



HAL
open science

Temporal Logic Modeling of Dynamical Behaviors: First-Order Patterns and Solvers

François Fages, Pauline Traynard

► **To cite this version:**

François Fages, Pauline Traynard. Temporal Logic Modeling of Dynamical Behaviors: First-Order Patterns and Solvers. Luis Fariñas del Cerro; Katsumi Inoue. Logical Modeling of Biological Systems, Wiley, 2014, Bioengineering and health science series, 978-1-84821-680-8. 10.1002/9781119005223.ch8 . hal-01103305

HAL Id: hal-01103305

<https://inria.hal.science/hal-01103305>

Submitted on 14 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contents

Chapter 1. Temporal Logic Modeling of Dynamical Behaviors: First-Order Patterns and Solvers	11
François FAGES, Pauline TRAYNARD	
1.1. Temporal Logic FO-LTL(\mathbb{R}_{lin})	12
1.1.1. Syntax	12
1.1.2. Semantics: Validity Domains of Free Variables	13
1.1.3. Generic Solver	15
1.1.4. Complexity	16
1.1.5. Trace Simplification	17
1.1.6. Continuous Satisfaction Degree in $[0,1]$	17
1.2. Formula Patterns and Dedicated Solvers	19
1.2.1. Temporal Operator Patterns	19
1.2.2. Thresholds	20
1.2.3. Amplitudes	21
1.2.4. Local Maxima	23
1.2.5. Monotony	24
1.2.6. Peaks	25
1.2.7. Oscillations	27
1.3. Study Case: Coupled Model of the Cell Cycle and the Circadian Clock	29
1.3.1. Circadian Molecular Clock Model	29
1.3.2. Cell Cycle Model	30
1.3.3. Coupling of the Cell Cycle with the Circadian Clock through <i>WEE1</i>	31
1.3.4. Successive Peak-to-Peak Distances	32
1.3.5. Oscillations with Precise Phaseshifts and Imprecise Amplitudes	33
1.3.6. Filtering out Damped Oscillations	35
1.3.7. Phase Constraints	35

10	Logical Modeling of Biological Systems	
	1.3.8. Model Calibration to Real Data	37
	1.3.9. Comparison of Solvers	38
	1.4. Related Work	39
	1.5. Conclusion	40
	1.6. Bibliography	41
	Index	45

Chapter 1

Temporal Logic Modeling of Dynamical Behaviors: First-Order Patterns and Solvers

This chapter describes how quantitative temporal logic formulae can be used to formalize imprecise dynamical behaviors of biological systems, and how such a formal specification of experimental observations can be used to calibrate models to real data, in a more versatile way than with curve fitting algorithms.

Temporal logics are modal logics which extend classical logic with modal operators to reason about time. After their initial introduction by philosophers in the 1950's, they were proposed in 1977 by Amir Pnueli [PNU 77] to reason about computer programs, either to verify their correctness, or to synthesize programs from their specifications in temporal logic. In the early days of systems biology, propositional temporal logic was soon proposed by computer scientists to formalize the Boolean properties of the behavior of biochemical reaction systems [EKE 02, CHA 03] or gene regulatory networks [BER 04, BAT 05]. In this approach, it is possible to evaluate qualitatively what may or must happen in interaction networks of large size (e.g. of one thousand reactions and species), and to compute initial conditions (i.e. design biological experiments) to exhibit particular behaviors. This can be achieved by using the powerful model-checking tools [CLA 99, CIM 02] designed over the last decades for circuit and program verification.

Chapter written by François FAGES and Pauline TRAYNARD.

Generalizing these techniques to quantitative models can be done in two ways: either by discretizing the different regimes of the dynamics in piecewise linear or affine models [JON 04, BAT 10], or by relying on numerical simulations and taking a first-order version of temporal logic with constraints on concentrations, as query language for the numerical traces [ANT 03, FAG 08]. In this chapter we present methods for the second approach. We illustrate their power for estimating parameters in high dimension [RIZ 09, RIZ 11], and calibrating quantitative models with respect to experimental data. We describe useful patterns of first-order temporal logic formulae to facilitate their use by the modelers, present efficient solvers dedicated to them, and illustrate their use to build a coupled model of the cell cycle and the circadian molecular clock with period and phase constraints.

1.1. Temporal Logic FO-LTL(\mathbb{R}_{lin})

1.1.1. Syntax

The *Linear Time Logic* LTL is a temporal logic [CLA 99] which extends classical logic with modal operators for qualifying when a formula is true in a series of timed states. The temporal operators are **X** (“next”, for at the next time point), **F** (“finally”, for at some time point in the future), **G** (“globally”, for at all time points in the future), **U** (“until”, for a first formula must be true until a second one becomes true), and **W** (“weak until”, a dual operator of **U**). These operators enjoy some simple duality properties, $\neg \mathbf{X}\phi = \mathbf{X}\neg\phi$, $\neg \mathbf{F}\phi = \mathbf{G}\neg\phi$, $\neg \mathbf{G}\phi = \mathbf{F}\neg\phi$, $\neg(\psi \mathbf{U} \phi) = (\neg\phi \mathbf{W} \neg\psi)$, $\neg(\psi \mathbf{W} \phi) = (\neg\psi \mathbf{U} \neg\phi)$, and we have $\mathbf{F}\phi = \text{true} \mathbf{U} \phi$, $\mathbf{G}\phi = \phi \mathbf{W} \text{false}$.

In this paper we consider a first-order version of LTL, denoted by FO-LTL(\mathbb{R}_{lin}), with variables and linear constraints over \mathbf{R} , and quantifiers. The grammar of FO-LTL(\mathbb{R}_{lin}) formulae is defined as follows:

$$\phi ::= c \mid \neg\phi \mid \phi \Rightarrow \psi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x \phi \mid \forall x \phi \mid \mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{W} \phi$$

where c denotes linear constraints between molecular concentrations (written with upper case letters) their first derivative (written dA/dt), free variables (written with lower case letters), real numbers, and the state time variable, denoted by *Time*.

For instance, the formula $\mathbf{F}(A \geq 0.2)$ expresses that the concentration of molecule A gets greater than 0.2 at some time point in the future (**F**). If needed, the precise time values where the concentration of A gets greater than the threshold value can be expressed by introducing a free variable t with an equality constraint to the real time variable, with the formula

$$\mathbf{F}(A \geq 0.2 \wedge t = \text{Time}).$$

Constraints between time variables can also relate the time of different events. For instance, the formula

$$\mathbf{G}(Time \leq t_1 \Rightarrow [A] < 1 \wedge Time \geq t_2 \Rightarrow [A] > 10) \wedge (t_2 - t_1 < 60)$$

expresses that the concentration of molecule A is always less than 1 up to some time t_1 , always greater than 10 after time t_2 , and the switching time between t_1 and t_2 is less than 60 units of time.

A local maximum for molecule concentration A can be defined with the formula

$$\mathbf{F}(A \leq x \wedge \mathbf{X}(A = x \wedge \mathbf{X}A \leq x))$$

where x is a free variable denoted to maximum value. Alternatively, local maxima can also be defined using the derivatives with the formula

$$\mathbf{F}(dA/dt > 0 \wedge \mathbf{X}(dA/dt \leq 0 \wedge x = A)).$$

These formulae can be used to define complex oscillation properties, with period constraints defined as time separation constraints between the local maxima of the molecule, as well as phase constraints between different molecules, as described in Section 1.2.

1.1.2. Semantics: Validity Domains of Free Variables

Temporal logic formulae are classically interpreted in a Kripke structure, i.e. a transition relation over a set of states such that each state has at least one successor [CLA 99]. In this paper, we consider finite traces obtained either by biological experiments in the case of real data, or by numerical integration in the case of simulated data over a finite time horizon. To give meaning to LTL formulae, a finite trace (s_0, \dots, s_n) is thus complemented in an infinite trace by adding a loop on the last state, $(s_0, \dots, s_n, s_n, \dots)$. In this interpretation over finite traces, the formula $\mathbf{G}\phi$ is thus true in the last state if ϕ is true in the last state.

The semantics of formulae containing free variables is given by the validity domains of the variables.

Definition 1. *The validity domain $\mathcal{D}_{(s_0, \dots, s_n), \phi}$ of the free variables of an FO-LTL(\mathbb{R}_{lin}) formula ϕ on a finite trace (s_0, \dots, s_n) , is a vector of least domains for the variables, noted $\mathcal{D}_{(s_0, \dots, s_n), \phi}$, satisfying the following equations:*

- $\mathcal{D}_{(s_0, \dots, s_n), \phi} = \mathcal{D}_{s_0, \phi}$,
- $\mathcal{D}_{s_i, c(\mathbf{x})} = \{\mathbf{v} \in \mathbb{R}^k \mid s_i \models c[\mathbf{v}/\mathbf{x}]\}$ for a constraint $c(\mathbf{x})$,
- $\mathcal{D}_{s_i, \phi \wedge \psi} = \mathcal{D}_{s_i, \phi} \cap \mathcal{D}_{s_i, \psi}$,

$$\begin{aligned}
- \mathcal{D}_{s_i, \phi \vee \psi} &= \mathcal{D}_{s_i, \phi} \cup \mathcal{D}_{s_i, \psi}, \\
- \mathcal{D}_{s_i, \neg \phi} &= \mathcal{C} \mathcal{D}_{s_i, \phi}, \\
- \mathcal{D}_{s_i, \exists x \phi} &= \Pi_x \mathcal{D}_{s_i, \phi}, \\
- \mathcal{D}_{s_i, \forall x \phi} &= \mathcal{D}_{s_i, \neg \exists x \neg \phi}, \\
- \mathcal{D}_{s_i, \mathbf{X} \phi} &= \mathcal{D}_{s_{i+1}, \phi} \text{ if } i < n, \\
- \mathcal{D}_{s_n, \mathbf{X} \phi} &= \mathcal{D}_{s_n, \phi}, \\
- \mathcal{D}_{s_i, \mathbf{F} \phi} &= \bigcup_{j=i}^n \mathcal{D}_{s_j, \phi}, \\
- \mathcal{D}_{s_i, \mathbf{G} \phi} &= \bigcap_{j=i}^n \mathcal{D}_{s_j, \phi}, \\
- \mathcal{D}_{s_i, \phi \mathbf{U} \psi} &= \bigcup_{j=i}^n (\mathcal{D}_{s_j, \psi} \cap \bigcap_{k=i}^{j-1} \mathcal{D}_{s_k, \phi}).
\end{aligned}$$

where \mathcal{C} is the set complement operator over domains, and Π_x is the domain projection operator out of x , restoring domain \mathbb{R} for x .

An FO-LTL(\mathbb{R}_{lin}) formula is false if the validity domain of one variable is empty, valid if the validity domains of all variables are \mathbb{R} , and satisfiable otherwise.

Example 1. For instance, on the numerical trace of Figure 1.1, the validity domain, depicted in Figure 1.2, of the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$, where y_1 and y_2 are free variables, is the domain $y_1 \leq 10 \wedge y_2 \geq 2$.

The two points $\phi_1 = (7, 3)$ and $\phi_2 = (7, 0)$ in this space correspond to the values of y_1 and y_2 in the closed formulae $\mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 3))$ and $\mathbf{F}(A \geq 7 \wedge \mathbf{F}(A \leq 0))$ respectively. In the first case, the formula is true since the point is inside the validity domain, and the second formula is false since the point is outside the validity domain. The distance to the validity domain is used in Section 2 to define a continuous satisfaction degree for FO-LTL(\mathbb{R}_{lin}) formulae.

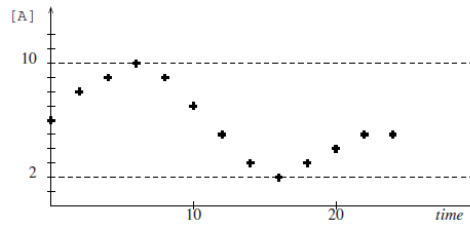


Figure 1.1: Numerical trace for the evolution of the concentration of molecule A over time.

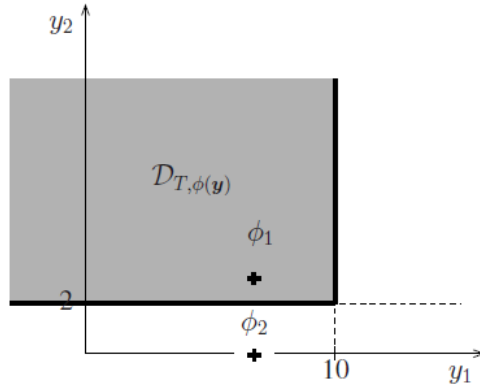


Figure 1.2: Validity domain of the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$ on the trace of Figure 1.1.

1.1.3. Generic Solver

The recursive definition of validity domains (Definition 1) can be directly implemented to compute the validity domains of the free variables in each point of the numerical trace, starting from the last to the first. In this computation the LTL subformulae are considered in the bottom-up order, i.e. first from the linear constraints at the leaves, to finally the root of the syntactic tree.

For instance, to evaluate the formula $\mathbf{F}(A \geq y_1 \wedge \mathbf{F}(A \leq y_2))$ of Example 1, the validity domains of y_1 , y_2 for the subformulae $A \geq y_1$ and $A \leq y_2$ are first computed in each time point of the trace, then the validity domains for $\mathbf{F}(A \leq y_2)$ in each time point from the last to the first, and last for the complete formula by composing the results for the subformulae from the last time point to the first.

Linear constraints over the Reals have a simple geometrical interpretation: a conjunctive state constraint represents a (possibly non-closed) convex polyhedron in the state variables' space and a disjunctive state constraint represents a (non convex) finite union of convex polyhedra. Domain operations with linear constraints can thus be implemented quite straightforwardly with a polyhedra manipulation library. In our implementation in BIOCHAM [CAL 06a], we use the Parma Polyhedra Library, PPL [BAG 08], using set operations for dealing with finite unions of polyhedra.

The conjunction of linear constraints is represented by the intersection polyhedron of the polyhedra associated to the constraints. The existential quantification of a variable is directly implemented by the projection operation of a

polyhedron on the subspace of the space without that variable. A disjunction of polyhedra is represented by a finite set of polyhedra. The complementary of a polyhedron p is computed by the unions of the negated constraints describing p . The universal quantification is implemented by double complementation of the existential quantification.

Crucial to the efficiency of the set-based implementation of disjunctive constraints is the elimination of redundant constraints. The subsumption check between finite sets of linear state constraints is co-NP hard, since checking whether one convex polyhedron is contained in a finite union of convex polyhedra is co-NP complete [SRI 93]. On the other hand, the local subsumption check, i.e. checking whether one convex polyhedron is contained in another one, can be done by linear programming in polynomial time.

1.1.4. Complexity

Bound constraints, i.e. constraints of the form $x \leq c$ or $x \geq c$ where x is a variable and c a constant, define boxes as a particular kind of polyhedra. In that case, the validity domains are finite union domains of boxes, since they are obtained by intersection, union, complementation and projection of boxes.

However, it is worth noticing that even in the case of bound constraints, the validity domain of a temporal formula can contain an exponential number of polyhedra [FAG 08]. Let us define the *size* of a finite union of boxes \mathcal{D} , as the least integer k such that $\mathcal{D} = \bigcup_{i=1}^k \mathcal{R}_i$ where the \mathcal{R}_i 's are boxes.

Proposition 1. *On a trace of length n , the validity domain of a FO-LTL(\mathbb{R}_{lin}) formula of size f containing v variables and only bound constraints, is a union of boxes of size less than $(nf)^{2v}$.*

Proof. Let us consider the number of possible bounds appearing in the validity domain \mathcal{D}_ϕ of a formula ϕ for a given variable x .

Let us first consider the case where ϕ is a bound constraint $x \leq c$ or $c \leq x$. Such a constraint is evaluated on each time point of the trace, by creating at most n different bounds for x . Hence the maximum number of bounds in that case for variable x , is n times the number of occurrences of x in ϕ which is less or equal to $n \times f$. Note that this maximum number of bounds is reached for the formula $F([A] = u \vee [A] + 1 = u \vee \dots \vee [A] + f = u)$ for instance.

Now, the validity domains for the logical connectives, quantifiers and temporal operators are defined by union, intersection, projection and complementation operations, which do not create new bound values for the variables.

As a box is a cartesian product of intervals, it is defined by two bounds for each variable. With less than $n \times f$ bounds per variable, one can thus form at most $(nf)^{2v}$ boxes. Therefore, the solution domain computed by the algorithm is a union of boxes of size less than $(nf)^{2v}$. \square

As for the tightness of these bounds, note that the following formula

$$F([A_1] = X_1 \vee [A_1] + 1 = X_1 \vee \dots \vee [A_1] + f = X_1) \wedge \dots \\ \wedge F([A_v] = X_v \vee [A_v] + 1 = X_v \vee \dots \vee [A_v] + f = X_v)$$

has a solution domain of size $(nf)^v$ on a trace of n values for the $[A_i]$'s such that the values $[A_i] + k$ are all different for $1 \leq i \leq v$ and $0 \leq k \leq f$.

We will see in Section 1.2 that this exponential complexity in the number of free variables does not show up in the formula patterns, since they contain only a limited number of variables, and that in many cases the dedicated solvers compute the validity domain in linear time.

1.1.5. Trace Simplification

A strategy to speed-up the computation of validity domains is to simplify the trace over which the domain is computed by filtering off redundant time points. Indeed, the validity domain is found by combining the subdomains of each time point, and this combination step can be very time consuming when the number of combinations is high. Thus discarding some irrelevant points can lead to significantly faster computation.

In Section 1.3.9, we give some performance figures obtained in a large model with the trace simplification strategy that consists in keeping only the points that are a local extremum for at least one of the molecules involved in the temporal logic formula, as shown in figure 1.3. For each extrema the two successive temporal points corresponding to the derivative change of sign are kept in the trace. It is worth noting however that these simplifications are not correct in general since they may change the validity domains of the variables in complex formulae.

1.1.6. Continuous Satisfaction Degree in $[0,1]$

The true/false valuation of FO-LTL(\mathbb{R}_{lin}) formulae makes it possible to scan the parameter space and check for each parameter set whether the temporal specification is satisfied. However, such generate-and-test methods have an

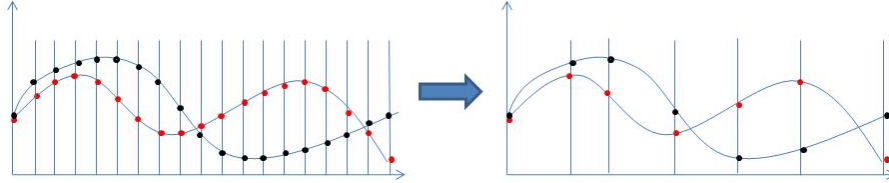


Figure 1.3: Trace simplification by discarding the temporal points that are not local extrema for at least one molecule

exponential complexity in the number of parameters. They are thus limited to two or three parameters and do not scale up.

Calculating a continuous satisfaction degree in the interval $[0, 1]$ for an FO-LTL(\mathbb{R}_{lin}) specification is particularly useful for parameter inference, since comparing the satisfaction degrees obtained with different parameter sets gives a direction to follow in the parameters' space. A continuous satisfaction degree able to measure progress towards satisfaction makes it possible to move from generate-and-test procedures for parameter scanning, to powerful continuous optimization methods for computing several tenths of parameter values in one run, for instance by using evolutionary algorithms with the satisfaction degree as fitness function.

This is the strategy implemented in BIOCHAM [CAL 06a] for parameter search, where we use the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [HAN 01] with the satisfaction degree of the temporal specification as fitness function [RIZ 11].

Definition 2. Let π be a numerical trace. The violation degree

$$vd(\phi, \mathbf{v}) = d(\mathbf{v}, D_{\pi, \psi})$$

of a formula ϕ with free variables \mathbf{x} with respect to objective values \mathbf{v} is the Euclidean distance between the point of coordinates \mathbf{v} and the validity domain ($D_{\pi, \psi}$) if it is not empty, and $+\infty$ if $D_{\pi, \psi}$ is empty. The satisfaction degree of ϕ w.r.t. \mathbf{v} is

$$sd(\phi, \mathbf{v}) = \frac{1}{1 + vd(\phi, \mathbf{v})}$$

The robustness degree

$$rb(\phi, \mathbf{v}) = vd(\neg\phi, \mathbf{x})$$

is the violation degree of the negation of ϕ , i.e. the distance to the complement of the validity domain.

Example 2. *Going back to example 1, we have*

$$vd(\phi_1) = 0, \quad sd(\phi_1) = 1, \quad ro(\phi_1) = 1,$$

and

$$vd(\phi_2) = 2, \quad sd(\phi_2) = 1/3, \quad ro(\phi_2) = 0).$$

The absolute values of the satisfaction degree are not meaningful but the relative values make it possible to compare different parameter sets, and improve them in the most promising direction.

The robustness degree is defined here as the distance between the objective and the complement of the validity domain. Indeed, this distance indicates a degree of robustness of the temporal properties under the assumption that if the validity domain boundaries are far from the objective, the validity domain obtained after slightly changing some parameters are close to the first one, and are thus likely to satisfy the specification. However, in non-linear systems this assumption may be violated, and the robustness of the system must be more accurately estimated by sampling the parameter space around the parameter set [RIZ 09].

1.2. Formula Patterns and Dedicated Solvers

Defining and implementing common patterns is useful for achieving two goals:

- provide user-friendly macros to define a dynamical behavior for biologists who are not familiar with temporal logic [MON 08],
- assign to each pattern a more efficient solver than the generic solver, in order to speed up the computation of validity domains and parameter search.

In this section, we first recall the temporal operator patterns of [MON 08] and introduce first-order patterns with free variables for some important behavior constraints. Each pattern is defined as a macro and is given with a dedicated solver which is compared to the generic solver.

1.2.1. Temporal Operator Patterns

First, it is useful to define macros for operators that apply to one or several temporal logic formulae, given as input to the macro. Table 1.1 lists the operators proposed in [MON 08].

Behavior	Formula	Macro
Occurrence of ϕ	$\mathbf{F}(\phi)$	Occurs(ϕ)
Exclusion	$\mathbf{G}(\neg\phi)$	Excludes(ϕ)
Invariance	$\mathbf{G}(\phi)$	Invariates(ϕ)
Sequence: ϕ occurs before ψ	$\mathbf{F}(\phi \wedge \mathbf{F}(\psi))$	WeakSequence(ϕ, ψ)
Sequence: ϕ immediately before ψ	$\mathbf{F}(\phi \wedge \mathbf{X}(\psi))$	ExactSequence(ϕ, ψ)
Sequence: ϕ always occurs until ψ	$\mathbf{G}(\phi \mathbf{U}(\psi))$	Sequence(ϕ, ψ)
Consequence: if ϕ then later ψ	$\mathbf{G}(\phi \Rightarrow \mathbf{F}(\psi))$	Consequence(ϕ, ψ)
Implication: ϕ implies ψ at the same time	$\mathbf{G}(\phi \Rightarrow \psi)$	Implication(ϕ, ψ)

Table 1.1: Some common operators for temporal logic formulae

1.2.2. Thresholds

We define a first pattern to test whether a concentration threshold is reached by a molecule: $reached(molecule, value)$. The equivalent $LTL(\mathbb{R}_{lin})$ formula and the dedicated solver are given below.

- **Macro: Reached(A,v)**
- Behavior: Reachability
- Equivalent $LTL(\mathbb{R}_{lin})$ formula: $\mathbf{F}(A \geq v)$
- Dedicated validity domain
- Dedicated solver (validity domain, violation degree and robustness):
 - $D =]-\infty; maxA]$
 - $vd = max(0, v - maxA)$
 - $ro = max(0, maxA - v)$
 - Time complexity: $\mathcal{O}(n)$

The figure 1.4 compares the solving procedures of the generic solver and the dedicated solver: while the generic solver computes a validity domain for the variable v for each time point and performs a set of unions to obtain the validity domain on the whole trace, the dedicated solver finds the maximum value of the trace and uses it to define direct functions for the validity domain, the violation degree and the robustness.

The dual pattern $unreached(molecule, value)$ is satisfied when the threshold is not reached. Its validity domain is complementary to the validity domain of $reached(molecule, value)$.

- **Macro: Unreached(A,v)**

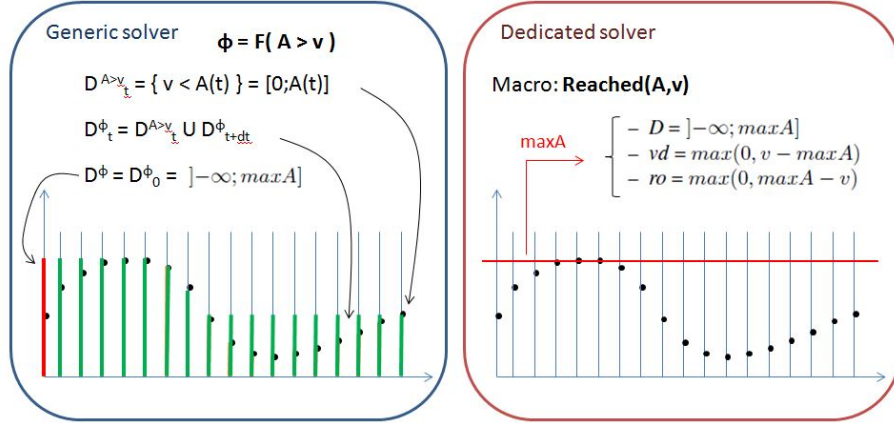


Figure 1.4: Left: the computation procedure followed by the generic solver to test a concentration threshold. Right: the dedicated functions defined by the dedicated solver

- Behavior: Non-reachability
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\mathbf{G}(A \leq v)$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D = [\max A; +\infty[$
 - $vd = \max(0, \max A - v)$
 - $ro = \max(0, v - \max A)$
 - Time complexity: $\mathcal{O}(n)$

1.2.3. Amplitudes

The pattern $\text{MinAmpl}(\text{molecule}, \text{value})$ specifies when the molecule has an amplitude bigger than the specified value or at least equals to it.

- **Macro: MinAmpl(A, a)**
- Behavior: Minimal amplitude
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\exists v \mathbf{F}(A \leq v) \wedge \mathbf{F}(A \geq v + a)$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D =]-\infty; \text{ampl} A[$
 - $vd = \max(0, a - \text{ampl} A)$
 - $ro = \max(0, \text{ampl} A - a)$
 - Time complexity: $\mathcal{O}(n)$

The equivalent LTL(\mathbb{R}_{lin}) formula need two variables to define the amplitude: a which is the amplitude and v which is the minimal value of the molecule. Therefore the validity domain computed with the generic solver has two dimensions, and finding the possible values for the amplitude requires a projection on the variable a , as shown on the figure 1.5. On the other hand the dedicated solver directly computes the one-dimension validity domain in one run.

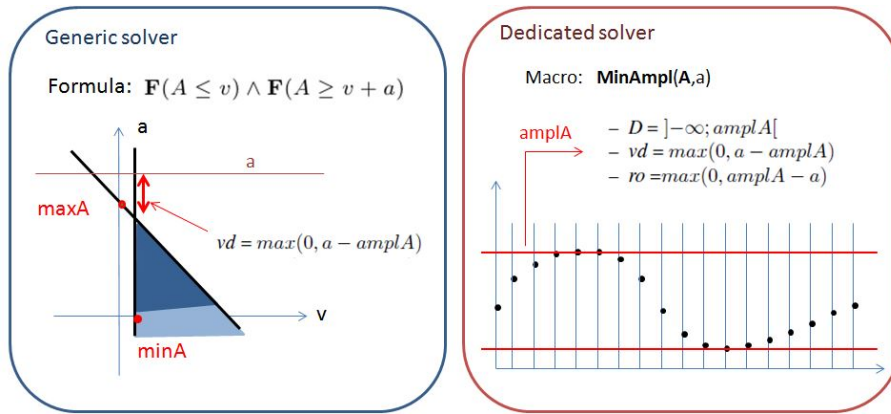


Figure 1.5: Left: the 2-dimensional validity domain computed by the generic solver for the minimal amplitude pattern, and the projection on the a -axis to obtain the violation degree. Right: the functions defined directly with the trace amplitude in the dedicated solver

The pattern $\text{MaxAmpl}(\text{molecule}, \text{value})$ expresses a stability property: it is satisfied when the amplitude of the molecule is equal to or smaller than the specified value. The validity domains of these two patterns are complementary.

- **Macro: MaxAmpl(A,a)**
- Behavior: Maximal amplitude
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\exists v \mathbf{G}(A \geq v \wedge A \leq v + a)$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D =]\text{ampl}A; +\infty[$
 - $vd = \max(0, \text{ampl}A - a)$
 - $ro = \max(0, a - \text{ampl}A)$
 - Time complexity: $\mathcal{O}(n)$

1.2.4. Local Maxima

The pattern $LocalMax(molecule)$ tests the existence of a local maximum: its validity domain is either empty or the universe. For this kind of validity domains, the robustness measure has no sense when the specification is satisfied since the domain is a singularity. In this case we define it arbitrarily to $+\infty$.

- **Macro: LocalMax(A):**
- Behavior: Existence of a local maximum
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\mathbf{F}(\frac{d}{dt}A > 0 \wedge \mathbf{X}(\frac{d}{dt}A \leq 0))$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D = \Omega$ if there is a local max, \emptyset otherwise
 - $vd = 0$ if there is a local max, $+\infty$ otherwise
 - $ro = +\infty$ if there is a local max, 0 otherwise
 - Time complexity: $\mathcal{O}(n)$

The pattern $LocalMax(molecule, time\ point)$ is satisfied when there exists a local maximum at a precise time point.

- **Macro: LocalMax(A,t)**
- Behavior: Existence of a local maximum at the time point t
- Equivalent LTL(\mathbb{R}_{lin}) formula:
 $\mathbf{F}(\frac{d}{dt}A > 0 \wedge \mathbf{X}(\frac{d}{dt}A \leq 0 \wedge Time = t))$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D = \bigcup(\{t'\} / \frac{d}{dt}A(t'-dt) > 0 \wedge \frac{d}{dt}A(t') \leq 0)$
 - $vd = +\infty$ if there is no local max, otherwise $\min_{t' \in D}(|t - t'|)$
 - $ro = +\infty$ if there is a local max at time t , 0 otherwise
 - Time complexity: $\mathcal{O}(n)$

The pattern $LocalMax(molecule, time\ interval\ lower\ bound, time\ interval\ upper\ bound)$ is satisfied when there exists a local maximum in the specified time interval.

- **Macro: LocalMax(A,t1,t2)**
- Behavior: Existence of a local maximum between the times $t1$ and $t2$
- Equivalent LTL(\mathbb{R}_{lin}) formula:
 $\mathbf{F}(\frac{d}{dt}A > 0 \wedge \mathbf{X}(\frac{d}{dt}A \leq 0 \wedge Time \geq t_1 \wedge Time \leq t_2))$
- Generic solver (validity domain, violation degree and robustness):
 - $D = \bigcup([-\infty, t'[\times]t', +\infty[/ \frac{d}{dt}A(t'-dt) > 0 \wedge \frac{d}{dt}A(t') \leq 0)$
 - $vd = +\infty$ if there is no local max,
 $\min_{t' \in local\ max} \sqrt{\max(0, t_1 - t')^2 + \max(0, t' - t_2)^2}$ otherwise

- $ro = 0$ if there is no local max,
 $\min(\max_{t' \in D}(\max(0, t' - t_1)), \max_{t' \in D}(\max(0, t_2 - t')))$ otherwise
- Time complexity: $\mathcal{O}(n)$

1.2.5. Monotony

The pattern *IncrInterv*(*molecule*, *time interval lower bound*, *time interval upper bound*) specifies an interval where the molecule derivative should be always positive.

- **Macro: IncrInterv**(**A**, t_1 , t_2)
- Behavior: Interval of increase
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\mathbf{G}((t_1 < \text{Time} < t_2) \Rightarrow \frac{d}{dt}A \geq 0)$
- Dedicated solver (validity domain, violation degree and robustness):
 - $D = \{(t_1, t_2) \in]-\infty; T_1[\times]T_2; +\infty[/ \forall t \in [T_1, T_2], \frac{d}{dt}A(t) > 0\}$
 - $vd = \min_{(T_1, T_2) \in D} \sqrt{\max(0, t_1 - T_1)^2 + \max(0, T_2 - t_2)^2}$
 - $ro = \min(\max_{T_1}(\max(0, T_1 - t_1)), \max_{T_2}(\max(0, t_2 - T_2)))$
 - Time complexity: $\mathcal{O}(n)$

The pattern *IncreasingSwitch*(*A*, *t*, *v1*, *v2*) specifies that the molecule concentration should increase from smaller than *v1* to greater than *v2* during a time interval of length *t*, without looking at the derivative.

However this pattern contains too many variables to have a directly computable validity domain. The validity domain of this pattern is a set of polyhedrons delineating the possible concentration values taken by the molecule and the time duration between both values:

$$D = \{(\{t\} \times [A(t); +\infty[\times]-\infty, A(t+d)] / A(t+d) > A(t)\}$$

Computing this validity domain requires to compute each possible combination of values that the variables (*t*, *v1*, *v2*) can take, which is exactly what the generic solver does.

- **Macro: IncreasingSwitch**(**A**, t , $v1$, $v2$)
- Concentration switch
- Equivalent LTL(\mathbb{R}_{lin}) formula: $\exists t_1, t_2 \mathbf{G}((\text{Time} \leq t_1 \Rightarrow A < v_1) \wedge (\text{Time} \geq t_2 \Rightarrow A > v_2) \wedge v_2 > v_1 \wedge t_2 - t_1 = t)$
- Validity domain
- $D = \{(\{t\} \times [A(t); +\infty[\times]-\infty, A(t+d)] / A(t+d) > A(t)\}$,
 computed with the generic solver
- The violation degree *vd* and the robustness *ro* are also computed with the generic solver

1.2.6. Peaks

Qualitative specifications of oscillating systems can be particularly various: one might search for the existence of any local maximum (peaks) in the trace, or for a precise time distance between either successive peaks or independent peaks, from one molecule trace or as phaseshifts between different traces.

DistancePeaks(molecule,d) checks that there are two local maxima distant by the time distance d .

– **Macro: DistancePeaks(A,d)**

– There should be two peaks of A distant by d

– Equivalent LTL(\mathbb{R}_{lin}) formula:

$$\begin{aligned} & \exists t_1, t_2 \mathbf{F}(\frac{d}{dt}A \geq 0 \\ & \wedge \mathbf{X}(\frac{d}{dt}A < 0 \wedge \text{Time} = t1 \\ & \wedge \mathbf{F}(\frac{d}{dt}A \geq 0 \\ & \wedge \mathbf{X}(\frac{d}{dt}A < 0 \wedge \text{Time} = t2 \\ & \wedge (t2 > t1) \wedge (t2 - t1 = d))) \end{aligned}$$

– Dedicated solver (validity domain, violation degree and robustness):

- $D = \{d_{i,j} = T_j - T_i, i \in [1;n], j \in [i;n]\}$ where $\{T_i\}$ is the set of local maxima times of the trace

- $vd = +\infty$ if there is no or one local max, $\min_{i,j \in [1;n]} (|d - d_{i,j}|)$ otherwise

- $ro = 0$ if there is no or one local max, otherwise $+\infty$

- Time complexity: $\mathcal{O}(n^2)$

DistancePeaks(molecule,d1,d2) checks that there are two local maxima distant by the time distance between $d1$ and $d2$.

– **Macro: DistancePeaks(A,d1,d2)**

– There should be two peaks of A distant by a value between d_1 and d_2

– Equivalent LTL(\mathbb{R}_{lin}) formula:

$$\begin{aligned} & \exists t_1, t_2 \mathbf{F}(\frac{d}{dt}A \geq 0 \\ & \wedge \mathbf{X}(\frac{d}{dt}A < 0 \wedge \text{Time} = t1 \\ & \wedge \mathbf{F}(\frac{d}{dt}A \geq 0 \\ & \wedge \mathbf{X}(\frac{d}{dt}A < 0 \wedge \text{Time} = t2 \\ & \wedge (t2 > t1) \wedge (t2 - t1 < d_2 \wedge t2 - t1 > d_1))) \end{aligned}$$

– Dedicated solver (validity domain, violation degree and robustness):

- $D = \{(d_1, d_2) \in]-\infty, \max_{i,j}(d_{i,j})[\times]\min_{i,j}(d_{i,j}), +\infty[\}$ with $d_{i,j} = T_j - T_i, i \in [1;n], j \in [i;n]$ where $\{T_i\}$ is the set of local maxima times of the trace

- $vd = +\infty$ if there is no or one local max,
otherwise $\sqrt{\max(0, d_1 - \max_{i,j} d_{i,j})^2 + \max(0, \min_{i,j} d_{i,j} - d_2)^2}$

- $ro = 0$ if there is no or one local max,
otherwise $\min(\max(0, \max_{i,j} d_{i,j} - d_1), \max(0, d_2 - \min_{i,j} d_{i,j}))$
- Time complexity: $\mathcal{O}(n^2)$

These two patterns do not require the local maxima to be successive, but it is needed in the pattern *DistanceSuccPeaks(molecule, d)* below.

- **Macro: DistanceSuccPeaks(A, d)**

- There should be successive two peaks of A distant by d

- Equivalent LTL(\mathbb{R}_{lin}) formula:

$\exists t_1, t_2, t_3, t_4, t_5$

$t_1 > 0 \wedge t_3 > t_2 + 1 \wedge t_2 > t_1 + 1 \wedge t_4 > t_3 + 1 \wedge t_5 > t_4 + 1 \wedge t_5 < 200 \wedge t_4 - t_2 = d$

$\wedge \mathbf{G}((t_1 < \text{Time} \wedge \text{Time} = \langle t_2 \wedge t_2 > t_1 + 1 \rangle - \langle d([A])/dt \rangle = 0))$

$\wedge \mathbf{G}((t_2 < \text{Time} \wedge \text{Time} = \langle t_3 \wedge t_3 > t_2 + 1 \rangle - \langle d([A])/dt \rangle = 0))$

$\wedge \mathbf{G}((t_3 < \text{Time} \wedge \text{Time} = \langle t_4 \wedge t_4 > t_3 + 1 \rangle - \langle d([A])/dt \rangle = 0))$

$\wedge \mathbf{G}((t_4 < \text{Time} \wedge \text{Time} = \langle t_5 \wedge t_5 > t_4 + 1 \rangle - \langle d([A])/dt \rangle = 0))$

- Dedicated solver (validity domain, violation degree and robustness):

- $D = \{d_i = T_{i+1} - T_i, i \in [1; n-1]\}$ where $\{T_i\}$ is the set of local maxima times of the trace

- $vd = +\infty$ if there is no or one local max, $\min_{i \in [1; n-1]} (|d_i - d|)$ otherwise

- $ro = 0$ if there is no or one local max, $+\infty$ otherwise

- Time complexity: $\mathcal{O}(n)$

Finally, the pattern *DistancePeaks(molecule, molecule, distance)* checks the time distance between two local maxima belonging to two different molecule traces. This pattern can be used to determine the phaseshift between two oscillating molecules.

- **Macro: DistancePeaks(A, B, d)**

- There should be a peak of A and a peak of B distant by d

- Equivalent LTL(\mathbb{R}_{lin}) formula:

$\exists t_1, t_2 \mathbf{F}(\frac{d}{dt} A \geq 0 \wedge \mathbf{X}(\frac{d}{dt} A < 0 \wedge \text{Time} = t_1))$

$\wedge \mathbf{F}(\frac{d}{dt} B \geq 0 \wedge \mathbf{X}(\frac{d}{dt} B < 0 \wedge \text{Time} = t_2 \wedge t_2 - t_1 = d))$

- Dedicated solver (validity domain, violation degree and robustness):

- $D = \{d_{i,j}\}$ with $d_{i,j} = T_{A_j} - T_{B_i}, i \in [1; n], j \in [i; n]$ where $\{T_{A_i}\}$ is the set of local maxima times for the trace of A and $\{T_{B_i}\}$ is the set of local maxima times for the trace of B

- $vd = +\infty$ if there is no or one local max,

otherwise $vd = \min_{i,j \in [1; n]} (|d - d_{i,j}|)$

- $ro = 0$ if there is no or one local max, $+\infty$ otherwise

- Time complexity: $\mathcal{O}(n^2)$

1.2.7. Oscillations

Additional constraints can be used to look only for peaks similar in amplitude or in time distance.

MaxDiffDistancePeaks(molecule, maxdifference) is a pattern constraining the maximal difference between the successive peak-to-peak distances in the trace. This maximal difference should be smaller or equal to the specified value. The satisfaction of this constraint ensures that the oscillations are regular enough.

Interestingly, this pattern sets a constraint on each peak of the trace, which is hard to transcribe into a LTL(\mathbb{R}_{lin}) formula. Indeed, in order to constrain every peaks in a trace with a formula one should either know the number of peaks in the trace before defining the formula, or should write a set of subformulas constraining each a different number of peaks and combine them with the operator *or*. Here we show such a formula taking into account only three successive peaks in the trace.

– **Macro: MaxDiffDistancePeaks(A,d)**

– Behavior: The variation of successive peak-to-peak distances for A should be smaller than d

– **LTL(\mathbb{R}_{lin}) formula for three peaks:**

$\exists t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, d_1, d_2, d_3$

$t_1 > 0 \wedge t_3 > t_2 + 1 \wedge t_2 > t_1 + 1 \wedge t_4 > t_3 + 1 \wedge t_5 > t_4 + 1 \wedge t_6 > t_5 + 1$
 $\wedge t_7 > t_6 + 1 \wedge t_8 > t_7 + 1 \wedge t_9 > t_8 + 1 \wedge t_9 < 200 \wedge t_4 - t_2 = d_1 \wedge t_6 - t_4 = d_2$
 $\wedge t_8 - t_6 = d_3 \wedge d_2 - d_1 = < d \wedge d_1 - d_2 = < d \wedge d_3 - d_2 = < d \wedge d_2 - d_3 = < d$
 $\wedge \mathbf{G}((t_1 < \text{Time} \wedge \text{Time} = < t_2 \wedge t_2 > t_1 + 1) \rightarrow (d([A])/dt \geq 0))$
 $\wedge \mathbf{G}((t_2 < \text{Time} \wedge \text{Time} = < t_3 \wedge t_3 > t_2 + 1) \rightarrow (d([A])/dt = < 0))$
 $\wedge \mathbf{G}((t_3 < \text{Time} \wedge \text{Time} = < t_4 \wedge t_4 > t_3 + 1) \rightarrow (d([A])/dt \geq 0))$
 $\wedge \mathbf{G}((t_4 < \text{Time} \wedge \text{Time} = < t_5 \wedge t_5 > t_4 + 1) \rightarrow (d([A])/dt = < 0))$
 $\wedge \mathbf{G}((t_5 < \text{Time} \wedge \text{Time} = < t_6 \wedge t_6 > t_5 + 1) \rightarrow (d([A])/dt \geq 0))$
 $\wedge \mathbf{G}((t_6 < \text{Time} \wedge \text{Time} = < t_7 \wedge t_7 > t_6 + 1) \rightarrow (d([A])/dt = < 0))$
 $\wedge \mathbf{G}((t_7 < \text{Time} \wedge \text{Time} = < t_8 \wedge t_8 > t_7 + 1) \rightarrow (d([A])/dt \geq 0))$
 $\wedge \mathbf{G}((t_8 < \text{Time} \wedge \text{Time} = < t_9 \wedge t_9 > t_8 + 1) \rightarrow (d([A])/dt \geq 0))$

– Dedicated solver (validity domain, violation degree and robustness):

- $D = [\max_{(i) \in [1, n]} |T_{i+2} - 2 * T_{i+1} + T_i|, +\infty[$
 where $\{T_i\}$ is the set of local maxima times of the trace
- $vd = +\infty$ if there is no or one local max,
 otherwise $\max(0, \max_{(i) \in [1, n]} |T_{i+2} - 2 * T_{i+1} + T_i| - d)$
- $ro = 0$ if there is no or one local max,
 otherwise $\max(0, d - \max_{(i) \in [1, n]} |T_{i+2} - 2 * T_{i+1} + T_i|)$
- Time complexity: $\mathcal{O}(n)$

MaxDiffAmplPeaks(molecule, maxdifference) works the same way for the maximal difference between the amplitudes of the peaks in the trace. The amplitude of each peak is computed as the difference between the peak level and the previous local minimum level. The satisfaction of this constraints ensures that the peaks are similar in amplitude, which is useful to filter out damped oscillations.

– **Macro: MaxDiffAmplPeaks(A,d)**

– Behavior: The variation of peaks amplitudes for A should be smaller than d

– **LTL(\mathbb{R}_{lin}) formula for three peaks:**

$\exists t_1, t_2, t_3, t_4, t_5, t_6, t_7, a_1, a_2, a_3, m_1, m_2, m_3, m_4, m_5, m_6$
 $t_1 > 0 \wedge t_3 > t_2 + 1 \wedge t_2 > t_1 + 1 \wedge t_4 > t_3 + 1 \wedge t_5 > t_4 + 1 \wedge t_6 > t_5 + 1 \wedge t_7 > t_6 + 1 \wedge t_7 < 200$

$\wedge \mathbf{G}((t_1 < \text{Time} \wedge \text{Time} = < t_2 \wedge t_2 > t_1 + 1) \rightarrow (d([A])/dt \geq 0))$

$\wedge \mathbf{G}((t_2 < \text{Time} \wedge \text{Time} = < t_3 \wedge t_3 > t_2 + 1) \rightarrow (d([A])/dt \leq 0))$

$\wedge \mathbf{G}((t_3 < \text{Time} \wedge \text{Time} = < t_4 \wedge t_4 > t_3 + 1) \rightarrow (d([A])/dt \geq 0))$

$\wedge \mathbf{G}((t_4 < \text{Time} \wedge \text{Time} = < t_5 \wedge t_5 > t_4 + 1) \rightarrow (d([A])/dt \leq 0))$

$\wedge \mathbf{G}((t_5 < \text{Time} \wedge \text{Time} = < t_6 \wedge t_6 > t_5 + 1) \rightarrow (d([A])/dt \geq 0))$

$\wedge \mathbf{G}((t_6 < \text{Time} \wedge \text{Time} = < t_7 \wedge t_7 > t_6 + 1) \rightarrow (d([A])/dt \leq 0))$

$\wedge \mathbf{F}(\text{Time} = t_1 \wedge [A] = m_1)$

$\wedge \mathbf{F}(\text{Time} = t_2 \wedge [A] = m_2) \wedge m_2 = m_1 + a_1$

$\wedge \mathbf{F}(\text{Time} = t_3 \wedge [A] = m_3)$

$\wedge \mathbf{F}(\text{Time} = t_4 \wedge [A] = m_4) \wedge m_4 = m_3 + a_2 \wedge a_2 - a_1 = < d \wedge a_1 - a_2 = < d$

$\wedge \mathbf{F}(\text{Time} = t_5 \wedge [A] = m_5)$

$\wedge \mathbf{F}(\text{Time} = t_6 \wedge [A] = m_6) \wedge m_6 = m_5 + a_3 \wedge a_2 - a_3 = < d \wedge a_3 - a_2 = < d$

– Dedicated solver (validity domain, violation degree and robustness):

- $D = \left[\max_{(i,j) \in [1,n]^2} |a_i - a_j|, +\infty \right]$ where $\{a_i\}$ is the set of amplitudes of the local maxima for the trace of A, defined as the difference between the local maximum value and the preceding local minimum value

- $vd = +\infty$ if there is no or one local max,

otherwise $\max(0, \max_{(i,j) \in [1,n]^2} |a_i - a_j| - d)$

- $ro = 0$ if there is no or one local max,

otherwise $\max(0, d - \max_{(i,j) \in [1,n]^2} |a_i - a_j|)$

- Time complexity: $\mathcal{O}(n)$

Example 3. An example of a pattern combination is:

$\text{DistanceSuccPeaks}(A, 24) \wedge \text{MaxDiffDistancePeaks}(A, 3)$

$\wedge \text{MaxDiffAmplPeaks}(A, 5) \wedge \text{DistancePeaks}(B, 24)$

$\wedge \text{IntervDistancePeaks}(A, B, 6, 10)$

This combination specifies that the trace of the molecule A should exhibit peaks similar in amplitude (with a maximum difference of 5) and in peak-to-peak distance (with a maximum difference of 3), with two successive peaks distant by 24 hours. The trace of the molecule B should exhibit at least two peaks distant

by 24 hours, not necessarily successive, and there should be a peak of A and a peak of B distant by a time between 6 and 10.

1.3. Study Case: Coupled Model of the Cell Cycle and the Circadian Clock

In many organisms, spontaneous gene expression oscillations with a period close to 24 hours have been observed. A biochemical clock present in each cell is responsible for maintaining these oscillations at this period. The central circadian clock in the suprachiasmatic nucleus (SCN) is sensitive to light and entrained by the day-night alternance, allowing molecular clocks in peripheral tissues to be synchronised by central signals. Indeed, Schibler and Nagoshi [NAG 04] have shown that in absence of synchronisation by the central clock, autonomous circadian oscillators are maintained in peripheral tissues with the same period, although they are progressively desynchronized.

Recent studies have put in evidence autonomous self-sustained circadian oscillators in individual fibroblasts [NAG 04], and proved the existence of several molecular links between the circadian clock and the cell cycle. All these links establish a control of the cell cycle by the circadian clock, and several models of these coupling have been studied to assess the conditions of entrainment in period of the cell cycle length by the circadian clock [GÉR 12].

ODE models yield precise dynamic properties but contain many kinetic parameters whose values can sometimes be roughly estimated with biological considerations but are usually unknown. Here as well, the parameter values have been chosen semi-arbitrarily in order to obtain the desired dynamical behavior. The question remains whether the model could yield a different relevant dynamic behavior with a different set of parameter values. Using $LTL(\mathbb{R}_{\text{lin}})$ constraints, we can query the model to know which qualitative properties are true or false for a given set of parameters, and how far the trace is from the specification. This provides a quantitative comparison between different sets of parameters. However as we will see on a few examples temporal logic formulae for oscillation constraints are very complex and solving the validity domain and satisfaction degree of a trace regarding this kind of property is very time consuming. In this context the use of defined patterns with specific solvers turns out to be particularly useful.

1.3.1. Circadian Molecular Clock Model

In mammalian cells, two major proteins are transcribed in a circadian manner, *CLOCK* and *BMAL1* which bind to form a heterodimer responsible for the transcription of *per* (period) and *cry* (cryptochrome). The two newly-formed

proteins then bind and as soon as the activity of the complex reaches a threshold, *PER/CRY* associates with the complex *CLOCK/BMAL1*, inhibiting its activity and therefore the transcription of the two proteins *PER* and *CRY*. This negative feedback loop gives rise to sustained oscillations.

We use a model proposed by Leloup and Goldbeter [LEL 03] for the mammalian circadian clock. This model consists of 50 reaction rules over 19 molecular species leading to 19 differential equations. The model incorporates the regulatory effects exerted on gene expression by the *PER*, *CRY*, *BMAL1*, *CLOCK*, and *REV-ERB α* proteins, as well as post-translational regulation on these proteins by reversible phosphorylation. We keep the parameter values as published so the period of the circadian rhythm is 24h.

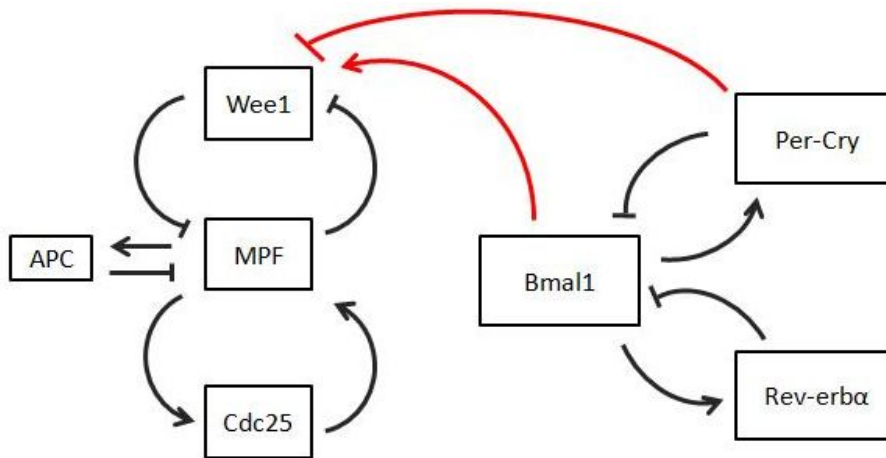


Figure 1.6: Schema of the circadian clock coupled to the cell cycle

1.3.2. Cell Cycle Model

The cell cycle of somatic cells is composed of four phases : DNA replication (S phase) and chromosome segregation or mitosis (M phase), separated by two gap phases (G1 and G2). At the center of the cell cycle regulation, there is a group of proteins, the cyclin-dependent kinases, which are complexes composed of a kinase and a cyclin partner determining the specificity of the complex. Each phase of the cell cycle is controlled by a specific cyclin-dependent kinase.

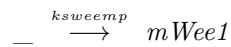
For our purpose, we use a model proposed by Qu et al. (2003) [QU 03], describing a generic cell cycle and focusing on the G2-M transition during which

the protein *WEE1* plays a significant role. The cell cycle is thus divided in two different phases, the G1-S-G2 and M phases. The M phase is triggered by the complex *CDC2/cyclinB*. This complex appears in two forms, an active form called *MPF* (M-phase Promoting Factor) and a phosphorylated, inactive form called *preMPF*. *MPF* is phosphorylated and inactivated by the kinase *WEE1*, and dephosphorylated and activated by the phosphatase *CDC25*. Both the kinase and phosphatase activities are themselves regulated by *MPF*, respectively inactivated and activated by the complex.

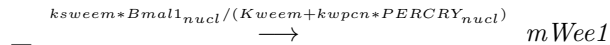
1.3.3. Coupling of the Cell Cycle with the Circadian Clock through *WEE1*

The kinase *WEE1* establishes a link between the cell and circadian cycles during the G2-M transition. Indeed, the *wee1* gene promoter is activated by the complex *CLOCK/BMAL1* and inhibited by *PER/CRY* [MAT 03]. There are other links by which the circadian rhythm influences the cell cycle. In particular, the protein *REV-ERB α* can inhibit the transcription of the Cdk inhibitor p21 [GRÉ 08], or repress the gene *c-myc* that induces the expression of cyclin E [PÉR 97]. However, the mechanisms involved require a very detailed description of the cell cycle. We will thus restrict our study to the link established through *WEE1*, which will allow us to rely on a simpler and more generic model of the cell cycle, focused on mitosis.

The coupling of the cell cycle model to Leloup and Goldbeter's circadian clock model via *WEE1* has been implemented by Calzone and Soliman in 2006 [CAL 06b], by adding to the basal synthesis rule of *mWee1* with kinetic parameter *ksweemp*



the following synthesis rule controlled by *BMAL1* as activator and *PERCRY* as inhibitor:



This coupling leads to a gating effect of the mitosis to precise circadian phases, as the Mitosis Promoting Factor (MPF) can reach its action level when its inhibitor *WEE1* is low, that is when the level of the activator *BMAL1* is low as well.

Under this rule the system exhibits an entrainment of the cell cycle by the circadian clock, as seen in the figure 1.7. In this figure the first plot represents a simulation of the model when the coupling is disabled by setting the parameter *ksweem* corresponding to the synthesis of *WEE1* activated by *BMAL1*,

to 0. The cell cycle (represented by *MPF*) and the circadian clock (the other molecules) are then independent and they have a slightly different period close to 24h. The second plot is the simulation of the coupled model, and the cell cycle is entrained to the same exact period as the circadian clock.

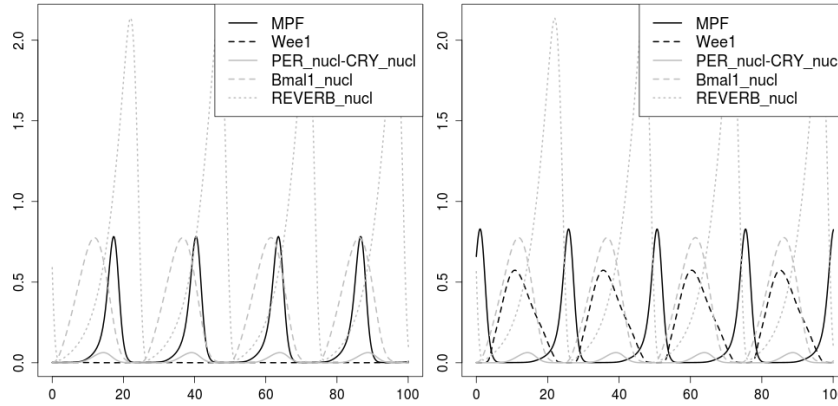


Figure 1.7: Trace of some components of the cell cycle and the circadian clock. Left: the cell cycle and the circadian clock are independent. Right: the cell cycle is entrained by the circadian clock through *WEE1*.

1.3.4. Successive Peak-to-Peak Distances

Temporal logic can be used to check the validity of any dynamic behavior and obtain quantitative values about this behavior. On the left plot of figure 1.7 showing the model without coupling, we could for example assess the stability of each component concentration, that is its amplitude, and its period of oscillation. Let us focus for example on the cell cycle component *MPF*.

The LTL(\mathbb{R}_{lin}) formula for the stability is:

$$\mathbf{G}([MPF] \geq v \wedge [MPF] \leq v + a)$$

The validity domain of this formula gives the values for v and a such that *MPF* amplitude is smaller than a . For this trace the result is:

```
domains (G ( ([MPF]>=v) & ([MPF]<=v+a) ) ).
v + a >= 0.837557, v <= 0.0016107
Time elapsed : 48 ms
```

This formula can be replaced by the pattern $\text{maxAmpl}(MPF, a)$ which gives the same result for a and is easier to write for users who are not familiar with $\text{LTL}(\mathbb{R}_{\text{lin}})$ syntax. The computation is also faster: less than 1ms.

```
domains(maxAmpl([MPF],[a])).
a >= 0.835946
Time elapsed : 0 ms
```

However the domain is directly defined in 1 dimension for the variable a and does not give the corresponding value for v , that is the minimum value of MPF .

The result for the oscillation period using the pattern $\text{distanceSuccPeaks}(MPF, d)$ is:

```
d = 23.3555
|
d = 23.1196
|
d = 23.0935
|
d = 23.119
```

The computation of the validity domain with the generic $\text{LTL}(\mathbb{R}_{\text{lin}})$ solver takes 27s, and only 0.33s with the dedicated solver.

1.3.5. Oscillations with Precise Phaseshifts and Imprecise Amplitudes

The previous formula takes into account successive peak-to-peak distances, which does not allow complex oscillations with small peaks alternating with high peaks. For some parameter values however this kind of behavior can appear, as can be seen on figure 1.8, and is biologically significant.

The pattern DistanceSuccPeaks applied on this formula for the complex Per-Cry in the nucleus (named CRY_n-PER_n in BIOCHAM) would return in the validity domain the distances between the successive peaks, but not the distances between the big peaks which is the real period. In this case it is better to use the pattern DistancePeaks which takes into account the distances between any peak.

We can compare the satisfaction degrees obtained in BIOCHAM with DistanceSuccPeaks and DistancePeaks when the objective is 17h:

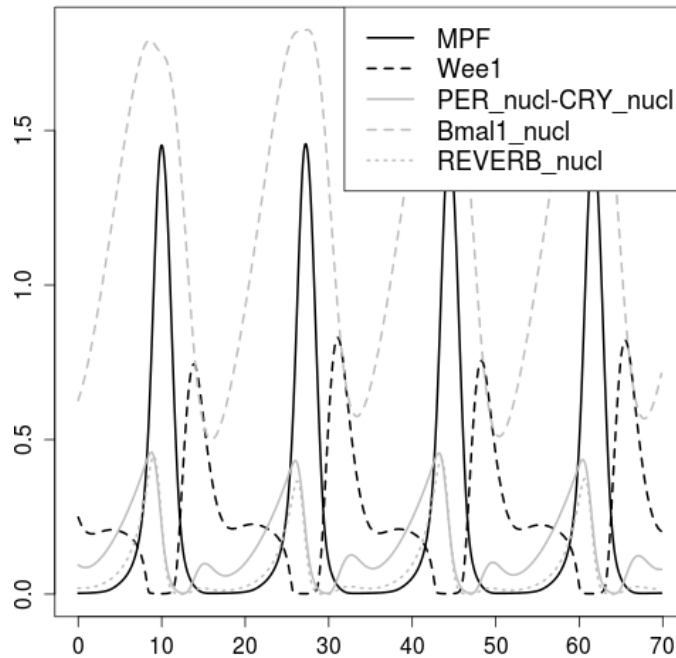


Figure 1.8: Trace for a set of parameters leading to 17h-period oscillations of the system with intermediate peaks for PerCry

```
satisfaction_degree(distanceSuccPeaks([CRY_n-PER_n],[d]],[d],
  ],[17],60).
Distance D=6.00455 computed in 0.29 s
Satisfaction degree 0.14

satisfaction_degree(distancePeaks([CRY_n-PER_n],[d]],[d],[17],60).
Distance D=0.206333 computed in 0.28 s
Satisfaction degree 0.82
```

The satisfaction degree is better with *distancePeaks*. It is not exactly 1 because the period is not exactly 17h but 17.2h.

This is the equivalent formula to be used with the generic solver:

```
satisfaction_degree(
F( d([CRY_nucl-PER_nucl])/dt>=0
  & X( d([CRY_nucl-PER_nucl])/dt<0
    & Time=t1
    & X( F( d([CRY_nucl-PER_nucl])/dt>=0
```

```

& X( d( [CRY_nucl-PER_nucl] ) / dt < 0
      & Time = t2
      & t2 > t1 & t2 - t1 = d ) ) ) )
, [d], [17], 60 ).
Distance D=0.206931 computed in 0.29 s
Satisfaction degree 0.82

```

For this pattern the computation time is roughly the same with the generic solver and the dedicated solver.

1.3.6. Filtering out Damped Oscillations

With the previous formulae one can evaluate the time distances between peaks in a trace, but the evaluation does not take into account the regularness of the oscillations. Thus a trace with damped oscillations like the one shown on figure 1.9 will be ranked well if there is a peak-to-peak distance satisfying the user's specification. To take into account as a rank penalty the variation between peak-to-peak distances and peak amplitudes, we use the patterns *maxDiffDistancePeaks* and *maxDiffAmplPeaks* presented in section 1.2.7.

We use it in combination with the previous macro:

```

domains( distanceSuccPeaks( [CRY_n-PER_n], [d1, d2] )
          & maxDiffDistancePeaks( [CRY_n-PER_n], [m1] )
          & maxDiffAmplPeaks( [CRY_n-PER_n], [m2] ) ).
d1 < 17.4984, d1 - d2 < 0, m1 > 6.54756, d2 > 10.9785, m2 >
1.90466
Time elapsed : 2836 ms

```

The validity domain shows that the successive peak-to-peak distances vary in a 6h range between 11h and 17h, while the peak amplitudes vary in a 2 units range.

1.3.7. Phase Constraints

To express a phase constraint between two molecules we use the pattern *distancePeaks(A,B,d)* which finds the time distance between the peaks of the molecules A and B.

In order to find a set of parameter values leading to oscillations for both the cell cycle and the circadian clock and a phase constraint between both system, we can use the following combination:

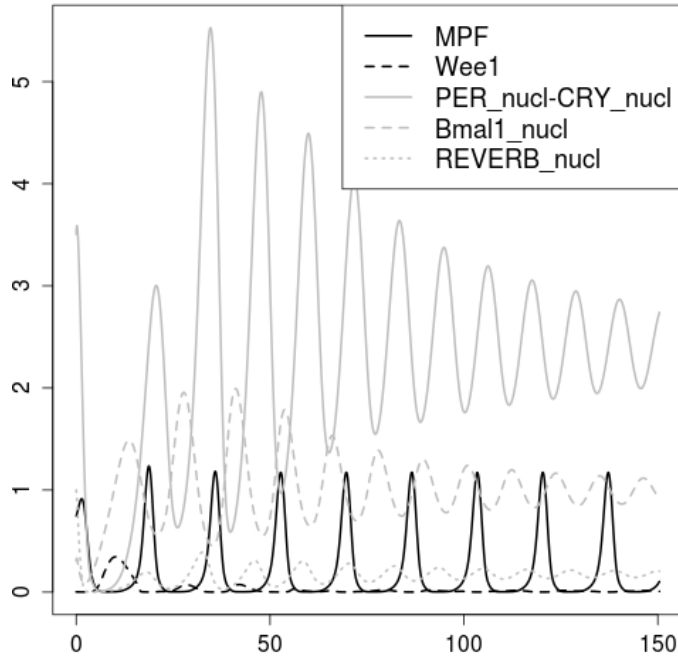


Figure 1.9: Trace for a set of parameters leading to damped oscillations for PerCry

```

distanceSuccPeaks ([MPF] , [d1, d2] ,
& maxDiffDistancePeaks ([MPF] , [m1])
& maxDiffAmplPeaks ([MPF] , [m2]) ) .
& distancePeaks ([CRY_n-PER_n] , [d3, d4] )
& maxDiffDistancePeaks ([CRY_n-PER_n] , [m3])
& maxDiffAmplPeaks ([CRY_n-PER_n] , [m4])
& distancePeaks ([MPF, CRY_n-PER_n] , [d] )

```

We assume that the molecules of each module oscillate synchronously, that is if we find a set of parameters such that PER/CRY , which is part of the circadian clock module, oscillates to a certain period then the other molecules of the circadian clock will do the same. Thus we can apply constraints on only two molecules, one from each module: PER/CRY from the circadian clock and MPF from the cell cycle.

The formula presented below is the kind of formula that we can use to express the same set of constraints:

the two coupled system will be consistent with the one observed experimentally. Indeed, it has been observed experimentally that cell divisions occur at preferential time windows regarding the circadian phase [NAG 04]. It is thus interesting to verify that the structure of the model allows such phases, by trying to find a set of parameter values that keep the dynamical behavior of the system with oscillations and entrainment of the cell cycle, and satisfy at the same time a given phaseshift between *MPF* as a marker of mitosis, and *REVERB* which is the circadian protein observed experimentally.

The optimization method chooses 95 sets of parameter values for 6 parameters involved in the control of the cell cycle by the circadian clock, and evaluate the violation degree of each simulation with the dedicated violation degree used with the combination of patterns:

$$\begin{aligned} &DistanceSuccPeaks(MPF, 24) \wedge MaxDiffAmplPeaks(MPF, 0.2) \\ &\wedge DistancePeaks(REVERB_n) \wedge MaxDiffAmplPeaks(REVERB_n, 0.2) \\ &\wedge DistancePeaks(MPF, REVERB_n, 10, 14). \end{aligned}$$

This specifies that MPF and REVERB should follow periods of 24h with peaks similar in amplitude and should have a phaseshift between 10 and 14h. The results on the population are used to choose the parameter distributions for the next iteration.

After 24 iterations, a satisfying result is found. The simulation obtained for this set of parameter values is shown in figure 1.10. Each molecule oscillates with a 24h period, and the phaseshift between *MPF* and *REVERB_nucl* is 13h.

1.3.9. Comparison of Solvers

In this section we use the introduced model to compare the different solvers on simulation traces over a time horizon 200 time units (hours), obtained using Rosenbrock's implicit method with variable step-size for numerical integration. This simulation contains 869 time points. The traces are periodic with a 24h-period. We evaluate the satisfaction degree of this trace for several dynamical behaviors specified for the molecule *MPF*, and shown in table 1.2. For each specification we evaluate the trace with three methods:

- with the generic solver on the whole trace (869 points).
- with the generic solver on the optimized trace: only two points for each extremum of the molecules present in the formula are kept in the trace. The optimization is performed in 8ms and keep 32 points out of 869 when only MPF is present in the formula, and is performed in 12ms and keep 64 points when REVERB_nucl is also present. The time in this column is the sum of the optimization time and the generic solver on the simplified trace.

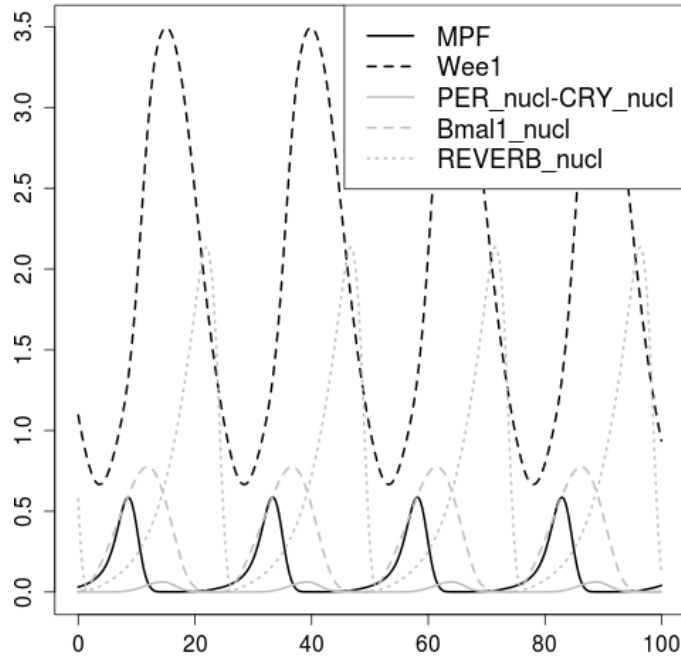


Figure 1.10: Result of the parameter search for a phaseshift between *MPF* and *REVERB_nucl* around 12h.

– with the dedicated solver applied on the whole trace (869 points).

1.4. Related Work

Compared to the patterns of temporal logic formulae defined in [MON 08], we have basically added a first-order setting with free variables, which make it possible to extract interesting values from a trace, define continuous satisfaction degrees, and beyond verification, see temporal formulae as constraints and use efficient parameter optimization procedures for model building.

The Signal Temporal Logic STL is a similar formalism that has been successfully used in [STO 13] to build a model of TRAIL-induced apoptosis and revisit the classification of T-cells.

A temporal logic specification of timing constraints in a model of a genetic switch was successfully used in [RIZ 09] with our method in BIOCHAM to compute global sensitivity indices and improve the design of a synthetic switch.

Pattern	Generic solver	with simpl. trace	Dedicated solver
Reached(MPF,1)	0.18	0.26	< 0.01
Unreached(MPF,1)	0.18	0.25	< 0.01
MinAmpl(MPF,1)	0.24	0.25	< 0.01
DistantPeaks(MPF,20)	1.76	0.32	0.52
DistantSuccPeaks(MPF,20)	17.97	3.01	0.88
DistantPeaks(MPF,20) & DistantPeaks(REVERB_n,20) & DistantPeaks(MPF,REVERB_n,6)	6.78	0.73	1.16
DistantPeaks(MPF,20) & MaxDiffAmplPeaks(MPF,0.1) & DistantPeaks(REVERB_n,20) & MaxDiffAmplPeaks(REVERB_n,0.1) & DistantPeaks(MPF,REVERB_n,6)	36.64	3.17	1.83

Table 1.2: Times in seconds for computing the validity domains of different formula patterns on a trace of 869 time points (32 with trace simplification), compared between the generic solver, without and with trace simplification, and the dedicated solver on the full trace.

The method described in this chapter has also been used in cell signaling to elucidate the complex dynamics of GPCR signaling in [HEI 12]. In this study, the failure to fit with these techniques some response curves in a model of GPCR signaling, was the key to revisit the structure of GPCR signaling interactions, and propose a different mechanism that has been verified experimentally.

1.5. Conclusion

Temporal logic provides a powerful language for describing the important properties of the dynamical behavior of a biological system. While curve fitting methods assume a precise specification of the dynamics given by a complete curve, it is possible with temporal logic formulae to specify imprecise behaviors in a semi-qualitative semi-quantitative manner.

The full first-order setting presented in this chapter makes it possible to extract interesting information from numerical traces, e.g. on periods and phases of irregular oscillations, through the computation of validity domains for the free variables of the formulae. In addition, we have shown how these validity domains can be used to define a continuous satisfaction degree in the interval

[0, 1] for temporal logic formulae, and how this can be combined with powerful continuous optimization algorithms to search parameter values in high dimension.

We believe that the first-order patterns of formulae defined in this chapter, together with their efficient dedicated constraint solvers, will facilitate the use of this approach by modeler, and its implementation in computational systems biology tools.

Acknowledgments

We are grateful to our colleagues of Dassault-Systèmes for fruitful discussions on this topic in the framework of the French OSEO BioIntelligence project.

1.6. Bibliography

- [ANT 03] ANTONIOTTI M., POLICRITI A., UGEL N., MISHRA B., “Model Building and Model Checking for Biochemical Processes”, *Cell Biochemistry and Biophysics*, vol. 38, p. 271–286, 2003.
- [BAG 08] BAGNARA R., HILL P. M., ZAFFANELLA E., “The Parma Polyhedra Library: Toward a Complete Set of Numerical Abstractions for the Analysis and Verification of Hardware and Software Systems”, *Science of Computer Programming*, vol. 72, num. 1–2, p. 3–21, 2008.
- [BAT 05] BATT G., ROPERS D., DE JONG H., GEISELMANN J., MATEESCU R., PAGE M., SCHNEIDER D., “Validation of qualitative models of genetic regulatory networks by model checking : Analysis of the nutritional stress response in *Escherichia coli*”, *Bioinformatics*, vol. 21, num. Suppl.1, p. i19-i28, 2005.
- [BAT 10] BATT G., PAGE M., CANTONE I., GOESSLER G., MONTEIRO P., DE JONG H., “Efficient parameter search for qualitative models of regulatory networks using symbolic model checking”, *Bioinformatics*, vol. 26, num. 18, p. i603-i610, 2010.
- [BER 04] BERNOT G., COMET J.-P., RICHARD A., GUESPIN J., “A Fruitful Application of Formal Methods to Biological Regulatory Networks: Extending Thomas’ Asynchronous Logical Approach with Temporal Logic”, *Journal of Theoretical Biology*, vol. 229, num. 3, p. 339–347, 2004.
- [CAL 06a] CALZONE L., FAGES F., SOLIMAN S., “BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge”, *Bioinformatics*, vol. 22, num. 14, p. 1805–1807, Oxford Univ. Press, 2006.
- [CAL 06b] CALZONE L., SOLIMAN S., Coupling the Cell cycle and the Circadian Cycle, Research Report num. 5835, INRIA, February 2006.

- [CHA 03] CHABRIER N., FAGES F., “Symbolic model checking of biochemical networks”, PRIAMI C., Ed., *CMSB’03: Proceedings of the first workshop on Computational Methods in Systems Biology*, vol. 2602 of *Lecture Notes in Computer Science*, Rovereto, Italy, Springer-Verlag, p. 149–162, March 2003.
- [CIM 02] CIMATTI A., CLARKE E., ENRICO GIUNCHIGLIA F. G., PISTORE M., ROVERI M., SEBASTIANI R., TACHELLA A., “NuSMV 2: An OpenSource Tool for Symbolic Model Checking”, *Proceedings of the International Conference on Computer-Aided Verification, CAV’02*, Copenhagen, Denmark, July 2002.
- [CLA 99] CLARKE E. M., GRUMBERG O., PELED D. A., *Model Checking*, MIT Press, 1999.
- [EKE 02] EKER S., KNAPP M., LADERROUTE K., LINCOLN P., MESEGUER J., SÖNMEZ M. K., “Pathway Logic: Symbolic Analysis of Biological Signaling”, *Proceedings of the seventh Pacific Symposium on Biocomputing*, p. 400–412, January 2002.
- [FAG 08] FAGES F., RIZK A., “On Temporal Logic Constraint Solving for the Analysis of Numerical Data Time series”, *Theoretical Computer Science*, vol. 408, num. 1, p. 55–65, November 2008.
- [GÉR 12] GÉRARD C., GOLDBETER A., “Entrainment of the Mammalian Cell Cycle by the Circadian Clock: Modeling Two Coupled Cellular Rhythms”, *PLoS Comput Biol*, vol. 8, num. 21, Pagee1002516, Public Library of Science, 05 2012.
- [GRÉ 08] GRÉCHEZ-CASSIAU A., RAYET B., GUILLAUMOND F., TEBOUL M., DE-LAUNAY F., “The circadian clock component Bmal1 is a critical regulator of p21WAF1/CIP1 expression and hepatocyte proliferation”, *J Biol Chem*, vol. 283, p. 4535-4542, 2008.
- [HAN 01] HANSEN N., OSTERMEIER A., “Completely derandomized self-adaptation in evolution strategies”, *Evolutionary Computation*, vol. 9, num. 2, p. 159–195, 2001.
- [HEI 12] HEITZLER D., DURAND G., GALLAY N., RIZK A., AHN S., KIM J., VIOLIN J. D., DUPUY L., GAUTHIER C., PIKETTY V., CRÉPIEU P., POUPON A., CLÉMENT F., FAGES F., LEFKOWITZ R. J., REITER E., “Competing G protein-coupled receptor kinases balance G protein and β -arrestin signaling”, *Molecular Systems Biology*, vol. 8, num. 590, June 2012.
- [JON 04] DE JONG H., GOUZÉ J.-L., HERNANDEZ C., PAGE M., SARI T., GEISELMANN J., “Qualitative simulation of genetic regulatory networks using piecewise-linear models”, *Bulletin of Mathematical Biology*, vol. 66, num. 2, p. 301–340, 2004.
- [LEL 03] LELOUP J.-C., GOLDBETER A., “Toward a detailed computational model for the mammalian circadian clock”, *Proceedings of the National Academy of Sciences*, vol. 100, p. 7051–7056, 2003.
- [MAT 03] MATSUI T., YAMAGUCHI S., MITSUI S., EMI A., SHIMODA F., OKAMURA H., “Control Mechanism of the Circadian Clock for Timing of Cell Division in Vivo”, *Science*, vol. 302, num. 5643, p. 255–259, October 2003.

- [MON 08] MONTEIRO P. T., ROPERS D., MATEESCU R., FREITAS A. T., DE JONG H., “Temporal logic patterns for querying dynamic models of cellular interaction networks”, *Bioinformatics*, vol. 24, num. 16, p. i227-i233, Oxford University Press, 2008.
- [NAG 04] NAGOSHI E., ET AL., “Circadian Gene Expression in Individual Fibroblasts: cell-autonomous and self-sustained oscillators pass time to daughter cells.”, *Cell*, vol. 119, p. 693–705, 2004.
- [PÉR 97] PÉREZ-ROGER I., SOLOMON D. L., SEWING A., LAND H., “Myc activation of cyclin E/Cdk2 kinase involves induction of cyclin E gene transcription and inhibition of p27(Kip1) binding to newly formed complexes.”, *Oncogene*, vol. 14, num. 20, p. 2373–81, May 1997.
- [PNU 77] PNUELI A., “The Temporal Logic of Programs”, *FOCS*, p. 46–57, 1977.
- [QU 03] QU Z., MACLELLAN W. R., WEISS J. N., “Dynamics of the cell cycle: checkpoints, sizers, and timers”, *Biophysics Journal*, vol. 85, num. 6, p. 3600–3611, 2003.
- [RIZ 09] RIZK A., BATT G., FAGES F., SOLIMAN S., “A general computational method for robustness analysis with applications to synthetic gene networks”, *Bioinformatics*, vol. 12, num. 25, p. il69–il78, June 2009.
- [RIZ 11] RIZK A., BATT G., FAGES F., SOLIMAN S., “Continuous Valuations of Temporal Logic Specifications with applications to Parameter Optimization and Robustness Measures”, *Theoretical Computer Science*, vol. 412, num. 26, p. 2827–2839, 2011.
- [SRI 93] SRIVASTAVA D., “Subsumption and indexing in constraint query languages with linear arithmetic constraints”, *Annals of Mathematics and Artificial Intelligence*, vol. 8, p. 315–343, 1993.
- [STO 13] STOMA S., DONZÉ A., BERTAUX F., MALER O., BATT G., “STL-based analysis of TRAIL-induced apoptosis challenges the notion of type I/type II cell line classification”, *PLoS Computational Biology*, vol. 9, num. 5, Pagee1003056, Public Library of Science, May 2013.

Index

amplitude constraint, 20
cell cycle, 28
circadian clock, 28
concentration threshold, 19
linear constraint, 15
Linear Time Logic, 12
local maximum, 21
model calibration, 36
monotony, 22
oscillation constraints, 25
peak constraints, 23
robustness degree, 18
satisfaction degree, 18
trace simplification, 17
validity domain, 13
violation degree, 18

