - 2)$

② $y > 0 \Rightarrow x \geq 2$
- $sv \geq 5$: $(2, sv - 2)$ $(2, sv - 3)$

③ $x \neq 3 \vee y \neq 1$

④ $x \neq 1$





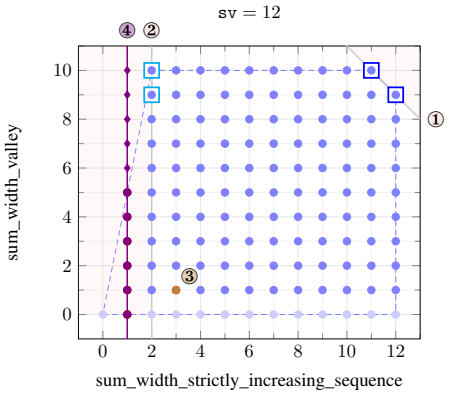
SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE($\mathtt{x}, \mathtt{VARIABLES}$) $\wedge$
SUM_WIDTH_VALLEY($\mathtt{y}, \mathtt{VARIABLES}$) with $\mathtt{sv} = |\mathtt{VARIABLES}|$

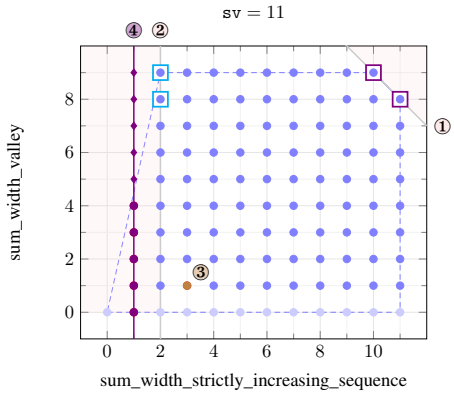
① $\mathtt{x} > 0 \wedge \mathtt{y} > 0 \Rightarrow \mathtt{x} + \mathtt{y} \leq 2 * \mathtt{sv} - 3$
- $\mathtt{sv} \bmod 2 = 0 \wedge \mathtt{sv} \geq 4$: $(\mathtt{sv}, \mathtt{sv}-3)$ $(\mathtt{sv}-1, \mathtt{sv}-2)$
- $\mathtt{sv} \bmod 2 = 1 \wedge \mathtt{sv} \geq 5$: $(\mathtt{sv}, \mathtt{sv}-3)$ $(\mathtt{sv}-1, \mathtt{sv}-2)$

② $\mathtt{y} > 0 \Rightarrow \mathtt{x} \geq 2$
- $\mathtt{sv} \geq 3$: $(2, \mathtt{sv}-2)$ $(2, \mathtt{sv}-3)$

③ $\mathtt{x} \neq 3 \vee \mathtt{y} \neq 1$

④ $\mathtt{x} \neq 1$





SUM_WIDTH_STRICTLY_INCREASING_SEQUENCE(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

① $y \leq x$
- sv mod 2 = 0 ∧ sv ≥ 3: (sv − 2, sv − 2) (0, 0)
- sv mod 2 = 1 ∧ sv ≥ 6: (sv − 3, sv − 3) (2, 2)

② $x = sv \wedge sv \bmod 2 = 1 \wedge sv > 1 \Rightarrow y \leq sv - 3$

③ $y > 0 \Rightarrow x \geq 2$

④ $$\bigvee \left(\begin{array}{l} y \neq (sv-2) * \min(1, \max(0, sv-3)), \\ x < 3, \\ x > sv * \min(1, \max(0, sv-1)) - 1, \\ 1 = sv \bmod 2, \\ 0 = x \bmod 2 \end{array} \right)$$

⑤ $x \neq 5 \vee y < 4$

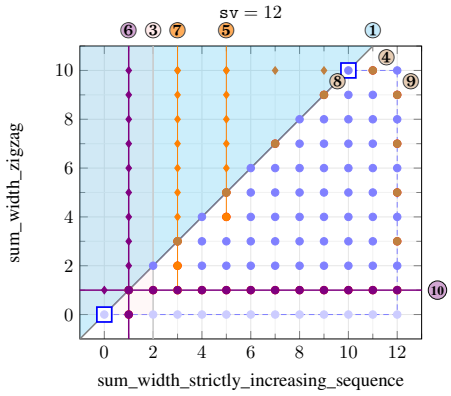
⑥ $x \neq 1$

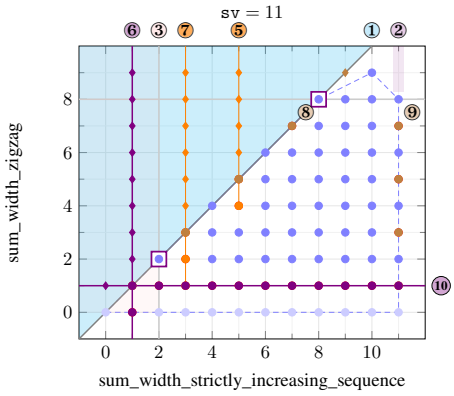
⑦ $x \neq 3 \vee y < 1$

⑧ $y \neq x \vee 0 = x \bmod 2$

⑨ $$\bigvee \left(\begin{array}{l} x \neq sv * \min(1, \max(0, sv-1)), \\ y < 1, \\ y > (sv-2) * \min(1, \max(0, sv-3)) - 1, \\ 0 = y \bmod 2 \end{array} \right)$$

⑩ $y \neq 1$





SUM_WIDTH_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_VALLEY(y, VARIABLES) with sv = |VARIABLES|

① $x > 0 \wedge y > 0 \Rightarrow 2 * x + y \leq 3 * sv - 9$

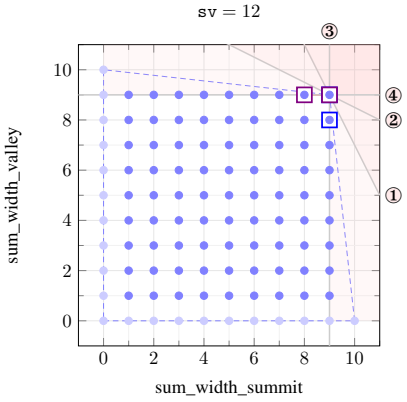
② $x > 0 \wedge y > 0 \Rightarrow x + 2 * y \leq 3 * sv - 9$

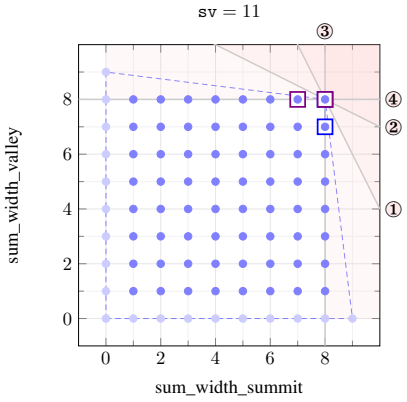
③ $x > 0 \wedge y > 0 \Rightarrow x \leq sv - 3$

- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 3, sv - 4)$

④ $x > 0 \wedge y > 0 \Rightarrow y \leq sv - 3$

- $sv \geq 4$: $(sv - 3, sv - 3)$ $(sv - 4, sv - 3)$





SUM_WIDTH_SUMMIT(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

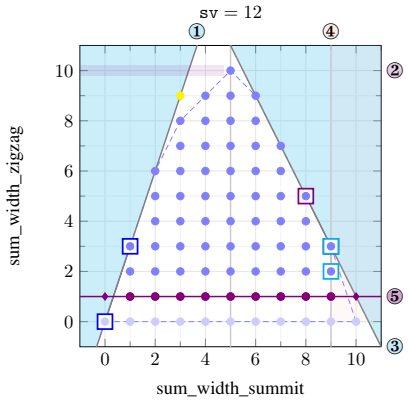
① $y \leq 3 * x$
- sv ≥ 5: (0, 0) (1, 3)

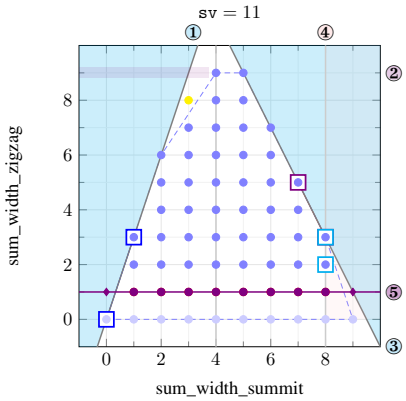
② $y = \max(0, sv - 2) \wedge sv > 1 \Rightarrow 2 * x \geq sv - 2 - sv \bmod 2$

③ $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3$
- sv ≥ 7: (sv − 3, 3) (sv − 4, 5)

④ $y > 0 \Rightarrow x \leq sv - 3$
- sv ≥ 5: (sv − 3, 2) (sv − 3, 3)

⑤ $y \neq 1$





SUM_WIDTH_VALLEY(x, VARIABLES) ∧
SUM_WIDTH_ZIGZAG(y, VARIABLES) with sv = |VARIABLES|

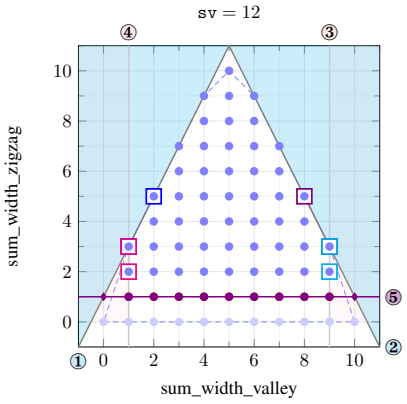
① $y \leq 2 * x + 1$
- $sv \geq 7$: $(1, 3)$ $(2, 5)$

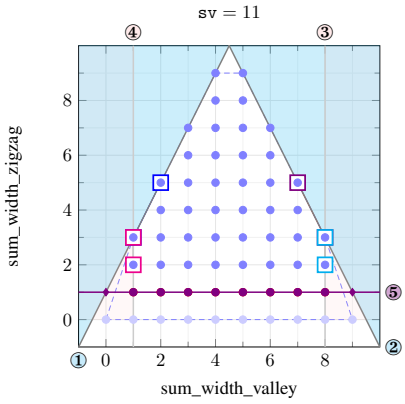
② $sv > 1 \Rightarrow 2 * x + y \leq 2 * sv - 3$
- $sv \geq 7$: $(sv - 3, 3)$ $(sv - 4, 5)$

③ $y > 0 \Rightarrow x \leq sv - 3$
- $sv \geq 5$: $(sv - 3, 2)$ $(sv - 3, 3)$

④ $y > 0 \Rightarrow x \geq 1$
- $sv \geq 5$: $(1, 2)$ $(1, 3)$

⑤ $y \neq 1$





Appendix A

Electronic Constraint Catalogue

The file attached to each constraint is available from a link located on the leftmost upper corner of the first page of each constraint provided you are using Adobe Acrobat Reader.

- The file `eval.pl`, containing all utilities, is available from this link  .
- The file `src.pl`, that allows one to download all time-series constraints in SICStus, is available from the link  .
- The MiniZinc implementation of the time-series constraints is available from this link  . It corresponds to a reformulation of the corresponding automata constraints.
- Besides the Prolog checkers available in the corresponding attached Prolog files, we also have Java and C synthesised code for the checkers from the following links:
  - `names.h`  , `names.c`  , `checker.h`  , `checker.c`  .
  - `TimeSeriesNames.java`  , `TimeSeriesChecker.java`  .
- Finally, the machine-readable version `invariants.pl` of the database of parameterised invariants described in Chapter 5 is available from the following link  .

Bibliography

[1] R. Alur, D. Fisman, and M. Raghothaman. Regular programming for quantitative properties of data streams. In P. Thiemann, editor, *Programming Languages and Systems - 25th European Symposium on Programming, ESOP*, volume 9632 of *LNCS*, pages 15–40. Springer, 2016. i

[2] E. Arafailova, N. Beldiceanu, M. Carlsson, P. Flener, M. A. F. Rodríguez, J. Pearson, and H. Simonis. Systematic derivation of bounds and glue constraints for time-series variables. In M. Rueher, editor, *Principles and Practice of Constraint Programming (CP'2016)*, volume 9892 of *LNCS*, pages 13–29, Toulouse, France, 2016. Springer-Verlag. iii, iv, 22

[3] E. Arafailova, N. Beldiceanu, R. Douence, P. Flener, M. A. Francisco Rodríguez, J. Pearson, and H. Simonis. Time-series constraints: Improvements and application in CP and MIP contexts. In C.-G. Quimper, editor, *International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR'16)*, volume 9676 of *LNCS*, pages 18–34, Banff, Canada, 2016. Springer-Verlag. iii, iv

[4] E. Arafailova, N. Beldiceanu, and H. Simonis. *Among* Implied Constraints for Two Families of Time-Series Constraints. In C. Beck, editor, *Principles and Practice of Constraint Programming (CP'2017)*, volume 10416 of *LNCS*, pages 38–54, Melbourne, Australia, 2017. Springer-Verlag. iv, 121

[5] E. Arafailova, N. Beldiceanu, and H. Simonis. Generating Linear Invariants for a Conjunction of Automata Constraints. In C. Beck, editor, *Principles and Practice of Constraint Programming (CP'2017)*, volume 10416 of *LNCS*, pages 21–37, Melbourne, Australia, 2017. Springer-Verlag. iv, 3201

[6] E. Arafailova, N. Beldiceanu, and H. Simonis. Deriving generic bounds for time-series constraints based on regular expressions characteristics. *Constraints*, 23(1):44–86, 2018. ii, iv, 121

[7] N. Beldiceanu, M. Carlsson, R. Debruyne, and T. Petit. Reformulation of Global Constraints Based on Constraint Checkers. *Constraints*, 10(3), 2005. i

[8] N. Beldiceanu, M. Carlsson, R. Douence, and H. Simonis. Using finite transducers for describing and synthesising structural time-series constraints. *Constraints*, 21(1):22–40, 2016. i, 21

[9] N. Beldiceanu, M. Carlsson, P. Flener, M. A. F. Rodríguez, and J. Pearson. Linking Prefixes and Suffixes for Constraints Encoded Using Automata with Accumulators. In B. O'Sullivan, editor, *Principles and Practice of Constraint Programming (CP'2014)*, volume 8656 of *LNCS*, pages 142–157, Lyon, France, 2014. Springer-Verlag. iv, 22

[10] N. Beldiceanu, M. Carlsson, and J.-X. Rampon. Global Constraint Catalog, 2nd Edition (revision a). Technical Report T2012-03, Swedish Institute of Computer Science, 2012. Available at http://soda.swedish-ict.se/5195/1/T2012-03.pdf. i, l

[11] N. Beldiceanu, P. Flener, J.-N. Monette, J. Pearson, and H. Simonis. Toward sustainable development in constraint programming. *Constraints*, 19(2):139–149, 2014. i

[12] J. Berstel. *Transductions and Context-Free Languages*. Teubner, 1979. i

[13] P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ii, 3201

[14] D. Q. Goldin and P. C. Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In U. Montanari and F. Rossi, editors, *Principles and Practice of Constraint Programming (CP 1995)*, volume 976 of *LNCS*, pages 137–153. Springer, 1995. i

[15] J.-L. Laurière. Toward efficiency through generality. In *6th Int. Joint Conf. on Artificial Intelligence (IJCAI-79)*, pages 519–521, 1979. i, ii

[16] C. Reutenauer and M. P. Schützenberger. Minimization of rational word functions. *SIAM J. Comput.*, 20(4):669–685, 1991. i

[17] M. A. Francisco Rodríguez, P. Flener, and J. Pearson. Implied constraints for automaton constraints. In *GCAI 2015*. EasyChair Epic Series in Computing, 2015. Available at http://www.easychair.org/publications/paper/Implied_Constraints_for_Automaton_Constraints. iv, 3201

[18] M. A. Francisco Rodríguez, P. Flener, and J. Pearson. Automatic generation of descriptions of time-series constraints. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*, pages 102–109. IEEE Computer Society, 2017. iv

[19] J. Sakarovitch. *Elements of Language Theory*. Cambridge University Press, 2009. i

[20] M. P. Schützenberger. A remark on finite transducers. *Information and Control*, 4(2-3):185–196, 1961. i

[21] T. Tantau. *The* TikZ & PGF *Packages: Manual for version 2.10-cvs*, 2.10 edition, 2012. iii

Index

Page numbers in bold face (as in **160**) point to a definition of a constraint, keyword, restriction or system. Page numbers in bold-italic face (as in ***160***) notify an abbreviation of a constraint name. Finally, page numbers in serif face (as in 160) indicate an occurrence of constraint name, keyword, system or author name.

A

D

F

H

I

L

M

N

P

R

S

W