A Decidable Recursive Logic for Weighted Transition Systems

Kim G. Larsen, Radu Mardare and Bingtian Xue

Aalborg University, Denmark

Abstract. Labelled weighted transition systems (LWSs) are transition systems labelled with actions and real valued quantities representing the costs of transitions with respect to various resources. We introduce Recursive Weighted Logic (RWL) being a multi-modal logic that expresses not only qualitative, but also quantitative properties of LWSs by using first-order variables that measure local costs, similar to the clocks in timed logics. In addition, RWL is endowed with simultaneous recursive equations, which specify the weakest properties satisfied by the recursive variables. It is proved that the satisfiability problem for the logic is decidable by applying a variant of the region technique developed for timed automata. This result is in contrast to corresponding temporal logics for real-time systems, where satisfiability is known to be undecidable.

Keywords: labelled weighted transition system, multi-modal logic, quantitative properties, maximal fixed point, satisfiability

1 Introduction

For industrial practice, especially in the area of embedded systems, an essential problem is how to deal with the growing complexity of the systems, while still meeting the requirements on correctness, predictability, performance and also resource constraints. In this respect, for embedded systems, verification should not only consider functional properties but also non-functional properties such as those related to resource constraints. Within the area of model checking a number of state-machine based modelling formalisms has emerged, which allow for such quantitative properties to be expressed. For instance, the formalisms of timed automata [AD90] and its extensions to weighted timed automata [BFH+01,ATP01] allow time-constraints to be modelled and efficiently analysed.

In order to specify and reason about not only the qualitative behaviours of (embedded) systems but also their quantitative consumptions of resources, we consider a multi-modal logic – Recursive Weighted Logic (RWL) – defined for a semantics based on labelled weighted transition systems (LWS) in this paper. LWSs are transition systems labelled with both actions and real numbers. The numbers represent the costs of the corresponding transitions in terms of resources. In order to use a variant of the region technique developed for timed automata [AD90,ACD90] to get the decidability, we restrict the numbers on the labels to be non-negative in this paper. Our notion of weighted transition systems is more than a simple instance of *weighted automata* [DKV09], since they can also be infinite (including infinitely branching) systems.

RWL is an extension of *weighted modal logic* [LM13] with only maximal fixed points. By introducing maximal fixed points defined by simultaneous recursive equations [Lar90,CKS92,CS93], RWL allows us to encode properties of infinite behaviours including safety and cost-bounded liveness properties. They specify the weakest properties satisfied by the recursive variables. RWL is also endowed with modal operators that predicate about the values of resource-variables, which allow us to specify and reason about the quantitative properties related to resources, e.g., energy and time. While in a LWS we can have real valued labels, the modalities of the logic only encodes rational values. This will not restrict too much the expressive power of RWL since we can characterize a transition using an infinite convergent sequences of rationals that approximate the real resource.

To encode various resource-constrains in RWL, we use resource-variables, similar to the clock-variables used in timed logics [ACD93a,HNSY92,AILS07]. We use *resource valuation* to assign non-negative values to resource-variables. In this paper we only discuss the event related variables, i.e., for each type of resource and each action, we associate one resource-variable. Every time the system does this action, all the resource-variables associated to this action will be reset, meaning that the resource valuation will map those resource-variables to zero. This is useful for encoding various interesting scenarios. In a former paper of the authors [LMXb], we restricted our attention to only one resource-variable for each type of resources. This guaranteed the decidability of the logic and finite model property. However, this restriction bounds the expressiveness of the logic. For example, ...(need to be finished). Here, we allow multiple resource-variables for each type of resource, more precisely one for each action, which measure the resource in different ways.

Even though RWL does not possess the finite model property, we may apply a variant of the region technique developed for timed automata [AD90,ACD90], to obtain a symbolic LWS of all satisfiable formulas. It provides an abstract semantics of LWSs in the form of finite labelled transition systems – symbolic model of LWS – with the truth value of RWL formulas being maintained. More precisely, we present a model construction algorithm, which constructs a symbolic LWS for a given RWL formula (provided that the formula is not a contradiction, i.e., without any model). From this symbolic model we can decide on the existence of a concrete LWS – might be infinite – which is a model of the given formula. Our construction is inspired from the region construction proposed in [LLW95] for timed automata, which adapts of the classical filtration-based model construction used in modal logics [HC96,HKT01,Wal00]. However, the result is in contrast to corresponding temporal logics for real-time systems (e.g., TCTL [ACD93b], T_μ [HNSY92], L_ν [LLW95] and timed modal logic (TML) [LMXa]), where satisfiability is known to be undecidable.

The rest of this paper is organized as follows: in the following section we present the notion of labelled weighted transition system; in Section 3, we introduce recursive weighted logic with its syntax and semantics; Section 4 is dedicated to the region theory and the symbolic models of LWSs; in Section 5 we prove the decidability of the satisfiability problem for RWL. we also present a conclusive section where we summarize the results and describe future research directions.

2 Labelled Weighted Transition Systems

A *labelled weighted transition system* (LWS) is a transition system that has several types of resources and has the transitions labelled both with actions and (non-negative) real numbers – as represented in Figure 1, in which there are three types of resource. Each number is used to present the costs of the corresponding transitions in terms of one type of resource, e.g., energy or time.

Definition 1 (Labelled Weighted Transition System). *A LWS is a tuple*

$$\mathcal{W} = (M, \mathcal{K}, \Sigma, \theta)$$

where M is a non-empty set of states, *$\mathcal{K} = \{e_1, \ldots, e_k\}$ is the finite set of (k types of) resources, Σ a non-empty set of* actions *and $\theta : M \times (\Sigma \times (\mathcal{K} \to \mathbb{R}_{\geq 0})) \to 2^M$ is the* labelled transition function.

For simplicity, hereafter we use a vector of real numbers instead of the function from the set of the resources $\mathcal{K}$ to real numbers, i.e., for $f : \mathcal{K} \to \mathbb{R}_{\geq 0}$ defined as $f(e_i) = r_i$ for all $i = 1, \ldots, k$, we write $\overline{u} = (r_1, \ldots, r_k) \in \mathbb{R}^k_{\geq 0}$ instead. On the other hand, for a vector of real numbers $\overline{u} \in \mathbb{R}^k_{\geq 0}$, $\overline{u}(e_i)$ denotes the i-th number of the vector $\overline{u}$, which represent the cost of the resource e_i during the transition.

Instead of $m' \in \theta(m, a, \overline{u})$ we write $m \xrightarrow{\overline{u}}_a m'$.

To clarify the role of the aforementioned concepts consider the following example.

Example 1. In Figure 1, we show the LWS

$$\mathcal{W} = (M, \Sigma, \mathcal{K}, \theta),$$

where $M = \{m_0, m_1, m_2\}$, $\mathcal{K} = \{e_1, e_2, e_3\}$, $\Sigma = \{a, b\}$, and θ defined as follows: $m_0 \xrightarrow{(3,4,5)}_a m_1$, $m_0 \xrightarrow{(\pi,\pi,0)}_b m_2$ and $m_1 \xrightarrow{(\sqrt{2},1.9,7)}_a m_2$.

$\mathcal{W}$ has three states m_0, m_1, m_2, three kinds of resource e_1, e_2, e_3 and two actions a, b. The state m_0 has two transitions: one a-transition – which costs 3 units of e_1, 4 units of e_2 and 5 units of e_3 – to m_1 and one b-transition – which costs π units of e_1 and e_2 respectively (and does not cost any e_3) – to m_2. If the system does an a-transition from m_0 to m_1, the amounts of the resource e_1, e_2 and e_3 increase with 3, 4 and 5 units respectively – that the system gains by doing the a-transition. ∎

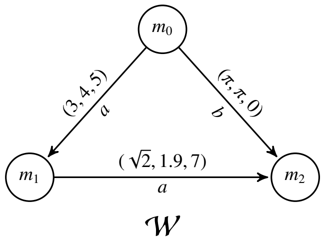

Fig. 1. Labelled Weighted Transition System

In the rest of this paper, we fix a set Σ of actions, and for simplicity we omit it in the description of LWSs and the logic defined in the next section.

The concept of *weighted bisimulation* is a relation between the states of a given LWS that equates states with identical (action- and weighted-) behaviors.

Definition 2 (Weighted Bisimulation). *Given a LWS* $\mathcal{W} = (M, \mathcal{K}, \theta)$*, a* weighted bisimulation *is an equivalence relation* $R \subseteq M \times M$ *such that whenever* $(m, m') \in R$*,*

- if $m \xrightarrow{\bar{u}}_a m_1$*, then there exists* $m'_1 \in M$ *s.t.* $m' \xrightarrow{\bar{u}}_a m'_1$ *and* $(m_1, m'_1) \in R$*;*
- if $m' \xrightarrow{\bar{u}}_a m'_1$*, then there exists* $m' \in M$ *s.t.* $m \xrightarrow{\bar{u}}_a m_1$ *and* $(m_1, m'_1) \in R$*.*

If there exists a weighted bisimulation relation R such that $(m, m') \in R$*, we say that m and m′ are* bisimilar*, denoted by* $m \sim m'$*.*

As for the other types of bisimulation, the previous definition can be extended to define the weighted bisimulation between distinct LWSs by considering bisimulation relations on their disjoint union. *Weighted bisimilarity* is the largest weighted bisimulation relation; if $\mathcal{W}_i = (M_i, \mathcal{K}_i, \theta_i)$, $m_i \in M_i$ for $i = 1, 2$ and m_1 and m_2 are bisimilar, we write $(m_1, \mathcal{W}_1) \sim (m_2, \mathcal{W}_2)$.

The next examples shows the role of the weighted bisimilarity.

Example 2. In Figure 2, $\mathcal{W}_1 = (M_1, \mathcal{K}_1, \theta_1)$ is a LWS with five states and one type of resources, where $M_1 = \{m_0, m_1, m_2, m_3, m_4\}$, $\mathcal{K}_1 = \{e\}$ and θ_1 is defined as: $m_0 \xrightarrow{3}_a m_1$, $m_0 \xrightarrow{2}_b m_2$, $m_1 \xrightarrow{0}_d m_2$, $m_1 \xrightarrow{3}_c m_3$, $m_2 \xrightarrow{0}_c m_1$ and $m_2 \xrightarrow{3}_c m_4$.

It is easy to see that $m_3 \sim m_4$ because both of them can not do any transition. Besides, $m_1 \sim m_2$ because both of them can do a c-transition with cost 3 to some states which are bisimular (m_3 and m_4), and a d-action transition with cost 0 to each other. m_0 is not bisimular to any states in $\mathcal{W}_1$.

$\mathcal{W}_2 = (M_2, \mathcal{K}_2, \theta_2)$ is a LWS with three states, where $M_2 = \{m'_0, m'_1, m'_2\}$, $\mathcal{K}_2 = \mathcal{K}_1$ and θ_2 is defined as: $m'_0 \xrightarrow{3}_a m'_1$, $m'_0 \xrightarrow{2}_b m'_1$, $m'_1 \xrightarrow{0}_d m'_1$ and $m'_1 \xrightarrow{3}_c m'_2$.

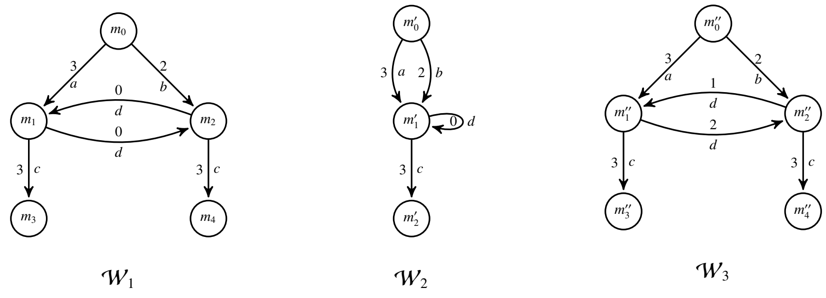

Fig. 2. Weighted Bisimulation

We can see that: $(m_0, \mathcal{W}_1) \sim (m'_0, \mathcal{W}_2)$, $(m_1, \mathcal{W}_1) \sim (m'_1, \mathcal{W}_2)$, $(m_2, \mathcal{W}_1) \sim (m'_1, \mathcal{W}_2)$, $(m_3, \mathcal{W}_1) \sim (m'_2, \mathcal{W}_2)$, $(m_4, \mathcal{W}_1) \sim (m'_2, \mathcal{W}_2)$.

Notice that $(m''_0, \mathcal{W}_3) \nsim (m'_0, \mathcal{W}_2)$, because $(m''_1, \mathcal{W}_3) \nsim (m'_1, \mathcal{W}_2)$. Besides, $m''_1 \nsim m''_2$, because m''_1 can do a d-action with weight 2 while m''_2 cannot and m''_2 can do a d-action with weight 1 while m''_1 cannot. ■

3 Recursive Weighted Logic

In this section we introduce a multi-modal logic that encodes properties of LWSs called *Recursive Weighted Logic* (RWL).

To encode various resource-constrains in RWL, we use resource-variables, similar to the clock-variables used in timed logics [ACD93a,HNSY92,AILS07]. In this paper, we introduce event related resource-variables to measure the resources in different ways corresponding to different actions, i.e., for each action $a \in \Sigma$, we associate resource-variables $x_a^1, \ldots, x_a^k$ for each types of resource $e_1, \ldots, e_k$ respectively. In the following, we use $\mathcal{V}_i = \{x_a^i \mid a \in \Sigma\}$ to denote the set of the resource-variables associated for the type of resource e_i, $\mathcal{V}_a = \{x_a^i \mid i = 1, \ldots, k\}$ to denote the set of the resource-variables associated with the action a and $\mathcal{V} = \bigcup_{i=1,\ldots,k} \mathcal{V}_i = \bigcup_{a\in\Sigma} \mathcal{V}_a$ to denote the set of all the resource-variables. Note that for any i, j such that $i \neq j$, $\mathcal{V}_i \cup \mathcal{V}_j = \emptyset$, and for any a, b such that $a \neq b$, $\mathcal{V}_a \cup \mathcal{V}_b = \emptyset$.

In addition to the classic boolean operators (except negation), our logic is firstly endowed with a class of recursive (formula) variables $X_1, \ldots, X_n$, which specify properties of infinite behaviours. We denote $\mathcal{X}$ the set of recursive formula variables.

Secondly, RWL is endowed with a class of modalities of arity 1, named *transition modalities*, of type $[a]$ or $\langle a \rangle$, for $a \in \Sigma$, which are defined as the classical transition modalities with reset operation of all the resource-variables associated with the corresponding action followed. More precisely, every time the system does an a-action, all the resource-variables $x \in \mathcal{V}_a$ will be reset, i.e., x is interpreted to zero after every a-action, for all $x \in \mathcal{V}_a$.

Besides, our logic is also endowed with a class of modalities of arity 0 called *state modalities* of type $x \bowtie r$, for $\bowtie \in \{\leq, \geq, <, >\}$, $r \in \mathbb{Q}_{\geq 0}$ and $x \in \mathcal{V}$, which predicates about the value of the resource-variable x at the current state.

Before proceeding with the maximal fixed points, we define the basic formulas of RWL and their semantics firstly. Based on them, we will eventually introduce the recursive definitions - the maximal equation blocks - which extend the semantics of the basic formulas.

Definition 3 (Syntax of Basic Formulas). *For arbitrary* $r \in \mathbb{Q}_{\geq 0}$*,* $a \in \Sigma$*,* $x \in \mathcal{V}$*,* $\bowtie \in \{\leq, \geq, <, >\}$ *and* $X \in \mathcal{X}$*, let*

$$\mathcal{L}: \quad \phi := \top \mid \bot \mid x \bowtie r \mid \phi \wedge \phi \mid \phi \vee \phi \mid [a]\phi \mid \langle a \rangle \phi \mid X\ .$$

Before looking at the semantics for the basic formulas, we define the notion of *resource valuation* and *extended states*.

Definition 4 (Resource Valuation). *A resource valuation is a function* $l : \mathcal{V} \to \mathbb{R}_{\geq 0}$ *that assigns (non-negative) real numbers to all the resource-variables in* $\mathcal{V}$*.*

A resource valuation assigns non-negative values to resource-variables and the assignment is interpreted as the amount of resources measured by the given resource-variable in a given state of the system. We denote by L the class of resource valuations.

We write l_i to denote the valuation for all resource-variables $x \in \mathcal{V}_i$ under the resource valuation l, i.e., for any $x \in \mathcal{V}$,

$$l_i(x) = \begin{cases} l(x), & x \in \mathcal{V}_i \\ \text{undefined}, & \text{otherwise} \end{cases}$$

Similar we write l_a to denote the valuation for all resource-variables $x \in \mathcal{V}_a$ under the resource valuation l, i.e., for any $x \in \mathcal{V}$,

$$l_a(x) = \begin{cases} l(x), & x \in \mathcal{V}_a \\ \text{undefined}, & \text{otherwise} \end{cases}$$

If l is a resource valuation and $x \in \mathcal{V}$, $s \in \mathbb{R}_{\geq 0}$ we denote by $l[x \mapsto s]$ the resource valuation that associates the same values as l to all variables except x, to which it associates the value s, i.e., for any $y \in \mathcal{V}$,

$$l[x \mapsto s](y) = \begin{cases} s, & y = x \\ l(y), & \text{otherwise} \end{cases}$$

Moreover for $\mathcal{V}' \subseteq \mathcal{V}$ and $s \in \mathbb{R}_{\geq 0}$, we denote by $l[\mathcal{V}' \mapsto s]$ the resource valuation that associates the same values as l to all variables except those in $\mathcal{V}'$, to which it associates the value s, i.e., for any $y \in \mathcal{V}$,

$$l[\mathcal{V}' \mapsto s](y) = \begin{cases} s, & y \in \mathcal{V}' \\ l(y), & \text{otherwise} \end{cases}$$

For $\overline{u} \in \mathbb{R}^k_{\geq 0}$, $l + \overline{u}$ is defined as: for any $i \in \{1, \dots, k\}$, for any $x \in \mathcal{V}_i$,

$$(l + \overline{u})(x) = l(x) + \overline{u}(e_i).$$

A pair (m, l) is called *extended state* of a given LWS $\mathcal{W} = (M, \mathcal{K}, \theta)$, where $m \in M$ and $l \in L$. Transitions between extended states are defined by:

$$(m, l) \longrightarrow_a (m', l') \text{ iff } m \xrightarrow{\overline{u}}_a m' \text{ and } l' = l + \overline{u}.$$

Given a LWS $\mathcal{W} = (M, \mathcal{K}, \theta)$, we interpret the RWL basic formulas over an extended state (m, l) and an environment ρ which maps each recursive formula variables to subsets of $M \times L$. The *LWS-semantics* of RWL basic formulas is defined inductively as follows.

$\mathcal{W}, (m, l), \rho \models \top$ – always,

$\mathcal{W}, (m, l), \rho \models \bot$ – never,

$\mathcal{W}, (m, l), \rho \models x \bowtie r$ iff $l(x) \bowtie r$,

$\mathcal{W}, (m, l), \rho \models \phi \wedge \psi$ iff $\mathcal{W}, (m, l), \rho \models \phi$ and $\mathcal{W}, (m, l), \rho \models \psi$,

$\mathcal{W}, (m, l), \rho \models \phi \vee \psi$ iff $\mathcal{W}, (m, l), \rho \models \phi$ or $\mathcal{W}, (m, l), \rho \models \psi$,

$\mathcal{W}, (m, l), \rho \models [a]\phi$ iff for arbitrary $(m', l') \in M \times L$ such that $(m, l) \longrightarrow_a (m', l')$, $\mathcal{W}, (m', l'[\mathcal{V}_a \mapsto 0]), \rho \models \phi$,

$\mathcal{W}, (m, l), \rho \models \langle a \rangle \phi$ iff exists $(m', l') \in M \times L$ such that $(m, l) \longrightarrow_a (m', l')$ and $\mathcal{W}, (m', l'[\mathcal{V}_a \mapsto 0]), \rho \models \phi$,

$\mathcal{W}, (m, l), \rho \models X$ iff $m \in \rho(X)$.

Definition 5 (Syntax of Maximal Equation Blocks). *Let $\mathcal{X} = \{X_1, \dots, X_n\}$ be a set of recursive formula variables. A maximal equation block B is a list of (mutually recursive) equations:*

$$X_1 = \phi_1$$
$$\vdots$$
$$X_n = \phi_n$$

in which X_i are pairwise-distinct over $\mathcal{X}$ and ϕ_i are basic formulas over $\mathcal{X}$, for all $i = 1, \dots, n$.

Each maximal equation block B defines an *environment* for the recursive formula variables $X_1, \dots, X_n$, which is the weakest property that the variables satisfy. We say that an arbitrary formula ϕ is *closed with respect to a maximal equation block B* if all the recursive formula variables appearing in ϕ are defined in B by some of its equations. If all the formulas ϕ_i that appears in the right hand side of some equation in B is closed with respect to B, we say that B is *closed*.

Given an environment ρ and $\overline{\Upsilon} = \langle \Upsilon_1, \dots, \Upsilon_n \rangle \in (2^{M \times L})^n$, let

$$\rho_{\overline{\Upsilon}} = \rho[X_1 \mapsto \Upsilon_1, \dots, X_n \mapsto \Upsilon_n]$$

be the environment obtained from ρ by updating the binding of X_i to Υ_i.

Given a maximal equation block B and an environment ρ, consider the function

$$f_B^{\rho} : (2^{M\times L})^n \longrightarrow (2^{M\times L})^n$$

defined as follows:

$$f_B^{\rho}(\overline{\Upsilon}) = \langle [\![\phi_1]\!]\rho_{\overline{\Upsilon}}, \ldots, [\![\phi_n]\!]\rho_{\overline{\Upsilon}}\rangle,$$

where $[\![\phi]\!]\rho = \{(m,l) \in M \times L \mid \mathcal{W}, (m,l), \rho \models \phi\}$.

Observe that $(2^{M\times L})^n$ forms a complete lattice with the ordering, join and meet operations defined as the point-wise extensions of the set-theoretic inclusion, union and intersection, respectively. Moreover, for any maximal equation block B and environment ρ, f_B^{ρ} is monotonic with respect to the order of the lattice and therefore, according to the Tarski fixed point theorem [Tar55], it has a greatest fixed point that we denote by $\nu\overline{X}.f_B^{\rho}$. This fixed point can be characterized as follows:

$$\nu\overline{X}.f_B^{\rho} = \bigcup\{\overline{\Upsilon} \mid \overline{\Upsilon} \subseteq f_B^{\rho}(\overline{\Upsilon})\}.$$

Consequently, a maximal equation block defines an environment that satisfies all its equations, i.e., $[\![B]\!]\rho = \nu\overline{X}.f_B^{\rho}$.

When B is closed, i.e. there is no free recursive formula variable in B, it is not difficult to see that for any ρ and ρ', $[\![B]\!]\rho = [\![B]\!]\rho'$. So, we just take $\rho = 0$ and write $[\![B]\!]$ in stead of $[\![B]\!]0$. In the rest of the paper we will only discuss this kind of equation blocks. (For those that are not closed, we only need to have the initial environment which maps the free recursive variables to subsets of $M \times L$.)

Now we are ready to define the general semantics of RWL: for an arbitrary LWS $\mathcal{W} = (M, \mathcal{K}, \theta)$ with $m \in M$, an arbitrary resource valuation $l \in L$ and arbitrary RWL-formula ϕ closed w.r.t. a maximal equation block B,

$$\mathcal{W}, (m,l) \models_B \phi \;\text{ iff }\; \mathcal{W}, (m,l), [\![B]\!] \models \phi.$$

The symbol $\models_B$ is interpreted as satisfiability for the block B. Whenever it is not the case that $\mathcal{W}, (m,l) \models_B \phi$, we write $\mathcal{W}, (m,l) \not\models_B \phi$. We say that a formula ϕ is *B-satisfiable* if there exists at least one LWS that satisfies it for the block B in one of its states under at least one resource valuation; ϕ is a *B-validity* if it is satisfied in all states of any LWS under any resource valuation - in this case we write $\models_B \phi$.

To exemplify the expressiveness of RWL, we propose the following example of system with recursively-defined properties.

Example 3.

4 Regions and Symbolic Models

In this section we introduce the region technique for LWS, which is inspired by that for timed automata of Alur and Dill [AD90,ACD90]. It provides an abstract semantics of LWSs in the form of finite labelled transition systems with the truth value of RWL formulas being maintained.

Here we introduce the regions defined for a given maximal constant $N \in \mathbb{N}$. For the case where the maximal constant is a rational number $\frac{p}{q}$ where $p, q \in \mathbb{N}$, we only need to get the regions for the maximal constant p first and divide all the regions by q.

First for $r \in \mathbb{R}_{\geq 0}$, let $\lfloor r \rfloor \stackrel{\text{def}}{=} \max\{z \in \mathbb{Z} \mid z \leq r\}$ denote the integral part of r, and let $\{r\} = r - \lfloor r \rfloor$ denote its fractional part. Moreover we have $\lceil r \rceil \stackrel{\text{def}}{=} \min\{z \in \mathbb{Z} \mid z \geq r\}$.

Definition 6. *Let $N \in \mathbb{N}$ be a given maximal constant and let $\mathcal{V}_i$ be a set of resource-variables for resource e_i. Then $l_i, l'_i : \mathcal{V}_i \to \mathbb{R}_{\geq 0}$ are equivalent with respect to N, denoted by $l_i \doteq l'_i$ iff:*

1. $\forall x \in \mathcal{V}_i$, $l_i(x) > N$ *iff* $l'_i(x) > N$;
2. $\forall x \in \mathcal{V}_i$ *s.t.* $0 \leq l_i(x) \leq N$, $\lfloor l_i(x) \rfloor = \lfloor l'_i(x) \rfloor$ *and* $\{l_i(x)\} = 0 \Leftrightarrow \{l'_i(x)\} = 0$;
3. $\forall x, y \in \mathcal{V}_i$ *s.t.* $0 \leq l_i(x), l_i(y) \leq N$, $\{(l_i(x)\} \leq \{l_i(y))\} \Leftrightarrow \{(l'_i(x)\} \leq \{l'_i(y))\}$.

The equivalence classes under $\doteq$ are called *regions*. $[l_i]$ denotes the region which contains the labelling l_i for resource-variables $x \in \mathcal{V}_i$ and $R^{\mathcal{V}_i}_{N_i}$ denotes the set of all regions for the set $\mathcal{V}_i$ of resource-variables for resource e_i and the constant N_i. Notice that for a given $N_i \in \mathbb{N}$, $R^{\mathcal{V}_i}_{N_i}$ is finite.

For a region $\delta \in R^{\mathcal{V}_i}_{N_i}$, we define the *successor* region as the region δ' – denoted by $\delta \rightsquigarrow \delta'$ – iff:

$$\text{for any } l_i \in \delta, \text{ there exists } d \in \mathbb{R}_{\geq 0} \text{ s.t. } l_i + d \in \delta'.$$

As we mentioned before, for the case where the maximal constant is a rational number $\frac{p_i}{q_i}$ where $p_i, q_i \in \mathbb{N}$, we only need to get the regions for the maximal constant p first and divide all the regions by q_i to achieve the set of all regions for the set $\mathcal{V}_i$ of resource-variables for resource e_i and the constant $\frac{p_i}{q_i}$ – denoted by $R^{\mathcal{V}_i}_{p_i/q_i}$. Let $\mathcal{R}^{\mathcal{V}} = \{\overline{[l]} = ([l_1], \ldots, [l_k]) \mid [l_i] \in R^{\mathcal{V}_i}_{p_i/q_i}, \frac{p_i}{q_i} \in \mathbb{Q}_{\geq 0}$ for any $i \in \{1, \ldots, k\}\}$.

We will now define the fundamental concept of a *symbolic model* of LWS. Every extended state (m, l) is replaced by a so-called *extended symbolic state* $(m, \overline{[l]})$. Whenever we have transition between two extended states, there should also be a transition between the corresponding symbolic states, i.e.:

$$(m, \overline{[l]}) \longrightarrow_a (m', \overline{[l']}) \text{ iff } (m, l) \longrightarrow_a (m', l').$$

Definition 7. *Given $\mathcal{R}^{\mathcal{V}}$ and a non-empty set of states M^s, a* symbolic LWS *is a tuple*

$$\mathcal{W}^s = (\Pi^s, \Sigma^s, \theta^s)$$

where $\Pi^s \subseteq M^s \times \mathcal{R}^{\mathcal{V}}$ is a non-empty set of symbolic states $\pi^s = (m, \overline{\delta})$*, Σ^s a non-empty set of actions and $\theta^s : \Pi^s \times (\Sigma^s) \to 2^{\Pi^s}$ is the* symbolic labelled transition function*, which satisfies the following:*

$$\textit{if } (m', \overline{\delta'}) \in \theta((m, \overline{\delta}), a), \textit{ then } \overline{\delta} \rightsquigarrow \overline{\delta'}.$$

Given a symbolic LWS, we can define the symbolic satisfiability relation $\models^s$ with $\pi = (m, \overline{\delta}) \in \Pi^s$ as follows:

$\mathcal{W}^s, \pi, \rho^s \models^s \top$ – always,

$\mathcal{W}^s, \pi, \rho^s \models^s \bot$ – never,

$\mathcal{W}^s, \pi, \rho^s \models^s x \bowtie r$ iff for any $w \in \overline{\delta}(x)$, $w \bowtie r$,

$\mathcal{W}^s, \pi, \rho^s \models^s \phi \wedge \psi$ iff $\mathcal{W}^s, \pi, \rho^s \models^s \phi$ and $\mathcal{W}^s, \pi, \rho^s \models^s \psi$,

$\mathcal{W}^s, \pi, \rho \models^s \phi \vee \psi$ iff $\mathcal{W}^s, \pi, \rho^s \models^s \phi$ or $\mathcal{W}^s, \pi, \rho^s \models^s \psi$,

$\mathcal{W}^s, \pi, \rho^s \models^s [a]\phi$ iff for any $\pi' = (m', \overline{\delta'}) \in \Pi^s$ s.t. $\pi \rightarrow_a \pi'$, $\mathcal{W}^s, (m', \overline{\delta'}[\mathcal{V}_a \mapsto 0]), \rho^s \models^s \phi$,

$\mathcal{W}^s, \pi, \rho^s \models^s \langle a\rangle\phi$ iff there exists $\pi' = (m', \overline{\delta'}) \in \Pi^s$ s.t. $\pi \rightarrow_a \pi'$ and $\mathcal{W}^s, (m', \overline{\delta'}[\mathcal{V}_a \mapsto 0]), \rho^s \models^s \phi$,

$\mathcal{W}^s, \pi, \rho^s \models^s X$ iff $m \in \rho^s(X)$,

where $\overline{\delta}[\mathcal{V}_a \mapsto 0]$ is defined as $\overline{\delta}[\mathcal{V}_a \mapsto 0](x) = 0$ for any $x \in \mathcal{V}_a$ and $\overline{\delta}[\mathcal{V}_a \mapsto 0](y) = \overline{\delta}(y)$ for any $y \in \mathcal{V}/\mathcal{V}_a$.

Similarly we can define the symbolic B-satisfiability relation $\models^s_B$ as in Section 3:

$$\mathcal{W}^s, \pi \models^s_B \phi \;\text{ iff }\; \mathcal{W}^s, \pi, [\![B]\!] \models^s \phi.$$

Lemma 1. *Let ϕ be a RWL formula closed w.r.t a maximal equation block B. If it is satisfied by a symbolic LWS $\mathcal{W}^s = (\Pi^s, \Sigma^s, \theta^s)$ i.e. $\mathcal{W}, \pi \models^s_B \phi$ with $\pi = (m, \overline{\delta}) \in \Pi^s$, then there exists a LWS $\mathcal{W} = (M, \Sigma, \mathcal{K}, \theta)$ and a resource valuation $l \in L$ such that $\mathcal{W}, (m, l) \models_B \phi$ with $m \in M$.*

Proof. Let $\Sigma = \Sigma^s$, $\mathcal{K}$ be the set of the resources appearing in $\mathcal{R}^{\mathcal{V}}$ and $l \in \overline{\delta}$. The transition function is defined as: $(m_1, \overline{\delta_1}, l_1, i) \xrightarrow{\overline{u}}_a (m_2, \overline{\delta_2}, l_2, j)$ iff,

$$(m_1, \overline{\delta_1}) \longrightarrow_a (m_2, \overline{\delta_2}), \text{ for } i = 1, 2, l_i \in \overline{\delta_i} \text{ and } l_2 = (l_1 + \overline{u})[\mathcal{V}_a \mapsto 0],$$

where $i, j \in \mathbb{N}$ are indexes for different states, where j is different from any existing indexes - equals to the last one plus 1.

We define the transition relation starting from $(m, \overline{\delta}, l, 0)$. Let M be the set of all the states defined for the transitions as the above. Note that it might be infinite. It is easy to verify that $\mathcal{W}, ((m, \overline{\delta}, l, 0), l) \models_B \phi$. ∎

5 Satisfiability of Recursive Weighted Logic

In this section we prove that it is decidable whether a given formula ϕ which is closed w.r.t. a maximal equation block B of RWL is satisfiable. We also present a decision procedure for the satisfiability problem of RWL. The results rely on a syntactic characterization of satisfiability that involves a notion of *mutually-consistent sets* that we define later.

Consider an arbitrary formula $\phi \in \mathcal{L}$ which is closed w.r.t. a maximal equation block B. In this context we define the following notions:

- Let $\Sigma[\phi, B]$ be the set of all actions $a \in \Sigma$ such that a appears in some transition modality of type $\langle a \rangle$ or $[a]$ in ϕ or B.

- For any $e_i \in \mathcal{K}$ and $x \in \mathcal{V}_i$, let $Q_i[\phi, B] \subseteq \mathbb{Q}_{\geq 0}$ be the set of all $r \in \mathbb{Q}_{\geq 0}$ such that r is in the label of some state or transition modality of type $x \bowtie r$ that appears in the syntax of ϕ or B.

- We denoted by g_i the *granularity of e_i in ϕ*, defined as the least common denominator of the elements of $Q_i[\phi, B]$. Let $R^{\mathcal{V}_i}_{p/g_i}[\phi, B]$ be the set of all regions for resource e_i, where $\frac{p}{g_i} = \max Q_i[\phi, B]$. Let

$$\mathcal{R}^{\mathcal{V}}[\phi, B] = \{\overline{\delta} = (\delta_1, \ldots, \delta_k) \mid \delta_i \in R^{\mathcal{V}_i}_{p/g_i}[\phi, B] \text{ for any } i \in \{1, \ldots, k\}\}.$$

For $r \in \mathbb{R}_{\geq 0}$, we use $r \in \overline{\delta}(x)$ to denote $r \in \delta_i(x)$, for any $i \in \{1, \ldots, k\}$ and $x \in \mathcal{V}_i$.

Observe that $\Sigma[\phi, B]$, $Q_i[\phi, B]$, $R^{\mathcal{V}_i}_{p/g_i}[\phi, B]$ and $\mathcal{R}^{\mathcal{V}}[\phi, B]$ are all finite (or empty).

At this point we can start our model construction. We fix a formula $\phi_0 \in \mathcal{L}$ that is closed w.r.t. a given maximal equation block B and, supposing that the formula admits a model, we construct a model for it. Let

$$\mathcal{L}[\phi_0, B] = \{\phi \in \mathcal{L} \mid \Sigma[\phi, B] \subseteq \Sigma[\phi_0, B], Q_i[\phi, B] \subseteq Q_i[\phi_0, B]\}.$$

Here we are going to construct a *symbolic model*. To construct the model we will use as symbolic states tuples of type $(\Gamma, \overline{\delta}) \in 2^{\mathcal{L}[\phi_0, B]} \times \mathcal{R}^{\mathcal{V}}[\phi_0, B]$, which are required to be maximal in a precise way. The intuition is that a state $(\Gamma, \overline{\delta}) \subseteq 2^{\mathcal{L}[\phi_0, B]} \times \mathcal{R}^{\mathcal{V}}[\phi_0, B]$ shall symbolically satisfy the formula ϕ in our model, whenever $\phi \in \Gamma$. From this symbolic model we can generalize a LWS - might be infinite - which is a model of the given formula. Our construction is inspired from the region construction proposed in [LLW95] for timed automata, which adapts of the classical filtration-based model construction used in modal logics [HC96,HKT01,Wal00].

Let $\Omega[\phi_0, B] \subseteq 2^{\mathcal{L}[\phi_0, B]} \times \mathcal{R}^{\mathcal{V}}[\phi_0, B]$. Since $\mathcal{L}[\phi_0, B]$ and $\mathcal{R}^{\mathcal{V}}[\phi_0, B]$ are both finite, $\Omega[\phi_0, B]$ is finite.

Definition 8. *For any* $(\Gamma, \overline{\delta}) \subseteq \Omega[\phi_0, B]$*,* $(\Gamma, \overline{\delta})$ *is said to be* maximal *iff:*

1. $\top \in \Gamma$, $\bot \notin \Gamma$ *;*

2. $x \bowtie r \in \Gamma$ *iff for any* $w \in \mathbb{R}_{\geq 0}$ *s.t.* $w \in \overline{\delta}(x)$*,* $w \bowtie r$*;*

3. $\phi \wedge \psi \in \Gamma$ *implies* $\phi \in \Gamma$ *and* $\psi \in \Gamma$*;*
$\phi \vee \psi \in \Gamma$ *implies* $\phi \in \Gamma$ *or* $\psi \in \Gamma$*;*

4. $X \in \Gamma$ *implies* $\phi \in \Gamma$*, for* $X = \phi \in B$*.*

The following definition establishes the framework on which we will define our model.

Definition 9. *Let $C \subseteq 2^{\Omega[\phi_0,B]}$. C is said to be mutually-consistent if for any $(\Gamma, \overline{\delta}) \in C$, whenever $\langle a\rangle\psi \in \Gamma$, then there exists $(\Gamma', \overline{\delta'}) \in C$ s.t.:*

1. there exists $\overline{\delta''}$ s.t. $\overline{\delta} \rightsquigarrow \overline{\delta''}$ and $\overline{\delta'} = \overline{\delta''}[\mathcal{V}_a \mapsto 0]$;

2. $\psi \in \Gamma'$;

3. for any $[a]\psi' \in \Gamma$, $\psi' \in \Gamma'$.

We say that $(\Gamma, \overline{\delta})$ is *consistent* if it belongs to some mutually-consistent set.

The following lemma proves a necessary precondition for the model construction.

Lemma 2. *Let $\phi \in \mathcal{L}$ be a formula closed w.r.t. a maximal equation block B. Then ϕ is satisfiable iff there exist $\Gamma \subseteq \mathcal{L}[\phi_0, B]$ and $\overline{\delta} \in \mathcal{R}^{\mathcal{V}}[\phi_0, B]$ s.t. $(\Gamma, \overline{\delta})$ is consistent and $\phi \in \Gamma$.*

Proof. ($\Longrightarrow$): Suppose ϕ is satisfied in the LWS $\mathcal{W} = (M, \Sigma, \mathcal{K}, \theta)$ under the resource valuation $l \in L$, i.e., there exists $m \in M$ s.t. $\mathcal{W}, (m, l) \models_B \phi$. We construct

$$C = \{(\Gamma, \overline{\delta}) \in \Omega[\phi_0, B] \mid \exists m \in M \text{ s.t. for any } \psi \in \Gamma, \exists l \in \overline{\delta} \text{ s.t. } \mathcal{W}, (m, l) \models_B \psi\}.$$

It is not difficult to verify that C is a mutually-consistent set.

($\Longleftarrow$): Let C be a mutually-consistent set. We construct a symbolic LWS $\mathcal{W}^s = (\Pi^s, \Sigma^s, \theta^s)$, where $\Pi^s = C$, $\Sigma^s = \Sigma[\phi_0, B]$ and for $(\Gamma, \overline{\delta}), (\Gamma', \overline{\delta'}) \in C$, the transition relation $(\Gamma, \overline{\delta}) \longrightarrow_a (\Gamma', \overline{\delta'})$ is defined iff

1. there exists $\overline{\delta''}$ s.t. $\overline{\delta} \rightsquigarrow \overline{\delta''}$ and $\overline{\delta'} = \overline{\delta''}[\mathcal{V}_a \mapsto 0]$;

2. whenever $[a]\psi \in \Gamma$ then $\psi' \in \Gamma'$.

Let $\rho^s(X) = \{(\Gamma, \overline{\delta}) \mid X \in \Gamma\}$ for $X \in \mathcal{X}$. With this construction we can prove the following implication by an simple induction on the structure of ϕ, where $(\Gamma, \overline{\delta}) \in \Pi^s$:

$$\phi \in \Gamma \text{ implies } \mathcal{W}^s, (\Gamma, \overline{\delta}), \rho^s \models^s \phi.$$

We prove that ρ^s is a fixed point of B under the assumption that $X = \phi_X \in B$:
$\Gamma \in \rho^s(X)$ implies $(X, \overline{\delta}) \in \Gamma$ by the construction of ρ^s, which implies $(\phi_X, \overline{\delta}) \in \Gamma$ by the definition of $\Omega[\phi_0, B]$. Then, by the implication we just proved, $\mathcal{W}^s, \Gamma, \rho^s \models^s \phi_X$.

Thus ρ^s is a fixed point of B. Since $[\![B]\!]$ is the maximal fixed point, $\rho^s \subseteq [\![B]\!]$.

Then for any $(\phi, \overline{\delta}) \in \Gamma \in C$, we have $\mathcal{W}^s, (\Gamma, \overline{\delta}), \rho^s \models^s \phi$, which further implies $\mathcal{W}^s, (\Gamma, \overline{\delta}), [\![B]\!] \models^s \phi$ because $\rho^s \subseteq [\![B]\!]$.

Hence, $\phi \in \Gamma$ and $(\Gamma, \overline{\delta}) \in C$ implies $\mathcal{W}^s, \Gamma \models^s_B \phi$.

By Lemma 1, there exists a LWS $\mathcal{W} = (M, \Sigma, \mathcal{K}, \theta)$ and a resource valuation $l \in L$ such that $\mathcal{W}, (m, l) \models_B \phi$ with $m \in M$. ∎

The above lemma allows us to conclude the finite model construction.

Theorem 1. *For any satisfiable RWL formula ϕ closed w.r.t. a maximal equation block B, there exists a finite symbolic LWS $\mathcal{W}^s = (\Pi^s, \Sigma^s, \theta^s)$ such that $\mathcal{W}^s, \pi \models^s_B \phi$ for some $\pi \in \Pi^s$. Further more, there exists a LWS $\mathcal{W} = (M, \Sigma, \mathcal{K}, \theta)$ and a resource valuation $l \in L$ such that $\mathcal{W}, (m, l) \models_B \phi$ for some $m \in M$.*

Theorem 2 (Decidability of B-satisfiability). *For an arbitrary maximal equation block B, the B-satisfiability problem for RWL is decidable.*

Example 4.

6 Conclusion

References

[ACD90] Rajeev Alur, Costas Courcoubetis, and David L. Dill. Model-checking for real-time systems. In *LICS*, pages 414–425, 1990.

[ACD93a] Rajeev Alur, Costas Courcoubetis, and David L. Dill. Model-checking in dense real-time. *Inf. Comput.*, 104(1):2–34, 1993.

[ACD93b] Rajeev Alur, Costas Courcoubetis, and David L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, May 1993.

[AD90] Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Mike Paterson, editor, *ICALP*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.

[AILS07] Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri Srba. *Reactive Systems: modelling, specification and verification*. Cambridge University Press, 2007.

[ATP01] Rajeev Alur, Salvatore La Torre, and George J. Pappas. Optimal paths in weighted timed automata. In Benedetto and Sangiovanni-Vincentelli [BSV01], pages 49–62.

[BFH+01] Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Guldstrand Larsen, Paul Pettersson, Judi Romijn, and Frits W. Vaandrager. Minimum-cost reachability for priced timed automata. In Benedetto and Sangiovanni-Vincentelli [BSV01], pages 147–161.

[BSV01] Maria Domenica Di Benedetto and Alberto L. Sangiovanni-Vincentelli, editors. *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001, Proceedings*, volume 2034 of *Lecture Notes in Computer Science*. Springer, 2001.

[CKS92] Rance Cleaveland, Marion Klein, and Bernhard Steffen. Faster model checking for the modal mu-calculus. In Gregor von Bochmann and David K. Probst, editors, *CAV*, volume 663 of *Lecture Notes in Computer Science*, pages 410–422. Springer, 1992.

[CS93] Rance Cleaveland and Bernhard Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2(2):121–147, 1993.

[DKV09] Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. Springer Verlag, 2009.

[HC96] G.E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic*. Routhledge, London, 1996.

[HKT01] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic Logic*. The MIT Press, 2001.

[HNSY92] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic model checking for real-time systems. In *LICS*, pages 394–406, 1992.

[Lar90] Kim Guldstrand Larsen. Proof systems for satisfiability in hennessy-milner logic with recursion. *Theor. Comput. Sci.*, 72(2&3):265–288, 1990.

[LLW95] François Laroussinie, Kim Guldstrand Larsen, and Carsten Weise. From timed automata to logic - and back. In Jirí Wiedermann and Petr Hájek, editors, *MFCS*, volume 969 of *Lecture Notes in Computer Science*, pages 529–539. Springer, 1995.

[LM13] Kim G. Larsen and Radu Mardare. Complete proof system for weighted modal logic. *Theoretical Computer Science*, to appear, 2013.

[LMXa] Kim G. Larsen, Radu Mardare, and Bingtian Xue. Adequacy and strongly-complete axiomatization for timed modal logic. under review.

[LMXb] Kim G. Larsen, Radu Mardare, and Bingtian Xue. Decidability and expressiveness of recursive weighted logic. under review.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.

[Wal00] Igor Walukiewicz. Completeness of kozen's axiomatisation of the propositional μ-calculus. *Inf. Comput.*, 157(1-2):142–182, 2000.