Behavior Analysis of Malicious Code by Weighted Behavior Abstraction

Philippe Beaucamps, Isabelle Gnaedig, Jean-Yves Marion

Behavior Analysis of Malicious Code by Weighted Behavior Abstraction

Philippe Beaucamps, Isabelle Gnaedig, Jean-Yves Marion

Université de Lorraine - INRIA Nancy Grand Est - LORIA
Campus Scientifique - BP 239 F54506 Vandoeuvre-lès-Nancy Cedex, France
Email: {Philippe.Beaucamps, Isabelle.Gnaedig, Jean-Yves.Marion}@loria.fr

Abstract. This work is a weighted generalization of the abstraction based analysis technique we previously proposed for the detection of high-level malware behaviors. Our approach, using a rewriting-based abstraction mechanism, produces abstracted forms of program traces, independent of the program implementation. The suspicious behaviors to be recognized, defined as combinations of patterns given in a signature, are detected by model-checking on the high-level representation of the program. Introducing weights in this approach allows us to express a pertinence degree of detection when analysis of the program results in an incomplete or uncertain program dataflow, or when abstraction cannot be performed reliably.

1 Introduction

In [2], we proposed a formal approach for high-level behavior analysis, underpinned by language theory, term rewriting and first-order temporal logic, allowing us to determine whether a program exhibits a given high-level behavior. Detection is achieved in two steps. First, traces of the program are abstracted in order to reveal the sequences of high-level functionalities they realize. Then, abstracted traces are compared with the behavior formula, using usual model-checking techniques.
More precisley, high-level behaviors we want to detect are expressed in a signature by combinations of functionalities and defined by first-order temporal logic formulas. Behavior patterns, expressing concrete realizations of functionalities in the observed programs, are also defined by first-order temporal logic formulas. These functionalities are then abstracted in program traces by term rewriting, to give normal forms revealing some abstract behavior. Via usual model checking techniques, we then test whether these normal forms satisfy a formula of the signature. In order to address the general intractability of the problem of constructing the normal form trace set for a given program by rewriting-based abstraction, we have identified a property of practical high-level behaviors allowing us to avoid computing normal forms and yielding a linear time detection algorithm.
In this paper, we extend our abstraction formalism in order to abstract traces that do not surely carry out a functionality. First, we would like to take into account static analysis shortcomings resulting in an incomplete or erroneous program dataflow being constructed. For instance, assume we want to identify a system file write and we observe a trace representing an opening of a system file followed by a writing of a file seemingly different from the opened one: rather than ignoring this behavior, we would like to nevertheless interpret it as a system file write but take into account the uncertainty stemming from a potential static analysis error. Also, an execution trace may not be expressive enough and induce a doubt with respect to the identification of a functionality. For instance, if the trace calls the function `sprintf` on some string, this string may not be

completely invalidated since, in C, the string is referenced by the address of one of its characters. Lastly, when a functionality is described by a complex action sequence and when this sequence is completely observed except for some isolated actions, one may attribute this absence to a static analysis error or to an incomplete functionality definition, and allow for abstracting this functionality all the same.

Thus, when an abstraction is uncertain and likely to induce an error in the identification of a high level behavior, our goal is to provide an alternative to classical abstraction by assigning a given probability to this abstraction. Then, when a high level behavior is searched for in a program, we look for a trace with an abstract form containing this behavior with a probability higher than a given threshold. Thus, detection of a malicious behavior is more accurate. Also, the probability that the program exhibits a behavior may complete other program characteristics computed by alternative static or dynamic analysis techniques, like packing characteristics, statistical profiling of code or network interactions.

To that end, we present a formalism relying on a weighted term rewriting mechanism, where a weight represents the probability that the realized abstraction be right.

2 Background

Term Algebras. Let $S = \{Trace, Action, Data\}$ be a set of sorts, $\mathcal{F} = \mathcal{F}_{\mathrm{t}} \cup \mathcal{F}_{\mathrm{a}} \cup \mathcal{F}_{\mathrm{d}}$ be a finite S-sorted signature, where $\mathcal{F}_{\mathrm{t}}$, $\mathcal{F}_{\mathrm{a}}$, $\mathcal{F}_{\mathrm{d}}$ are mutually distinct and:

- $\mathcal{F}_{\mathrm{t}} = \{\epsilon, \cdot\}$ is the set of the trace constructors, where $\epsilon :\rightarrow Trace$ denotes the empty trace, . has profile $Data\ Trace \rightarrow Trace$;
- $\mathcal{F}_{\mathrm{a}}$ is a set of function symbols or constants, with profile $Data^n \rightarrow Action$, $n \in \mathbb{N}$, describing actions;
- $\mathcal{F}_{\mathrm{d}}$ is a set of data constructors, with profile $\rightarrow Data$ or $Data^n \rightarrow Data$, $n \in \mathbb{N}$.

Let $\mathbb{N}_+^*$ be the set of finite strings of positive natural numbers, called *positions*. The empty string is denoted by λ, and $u \leq v$ means that u is prefix of v. Let X be a set of S-sorted variables. A S-sorted *term* over $(\mathcal{F}, X)$ is a partial function $t : \mathbb{N}_+^* \rightarrow \mathcal{F} \cup X$, such that the domain of definition of t, denoted by $\mathcal{P}os(t)$, is finite and satisfies, for $w \in \mathbb{N}_+^*$ and $i \in \mathbb{N}$: (1) $wi \in \mathcal{P}os(t) \Rightarrow w \in \mathcal{P}os(t)$, (2) $w \in \mathcal{P}os(t) \Rightarrow t(w) \in \mathcal{F} \cup X$. $\mathcal{P}os(t)$ is called the set of positions of t. We denote by $T(\mathcal{F}, X)$ (resp. $T(\mathcal{F})$) the set of S-sorted terms over $(\mathcal{F}, X)$ (resp. the set of finite ground terms over $\mathcal{F}$). For any sort $s \in S$, and any of the above sets of terms T we denote by T_s *the restriction* of T to terms of sort s and by X_s *the subset of variables* of X of sort s. For a term t with $p \in \mathcal{P}os(t)$, we denote by $t|_p$ the subterm of t at position p. We denote by $t[t']_p$ the term obtained by replacing by t' the *subterm* at position p in t. We use the abbreviated notation $\overline{x}$ for variables $x_1, \ldots, x_n$. So $\overline{x} \in X$ stands for $x_1, \ldots, x_n \in X$, and if $f \in \mathcal{F}$ is a symbol of arity $n \in \mathbb{N}$, we denote by $f(\overline{x})$ the term $f(x_1, \ldots, x_n)$.

The elements of $T_{Trace}(\mathcal{F})$ are called *traces*, the elements of $T_{Action}(\mathcal{F})$ are called *actions*. We distinguish the sort *Action* from the sort *Trace* but, for a sake of readability, we may denote by a the trace $\cdot(a, \epsilon)$, for some action a. Similarly, we use the $\cdot$ symbol with infix notation and right associativity, and ϵ is understood when the context is unambiguous. For instance, if a, b, c are actions, $a \cdot b \cdot c$ denotes the trace $\cdot(a, \cdot(b, \cdot(c, \epsilon)))$.

We partition $\mathcal{F}_{\mathrm{a}}$ in a set Σ of symbols, denoting concrete program-level actions, and a set Γ, denoting abstract actions identifying abstracted functionalities. To construct purely concrete (resp. abstract) terms, we use $\mathcal{F}_\Sigma = \mathcal{F} \setminus \Gamma$ (resp. $\mathcal{F}_\Gamma = \mathcal{F} \setminus \Sigma$). The *projection* $t|_{\Sigma'}$, also denoted $\pi_{\Sigma'}(t)$, of a trace t on an alphabet $\Sigma' \subseteq \mathcal{F}_{\mathrm{a}}$ corresponds to keeping in a trace only actions from Σ'. If X is a set of variables of sort *Data*, we define the projection on an alphabet $\Sigma' \subseteq \mathcal{F}_{\mathrm{a}}$ of a term $t \in T_{Trace}(\mathcal{F}, X)$, denoted by $\pi_{\Sigma'}(t)$ or,

equivalently, by $t|_{\Sigma'}$, in the following way:

$$\pi_{\Sigma'}(\epsilon) = \epsilon$$
$$\pi_{\Sigma'}(b \cdot u) = \begin{cases} b \cdot \pi_{\Sigma'}(u) & \text{if } b \in T_{Action}(\mathcal{F}_{\Sigma'}, X) \\ \pi_{\Sigma'}(u) & \text{otherwise} \end{cases}$$

with $b \in T_{Action}(\mathcal{F}, X)$ and $u \in T_{Trace}(\mathcal{F}, X)$. The projection is naturally extended to sets of traces.
We define in a natural way the *concatenation* $t \cdot t'$ of two traces t and t'. The concatenation of two terms t and t' of $T_{Trace}(\mathcal{F}, X)$, where X is a set of S-sorted variables and $t \notin X$, is denoted by $t \cdot t' \in T_{Trace}(\mathcal{F}, X)$ and defined by $t \cdot t' = t[t']_p$, where p is the position of ϵ in t, i.e., $t|_p = \epsilon$. Projection and concatenation are naturally extended to sets of terms of sort *Trace*. We also extend concatenation to $2^{T_{Trace}(\mathcal{F},X)} \times 2^{T_{Trace}(\mathcal{F},X)}$ with $L \cdot L' = \{t \cdot t' \mid t \in L, t' \in L'\}$ and to $2^{T_{Trace}(\mathcal{F},X)} \times T_{Action}(\mathcal{F}, X)$ with $L \cdot a = L \cdot \{a \cdot \epsilon\}$.
Substitutions are defined as usual. A *ground substitution* on a finite set X of S-sorted variables is a mapping $\sigma : X \to T(\mathcal{F})$ such that: $\forall s \in S, \forall x \in X_s$, $\sigma(x) \in T_s(\mathcal{F})$. σ can be naturally extended to a mapping $T(\mathcal{F}, X) \to T(\mathcal{F})$ in such a way that:

$$\forall f(t_1, \ldots, t_n) \in T(\mathcal{F}, X),$$
$$\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n)) \quad .$$

By convention, we denote by $t\sigma$ or by $\sigma(t)$ the application of a substitution σ to a term $t \in T(\mathcal{F}, X)$ and by $L\sigma$ the application of σ to a set of terms $L \subseteq T(\mathcal{F}, X)$. The *set of ground substitutions* over X is denoted by $Subst_X$.

Program Behavior. The representation of a program is chosen to be its set of traces. When executing a program, the captured data is represented on the alphabets Σ, denoting the concrete actions, and $\mathcal{F}_d$, describing the data. In this paper, we consider that the captured data is the library calls along with their arguments. Σ therefore represents the finite set of library calls, while terms built on $\mathcal{F}_d$ identify the arguments and the return values of these calls. A *program execution trace* then consists of a sequence of library calls and is defined by a term of $T_{Trace}(\mathcal{F}_\Sigma)$. A *program behavior* is defined by the set of its execution traces, that is a possibly infinite subset of $T_{Trace}(\mathcal{F}_\Sigma)$. For instance, the term $fopen(1,2) \cdot fwrite(1,3)$ represents the execution trace of a file open call $fopen(1,2)$ followed by a file write call $fwrite(1,3)$, where $1 \in \mathcal{F}_d$ identifies the file handle returned by $fopen$, $2 \in \mathcal{F}_d$ identifies the file path and $3 \in \mathcal{F}_d$ identifies the written data.

First-Order Linear Temporal Logic (FOLTL) We consider the First-Order Logic (FOLTL) defined in [9], without the equality predicate, where the set of atomic predicates AP is a set of terms with variables in a set X. FOLTL is an extension of the LTL Logic (see also [9]) such that:

- If φ is an LTL formula, then φ is an FOLTL formula;
- If φ is an FOLTL formula and $Y \subseteq X$ is a set of variables, then: $\exists Y.\varphi$ and $\forall Y.\varphi$ are FOLTL formulas, where as usual: $\forall Y.\varphi \equiv \neg\exists Y.\neg\varphi$.

Notation $\varphi_1 \odot \varphi_2$ stands for $\varphi_1 \wedge \mathbf{X}(\top \mathbf{U} \varphi_2)$.
We say that a FOLTL formula is *closed* when it has no free variable, i.e., every variable is bound by a quantifier.
Let $Y \subseteq X$ be a set of variables of sort *Data* and $\sigma \in Subst_Y$ be a ground substitution over Y. The *application of* σ to an FOLTL formula φ is naturally defined by the formula $\varphi\sigma$ where any free variable x in φ which is in Y has been replaced by its value $\sigma(x)$.

A formula φ is *satisfied* on infinite sequences of sets of ground instances of atomic predicates, denoted by $\xi = (\xi_0, \xi_1, \ldots)$. $\xi \models \varphi$ (ξ satisfies φ) is defined in the same way as for the LTL logic, with the additional rule: $\xi \models \exists Y.\varphi$ iff there exists $\sigma \in Subst_Y$ such that $\xi \models \varphi\sigma$.
In our context, a formula is satisfied over traces of $T_{Trace}(\mathcal{F})$ identified with sequences of singleton sets of atomic predicates. A finite trace $t = a_0 \cdots a_n$ is identified with the infinite sequence of sets of atomic predicates $\xi_t = (\{a_0\}, \ldots, \{a_n\}, \{\}, \{\}, \ldots)$, and t satisfies φ, denoted by $t \models \varphi$, iff $\xi_t \models \varphi$.
We consider two distinct instances of this logic, depending on the fact that we consider concrete traces or abstract traces. We denote by FOLTL_Σ the FOLTL logic, where the set of atomic predicates is $AP_\Sigma = T_{Action}(\mathcal{F}_\Sigma, X)$ and ξ is in $\left(2^{T_{Action}(\mathcal{F}_\Sigma)}\right)^\omega$. We denote by FOLTL_Γ the FOLTL logic, where the set of atomic predicates is $AP_\Gamma = T_{Action}(\mathcal{F}_\Gamma, X)$ and ξ is in $\left(2^{T_{Action}(\mathcal{F}_\Gamma)}\right)^\omega$.
Note that in practice, to express behaviors, we only use FOLTL formulas that are negations of safety properties. We do not use properties with liveness aspects, which would note make sense on finite traces. Using FOLTL on finite traces allows us a correct balance between behavior expresivity and decidability.
Tree automata and tree transducers are defined as usual [4].

Weighted sets and transformations

Weighted sets A weighted set over a semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$ is a mapping s_A of $A \to S$ where A is a set.
The support of a weighted set $s_A : A \to S$ is denoted by $Supp(s_A)$ and defined by: $Supp(s_A) = \{a \in A \mid s_A(a) \neq \overline{0}\}$.
The union of two weighted sets s_A and s_B over $(S, \oplus, \otimes, \overline{0}, \overline{1})$ is the weighted set denoted by $s_A \cup s_B$ and defined by the mapping $s : A \cup B \to S$ such that: $\forall c \in A \cup B$, $s(c) = s_A(c) \oplus s_B(c)$.

Weighted transformations A weighted transformation from a set A to a set B over a semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$ is a weighted set $s_{A\times B}$ over S, denoted by:

$$a \overset{w}{\rightsquigarrow}_{s_{A\times B}} b$$

for all $a \in A$, $b \in B$ and $w \in S$ such that $w = s_{A\times B}(a, b)$. The subscript $s_{A\times B}$ will be understood when there is no ambiguity.
The union of two weighted transformations μ and μ' of $A \times B \to S$, denoted by $\mu \cup \mu'$, is the weighted transformation defined, for all $a \in A$ and $b \in B$, by:

$$(\mu \cup \mu')(a, b) = \mu(a, b) \oplus \mu'(a, b).$$

The functional composition of two weighted transformations $\mu \in A \times B \to S$ and $\mu' \in B \times C \to S$ is the weighted transformation $\mu; \mu' \in A \times C \to S$ defined, for all $a \in A$ and $c \in C$, by:

$$(\mu; \mu')(a, c) = \bigoplus_{b\in B} \mu(a, b) \otimes \mu'(b, c).$$

Let μ be a weighted transformation from a set A to a set B, and $L \subseteq A$ be a set. We denote by $\mu(L)$ the weighted set of $B \to S$ such that:

$$\forall b \in B,\ \mu(L)(b) = \bigoplus_{a\in L, a\overset{w}{\rightsquigarrow}_\mu b} w.$$

Similarly, we define, for all weighted transformation μ from A to B and all weighted set $s : A \to S$, the weighted set $\mu(s) : B \to S$ by:

$$\forall b \in B,\ \mu(s)(b) = \bigoplus_{a \in A, a \overset{w}{\rightsquigarrow}_{\mu} b} s(a) \otimes w.$$

Let $s : A \to S$ be a weighted set. The identity transformation over s is the weighted transformation from A to A over S denoted by Id_s and defined by:

$$\forall a \in A,\ a \overset{s(a)}{\rightsquigarrow}_{Id_s} a.$$

Weighted tree automata Weighted (tree or word) automata and transducers have been intensively studied in the literature [11,3,15,16,5], and have been in particular applied to natural language processing [14,8]. The addition operation $\oplus$ of the semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ represents the aggregation of weights of alternative paths in a weighted automaton, while the multiplication operation $\otimes$ represents the aggregation of transition weights along a path.
Our definition of weighted tree automata is an immediate translation in terms of tree automata [4] of the weighted tree grammars defined by Alexandrakis [1] and Knight, May and Vogler [12].
X is a set of variables.

Definition 1. *A* weighted (top-down) tree automaton *over a semiring* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ *is a tuple* $A = (\mathcal{F}, Q, q_0, \Delta)$ *where* $\mathcal{F}$ *is a finite alphabet,* Q *is a finite set of states,* $q_0 \in Q$ *is an initial state and* Δ *is a finite set of rules of the form:*

$$q(f(x_1, \ldots, x_n)) \overset{w}{\to} f(q_1(x_1), \ldots, q_n(x_n))$$

where $f \in \mathcal{F}$ *is a symbol of arity* $n \in \mathbb{N}$, $q, q_1, \ldots, q_n \in Q$, $x_1, \ldots, x_n$ *are distinct variables of a set* X *of variables and* $w \in S$.

The transition relation $\to_A$ associated to A is defined by:

$$\begin{gathered}
\forall t, t' \in T(\mathcal{F} \cup Q), \\
t \overset{w}{\to}_A t' \\
\Leftrightarrow \\
\exists\, q(f(x_1, \ldots, x_n)) \overset{w}{\to} f(q_1(x_1), \ldots, q_n(x_n)) \in \Delta, \\
\exists p \in Pos(t), \exists u_1, \ldots, u_n \in T(\mathcal{F}), \\
t|_p = q(f(u_1, \ldots, u_n)) \\
\text{and } t' = t\left[f(q_1(u_1), \ldots, q_n(u_n))\right]_p.
\end{gathered}$$

The weight of a sequence of reductions by $\to_A$ is defined as the product (by $\otimes$) of the weights of each reduction. The restriction of $\to_A$ to leftmost reductions is denoted by $\to^l_A$. Observe that, in a commutative semiring:

$$\forall t, t' \in T(\mathcal{F} \cup Q), \forall w \in S,\ t \overset{w}{\to}{}^*_A\ t' \Leftrightarrow t \overset{w}{\to}{}^{l*}_A\ t'.$$

The weighted tree language recognized by A is the weighted set $\|A\| : T(\mathcal{F}) \to S$ such that, for all term $t \in T(\mathcal{F})$, $\|A\|(t)$ is the sum (by $\oplus$) of the weights of the set $Red_l(t)$ of sequences of reductions by $\to^l_A$ from $q_0(t)$ to t:

$$\forall t \in T(\mathcal{F}),\ \|A\|(t) = \bigoplus_{q_0(t) \overset{w_1}{\to}{}^l_A \cdots \overset{w_n}{\to}{}^l_A t \in Red_l(t)} w_1 \otimes \cdots \otimes w_n.$$

Weighted tree automata recognize the set of regular weighted tree languages.
The size of A is defined by: $|A| = |Q| + |\Delta|$.
An unweighted tree automaton can be seen as a weighted tree automaton A whose rules are weighted by $\overline{1}$. The unweighted set recognized by A is then defined as the support of the weighted set recognized by A. Unweighted tree automata recognize the set of regular unweighted tree languages.

Definition 2. *A* weighted trace language *over an alphabet $\mathcal{F}$ is a weighted tree language of $T_{Trace}(\mathcal{F})$. A* weighted trace automaton *over $\mathcal{F}$ is a weighted tree automaton recognizing a weighted trace language.*

Weighted tree transducers

Definition 3. *A* weighted (top-down) tree transducer *over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ is a tuple $\tau = (\mathcal{F}, \mathcal{F}', Q, q_0, \Delta)$ where $\mathcal{F}$ is a finite set of input symbols, $\mathcal{F}'$ is a finite set of output symbols, Q is a finite set of states, $q_0 \in Q$ is an initial state and Δ is a finite set of rules of the form:*

$$q\left(f\left(x_1, \ldots, x_n\right)\right) \xrightarrow{w} u$$
$$or$$
$$q\left(x_1\right) \xrightarrow{w} u \qquad (\epsilon\text{-rule})$$

where $f \in \mathcal{F}$ is a symbol of arity $n \in \mathbb{N}$, $q \in Q$, $x_1, \ldots, x_n$ are distinct variables of a set X of variables, $u \in T\left(\mathcal{F}' \cup Q, \{x_1, \ldots, x_n\}\right)$ and $w \in S$.

The transition relation $\rightarrow_\tau$ associated to the transducer τ is defined by:

$$\begin{gathered}\forall t, t' \in T\left(\mathcal{F} \cup \mathcal{F}' \cup Q\right), \\ t \xrightarrow{w}_\tau t' \\ \Leftrightarrow \\ \exists\, q(f\left(x_1, \ldots, x_n\right)) \xrightarrow{w} u \in \Delta, \\ \exists p \in Pos\left(t\right), \exists u_1, \ldots, u_n \in T\left(\mathcal{F}'\right), \\ t|_p = q\left(f\left(u_1, \ldots, u_n\right)\right) \\ \text{and } t' = t\left[u\left\{x_1 \leftarrow u_1, \ldots, x_n \leftarrow u_n\right\}\right]_p.\end{gathered}$$

ϵ-rules are a particular case of this definition.
The weight of a sequence of reductions by $\rightarrow_\tau$ is defined as the product (by $\otimes$) of the weights of each reduction. The restriction of $\rightarrow_\tau$ to leftmost reductions is denoted by $\rightarrow_\tau^l$.
The weighted transformation realized by τ is the weighted transformation $\|\tau\|$ from $T\left(\mathcal{F}\right)$ to $T\left(\mathcal{F}'\right)$ over S such that, for all terms $t \in T\left(\mathcal{F}\right)$ and $t' \in T\left(\mathcal{F}'\right)$, $\|\tau\|\left(t, t'\right)$ is equal to the sum of the weights of the set $Red_l\left(t\right)$ of sequences of reductions by $\rightarrow_\tau^l$ from $q_0\left(t\right)$ to t':

$$\begin{gathered}\forall t \in T\left(\mathcal{F}\right), \forall t' \in T\left(\mathcal{F}'\right), \\ \|\tau\|\left(t, t'\right) = \bigoplus\nolimits_{q_0(t) \xrightarrow{w_1}_\tau^l \cdots \xrightarrow{w_n}_\tau^l t' \in Red_l(t)} w_1 \otimes \cdots \otimes w_n.\end{gathered}$$

A weighted top-down tree transducer is *linear* if no variable appears twice in a left-hand side or in a right-hand side of rule. It is *nondeleting* if, for each rule $q\left(f\left(x_1, \ldots, x_n\right)\right) \xrightarrow{w} u$ and for all $i \in [1..n]$, the variable x_i appears in u.
A weighted transformation from $T\left(\mathcal{F}\right)$ to $T\left(\mathcal{F}'\right)$ over S is called *rational* iff there exists a weighted linear nondeleting top-down tree transducer realizing it.
Weighted linear nondeleting top-down tree transducers preserve regularity [12,10] and are closed by union and functional composition [12,6]. In the following, we only consider weighted linear nondeleting top-down tree transducers, because of their good properties.
An unweighted tree transducer can be seen as a weighted tree transducer τ whose rules are weighted by $\overline{1}$. The unweighted transformation $\|\tau\|$ recognized by τ is then defined as the support of the weighted transformation it recognizes.
The size of τ is defined by: $|\tau| = |Q| + |\Delta|$.

Definition 4. *A* weighted trace transducer *is a weighted transducer transforming weighted trace languages into weighted trace languages.*

Definition 5. *Let $\mathcal{F}$ and $\mathcal{F}'$ be two alphabets not necessarily distinct. A* relabeling *from $\mathcal{F}$ to $\mathcal{F}'$ is a tree homomorphism of $T(\mathcal{F}, X) \to T(\mathcal{F}', X)$ defined by a total function of $\mathcal{F} \to \mathcal{F}'$.*

Proposition 1. *Let $\mathcal{F}$ and $\mathcal{F}'$ be two alphabets. Every relabeling from $\mathcal{F}$ to $\mathcal{F}'$ is realized by a top-down tree transducer $(\mathcal{F}, \mathcal{F}', Q, q_0, \Delta)$ such that Q contains a unique state $*$ and the rules of Δ are of the form: $*(f(x_1, \ldots, x_k)) \to g(*(x_1), \ldots, *(x_k))$ with $k \in \mathbb{N}$, $f \in \mathcal{F}$ and $g \in \mathcal{F}'$ of arity k.*

Proof. This transducer simply needs to transform every symbol of $\mathcal{F}$ into its associated symbol of $\mathcal{F}'$.

Complexity results

Proposition 2. *Let A be a weighted tree automaton over an alphabet $\mathcal{F}$. Then the identity transformation over $\|A\|$ is realized by a weighted tree transducer without ϵ-rules of size $O(|A|)$.*

Proof. Straightforward as a weighted tree automaton already has a form of weighted tree transducer precisely realizing the identity transformation over the weighted tree language recognized by this automaton.

Proposition 3. *Let A be a weighted trace automaton over an alphabet $\mathcal{F}$ and a semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$ and let $\tau = (\mathcal{F}, \mathcal{F}', Q, q_0, \Delta)$ be a weighted trace transducer. Then the weighted trace language $\tau(\|A\|)$ is recognized by a weighted trace automaton of size $O(|\tau| \cdot |A|)$.*

Proof. Define two word alphabets Ω and Ω' in bijection with the finite sets $T_{Action}(\mathcal{F})$ and $T_{Action}(\mathcal{F}')$. This induces a bijection between words of Ω^* (resp. Ω'^*) and traces of $T_{Trace}(\mathcal{F})$ (resp. $T_{Trace}(\mathcal{F}')$).
Then, since A is a weighted trace automaton and τ is a weighted trace transducer, the result follows by direct analogy with the case of weighted string transducers on the word alphabets Ω and Ω': the equivalent result on words is proved in [13].

3 Weighted abstraction

We define weighted abstraction as an extension of the unweighted abstraction formalism. As in the unweighted case, we define the functionalities to recognize using the notion of behavior patterns. A functionality is described using an FOLTL formula, so that traces validating this formula are traces realizing the functionality. A behavior pattern is then defined as the set of traces realizing a given functionality, i.e. as the set of traces validating the FOLTL formula describing this functionality.

Definition 6. *A* behavior pattern *is a set of traces $B \subseteq T_{Trace}(\mathcal{F}_\Sigma)$ satisfying a closed* FOLTL_Σ *formula φ: $B = \{t \in T_{Trace}(\mathcal{F}_\Sigma) \mid t \models \varphi\}$.*

3.1 Weighted abstraction transformation

Abstraction of a behavior pattern in a trace consists in identifying an occurrence of this behavior pattern in the trace and in marking this occurrence by inserting an abstract action at its level. This abstract action has the form $\lambda(d_1, \ldots, d_n)$, where λ is the symbol of Γ associated to the behavior pattern and $d_1, \ldots, d_n$ are constants of $\mathcal{F}_d$. The difference with the unweighted case is that now some occurrences of the behavior pattern may realize the functionality with some uncertainty. An abstract form of a program trace has an associated degree of uncertainty and the detection problem then consists in determining whether there exists some trace with an abstract form realizing a given abstract behavior with low enough uncertainty.

Abstraction uncertainty is naturally described by the probability that a behavior pattern occurrence actually realizes the functionality.

Example 1. Define three behavior patterns $\lambda_1 := a$, $\lambda_2 := b$ and $\lambda_3 := c$, such that the trace b realizes the functionality associated to the behavior pattern λ_2 only with probability 0.1.
Then, assume we want to determine whether a given program exhibits, with probability greater than 0.5, the abstract behavior described by the FOLTL formula: $\lambda_1 \wedge \neg\lambda_2 \,\mathbf{U}\, \lambda_3$. One solution is to compute the set of completely abstracted forms of program traces and then to check whether one of these abstract forms is an instance of this behavior and has an abstraction probability greater than the threshold 0.5.
Thus, consider a program whose unique trace is $a \cdot b \cdot c$. The set of completely abstracted forms of its traces contains the trace $a \cdot \lambda_1 \cdot b \cdot \lambda_2 \cdot c \cdot \lambda_3$, with an abstraction probability equal to 0.1, and the trace $a \cdot \lambda_1 \cdot b \cdot c \cdot \lambda_3$, with an abstraction probability equal to 0.9. Thus, we deduce that the program exhibits the specified behavior.

Though abstraction uncertainty is intuitively represented by a probability, we formalize the resulting abstraction relation using a weighted rewriting system, rather than a probabilistic rewriting system. Indeed, a probability measure is too restrictive in our formalism. For instance, define a behavior pattern $\lambda := a \cdot c$ and a behavior pattern $\lambda' := b \cdot c$, both of them being abstracted with probability 1. The trace $a \cdot b \cdot c$ can then be abstracted into $a \cdot b \cdot c \cdot \lambda \cdot \lambda'$ with some probability p and into $a \cdot b \cdot c \cdot \lambda' \cdot \lambda$ with probability $1 - p$. Assume we want to detect the behavior $\lambda \cdot \lambda'$ with a probability strictly greater than p, then detection would unexpectedly fail. This comes from the fact that we are not interested in the probability of an abstract form with respect to other abstract forms but in the probability of a chain of abstractions resulting in this abstract form.
For this reason, we represent the abstraction uncertainty by a weight defined over a particular commutative semiring, the tropical semiring $(\mathbb{R}^+ \cup \{+\infty\}, \min, +, +\infty, 0)$, where a weight w is naturally associated to a probability p by the formula: $w = -\log(p)$. Thus, if an abstraction has a probability p_1, associated to the weight $-\log(p_1)$, and if another abstraction has a probability p_2, associated to the weight $-\log(p_2)$, the weight resulting from the combined abstractions is given by: $w_1 \otimes w_2 = w_1 + w_2 = -\log(p_1 \cdot p_2)$, which represents as expected the resulting probability of the combined abstractions. Moreover, when an abstract form can be obtained using two different chains of abstractions, with different weights, we are only interested in the abstraction chain with the lowest uncertainty, as in the end we want to discover an abstract form whose weight does not exceed a given threshold: the choice of min to combine weights associated to an abstract form is therefore natural, given that the weight decreases when the probability increases. Finally, the tropical semiring is instrumental in obtaining the detection decidability results in Section 4.

Note that weighted abstraction could be defined in a generic way with respect to the chosen commutative semiring. For this reason and for the sake of simplicity, we keep the notation $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ to represent the tropical semiring.

We then formalize weighted abstraction using a weighted rewriting system, which we call weighted abstraction system.
As in the unweighted abstraction formalism, we insert an abstract action of $T_{Action}\left(\mathcal{F}_{\Gamma}\right)$ whenever an occurrence of some behavior pattern is discovered. For instance, the trace $a \cdot b \cdot c$ could be rewritten into $a \cdot \lambda_1 \cdot b \cdot c$.
When a behavior pattern occurrence is not recognized with certainty, we abstract it with a probability p. Then we also have to consider the complementary case, associated to the complementary probability, that the pattern be not recognized. Thus, if a behavior pattern occurrence is abstracted into λ with a probability p, we define the complementary abstraction of this occurrence into a dual symbol of λ in Γ, denoted by $\overline{\lambda}$, with probability $1 - p$. The previous example illustrates the importance of such a complementary abstraction. Indeed, if we had chosen to simply rewrite $a \cdot b \cdot c$ into $a \cdot b \cdot \lambda_2 \cdot c$ with probability 0.1, the trace $a \cdot b \cdot c$ would admit a single normal form, $a \cdot \lambda_1 \cdot b \cdot \lambda_2 \cdot c \cdot \lambda_3$ with probability 0.1, and we would then wrongly infer that the program does not exhibit the behavior $\lambda_1 \wedge \neg\lambda_2 \,\mathbf{U}\, \lambda_3$. But using the complementary abstraction, trace $a \cdot b \cdot c$ can also be rewritten into $a \cdot b \cdot \overline{\lambda_2} \cdot c$ with probability 0.9, where $\overline{\lambda_2}$ is a symbol of Γ uniquely associated to λ_2, which yields the normal form $a \cdot \lambda_1 \cdot b \cdot \overline{\lambda_2} \cdot c \cdot \lambda_3$, with probability 0.9. Thus Γ contains both abstraction symbols and their dual ones.
The weighted abstraction system we considered in the example is then composed of the three following rules:

$$\begin{aligned} a &\to \left\{a \cdot \lambda_1 : \overline{1}\right\} \\ b &\to \left\{b \cdot \lambda_2 : 0.1, b \cdot \overline{\lambda_2} : 0.9\right\} \\ c &\to \left\{c \cdot \lambda_1 : \overline{1}\right\}. \end{aligned}$$

When no ambiguity is possible, as for the previous system, we label a rule by its probability p rather than its weight $w = -\log(p)$.

Definition 7 (Weighted Abstraction System). *Let $\lambda \in \Gamma$ be an abstraction symbol, $\overline{\lambda} \in \Gamma$ be its dual symbol, X be a set of variables of sort Data, $\overline{x}$ be a sequence of variables in X and y be a variable of sort Trace. A* weighted abstraction system *R on $T_{Trace}\left(\mathcal{F}\right)$ over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ is a finite set of rewrite rules of the form:*

$$\begin{aligned} A_i(X) \cdot B_i(X) \cdot y \to \{\, & A_i(X) \cdot \lambda(\overline{x}) \cdot B_i(X) \cdot y : w_i, \\ & A_i(X) \cdot \overline{\lambda}(\overline{x}) \cdot B_i(X) \cdot y : w_i' \,\} \end{aligned}$$

or of the form:

$$A_i(X) \cdot B_i(X) \cdot y \to A_i(X) \cdot \lambda(\overline{x}) \cdot B_i(X) \cdot y : \overline{1}$$

where the sets $A_i(X)$ and $B_i(X)$ are sets of concrete traces of $T_{Trace}\left(\mathcal{F}_{\Sigma}, X\right)$, $w_i, w_i' \in S$. The associated unweighted abstraction system is the system R_u composed of the same rules, where weights are not considered.

Note that, since the weight represents the abstraction probability of an occurrence of the behavior pattern, there must exist for all i a probability $p_i \in [0..1]$ such that: $w_i = -\log(p_i)$ and $w_i' = -\log(1 - p_i) = -\log\left(1 - e^{-w_i}\right)$.

Definition 8. *The* weighted reduction relation *on $T_{Trace}\left(\mathcal{F}\right)$ generated by a weighted abstraction system R over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ with n rewrite rules $A_i(X) \cdot B_i(X) \cdot$*

$y \rightarrow \left\{A_i\left(X\right) \cdot \lambda\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i, A_i\left(X\right) \cdot \overline{\lambda}\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i'\right\}$ *is the relation, also denoted by* R*, such that, for all* $t, t' \in T_{Trace}\left(\mathcal{F}\right)$ *and for all* $w \in S$*:*

$$t \overset{w}{\rightarrow}_R t'$$
$$\Leftrightarrow$$
$$\exists \sigma \in Inst_{X \cup \{y\}}, \exists p \in Pos\left(t\right), \exists i \in [1..n], \exists \mu \in \Gamma, \left(\mu, w\right) \in \left\{\left(\lambda, w_i\right), \left(\overline{\lambda}, w_i'\right)\right\},$$
$$\exists a \in T_{Trace}\left(\mathcal{F}\right) \cdot T_{Action}\left(\mathcal{F}_\Sigma\right), \exists b \in T_{Trace}\left(\mathcal{F}\right),$$
$$a|_\Sigma \in A_i\left(X\right)\sigma,\, b|_\Sigma \in B_i\left(X\right)\sigma,$$
$$t|_p = a \cdot b \cdot y\sigma \text{ and } t' = t\left[a \cdot \mu\left(\overline{x}\right)\sigma \cdot b \cdot y\sigma\right]_p.$$

The associated unweighted reduction relation is the reduction relation induced by the unweighted abstraction system R_u *and is also denoted by* R_u*.*

The transformation generated by a weighted abstraction system is the weighted transformation such that a term $t \in T_{Trace}\left(\mathcal{F}\right)$ is transformed into a term $t' \in T_{Trace}\left(\mathcal{F}\right)$ with a weight w corresponding to the sum of the weights of the rewriting steps by which t can be rewritten into t'.

Definition 9. *The* weighted transformation *on* $T_{Trace}\left(\mathcal{F}\right)$ *generated by a weighted abstraction system* R *over a semiring* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ *with* n *rewrite rules* $A_i\left(X\right) \cdot B_i\left(X\right) \cdot y \rightarrow \left\{A_i\left(X\right) \cdot \lambda\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i, A_i\left(X\right) \cdot \overline{\lambda}\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i'\right\}$ *is the weighted transformation defined as follows:*

$$\forall t, t' \in T_{Trace}\left(\mathcal{F}\right),$$
$$t \overset{w}{\rightsquigarrow}_R t'$$
$$\Leftrightarrow$$
there exists n *weighted reduction steps* $t \overset{w_i}{\rightarrow}_R t'$ *and:* $w = \bigoplus_{i \in [1..n]} w_i$.

Then, a weighted abstraction transformation with respect to a given behavior pattern is the weighted transformation generated over $T_{Trace}\left(\mathcal{F}\right)$ by a weighted abstraction system, such that the set of instances of left-hand sides of rules of the abstraction system cover the behavior pattern set of traces.

Definition 10 (Weighted Abstraction Transformation). *Let* B *be a behavior pattern associated with an abstraction symbol* $\lambda \in \Gamma$*. Let* X *be a set of variables of sort Data. The* weighted abstraction transformation *with respect to this behavior pattern over the semiring* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ *is the weighted transformation on* $T_{Trace}\left(\mathcal{F}\right)$ *generated by a weighted abstraction system over* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ *composed of* n *rules* $A_i\left(X\right) \cdot B_i\left(X\right) \cdot y \rightarrow \{\, A_i\left(X\right) \cdot \lambda\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i,\ A_i\left(X\right) \cdot \overline{\lambda}\left(\overline{x}\right) \cdot B_i\left(X\right) \cdot y : w_i' \,\}$ *verifying:*

$$B = \bigcup_{i \in [1..n]} \bigcup_{\sigma \in Inst_{X \cup \{y\}}} \left(A_i\left(X\right) \cdot B_i\left(X\right)\right)\sigma.$$

Then we generalize the definition of a weighted abstraction transformation to a set of behavior patterns.

Definition 11. *Let* C *be a finite set of behavior patterns. The* weighted abstraction transformation *with respect to* C *over a semiring* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ *is the union of the weighted abstraction transformations with respect to each behavior pattern of* C *over the semiring* $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$*.*

From now on, if a behavior pattern is defined using an FOLTL formula φ and associated to an abstraction symbol λ, we may describe it using the notation $\lambda := \varphi$.

3.2 Sound weighted abstraction

Being able to abstract a behavior pattern occurrence successively into an abstract symbol and into its dual one is not consistent. Similarly, abstracting twice the same behavior pattern occurrence is not consistent and may distort the computation of the abstraction weight.
Therefore we define a notion of sound weighted abstraction transformation. In this definition, for an abstract action $\alpha \in T_{Action}(\mathcal{F}_\Gamma)$, we denote by $\overline{\alpha}$ the dual action obtained by replacing in α the symbol of Γ by its dual one. For instance, for $\alpha = \lambda(d)$, with $d \in \mathcal{F}_d$, we define $\overline{\alpha} = \overline{\lambda}(d)$, and for $\alpha = \overline{\lambda}(d)$, we define $\overline{\alpha} = \lambda(d)$.

Definition 12 (Sound Weighted Abstraction Transformation). *The* sound weighted abstraction transformation *over a semiring* $(S, \oplus, \otimes, \overline{0}, \overline{1})$ *for a weighted abstraction transformation* $\rightsquigarrow_R$ *is the weighted transformation* $\rightsquigarrow_{R_s}$ *defined by:*

$$\begin{gathered}\forall t_1, t_2 \in T_{Trace}(\mathcal{F}), \forall \alpha \in T_{Action}(\mathcal{F}_\Gamma), \forall w \in S \\ t_1 \cdot t_2 \overset{w}{\rightsquigarrow}_{R_s} t_1 \cdot \alpha \cdot t_2 \\ \Leftrightarrow \\ t_1 \cdot t_2 \overset{w}{\rightsquigarrow}_{R} t_1 \cdot \alpha \cdot t_2 \\ \textit{and} \\ \nexists (u, u') \in T_{Trace}(\mathcal{F}_\Gamma) \times T_{Trace}(\mathcal{F}), \\ t_2 = u \cdot \alpha \cdot u' \textit{ or } t_2 = u \cdot \overline{\alpha} \cdot u'.\end{gathered}$$

Observe that a sound weighted abstraction transformation is terminating.
From now on, for a weighted abstraction system R, we denote by $R_{\rightsquigarrow}$ the sound weighted abstraction transformation generated by this system over $T_{Trace}(\mathcal{F})$. Then we define the weighted transformation $R^*_{\rightsquigarrow}$ by: $R^*_{\rightsquigarrow} = \bigcup_{i \in \mathbb{N}} R^i_{\rightsquigarrow}$.

4 Detection problem

From now on, we assume that $R_{\rightsquigarrow}$ is a sound weighted abstraction transformation over the semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$.
As in the unweighted case, an abstract behavior describes combinations of high-level functionalities, that is sequences of abstract actions, and is then defined by combinations of abstraction symbols associated to behavior patterns, using an FOLTL formula φ on $AP_\Gamma = T_{Action}(\mathcal{F}_\Gamma, X)$ instead of $AP_\Sigma = T_{Action}(\mathcal{F}_\Sigma, X)$.

Definition 13. *An* abstract behavior *is a set of traces* $M \subseteq T_{Trace}(\mathcal{F}_\Gamma)$ *satisfying a closed* FOLTL$_\Gamma$ *formula* φ_M*:* $M = \{t \in T_{Trace}(\mathcal{F}_\Gamma) \mid t \models \varphi_M\}$*. When* M *is defined by a formula* φ_M*, we write:* $M := \varphi_M$*.*

In the following, for the sake of simplicity, the initial **F** operator will be implicit in the definitions of abstract behaviors.
Detection of an abstract behavior is defined with respect to a threshold $\rho \in S \setminus \{\overline{0}\}$ and a program p exhibits an abstract behavior M iff one of its traces admits an abstract form realizing M with a weight not exceeding this threshold ρ. But rather than looking for this abstract form among the normal forms of traces of p, as was done in the unweighted case, we look for it among partially abstracted forms of traces of p and we then require that the descendants of this abstract form remain infected, independently of their weight. Indeed, working on normal forms entails that every occurrence of a behavior pattern has been abstracted, even occurrences which do not play any role in the detection of this abstract behavior. However, every abstraction is liable to alter the final weight and thus to compromise detection by exceeding the threshold ρ. Moreover, one may observe that in the unweighted case these two definitions are equivalent.

Example 2. Define the behavior $M := \lambda_1 \wedge \neg\lambda_2 \,\mathbf{U}\, \lambda_3$. Detecting M then amounts to finding a partially abstract trace realizing M i.e., containing an action λ_1 followed by an action λ_3, and such that occurrences of the behavior pattern λ_2 appearing between actions λ_1 and λ_3 have already been abstracted into $\overline{\lambda_2}$.

Interpretation of the detection threshold depends on the chosen semiring. Thus we denote by $\preceq$ the order relation to verify: in the tropical semiring, we have $\preceq = \leq$. We choose to include equality in order to consider "certain" detection as a particular case, i.e. $\rho = \overline{1}$.

Definition 14. *A set of traces L* exhibits an abstract behavior *M defined by a formula φ_M with respect to a threshold $\rho \in S \setminus \{\overline{0}\}$, which is denoted by $L \Subset_{\preceq\rho} M$, iff:*

$$\begin{gathered}\exists t \in L, \exists t' \in T_{Trace}(\mathcal{F}),\\ t' \models \varphi_M,\ R^*_{\rightsquigarrow}(t,t') \preceq \rho\\ \textit{and}\\ \forall t'' \in R^*_u(t'),\ t''|_\Gamma \models \varphi_M.\end{gathered}$$

As the weighted set $R^*_{\rightsquigarrow}(t,t')$ is not computable in general, we generalize the (m,n)-completeness property to the weighted case. This property conveys the fact that if there exists a partially abstracted trace t' as in Definition 14, then it can be discovered in at most m abstraction steps and then n abstraction steps are sufficient to verify that its descendants still realize M.
Intuitively, the m steps are used to insert in a trace exhibiting M the abstract actions ensuring recognition of M (for instance, actions λ_1 and λ_3 for the signature $\lambda_1 \wedge \neg\lambda_2 \,\mathbf{U}\, \lambda_3$), and the n steps are used to check that every important abstraction has been carried out (for instance, that no abstraction could insert an action λ_2 between actions λ_1 and λ_3). Thus, we can consider that the cost (and accordingly the threshold) of abstraction only concerns the first m steps.

Definition 15 ((m,n)-completeness). *Let M be an abstract behavior and m and n be positive numbers. M has the property of* (m,n)-completeness *iff for every threshold $\rho \in S \setminus \{\overline{0}\}$ and for every set of traces $L \subseteq T_{Trace}(\mathcal{F}_\Sigma)$:*

$$\begin{gathered}L \Subset_{\preceq\rho} M\\ \Leftrightarrow\\ \exists t \in L, \exists t' \in T_{Trace}(\mathcal{F}), \exists i \leq m,\ R^i_{\rightsquigarrow}(t,t') \preceq \rho\\ \textit{and}\\ \forall t'' \in R^{\leq n}_u(t'),\ t''|_\Gamma \models \varphi_M.\end{gathered}$$

Example 3. Consider the behavior $M := \lambda_1 \wedge \neg\lambda_2 \,\mathbf{U}\, \lambda_3$ and let w_2 be the minimum weight of a rule inserting an action $\overline{\lambda_2}$. Values of m and n are defined as follows.
Intuitively, m must be at least equal to 2 in order to insert in a trace the actions λ_1 and λ_3. Moreover, if occurrences of the behavior pattern λ_2 appear between both actions, they must have been abstracted into $\overline{\lambda_2}$ during the m steps (otherwise, they could be abstracted into λ_2 during the n steps, which would result in a descendant not realizing M). Yet, if the trace t' exists, its weight does not exceed the threshold ρ so at most ρ/w_2 occurrences of the behavior pattern λ_2 were abstracted into $\overline{\lambda_2}$. Thus we choose: $m = 2 + \rho/w_2$.
Assume now that, for a certain $i \leq m$, we have such a trace t' in $R^i_{\rightsquigarrow}(L)$ which realizes M, with a weight not exceeding the threshold ρ. In order to determine whether its descendants still realize M, it is enough to only consider its descendants at the order 1, so as to check that no occurrence of the behavior pattern λ_2 was forgotten between actions λ_1 and λ_3. Indeed, if an occurrence of the behavior pattern λ_2 can be abstracted after

several reductions of t', it can be abstracted directly in t'. This permutation property of abstraction steps will be formally established later. Thus we choose: $n = 1$.
We will show in Theorem 3 that these values are correct. Notice incidentally that, when $w_2 = \overline{0}$, i.e. occurrences of the behavior pattern λ_2 are never abstracted into $\overline{\lambda_2}$, we get the $(2, 1)$-completeness from the unweighted case.

We now show that detection of a behavior having a property of (m, n)-completeness is decidable in the case of a rational weighted abstraction transformation.
To this end, we define the unweighted set of traces realizing M and whose descendants up to the order n still realize M.

Definition 16. *Let R be a sound weighted abstraction transformation. Let M be a behavior defined by a formula φ_M having the property of (m, n)-completeness. The set of* traces n-exhibiting M *with respect to R is the set:*

$$\left\{ t \in T_{Trace}\left(\mathcal{F}\right) \mid \forall t' \in R_u^{\leq n}(t),\ t'\big|_{\Gamma} \models \varphi_M \right\}.$$

When M is regular and when the inverse of the unweighted abstraction relation R_u is rational, the set of traces n-exhibiting M is regular.

Lemma 1. *Let $R_{\rightsquigarrow}$ be a sound weighted abstraction transformation over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$, such that the relation R_u^{-1} is rational. Let M be a regular abstract behavior with the property of (m, n)-completeness for some positive integers m and n.*
Then the set of traces n-exhibiting M is regular.

Proof. As the set of traces n-exhibiting M is defined by:

$$\left\{ t \in T_{Trace}\left(\mathcal{F}\right) \mid \forall t' \in R_u^{\leq n}(t),\ t'\big|_{\Gamma} \models \varphi_M \right\}$$

and as R_u is an unweighted abstraction relation, this result is an instance of Lemma 3XX in [REF FMCAD].

Detection decidability in the case of a behavior having the property of (m, n)-completeness follows.

Theorem 1. *Let $R_{\rightsquigarrow}$ be a sound weighted abstraction transformation over the tropical semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$ such that $R_{\rightsquigarrow}$ and R_u^{-1} are rational. There exists a detection procedure deciding whether $L \Cap_{\preceq \rho} M$, for every regular set of traces L, for every threshold $\rho \in S \setminus \left\{\overline{0}\right\}$ and for every regular abstract behavior M having the property of (m, n)-completeness for some positive integers m and n.*

Proof. We define, in the tropical semiring:

$$R_{\rightsquigarrow}^{\leq m} = \bigcup_{0 \leq i \leq m} R_{\rightsquigarrow}^{i}.$$

Then, for all $t' \in T_{Trace}\left(\mathcal{F}\right)$:

$$R_{\rightsquigarrow}^{\leq m}(L)(t') = \bigoplus_{t \in L} R_{\rightsquigarrow}^{\leq m}(t, t') = \bigoplus_{t \in L} \bigoplus_{0 \leq i \leq m} R_{\rightsquigarrow}^{i}(t, t').$$

Observing that $\oplus = \min$ and $\preceq = \leq$ in the tropical semiring, the property of (m, n)-completeness can then be stated as follows:

$$\begin{gathered} L \Cap_{\preceq \rho} M \\ \Leftrightarrow \\ \exists t' \in T_{Trace}\left(\mathcal{F}\right),\ R_{\rightsquigarrow}^{\leq m}\left(L\right)\left(t'\right) \preceq \rho \\ \text{and} \\ \forall t'' \in R_u^{\leq n}\left(t'\right),\ t''\big|_{\Gamma} \models \varphi_M. \end{gathered}$$

Denoting by M'' the set of traces n-exhibiting M and by $Id_{M''}$ the identity transformation over M'', (m,n)-completeness of M can be restated as:

$$\begin{gathered} L \Cap_{\preceq \rho} M \\ \Leftrightarrow \\ \exists t' \in M'',\, R_{\rightsquigarrow}^{\leq m}(L)(t') \preceq \rho \\ \Leftrightarrow \\ \exists t' \in T_{Trace}(\mathcal{F}),\, Id_{M''}\left(R_{\rightsquigarrow}^{\leq m}(L)\right)(t') \preceq \rho. \end{gathered}$$

M'' is regular by Lemma 1, so $Id_{M''}$ is rational by Proposition 2. Moreover, $R_{\rightsquigarrow}$ is rational and L is regular, so $R_{\rightsquigarrow}^{\leq m}(L)$ is regular too, as well as $Id_{M''}\left(R_{\rightsquigarrow}^{\leq m}(L)\right)$.
Finally, in the tropical semiring, searching for the path of minimum weight in a weighted tree automaton takes linear time [7].

We now show that abstract behaviors considered in practice have the property of (m,n)-completeness. To this end, we establish several preliminary results.

Definition 17 (Concrete Position). *Let t be a term of $T_{Trace}(\mathcal{F})$ and t' be a subterm of t of sort Trace. The* concrete position of t' *in t is the position of $t'|_{\Sigma}$ in $t|_{\Sigma}$.*

Definition 18 (Reduction at a Concrete Position). *Let R be a weighted abstraction system over a semiring $\left(S,\oplus,\otimes,\overline{0},\overline{1}\right)$. We say that a trace $t = t_1 \cdot t_2$ is* reduced by R into $t_1 \cdot \alpha \cdot t_2$ at the concrete position p with weight w, *denoted by $t_1 \cdot t_2 \xrightarrow{w}_p t_1 \cdot \alpha \cdot t_2$, iff $t_1 \cdot t_2 \xrightarrow{w}_R t_1 \cdot \alpha \cdot t_2$ and p is the concrete position of t_2 in t.*

Proposition 4. *Let $R_{\rightsquigarrow}$ be a sound weighted abstraction transformation over the tropical semiring $\left(S,\oplus,\otimes,\overline{0},\overline{1}\right)$. Let $t \in T_{Trace}(\mathcal{F}_{\Sigma})$ and $t' \in T_{Trace}(\mathcal{F})$ be two terms and let two abstraction sequences by $R_{\rightsquigarrow}$ be:*

$$t \overset{w_1}{\rightsquigarrow}_R \ldots \overset{w_n}{\rightsquigarrow}_R t'$$

and:

$$t \overset{w'_1}{\rightsquigarrow}_R \ldots \overset{w'_{n'}}{\rightsquigarrow}_R t'.$$

Then $n = n'$ and:

$$w_1 \otimes \ldots \otimes w_n = w'_1 \otimes \ldots \otimes w'_{n'}.$$

Proof. By definition of $\rightsquigarrow_R$, in the tropical semiring, given that w_i is the smallest w_i^j such that $t_i \overset{w_i^j}{\rightarrow} t_{i+1}$, there exists two reduction sequences by R of the form:

$$t = t_1 \xrightarrow{w_1}_{p_1} \ldots \xrightarrow{w_n}_{p_n} t'$$

and:

$$t \xrightarrow{w'_1}_{p'_1} \ldots \xrightarrow{w'_{n'}}_{p'_{n'}} t'$$

where $p_1,\ldots,p_n$ and $p'_1,\ldots,p'_{n'}$ are the concrete positions at which reductions are carried out.
Both sequences transform t into t' so the second sequence insert the same abstract actions, at the same concrete positions, than the first sequence, but in a different order. In other words, $n = n'$ and there exists a permutation $\sigma : [1..n] \to [1..n]$ such that: $\forall i \in [1..n],\, p'_{\sigma(i)} = p_i$.
Moreover, since by definition of $\rightsquigarrow_R$, the weight w_i and w'_i are respectively the smallest weights associated to the i-th abstraction, for all i in $[1..n]$, we have as expected: $w_1 \otimes \ldots \otimes w_n = w'_1 \otimes \ldots \otimes w'_{n'}$.

We deduce the following corollary, in the tropical semiring.

Corollary 1. *Let $R_{\rightsquigarrow}$ be a sound weighted abstraction transformation over the tropical semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$. Let $t \in T_{Trace}\left(\mathcal{F}_{\Sigma}\right)$ and $t' \in T_{Trace}\left(\mathcal{F}\right)$ be two terms. For every abstraction sequence $t \overset{w_1}{\rightsquigarrow} \ldots \overset{w_n}{\rightsquigarrow} t'$ by $R_{\rightsquigarrow}$, we have:*

$$w_1 \otimes \ldots \otimes w_n = R^*_{\rightsquigarrow}\left(t, t'\right).$$

Proof. $R^*_{\rightsquigarrow} = \bigcup_{i \in \mathbb{N}} R^i_{\rightsquigarrow}$ so: $R^*_{\rightsquigarrow}\left(t, t'\right) = \min\limits_{i \in \mathbb{N}} R^i_{\rightsquigarrow}\left(t, t'\right)$.
Moreover, $R^i_{\rightsquigarrow}\left(t, t'\right) = \min\limits_{t \overset{w_1}{\rightsquigarrow}_R \ldots \overset{w_i}{\rightsquigarrow}_R t'} w_1 \otimes \ldots \otimes w_i$ so:

$$R^*_{\rightsquigarrow}\left(t, t'\right) = \min_{i \in \mathbb{N},\, t \overset{w_1}{\rightsquigarrow}_R \ldots \overset{w_i}{\rightsquigarrow}_R t'} w_1 \otimes \ldots \otimes w_i.$$

The result then follows from the previous proposition.

We now extend the definition of a reduction at a concrete position (see Definition 18) to an abstraction at a concrete position.

Definition 19 (Abstraction at a Concrete Position). *Let R be a weighted abstraction system over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$. We say that a trace $t = t_1 \cdot t_2$ is* abstracted by R (resp. R_u) into $t_1 \cdot \alpha \cdot t_2$ at the concrete position p with weight w, *denoted by $t_1 \cdot t_2 \overset{w}{\rightsquigarrow}_p t_1 \cdot \alpha \cdot t_2$, iff $t_1 \cdot t_2 \overset{w}{\rightsquigarrow} t_1 \cdot \alpha \cdot t_2$ by R (resp. R_u) and p is the concrete position of t_2 in t.*

Lemma 2. *Let R be a weighted abstraction system over the tropical semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$. Let $t \in T_{Trace}\left(\mathcal{F}\right)$ be a trace and $\alpha_1, \ldots, \alpha_k \in T_{Action}\left(\mathcal{F}_{\Gamma}\right)$ be abstract actions. Let a weighted abstraction chain by R (resp. R_u) from t be: $t \overset{w_1}{\rightarrow}^* t_1 \cdot t'_1 \overset{w'_1}{\rightarrow}_{p_1} t_1 \cdot \alpha_1 \cdot t'_1 \overset{w_2}{\rightarrow}^* t_2 \cdot t'_2 \overset{w'_2}{\rightarrow}_{p_2} t_2 \cdot \alpha_2 \cdot t'_2 \overset{w_3}{\rightarrow}^* \ldots \overset{w_k}{\rightarrow}^* t_k \cdot t'_k \overset{w'_k}{\rightarrow}_{p_k} t_k \cdot \alpha_k \cdot t'_k$ where we distinguish k abstraction steps. Then, we have the following reduction sequence by R (resp. R_u):*

$$\exists u_1, \ldots u_k, u'_1, \ldots u'_k \in T_{Trace}\left(\mathcal{F}\right),$$
$$t \overset{w'_1}{\rightarrow}_{p_1} u_1 \cdot \alpha_1 \cdot u'_1 \overset{w'_2}{\rightarrow}_{p_2} u_2 \cdot \alpha_2 \cdot u'_2 \overset{w'_3}{\rightarrow}_{p_3} \ldots \overset{w'_k}{\rightarrow}_{p_k} u_k \cdot \alpha_k \cdot u'_k.$$

Proof. By induction on the length l of the derivation $t \rightarrow^* t_k \cdot \alpha_k \cdot t'_k$.

- For the case $l = 1$, we have: $t \overset{w_1}{\rightarrow}_{p_1} t_1 \cdot \alpha_1 \cdot t'_1$. Hence, we define $u_1 = t_1$ and $u'_1 = t'_1$.
- For the general induction step, assume the property for $l = n$. We prove the property for $l = n + 1$. By the induction hypothesis applied to $t \overset{w_1}{\rightarrow}^* t_1 \cdot t'_1 \overset{w'_1}{\rightarrow}_{p_1} t_1 \cdot \alpha_1 \cdot t'_1 \overset{w_2}{\rightarrow}^* t_2 \cdot t'_2 \ldots \overset{w_k}{\rightarrow}^* t_k \cdot t'_k \overset{w'_k}{\rightarrow}_{p_k} t_k \cdot \alpha_k \cdot t'_k$, we have: $\exists u_1, \ldots u_k, u'_1, \ldots, u'_k \in T_{Trace}\left(\mathcal{F}\right)$, $t \overset{w'_1}{\rightarrow}_{p_1} u_1 \cdot \alpha_1 \cdot u'_1 \overset{w'_2}{\rightarrow}_{p_2} \ldots \overset{w'_k}{\rightarrow}_{p_k} u_k \cdot \alpha_k \cdot u'_k$.

  For $l = n + 1$, the chain of length n is extended by $t_k \cdot \alpha_k \cdot t'_k \overset{w_{k+1}}{\rightarrow}^* t_{k+1} \cdot t'_{k+1} \overset{w'_{k+1}}{\rightarrow} t_{k+1} \cdot \alpha_{k+1} \cdot t'_{k+1}$.
  We want to rewrite $u_k \cdot \alpha_k \cdot u'_k$ into $u_{k+1} \cdot \alpha_{k+1} \cdot u'_{k+1}$.

  Now, existence of the reduction $t_{k+1} \cdot t'_{k+1} \overset{w'_{k+1}}{\rightarrow} t_{k+1} \cdot \alpha_{k+1} \cdot t'_{k+1}$ entails the existence of an occurrence of the behavior pattern B_{k+1} in $t_{k+1} \cdot t'_{k+1}$. This occurrence also appears in $u_k \cdot \alpha_k \cdot u'_k$ and can therefore be abstracted at the same concrete position p_{k+1} with the same weight w'_{k+1}, hence the existence of terms u_{k+1} and u'_{k+1} such that: $u_k \cdot \alpha_k \cdot u'_k \overset{w'_{k+1}}{\rightarrow}_{p_{k+1}} u_{k+1} \cdot \alpha_{k+1} \cdot u'_{k+1}$.

Lemma 3. *Let R be a weighted abstraction system over the tropical semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$. Let $t \in T_{Trace}\left(\mathcal{F}\right)$ be a trace and $\alpha_1, \ldots, \alpha_k \in T_{Action}\left(\mathcal{F}_\Gamma\right)$ be actions. Let a weighted abstraction chain by $R_{\rightsquigarrow}$ from t be: $t \overset{w_1}{\rightsquigarrow}^* t_1 \cdot t'_1 \overset{w'_1}{\rightsquigarrow}_{p_1} t_1 \cdot \alpha_1 \cdot t'_1 \overset{w_2}{\rightsquigarrow}^* t_2 \cdot t'_2 \overset{w'_2}{\rightsquigarrow}_{p_2} t_2 \cdot \alpha_2 \cdot t'_2 \overset{w_3}{\rightsquigarrow}^* \ldots \overset{w_k}{\rightsquigarrow}^* t_k \cdot t'_k \overset{w'_k}{\rightsquigarrow}_{p_k} t_k \cdot \alpha_k \cdot t'_k$ where we distinguish k abstraction steps. Then:*

$$\exists u_1, \ldots u_k, u'_1, \ldots u'_k \in T_{Trace}\left(\mathcal{F}\right),$$
$$t \overset{w'_1}{\rightsquigarrow}_{p_1} u_1 \cdot \alpha_1 \cdot u'_1 \overset{w'_2}{\rightsquigarrow}_{p_2} u_2 \cdot \alpha_2 \cdot u'_2 \overset{w'_3}{\rightsquigarrow}_{p_3} \ldots \overset{w'_k}{\rightsquigarrow}_{p_k} u_k \cdot \alpha_k \cdot u'_k.$$

Proof. By definition of $\rightsquigarrow_R$, over the tropical semiring (where $\oplus = \min$), there exists a reduction sequence $t \overset{w_1}{\rightarrow}^* t_1 \cdot t'_1 \overset{w'_1}{\rightarrow}_{p_1} t_1 \cdot \alpha_1 \cdot t'_1 \overset{w_2}{\rightarrow}^* t_2 \cdot t'_2 \overset{w'_2}{\rightarrow}_{p_2} t_2 \cdot \alpha_2 \cdot t'_2 \overset{w_3}{\rightarrow}^* \ldots \overset{w_k}{\rightarrow}^* t_k \cdot t'_k \overset{w'_k}{\rightarrow}_{p_k} t_k \cdot \alpha_k \cdot t'_k$.
By Lemma 2, we have the following reduction sequence by R:

$$\exists u_1, \ldots u_k, u'_1, \ldots u'_k \in T_{Trace}\left(\mathcal{F}\right),$$
$$t \overset{w'_1}{\rightarrow}_{p_1} u_1 \cdot \alpha_1 \cdot u'_1 \overset{w'_2}{\rightarrow}_{p_2} u_2 \cdot \alpha_2 \cdot u'_2 \overset{w'_3}{\rightarrow}_{p_3} \ldots \overset{w'_k}{\rightarrow}_{p_k} u_k \cdot \alpha_k \cdot u'_k.$$

We now show, by contradiction, that for each reduction $u_{i-1} \cdot \alpha_{i-1} \cdot u'_{i-1} \overset{w'_i}{\rightarrow}_{p_i} u_i \cdot \alpha_i \cdot u'_i$ by R with $i \in [1..k]$, the weight w'_i is the minimum weight of such a reduction and that we therefore have, by Definition 9 of the weighted abstraction transformation $\rightsquigarrow_R$ in the tropical semiring (where $\oplus = \min$): $u_{i-1} \cdot \alpha_{i-1} \cdot u'_{i-1} \overset{w'_i}{\rightsquigarrow}_{p_i} u_i \cdot \alpha_i \cdot u'_i$.
Assume there exists an index $i \in [1..k]$ such that there exists a reduction $u_{i-1} \cdot \alpha_{i-1} \cdot u'_{i-1} \overset{w''_i}{\rightarrow}_{p_i} u_i \cdot \alpha_i \cdot u'_i$ with $w''_i < w'_i$. This entails that there exists an occurrence of the behavior pattern B_i in $u_{i-1} \cdot \alpha_{i-1} \cdot u'_{i-1}$ such that we can apply a rewrite rule of weight strictly lower than w'_i. So this rule could also be applied to the term $t_i \cdot t'_i$, at the same concrete position of $t_i \cdot t'_i \overset{w''_i}{\rightarrow}_{p_i} t_i \cdot \alpha_i \cdot t'_i$. Yet, by hypothesis, $t_i \cdot t'_i \overset{w'_i}{\rightsquigarrow}_R t_i \cdot \alpha_i \cdot t'_i$ and weights are defined over the tropical semiring where $\oplus = \min$. So, by definition of the weighted abstraction transformation $\rightsquigarrow_R$ (see Definition 9), w'_i must be the smallest weight of a reduction of $t_i \cdot t'_i$ into $t_i \cdot \alpha_i \cdot t'_i$, contradicting hypothesis $w''_i < w'_i$.
We deduce that, for all $i \leq k$, w'_i is the smallest weight of a reduction of $u_{i-1} \cdot \alpha_{i-1} \cdot u'_{i-1}$ into $u_i \cdot \alpha_i \cdot u'_i$. Hence, by Definition 9 of the weighted abstraction transformation $\rightsquigarrow_R$, we have, in the tropical semiring, by R:

$$t \overset{w'_1}{\rightsquigarrow}_{p_1} u_1 \cdot \alpha_1 \cdot u'_1 \overset{w'_2}{\rightsquigarrow}_{p_2} u_2 \cdot \alpha_2 \cdot u'_2 \overset{w'_3}{\rightsquigarrow}_{p_3} \ldots \overset{w'_k}{\rightsquigarrow}_{p_k} u_k \cdot \alpha_k \cdot u'_k.$$

We can now prove that abstract behaviors considered in practice have a property of (m, n)-completeness.

Theorem 2. *Let Y be a set of variables of sort Data. Let $\alpha_1, \ldots, \alpha_m \in T_{Action}\left(\mathcal{F}_\Gamma, Y\right)$. Then the abstract behavior $M := \exists Y.\, \alpha_1 \odot \alpha_2 \odot \ldots \odot \alpha_m$ has the property of $(m, 0)$-completeness.*

Proof. Let $\varphi_M = \exists Y.\, \mathbf{F}(\alpha_1 \odot \alpha_2 \odot \ldots \odot \alpha_m)$.
We need to show that:

$$L \Cap_{\preceq \rho} M$$
$$\Leftrightarrow$$
$$\exists t \in L, \exists t' \in T_{Trace}\left(\mathcal{F}\right), \exists i \leq m,\ R^i_{\rightsquigarrow}(t, t') \preceq \rho$$
and
$$t'|_\Gamma \models \varphi_M.$$

First, by the semantics of FOLTL, a trace $t \in T_{Trace}(\mathcal{F}_\Gamma)$ validates the formula $\exists Y.\, \mathbf{F}(\alpha_1 \odot \alpha_2 \odot \ldots \odot \alpha_m)$ iff t can be written $t = t'_1 \cdot t'_2$ with $t'_1, t'_2 \in T_{Trace}(\mathcal{F}_\Gamma)$ and t'_2 validating the formula $\exists Y.\, \alpha_1 \odot \alpha_2 \odot \ldots \odot \alpha_m$. Then, t'_2 validates the formula $\exists Y.\, \alpha_1 \odot \alpha_2 \odot \ldots \odot \alpha_m$ iff there exists an instantiation $\sigma_Y \in Inst_Y$ such that t'_2 validates the formula $\alpha_1\sigma_Y \odot \alpha_2\sigma_Y \odot \ldots \odot \alpha_m\sigma_Y$, equivalent to the formula $\alpha_1\sigma_Y \wedge \mathbf{X}(\top \mathbf{U}\, \alpha_2\sigma_Y \wedge \mathbf{X}(\top \mathbf{U}\, \ldots \wedge \mathbf{X}(\top \mathbf{U}\, \alpha_m\sigma_Y)))$. So the trace t'_2 validates formulas $\alpha_1\sigma_Y$ and $\mathbf{X}(\top \mathbf{U}\, \alpha_2\sigma_Y \wedge \mathbf{X}(\top \mathbf{U}\, \ldots \wedge \mathbf{X}(\top \mathbf{U}\, \alpha_m\sigma_Y)))$. Hence, t'_2 is of the form:

$$t'_2 = \alpha_1\sigma_Y \cdot t_1 \cdot \alpha_2\sigma_Y \cdot t_2 \cdots \alpha_m\sigma_Y \cdot t_m$$

where $t_1, \ldots, t_m \in T_{Trace}(\mathcal{F}_\Gamma)$.

$\Rightarrow$: By definition of the exhibition of M by L, there exists a trace $t \in L$ with a partially abstracted form $\hat{t}$ by $R_{\rightsquigarrow}$ such that $\hat{t}\big|_\Gamma \models \varphi_M$, $w = R^*_{\rightsquigarrow}(t, \hat{t}) \preceq \rho$ and $\forall t'' \in R^*_u(\hat{t})\big|_\Gamma$, $t'' \models \varphi_M$.
Since $\hat{t}\big|_\Gamma \models \varphi_M$, there exists an instantiation $\sigma_Y \in Inst_Y$ such that:

$$\hat{t} = t_0 \cdot \alpha_1\sigma_Y \cdot t_1 \cdot \alpha_2\sigma_Y \cdot t_2 \cdots \alpha_m\sigma_Y \cdot t_m$$

where $t_0, \ldots, t_m \in T_{Trace}(\mathcal{F})$.
By Corollary 1, any sequence of transformations $t \overset{w_1}{\rightsquigarrow}_R \ldots \overset{w_n}{\rightsquigarrow}_R \hat{t}$ has the weight $w = w_1 \otimes \ldots \otimes w_n$. Since by hypothesis $\hat{t}$ is a partially abstracted form of t, there exists at least one such sequence.
Then, by applying Lemma 3 to this sequence, there exists $u_0, \ldots, u_m \in T_{Trace}(\mathcal{F})$ such that t is transformed by R into a trace $t' = u_0 \cdot \alpha_1\sigma_Y \cdot u_1 \cdots \alpha_m\sigma_Y \cdot u_m$ in exactly m steps with a weight $w' = w_{i_1} \otimes \ldots \otimes w_{i_m}$, for some sequence of distinct indices $(i_j)_j$ in $[1..n]$.
Therefore, in the tropical semiring: $R^m(t, t') \preceq w'$ and $w' \preceq w \preceq \rho$. Moreover $t'|_\Gamma \models \varphi_M$.

$\Leftarrow$: Let $t \in L$, $i \leq m$ and $t' \in T_{Trace}(\mathcal{F})$ be a partially abstracted form of a trace of L such that: $R^i(t, t') \preceq \rho$ and $t'|_\Gamma \models \varphi_M$.
So t' can be written $t' = t_0 \cdot \alpha_1\sigma_Y \cdot t_1 \cdots \alpha_m\sigma_Y \cdot t_m$, where $t_0, \ldots, t_m \in T_{Trace}(\mathcal{F})$ and $\sigma_Y \in Inst_Y$. Clearly, any future abstraction of t' by R_u will still be of the form $u_0 \cdot \alpha_1\sigma_X \cdot u_1 \cdots \alpha_m\sigma_X \cdot u_m$ and will hence validate φ_M. So t' satisfies the condition of Definition 14, entailing: $L \Subset_{\preceq\rho} M$.

For a behavior pattern λ, let R_λ denote the restriction of the weighted abstraction transformation R to abstraction with respect to λ. We say that two behavior patterns λ and λ' are *independent* iff: $R_\lambda; R_{\lambda'} = R_{\lambda'}; R_\lambda$.

Example 4. Let $\lambda := a \cdot c$ and $\lambda' := b \cdot c$ be two behavior patterns such that abstraction inserts the abstraction symbol after action c. Trace $a \cdot b \cdot c$ is abstracted into $a \cdot b \cdot c \cdot \lambda' \cdot \lambda$ by $R_\lambda; R_{\lambda'}$ and into $a \cdot b \cdot c \cdot \lambda \cdot \lambda'$ by $R_{\lambda'}; R_\lambda$ so these behavior patterns are not independent.

Then we get the following result.

Theorem 3. *Let $M := \exists Y.\, \lambda_1(\overline{x_1}) \wedge \neg(\exists Z.\, \lambda_2(\overline{x_2}))\, \mathbf{U}\, \lambda_3(\overline{x_3})$ be an abstract behavior where Y and Z are two disjoint sets of variables of sort Data and where $\lambda_2 \neq \lambda_1$, $\lambda_2 \neq \lambda_3$ and λ_2 is independent from λ_3. Let $w_2 \in S$ be the smallest weight of an abstraction rule inserting $\overline{\lambda_2}$. Then M has the property of $(2 + \rho/w_2, 1)$-completeness.*

Proof. Let $\varphi_M = \exists Y.\, \mathbf{F}(\lambda_1(\overline{x_1}) \wedge \neg(\exists Z.\, \lambda_2(\overline{x_2}))\, \mathbf{U}\, \lambda_3(\overline{x_3}))$.
Let's denote the actions $\lambda_1(\overline{x_1})$, $\lambda_2(\overline{x_2})$ and $\lambda_3(\overline{x_3})$ by α_1, α_2 and α_3 respectively.

Let $L \subseteq T_{Trace}(\mathcal{F}_\Sigma)$ be a set of traces. We need to show that:

$$\begin{gathered} L \Cap_{\preceq \rho} M \\ \Leftrightarrow \\ \exists t \in L, \exists t' \in T_{Trace}(\mathcal{F}), \exists i \leq 2 + \rho/w_2,\ R^i_{\rightsquigarrow}(t,t') \preceq \rho \\ \text{and} \\ \forall t'' \in R_u^{\leq 1}(t'),\ t''|_\Gamma \models \varphi_M. \end{gathered}$$

$\Rightarrow$: By definition of the exhibition of M by L, there exists a trace $t \in L$ with a partially abstracted form $\hat{t}$ by R such that $\hat{t}|_\Gamma$ validates φ_M, $w = R^*_{\rightsquigarrow}(t,\hat{t}) \preceq \rho$ and $\forall t'' \in R^*_u(\hat{t}),\ t''|_\Gamma \models \varphi_M$. So there exists an instantiation $\sigma_Y \in Inst_Y$ such that:

$$\hat{t} = t_1 \cdot \alpha_1\sigma_Y \cdot t_2 \cdot \alpha_3\sigma_Y \cdot t_3$$

where $t_1, t_2, t_3 \in T_{Trace}(\mathcal{F})$ and t_2 contains no instance of $\alpha_2\sigma_Y$.
Moreover, we decompose t_2 in order to identify all possible occurrences of $\overline{\alpha_2}\sigma_Y$:

$$t_2 = t_2^1 \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdot t_2^2 \cdots t_2^n \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot t_2^{n+1}$$

where $t_2^1 \ldots, t_2^{n+1} \in T_{Trace}(\mathcal{F})$, $\sigma_{Z,1}, \ldots, \sigma_{Z,n} \in Inst_Z$ and no instance of $\overline{\alpha_2}\sigma_Y$ appears in $t_2^1 \ldots, t_2^{n+1}$.
Observe that $n \leq \rho/w_2$. Indeed, if $w_2 = \overline{0}$, then no instance of $\overline{\alpha_2}$ could appear, whereas if $w_2 \neq \overline{0}$, each instance of $\overline{\alpha_2}\sigma_Y$ adds a weight of at least w_2 to the final weight of $\hat{t}$, which must be lower than ρ.

We first define a term t' with a weight less than ρ in $R^i_{\rightsquigarrow}(t)$ for some $i \leq 2 + \rho/w_2$ such that t' contains the same occurrence $\alpha_1\sigma_Y \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdots \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot \alpha_3\sigma_Y$ of M than $\hat{t}$ and then we show that its future abstractions until the order 1 still realize M.
By Corollary 1, any sequence of transformations by $\rightsquigarrow_R$ from t to $\hat{t} = t_1 \cdot \alpha_1\sigma_Y \cdot t_2^1 \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdot t_2^2 \cdots t_2^n \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot t_2^{n+1} \cdot \alpha_3\sigma_Y \cdot t_3$ has the weight $w = R^*(t,\hat{t}) \preceq \rho$. Since $w \preceq \rho \neq \overline{0}$, there exists at least one such sequence.
Then, by applying Lemma 3 to this sequence, there exists traces u_1, u_2^1, …, u_2^{n+1}, u_3 and a weight w' such that: $t \overset{w'}{\rightsquigarrow}{}^*_R u_1 \cdot \alpha_1\sigma_Y \cdot u_2^1 \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdot u_2^2 \cdots u_2^n \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot u_2^{n+1} \cdot \alpha_3\sigma_Y \cdot u_3$ in $n+2$ steps. Moreover, in the tropical semiring, $w' \preceq w \preceq \rho$.
Therefore we define:

$$t' = u_1 \cdot \alpha_1\sigma_Y \cdot u_2^1 \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdot u_2^2 \cdots u_2^n \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot u_2^{n+1} \cdot \alpha_3\sigma_Y \cdot u_3.$$

Since $t \in L$ is concrete, $u_1, u_2^1, \ldots, u_2^{n+1}, u_3$ contain no abstract action, so $t'|_\Gamma = \alpha_1\sigma_Y \cdot \overline{\alpha_2}\sigma_Y\sigma_{Z,1} \cdots \overline{\alpha_2}\sigma_Y\sigma_{Z,n} \cdot \alpha_3\sigma_Y \models \varphi_M$. Finally, let $i = n+2$. Therefore, $t \overset{w'}{\rightsquigarrow}{}^i_R t'$ and, by Corollary 1, $R^i_{\rightsquigarrow}(t,t')$ is the weight of any sequence of transformations of t in t' by R, so $R^i_{\rightsquigarrow}(t,t') = w'$. Moreover, we observed previously that $n \leq \rho/w_2$, so $i \leq 2 + \rho/w_2$. Therefore, with our previous observation that $w' \preceq \rho$:

$$\exists t' \in T_{Trace}(\mathcal{F}), \exists i \leq 2 + \rho/w_2,\ R^i(t,t') \preceq \rho.$$

We now show that: $\forall t'' \in R_u^{\leq 1}(t'),\ t''|_\Gamma \models \varphi_M$. We already have: $t'|_\Gamma \models \varphi_M$. Assume that there exists a $t'' \in R_u(t')|_\Gamma$ such that $t'' \not\models \varphi_M$ i.e., that t' can be rewritten by R_u in such a way that an action $\alpha_2\sigma_Y\sigma'_Z$ is inserted within a subterm u_2^i of t' for some instantiation $\sigma'_Z \in Inst_Z$. Let p_1 be the concrete position of $\alpha_1\sigma_Y$ in t' and p_3 be the concrete position of $\alpha_3\sigma_Y$ in t'. The occurrence of the behavior pattern related to this insertion appears in t and can be abstracted in t at the same concrete position, that is between p_1 excluded and p_3 included. So there are two cases for $\hat{t}$:

- This occurrence has already been abstracted in $\hat{t}$, so an action $\alpha_2\sigma_Y\sigma'_Z$ or $\overline{\alpha_2}\sigma_Y\sigma'_Z$ has been inserted at a concrete position between p_1 excluded and p_3 included in a term from the abstract derivation from t to $\hat{t}$. Since behavior patterns λ_2 and λ_3 are independent, their abstractions cannot take place at the same concrete position, so the occurrence could not be abstracted at the position p_3. Hence, the abstract action was inserted at a concrete position between p_1 excluded and p_3 excluded. Hence it appears in t_2.
  Finally, as t_2 cannot contain, by hypothesis, any instance of $\alpha_2\sigma_Y$, the inserted abstract action must be one of the actions $\overline{\alpha_2}\sigma_Y\sigma_{Z,i}$ identified in $\hat{t}$. But when applying Lemma 3 to construct t', we already considered the associated behavior pattern occurrence. Hence, the inserted abstract action cannot be an action $\overline{\alpha_2}\sigma_Y\sigma'_Z$ either.
- This occurrence has not yet been abstracted in $\hat{t}$. Then it can be abstracted in $\hat{t}$ at a concrete position between p_1 excluded and p_3 included, that is after a concrete action of t_2 and before the action $\alpha_3\sigma_Y$. So this results in a trace t'' whose projection on Γ does not realize M, contradicting the hypothesis on $\hat{t}$: $\forall t'' \in R_u^*(\hat{t}),\ t''|_\Gamma \models \varphi_M$.

$\Leftarrow$: We reason by contradiction. Let $t \in L$, $t' \in T_{Trace}(\mathcal{F})$ and $i \leq 2 + \rho/w_2$ such that: $R^i(t,t') = w \preceq \rho$ and $\forall t'' \in R_u^{\leq 1}(t'),\ t''|_\Gamma \models \varphi_M$. Assuming that L does not exhibit M, we construct a trace $t'_1 \in R_u^{\leq 1}(t')$ which does not realize M, contradicting the fact that $\forall t'' \in R_u^{\leq 1}(t'),\ t''|_\Gamma \models \varphi_M$.
In particular, $t'|_\Gamma \models \varphi_M$ so there exists an instantiation $\sigma_Y \in Inst_Y$ such that we can decompose t' into

$$t' = t_1 \cdot \alpha_1\sigma_Y \cdot t_2 \cdot \alpha_3\sigma_Y \cdot t_3$$

where $t_1, t_2, t_3 \in T_{Trace}(\mathcal{F})$ and t_2 contains no instance of $\alpha_2\sigma_Y$.
Assume L does not exhibit M. Then, by Definition 14, since $R^*(t,t') \preceq R^i(t,t') \preceq \rho$ in the tropical semiring, there must exist a trace $t'' \in R_u^*(t')$ such that $t''|_\Gamma \not\models \varphi_M$. By definition of M, there must exist an instantiation $\sigma_Z \in Inst_Z$ such that an abstract action $\alpha_2\sigma_Y\sigma_Z$ has been inserted into a term of the derivation from t' to t'' by $\rightarrow_{R_u}$, at a concrete position p between $\alpha_1\sigma_Y$ and $\alpha_3\sigma_Y$. By Lemma 3, we could have inserted this action $\alpha_2\sigma_Y\sigma_Z$ directly in the term t', at the same concrete position p, that is between actions $\alpha_1\sigma_Y$ and $\alpha_3\sigma_Y$. Let t''' be the term we would have obtained. Then $t'''|_\Gamma \not\models \varphi_M$, contradicting the hypothesis $\forall t'' \in R_u^{\leq 1}(t')\big|_\Gamma,\ t'' \models \varphi_M$.

Observe that we can refine the bound in the previous theorem by considering the weights w_1 and w_3 representing the minimum weights associated to the abstraction of an occurrence of λ_1 and λ_3 respectively. Then, the considered behavior has a property of $(2 + (\rho - w_1 - w_3)/w_2, 1)$-completeness.
We notice moreover that, when abstraction rules of the behavior pattern associated to λ_2 are weighted by $\overline{1}$ (i.e. abstraction is certain for λ_2), the action $\overline{\lambda_2}$ is never inserted so $w_2 = \overline{0} = +\infty$ in the tropical semiring, which entails that the behavior has the property of $(2, 1)$-completeness.

5 Rational Abstraction

By Definition 10, the abstraction transformation is a weighted transformation of $T_{Trace}(\mathcal{F}) \times T_{Trace}(\mathcal{F}) \rightarrow S$. We show that this transformation is realized by a weighted linear non-deleting top-down tree transducer, i.e. that it is rational.
In order to prove the rationality results, we define an alphabet Γ' distinct from Γ in bijection with alphabet Γ and we adapt abstraction, so that the inserted abstract action is defined on Γ'. This allows us to isolate in an abstract term the inserted abstract action from abstract actions existing prior to the abstraction. Thus, let's denote by

$\mathcal{F}'$ the extended alphabet $\mathcal{F} \cup \Gamma'$. For $\lambda \in \Gamma$, we denote by $\lambda' \in \Gamma'$ its associated symbol, and for an action $\alpha \in T_{Action}(\mathcal{F}_\Gamma)$, we denote by $\alpha' \in T_{Action}(\mathcal{F}'_{\Gamma'})$ its associated action on Γ'. Finally, we define the relabelings $l_{\Gamma' \to \Gamma} : T_{Trace}(\mathcal{F}') \to T_{Trace}(\mathcal{F})$ and $l_{\Gamma \to \Gamma'} : T_{Trace}(\mathcal{F}) \to T_{Trace}(\mathcal{F}')$.
The abstraction transformation inserting actions of Γ' is then defined in the following way.

Definition 20. *Let R be a sound weighted abstraction transformation over a semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$. The* weighted abstraction transformation induced by R on Γ' *over the semiring $(S, \oplus, \otimes, \overline{0}, \overline{1})$ is the weighted transformation $R' : T_{Trace}(\mathcal{F}) \times T_{Trace}(\mathcal{F}') \to S$ defined by:*

$$\begin{gathered}\forall t_1, t_2 \in T_{Trace}(\mathcal{F}), \forall \alpha \in T_{Action}(\mathcal{F}_\Gamma), \\ t_1 \cdot t_2 \overset{w}{\rightsquigarrow}_{R'} t_1 \cdot \alpha' \cdot t_2 \\ \Leftrightarrow \\ t_1 \cdot t_2 \overset{w}{\rightsquigarrow}_{R} t_1 \cdot \alpha \cdot t_2.\end{gathered}$$

We first show the rationality of R' and we then deduce the rationality of R. To this end, we use the following definition and lemma.

Definition 21. *Let $\Omega \subseteq \mathcal{F}_a$ and $\Omega' \subseteq \mathcal{F}_a$ be two disjoint sets of function symbols. Let $s : T_{Trace}(\mathcal{F}_{\Omega'}) \to S$ be a weighted set. The* Ω-generalized form *of s, denoted by $\amalg_\Omega(s)$, is the weighted set s' of $T_{Trace}(\mathcal{F}_{\Omega \cup \Omega'})$ defined by:*

$$\forall t \in T_{Trace}(\mathcal{F}_{\Omega \cup \Omega'}), \; s'(t) = s\left(t|_{\Omega'}\right).$$

$\amalg_\Omega(s)$ denotes the inverse projection of the weighted set. In a sense, this amounts to randomly injecting actions of Ω into the terms of s, without modifying their weight.
For instance, on unweighted sets, let $\Omega' = \{a, b\}$ and $\Omega = \{c, d\}$ and let s be the set $s = \{a \cdot b, c\}$. Then:

$$\begin{aligned}\amalg_\Omega(s) = {} & T_{Trace}(\mathcal{F}_\Omega) \cdot a \cdot T_{Trace}(\mathcal{F}_\Omega) \cdot b \cdot T_{Trace}(\mathcal{F}_\Omega) \\ & \cup \\ & T_{Trace}(\mathcal{F}_\Omega) \cdot c \cdot T_{Trace}(\mathcal{F}_\Omega).\end{aligned}$$

Lemma 4. *Let $\Omega \subseteq \mathcal{F}_a$ and $\Omega' \subseteq \mathcal{F}_a$ be two disjoint sets of function symbols and let $s : T_{Trace}(\mathcal{F}_{\Omega'}) \to S$ be a weighted set recognized by a weighted tree automaton A. Then $\amalg_\Omega(s)$ is recognized by a weighted tree automaton of size $O(|A|)$.*

Proof. We define a weighted top-down tree transducer $\tau = (\mathcal{F}_{\Omega'}, \mathcal{F}_{\Omega \cup \Omega'}, \{q_t, q_a, q_d\}, q_t, \Delta)$ such that: $\|\tau\|(s) = \amalg_\Omega(s)$.
Δ is composed of the following rules:

- $q_t(\cdot(x_1, x_2)) \to \cdot(q_a(x_1), q_t(x_2))$;
- $q_t(\epsilon) \to \epsilon$;
- for all $f \in \Omega'$ of arity $k \in \mathbb{N}$, Δ contains the rule: $q_a(f(x_1, \ldots, x_k)) \to f(q_d(x_1), \ldots, q_d(x_k))$;
- for all $d \in \mathcal{F}_d$, Δ contains the rule: $q_d(d) \to d$;
- for all term $\alpha \in T_{Action}(\mathcal{F}_\Omega)$, Δ contains the rule: $q_t(x) \to \cdot(\alpha, q_t(x))$.

The transducer τ realizes $\amalg_\Omega$ and is linear, nondeleting and of constant size with respect to A. Application of Proposition 3 thereby entails that the weighted tree language $\amalg_\Omega(s) = \tau(s)$ is recognized by a weighted tree automaton of size $O(|\tau| \cdot |A|) = O(|A|)$.

Lemma 5. *Let R be a sound weighted abstraction transformation and R' be the weighted transformation induced by R over Γ'. Let $I = R'(T_{Trace}(\mathcal{F}))$ be the image of $T_{Trace}(\mathcal{F})$ by $\rightsquigarrow_{R'}$. Then:*

1. *For all $t \in T_{Trace}(\mathcal{F})$, $t' \in T_{Trace}(\mathcal{F}')$ and for all $w \in S \setminus \{\overline{0}\}$:*
$$t \overset{w}{\rightsquigarrow}_{R'} t' \Rightarrow t = t'|_{\Sigma \cup \Gamma}.$$
2. *For all $t' \in T_{Trace}(\mathcal{F}')$:*
$$I(t') = R'(t'|_{\Sigma \cup \Gamma}, t').$$
3. *For all $t' \in T_{Trace}(\mathcal{F}')$:*
$$t'|_{\Sigma \cup \Gamma} \overset{I(t')}{\rightsquigarrow}_{R'} t'.$$

Proof. Let's prove the first result.
As $w \neq \overline{0}$, reduction by R' operates as defined in Definition 20, so there exists $t_1, t_2 \in T_{Trace}(\mathcal{F})$ and $\alpha \in T_{Action}(\mathcal{F}')$ such that: $t = t_1 \cdot t_2$ and $t' = t_1 \cdot \alpha' \cdot t_2$. Hence, the trace t' only differs from t by an abstract action of $T_{Action}(\mathcal{F}'_{\Gamma'})$. As $t \in T_{Trace}(\mathcal{F})$ and $\alpha' \in T_{Action}(\mathcal{F}_{\Gamma'})$, we have: $t'|_{\Sigma \cup \Gamma} = t$.

Let's prove the second result.

$$I(t') = R'(T_{Trace}(\mathcal{F}))(t') = \bigoplus_{\substack{t \in T_{Trace}(\mathcal{F}), \\ t \overset{w}{\rightsquigarrow}_{R'} t'}} w = \bigoplus_{t \in T_{Trace}(\mathcal{F})} R'(t, t').$$

Two cases are then possible:

- Either $I(t') = \overline{0}$, in which case, by the above equality: $\forall t \in T_{Trace}(\mathcal{F}), R'(t, t') = \overline{0}$ and in particular: $R'(t'|_{\Sigma \cup \Gamma}, t') = \overline{0}$. Hence the result.
- Or $I(t') \neq \overline{0}$, in which case, by the above equality: $\exists t \in T_{Trace}(\mathcal{F}), R'(t, t') \neq \overline{0}$, i.e. $t \overset{R'(t,t')}{\rightsquigarrow}_{R'} t'$ with $R'(t, t') \neq \overline{0}$. By the first result, the only term t such that $R'(t, t') \neq \overline{0}$ is the term $t'|_{\Sigma \cup \Gamma}$. So the sum $\bigoplus_{t \in T_{Trace}(\mathcal{F})} R'(t, t')$ is reduced to $R'(t'|_{\Sigma \cup \Gamma}, t')$.

Finally, the third result is a reformulation of the second result.

Let R be a weighted abstraction system associated to a behavior pattern B. We define the weighted set of instances of the right-hand side of a rule $A_i(X) \cdot B_i(X) \cdot y \rightarrow \{ A_i(X) \cdot \lambda(\overline{x}) \cdot B_i(X) \cdot y : w_i, A_i(X) \cdot \overline{\lambda}(\overline{x}) \cdot B_i(X) \cdot y : w'_i \}$ as the weighted set associating to each term of $\bigcup_{\sigma \in Inst_{X \cup \{y\}}} A_i(X) \cdot \lambda(\overline{x}) \cdot B_i(X) \cdot y$ a weight w_i and to each term of $\bigcup_{\sigma \in Inst_{X \cup \{y\}}} A_i(X) \cdot \overline{\lambda}(\overline{x}) \cdot B_i(X) \cdot y$ a weight w'_i. We then define the weighted set of instances of right-hand sides of rules as the union of weighted sets of each right-hand side of rule. The weight of a term t in this set is thus the sum of the weights of t in each set of instances.

Lemma 6. *Let B be a behavior pattern and R be a sound weighted abstraction transformation with respect to B defined from an abstraction system whose set of weighted instances of right-hand sides of rules is recognized by a weighted tree automaton A_R. Then:*

- *The weighted transformation R' induced by R on Γ' is rational.*
- *For any weighted set s of $T_{Trace}(\mathcal{F})$ recognized by a weighted trace automaton A, $R'(s)$ is recognized by a weighted trace automaton of size $O(|A| \cdot |A_R|)$.*

Proof. First, in order to simplify the proof, we slightly modify the abstraction system associated to B so that filtering of left-hand sides of rules during abstraction covers whole traces. Observe first that this filtering already covers traces on their right. So we slightly modify the sets defining the rules associated to the behavior pattern B by concatenating on their left the sets A_i with $T_{Trace}(\mathcal{F}_\Sigma)$. For instance, for the behavior pattern $\lambda := a$, we define $A_i = T_{Trace}(\mathcal{F}_\Sigma) \cdot a$ and $B_i = \{\epsilon\}$, instead of $A_i = \{a\}$ and $B_i = \{\epsilon\}$. Recall that a behavior pattern is defined on $T_{Trace}(\mathcal{F}_\Sigma)$. This modification does not alter the abstraction transformation, it has a constant impact on the size of automatonA_R and, above all, it allows us to simplify the definition of the weighted reduction relation generated by R (see Definition 8) in the following way:

$$\begin{array}{c} \forall t, t' \in T_{Trace}(\mathcal{F}), \forall w \in S, \\ t \xrightarrow{w}_R t' \\ \Leftrightarrow \\ \exists \sigma \in Inst_{X \cup \{y\}}, \exists i \in [1..n], \exists \mu \in \Gamma, (\mu, w) \in \left\{(\lambda, w_i), \left(\overline{\lambda}, w_i'\right)\right\}, \\ \exists a \in T_{Trace}(\mathcal{F}) \cdot T_{Action}(\mathcal{F}_\Sigma), \exists b \in T_{Trace}(\mathcal{F}), \\ a|_\Sigma \in A_i(X)\sigma, b|_\Sigma \in B_i(X)\sigma, t = a \cdot b \cdot y\sigma \text{ and } t' = a \cdot \mu(\overline{x})\sigma \cdot b \cdot y\sigma. \end{array} \tag{1}$$

We deduce the following intermediary result when trace t is concrete:

$$\begin{array}{c} \forall t \in T_{Trace}(\mathcal{F}_\Sigma), \forall t' \in T_{Trace}(\mathcal{F}), \forall w \in S, \\ t \xrightarrow{w}_R t' \\ \Leftrightarrow \\ t' \text{ is an instance of a right-hand side of rule of } R \text{ of weight } w. \end{array} \tag{2}$$

Indeed, by (1), t' can be written $t' = a \cdot \mu(\overline{x})\sigma \cdot b \cdot y\sigma$ with $a|_\Sigma \in A_i(X)\sigma$ and $b|_\Sigma \in B_i(X)\sigma$. And since $t = a \cdot b \cdot y\sigma$ is concrete, we have $a|_\Sigma = a$, $b|_\Sigma = b$, $y\sigma \in T_{Trace}(\mathcal{F}_\Sigma)$ and: $t' \in \sigma(A_i(X) \cdot \mu(\overline{x}) \cdot B_i(X) \cdot y)$.

Let C be the weighted set of instances of right-hand sides of rules of R.
We construct a transducer realizing $\rightsquigarrow_{R'}$, in the following way.
Let I be the image of $T_{Trace}(\mathcal{F})$ by $\rightsquigarrow_{R'}$: $I = R'(T_{Trace}(\mathcal{F}))$.
First, we show that I is a regular set which can be expressed in terms of the weighted set $l_{\Gamma \to \Gamma'}(C)$.
Second, we simulate the weighted transformation $\rightsquigarrow_{R'}$ by the functional composition of two transformations:

- a first unweighted transformation R_{pick}, which injects, at a random location, an action α' of $T_{Action}(\mathcal{F}'_{\Gamma'})$ into the trace $t \in T_{Trace}(\mathcal{F})$ to abstract, yielding a trace t';
- a second weighted transformation Id_I which realizes the identity transformation over the weighted set I previously constructed and which therefore guarantees that t' is in the image of $T_{Trace}(\mathcal{F})$ by R'.

Let's prove the first point. We express the set $I = R'(T_{Trace}(\mathcal{F}))$ in terms of the set C of instances of right-hand sides of rules of R.
To this end, we define the unweighted set $Valid$ of traces verifying the soundness condition of Definition 12 applied to R':

$$\begin{array}{rl} Valid = & \bigcup_{\alpha \in T_{Action}(\mathcal{F}_\Gamma)} T_{Trace}(\mathcal{F}) \cdot T_{Action}(\mathcal{F}_\Sigma) \cdot \alpha' \cdot \\ & (T_{Trace}(\mathcal{F}) \setminus (T_{Trace}(\mathcal{F}_\Gamma) \cdot \{\alpha, \overline{\alpha}\} \cdot T_{Trace}(\mathcal{F}))). \end{array}$$

Let $u, v \in T_{Trace}(\mathcal{F})$ be two terms and $\alpha \in T_{Action}(\mathcal{F}_\Gamma)$ be an action. We denote by α' its associated action in $T_{Action}(\mathcal{F}'_{\Gamma'})$.

Assume we have, for some non-null weight $w \in S \setminus \{\overline{0}\}$:

$$u \cdot v \overset{w}{\rightsquigarrow}_{R'} u \cdot \alpha' \cdot v.$$

Then, as w is not null and by definition of R', we have, equivalently:

$$u \cdot v \overset{w}{\rightsquigarrow}_{R} u \cdot \alpha \cdot v.$$

As R is a sound abstraction transformation, we then know, equivalently, that:

- First, the action α, inserted by R in $u \cdot v$, verifies the soundness condition, i.e. the following abstract actions do not contain α or $\overline{\alpha}$. In other words, by definition of $Valid$:
$$u \cdot \alpha' \cdot v \in Valid.$$
- Second, there exists n reduction steps $u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v$ and w is given by: $w = \bigoplus_{i \in [1..n]} w_i$.

Finally, we have:

$$\begin{array}{c} u \cdot v \overset{w}{\rightsquigarrow}_{R'} u \cdot \alpha' \cdot v \\ \Leftrightarrow \\ u \cdot \alpha' \cdot v \in Valid \text{ and} \\ \text{there exists } n \text{ reduction steps } u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v \text{ and } w = \bigoplus_{i \in [1..n]} w_i. \end{array} \tag{3}$$

Let's consider one of these reduction steps: $u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v$. Then, by definition of the reduction relation R, we have:

$$u \cdot \alpha' \cdot v \in Valid \text{ and } \ u|_{\Sigma} \cdot v|_{\Sigma} \overset{w_i}{\rightarrow}_R \ u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}\,.$$

Conversely, if $u|_{\Sigma} \cdot v|_{\Sigma} \overset{w_i}{\rightarrow}_R u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}$, then $u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v$, under the condition that the inserted action α verifies the soundness condition, which is the case since, as was said previously, this amounts to requiring that $u \cdot \alpha' \cdot v \in Valid$, which is true by hypothesis. Hence, applying formula (2), $u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}$ is equivalently an instance of a right-hand side of a rule of R of weight w_i.

To sum up:

$$\begin{array}{c} u \cdot \alpha' \cdot v \in Valid \text{ and } u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v \\ \Leftrightarrow \\ u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma} \text{ is an instance of a right-hand side} \\ \text{of a rule of } R \text{ of weight } w_i. \end{array}$$

In other words, when $u \cdot \alpha' \cdot v \in Valid$, each instance $u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}$ of a right-hand side of a rule of R of weight w_i identifies a reduction step $u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v$, and conversely.

Yet, by definition of C, $C(u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma})$ is the sum of the weights w_i such that $u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}$ is an instance of a right-hand side of a rule of R of weight w_i. Thus $C(u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma})$ is also the sum of the weights w_i of each reduction step $u \cdot v \overset{w_i}{\rightarrow}_R u \cdot \alpha \cdot v$.

Going back to (3), we deduce:

$$\begin{array}{c} u \cdot v \overset{w}{\rightsquigarrow}_{R'} u \cdot \alpha' \cdot v \\ \Leftrightarrow \\ u \cdot \alpha' \cdot v \in Valid \text{ and} \\ w = C(u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}). \end{array}$$

Finally, we observe that:

$$C(u|_{\Sigma} \cdot \alpha \cdot v|_{\Sigma}) = l_{\Gamma \to \Gamma'}(C)(u|_{\Sigma} \cdot \alpha' \cdot v|_{\Sigma}). \tag{4}$$

Indeed, by definition of the application of a weighted transformation to a weighted set (see Section 2):

$$l_{\Gamma\to\Gamma'}(C)(u|_\Sigma\cdot\alpha'\cdot v|_\Sigma)=\bigoplus_{t\in T_{Trace}(\mathcal{F}),\, t\overset{w}{\rightsquigarrow}l_{\Gamma\to\Gamma'}\, u|_\Sigma\cdot\alpha'\cdot v|_\Sigma} w\otimes C(t)$$

and the only term t such that $t\overset{w}{\rightsquigarrow}l_{\Gamma\to\Gamma'}\ u|_\Sigma\cdot\alpha'\cdot v|_\Sigma$ with $w\neq\overline{0}$ is $t=u|_\Sigma\cdot\alpha\cdot v|_\Sigma$. And in this case, $w=\overline{1}$ since the transformation is a relabeling which does not modify weights.
Finally, applying (4):

$$\begin{array}{c} u\cdot v\overset{w}{\rightsquigarrow}_{R'}\ u\cdot\alpha'\cdot v\\ \Leftrightarrow\\ u\cdot\alpha'\cdot v\in Valid \text{ and}\\ w=l_{\Gamma\to\Gamma'}(C)(u|_\Sigma\cdot\alpha'\cdot v|_\Sigma).\end{array}\tag{5}$$

Let Id_{Valid} be the identity transformation over the set $Valid$. We show that the weighted set $I=R'\left(T_{Trace}\left(\mathcal{F}\right)\right)$ is identical to the weighted set I' defined by:

$$I'=Id_{Valid}\left(\Pi_\Gamma\left(l_{\Gamma\to\Gamma'}\left(C\right)\right)\right).$$

First, for all $t'\in T_{Trace}\left(\mathcal{F}'\right)$ which cannot be written $t'=u\cdot\alpha'\cdot v$ with $u,v\in T_{Trace}\left(\mathcal{F}\right)$ and $\alpha'\in T_{Action}\left(\mathcal{F}'_{\Gamma'}\right)$, we observe that:

$$I(t')=I'(t')=\overline{0}.\tag{6}$$

Indeed, t' cannot be in the image of $T_{Trace}\left(\mathcal{F}\right)$ by R' so $I(t')=\overline{0}$ and, moreover, t' cannot be in $Valid$ so $I'(t')=\overline{0}$.
Let's now consider a term $t'\in T_{Trace}\left(\mathcal{F}'\right)$ which can be written: $t'=u\cdot\alpha'\cdot v$.
Let s be a weighted set of $T_{Trace}\left(\mathcal{F}'\right)$. We first observe that, by definition of the identity transformation and given that $Valid$ is an unweighted set (i.e. its element all have weight $\overline{1}$), we have:

$$\begin{aligned} Id_{Valid}(s)(t')&=\textstyle\bigoplus_{t\in T_{Trace}(\mathcal{F}\cup\Gamma'),\, t\overset{w}{\rightsquigarrow}_{Id_{Valid}}t'} s(t)\otimes w\\ &=\begin{cases}s(t') & \text{if } t'\in Valid\\ \overline{0} & \text{otherwise.}\end{cases}\end{aligned}$$

Applying this result to $I'(t')$, we have:

$$I'\left(t'\right)=\begin{cases}\Pi_\Gamma\left(l_{\Gamma\to\Gamma'}\left(C\right)\right)\left(t'\right) & \text{if } t'\in Valid\\ \overline{0} & \text{otherwise}\end{cases}$$

$$\Leftrightarrow$$

$$I'\left(t'\right)=\begin{cases}w & \text{if } t'\in Valid, \text{ if } w=\Pi_\Gamma\left(l_{\Gamma\to\Gamma'}\left(C\right)\right)\left(u\cdot\alpha'\cdot v\right) \text{ and if } w\neq\overline{0}\\ \overline{0} & \text{otherwise}\end{cases}$$

$$\Leftrightarrow$$

$$\text{by definition of } \Pi_\Gamma$$

$$I'\left(t'\right)=\begin{cases}w & \text{if } t'\in Valid, \text{ if } w=l_{\Gamma\to\Gamma'}\left(C\right)\left(u|_\Sigma\cdot\alpha'\cdot v|_\Sigma\right) \text{ and if } w\neq\overline{0}\\ \overline{0} & \text{otherwise}\end{cases}$$

$$\Leftrightarrow$$

$$\text{by (5)}$$

$$I'\left(t'\right)=\begin{cases}w & \text{if } u\cdot v\overset{w}{\rightsquigarrow}_{R'}\ u\cdot\alpha'\cdot v \text{ and } w\neq\overline{0}\\ \overline{0} & \text{otherwise}\end{cases}$$

$$\Leftrightarrow$$

$$\text{by Lemma 5}$$

$$I'\left(t'\right)=\begin{cases}I\left(u\cdot\alpha'\cdot v\right) & \text{if } I\left(u\cdot\alpha'\cdot v\right)\neq\overline{0}\\ \overline{0} & \text{otherwise.}\end{cases}$$

Thus, with (6), we have as expected, on $T_{Trace}(\mathcal{F}')$: $I' = I$. So, by definition of I':

$$I = Id_{Valid}\left(\amalg_{\Gamma}\left(l_{\Gamma\to\Gamma'}(C)\right)\right). \tag{7}$$

Let's now prove the second point. We show that:

$$R' = R_{pick}; Id_I. \tag{8}$$

Recall that the transformation R_{pick} is the transformation which randomly injects an action of $T_{Action}(\mathcal{F}'_{\Gamma'})$ into the trace to abstract. In other words, for all $t \in T_{Trace}(\mathcal{F})$ and $t' \in T_{Trace}(\mathcal{F}')$:

$$\begin{gathered} t \overset{w}{\rightsquigarrow}_{R_{pick}} t' \\ \Leftrightarrow \\ w = \begin{cases} \overline{1} & \text{if } \exists \alpha' \in T_{Action}(\mathcal{F}'_{\Gamma'}), \exists t_1, t_2 \in T_{Trace}(\mathcal{F}), \\ & \quad t = t_1 \cdot t_2, t' = t_1 \cdot \alpha' \cdot t_2 \\ \overline{0} & \text{otherwise.} \end{cases} \end{gathered} \tag{9}$$

By definition of the functional composition of weighted transformations, we have, for all $t \in T_{Trace}(\mathcal{F})$ and $t' \in T_{Trace}(\mathcal{F}')$:

$$\begin{gathered} t \overset{w}{\rightsquigarrow}_{R_{pick};Id_I} t' \\ \Leftrightarrow \\ w = \bigoplus_{t'' \in T_{Trace}(\mathcal{F}')} R_{pick}(t, t'') \otimes Id_I(t'', t') \end{gathered}$$

By definition of the identity transformation over a set, we have, for all $t'' \neq t'$ of $Id_I(t'', t') = \overline{0}$. We deduce:

$$\begin{gathered} t \overset{w}{\rightsquigarrow}_{R_{pick};Id_I} t' \\ \Leftrightarrow \\ w = R_{pick}(t, t') \otimes Id_I(t', t') = R_{pick}(t, t') \otimes I(t') \\ \Leftrightarrow \\ \text{by (9)} \\ w = \begin{cases} \overline{0} & \text{if } R_{pick}(t, t') = \overline{0} \\ I(t') & \text{otherwise} \end{cases} \\ \Leftrightarrow \\ \text{by Lemma 5} \\ w = \begin{cases} \overline{0} & \text{if } R_{pick}(t, t') = \overline{0} \\ R'\left(t'|_{\Sigma\cup\Gamma}, t'\right) & \text{otherwise.} \end{cases} \end{gathered}$$

But, if $R_{pick}(t, t') = \overline{0}$, then, by definition of R_{pick}, t' cannot be obtained from t by inserting an abstract action of $T_{Action}(\mathcal{F}'_{\Gamma'})$ in t so, necessarily: $R'(t, t') = \overline{0}$. And if $R_{pick}(t, t') \neq \overline{0}$, then, by definition of R_{pick}: $t'|_{\Sigma\cup\Gamma} = t$.
Thus, we get:

$$\begin{gathered} t \overset{w}{\rightsquigarrow}_{R_{pick};Id_I} t' \\ \Leftrightarrow \\ w = R'(t, t') \\ \Leftrightarrow \\ t \overset{w}{\rightsquigarrow}_{R'} t'. \end{gathered}$$

To sum up, we have, with (7):

$$I = Id_{Valid}\left(\amalg_{\Gamma}\left(l_{\Gamma\to\Gamma'}(C)\right)\right)$$

and, with (8):

$$R' = R_{pick}; Id_I.$$

We can now **conclude**.
First, we show that I is regular and recognized by a weighted trace automaton of size $O(|A_R|)$. As the homomorphism $l_{\Gamma\to\Gamma'}$ is a relabeling, by Proposition 1, it is realized by a trace transducer of constant size. As the weighted set C is recognized by hypothesis by automatonA_R, by Proposition 3, the weighted set $l_{\Gamma\to\Gamma'}(C)$ is recognized by a weighted trace automaton of size $O(|A_R|)$. By Lemma 4, the weighted set $\amalg_\Gamma(l_{\Gamma\to\Gamma'}(C))$ is thus recognized by a weighted trace automaton of size $O(|A_R|)$. As the set $Valid$ is recognized by an automaton of constant size, by Proposition 2, the transformation Id_{Valid} is realized by a transducer of constant size. The weighted set $I = Id_{Valid}(\amalg_\Gamma(l_{\Gamma\to\Gamma'}(C)))$ is thus recognized by a weighted trace automaton of size $O(|A_R|)$, by Proposition 3.
Hence, by Proposition 2, there exists a weighted trace transducer realizing Id_I, of size $O(|A_R|)$. Finally, R_{pick} is rational and realized by a weighted trace transducer of constant size.
First, as rational weighted transformations are closed by functional composition, we deduce that the relation $R' = R_{pick}; Id_I$ is rational.
Second, we deduce that, for all weighted trace automaton A, the weighted set $R'(\|A\|) = Id_I(R_{pick}(\|A\|))$ is recognized by a weighted trace automaton of size $O(|A_R| \cdot |A|)$, by applying twice Proposition 3.

We deduce from Lemma 6 rationality of R, which is expressed in the following theorem.

Theorem 4. *Let B be a behavior pattern and R be a sound weighted abstraction transformation with respect to B defined from an abstraction system whose union of the weighted sets of instances of right-hand sides of rules is recognized by a weighted tree automaton A_R. Then:*
- *R_u and R_u^{-1} are rational and realized by tree transducers of size $O(|A_R|)$.*
- *R is rational and, for any weighted tree automaton A, $R(\|A\|)$ is recognized by a weighted tree automaton of size $O(|A| \cdot |A_R|)$.*

Proof. As relation R_u is an unweighted abstraction relation in the sense of our previous article [REF], rationality of R_u and of R_u^{-1} and existence of two unweighted transducers of size $O(|A_{R_u}|) = O(|A_R|)$ realizing them are direct consequences of Theorem XX in [REF].
Let R' be the weighted transformation induced by R on Γ'. Then: $R = R'; l_{\Gamma'\to\Gamma}$.
By Proposition 1, the relabeling $l_{\Gamma'\to\Gamma}$ is realized by a tree transducer of constant size.
By Lemma 6, R' is rational and, for all weighted tree automaton A, the weighted set $R'(\|A\|)$ is recognized by a weighted tree automaton of size $O(|A| \cdot |A_R|)$. We deduce that the weighted set $R(\|A\|) = l_{\Gamma'\to\Gamma}(R'(\|A\|))$ is recognized by a weighted tree automaton of size $O(|A| \cdot |A_R|)$, by Proposition 3.
As transformations $l_{\Gamma'\to\Gamma}$ and R' are rational, the transformation R is also rational.

Using the set of traces n-exhibiting M, we obtain the following complexity for the detection of M, which remains linear in the size of the automaton recognizing the program set of traces, as was the case in the unweighted approach.

Theorem 5. *Let R be a rational sound weighted abstraction transformation over a semiring $\left(S, \oplus, \otimes, \overline{0}, \overline{1}\right)$, defined from an abstraction system whose union of the weighted sets of instances of right-hand sides of rules is recognized by a weighted tree automaton A_R. Let M be a regular abstract behavior with the property of (m,n)-completeness and A_M be a tree automaton recognizing the set of traces n-exhibiting M.*
Deciding, for some regular set of traces L recognized by a tree automaton A, whether L exhibits M with respect to a threshold $\rho \in S \setminus \left\{\overline{0}\right\}$ takes
$O\left(|A_R|^{m\cdot(m+1)/2} \times |A| \times |A_M|\right)$ *time and space.*

Proof. The proof of Theorem 1 relied on the following result:

$$L \Cap_{\preceq \rho} M \\ \Leftrightarrow \\ \exists t \in T_{Trace}\left(\mathcal{F}\right), Id_{\|A_M\|}\left(R_{\rightsquigarrow}^{\leq m}\left(L\right)\right)\left(t\right) \preceq \rho.$$

For $i \in [1..m]$, $R_{\rightsquigarrow}^{i}\left(L\right)$ is recognized by a weighted tree automaton of size $O\left(|A_R|^i \times |A|\right)$, by applying i times Theorem 4. So there exists a weighted tree automaton of size $O\left(|A_R|^{m\cdot(m+1)/2} \times |A|\right)$ recognizing $R_{\rightsquigarrow}^{\leq m}\left(L\right)$.
By Proposition 2, the transformation $Id_{\|A_M\|}$ is realized by a weighted tree transducer of size $O\left(|A_M|\right)$.
By Proposition 3, the set $Id_{\|A_M\|}\left(R_{\rightsquigarrow}^{\leq m}\left(L\right)\right)$ is thus recognized by a weighted tree automaton of size $O\left(|A_R|^{m\cdot(m+1)/2} \times |A| \times |A_M|\right)$.
Finally, in the tropical semiring, searching for a path of minimum weight in a weighted tree automaton takes linear time [7].

6 Conclusion

The weighted abstraction formalism we presented has the advantage of providing a detection algorithm with the same complexity as in the unweighted case, that is linear in the size of the trace automaton. Thus, without any overhead, we can take into account the abstraction uncertainty, whether they stem from static analysis errors or whether they are related to the way of carrying out a functionality.
Besides, behavior pattern definitions can be refined by adapting the abstraction weight depending on the context: if the recognized occurrence appears in a context where the associated functionality is likely to be encountered, we will decrease uncertainty, whereas otherwise we will increase it. Similarly, if the program is obfuscated, we may be more tolerant with respect to the abstraction uncertainty since the dataflow is likely to contain more errors.
Finally, we could express the problem of abstract behavior detection in a slightly different way: rather than determining whether a program exhibits an abstract behavior with a probability not exceeding a given threshold, we could compute the exact probability that the program exhibits this behavior. Such an approach of the detection problem opens new perspectives. First, a human analyst would get more precise information with respect to the exhibited behavior. Second, assume exhibition of a behavior is decision criterion among others in order to assess the maliciousness of a program: the weight of this criterion in the final decision could be a function of the probability that the program exhibits this behavior.

References

1. Athanasios Alexandrakis and Symeon Bozapalidis. Weighted grammars and kleene's theorem. *Inf. Process. Lett.*, 24(1):1–4, January 1987.
2. Philippe Beaucamps, Isabelle Gnaedig, and Jean-Yves Marion. Abstraction-based Malware Analysis Using Rewriting and Model Checking. In Sara Foresti, MotiYung, and Fabio Martinelli, editors, *ESORICS - 17th European Symposium on Research in Computer Security - 2012*, volume 7459 of *Lecture Notes in Computer Science*, pages 806–823, Pisa, Italie, 2012. Springer.

3. Jean Berstel and Christophe Reutenauer. *Rational series and their languages*, volume 12 of *Monographs in Theoretical Computer Science. An EATCS Series*. Springer, 1988.
4. H. Comon, M. Dauchet, R. Gilleron, D. Lugiez, S. Tison, and Tommasi. Tree automata techniques and applications. Preliminary Version, `http://l3ux02.univ-lille3.fr/tata/`, 1997.
5. Joost Engelfriet, Zoltán Fülöp, and Heiko Vogler. Bottom-up and top-down tree series transformations. *J. Autom. Lang. Comb.*, 7(1):11–70, July 2001.
6. Zoltán Fülöp and Heiko Vogler. Weighted tree automata and tree transducers. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science. An EATCS Series, pages 313–403. Springer Berlin Heidelberg, 2009.
7. Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 53–64, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
8. Kevin Knight and Jonathan May. Applications of weighted automata in natural language processing. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science. An EATCS Series, pages 571–596. Springer Berlin Heidelberg, 2009.
9. Fred Kröger and Stephan Merz. *Temporal Logic and State Systems*. Texts in Theoretical Computer Science. An EATCS Series. 2008.
10. Werner Kuich. Tree transducers and formal tree series. *Acta Cybern.*, 14(1):135–164, 1999.
11. Werner Kuich and Arto Salomaa. *Semirings, Automata and Languages*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1985.
12. Jonathan May, Kevin Knight, and Heiko Vogler. Efficient inference through cascades of weighted tree transducers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, pages 1058–1066, Stroudsburg, PA, USA, 2010.
13. Mehriar Mohri. *Applied Combinatorics on Words*, chapter Statistical natural language processing. Chapter 4. Cambridge University Press, New York, NY, USA, 2005.
14. Mehryar Mohri. Finite-state transducers in language and speech processing. *Comput. Linguist.*, 23(2):269–311, June 1997.
15. William C. Rounds. Mappings and grammars on trees. *Mathematical Systems Theory*, 4(3):257–287, 1970.
16. James W. Thatcher. Generalized2 sequential machine maps. *J. Comput. Syst. Sci.*, 4(4):339–367, August 1970.