Tableaux For Event-Recording Logic

On deciding the existence of deterministic models

Nguena-Timo Omer

IRIT - INPT/ENSEEIHT
Université Toulouse 3
Toulouse
France
nguena@enseeiht.fr

ABSTRACT. Given a real-time system specification, the satisfiability problem is to decide the existence of a system that models the specification, and eventually to synthesise a witness system. Usually, the systems are required to be deterministic. This paper consider the deterministic-satisfiability problem for the timed μ-calculus called Event-Recording Logic (ERL). ERL is adapted for specifying timed properties of real-time systems described with Event-Recording Automata (ERA). Thus, we want to know whether there exists a procedure that decides whether ERL formulae have deterministic ERA models. Assuming some restrictions on the timing resources of models, we propose an EXPTIME decision procedure. The general case is left open.

RÉSUMÉ. Étant donnée une spécification de systèmes temps-réel, il est important de décider de l'existence d'un système qui la modélise ou la satisfait, et éventuellement construire un modèle: c'est le problème de satisfaisabilité. En pratique, les systèmes sont déterministes. Event-Recording Logic (ERL) est une adaptation temporisée du μ-calcul pour décrire les propriétés des systèmes temps-réel modélisés par des Event-Recording Automata (ERA). Nous étudions la satisfaisabilité-déterministe de ERL : nous voulons décider de l'existence de modèles ERA déterministes. Il s'agit d'une étude pionnière sur la recherche de modèles déterministes d'extensions temporisées du μ-calcul. Nous proposons des règles de tableaux qui permettent un raisonnement inductif pour la décision. Lorsque la granularité des modèles est donnée à l'avance, nous proposons un algorithme de décision EXPTIME. La décision est laissée ouverte lorsque la granularité n'est pas connue.

KEYWORDS : Satisfiability, deterministic models, timed μ-calculus, Event-recording Automata, tableaux

MOTS-CLÉS : Satisfiabilité, modèles déterministes, μ-calcul temporisé, automates temporisés, tableaux

1. Introduction

The satisfiability problem (SAT) amounts to decide whether systems specifications can be modelled/implemented, and eventually to synthesise witness models/implementations. Very often, only deterministic systems are of our interest: this is the d-satisfiability (DSAT) problem . Formal methods consider temporal logics formulae for specifying the systems modelled with transitions systems. We consider DSAT for the timed Branching-Time Temporal Logic (BTTL) called Event-Recording Logic (ERL) [8].

SAT and DSAT has been studied for varieties of untimed and timed BTTL and models. The Kozen's μ-calculus [5] is one of the most studied untimed BTTL and it is adequate for Kripke Structures. SAT and DSAT for the μ-calculus are EXPTIME Complete and the synthesis of witness models is effective [7]. An interesting proof for the (deterministic) satisfiability is based on the notion of tableau. Tableaux are proof trees constructed by applying reduction rules and Rabin/parity automata are used to check well-foundeness properties over paths of tableaux [7]. Besides, Timed Automata (TA) [1] and Event-Recording Automata (ERA) [2] are famous timed extensions of Kripke Structures. They use constraints to restrict the firings of actions and transitions. Constraints compare real-valued clock variables with rationals. ERA are less expressive than TA [2]. But ERA are determinizable and closed under Boolean operations, just like Kripke Structures. Unfortunately, there are few decidable results for timed μ-calculus. SAT for $\mathcal{L}_\nu$, WT_μ are still open [4, 6] and SAT of a fragment of WT_μ containing ERL is EXPTIME Complete [6]. ERL is more expressive than MECS [3]. Early tableau-based decision procedures for the fragment of WT_μ and ERL [8, 6] allow to build witness non deterministic ERA models.

We consider DSAT for ERL. Awkwardly, building deterministic models by determinizing non deterministic ones (obtained from [8, 6]) is not a correct solution. Indeed, determinization procedures (for example [2]) do not preserve branching-time properties. But, adapting former [7, 8, 6] tableau-based procedures to DSAT for ERL (and probably SAT for WT_μ, SAT for $\mathcal{L}_\nu$) is not immediate. In [8, 6], non deterministic models compare clocks with constants specified in formulae only. But as we will show, deterministic models may required constant not specified in formulae. Thus, we provide tableau rules adapted for the case when the constants are also given in advance. In this case, our tableau-based procedure allows to decide DSAT.

The paper is organised as follows: ERA and ERL are defined in the next section. In Section 3, we present technical constructions, including regions and normalised formulae. In section 4, we define two systems of rules for SAT and DSAT; then we present their properties. We provide a decision procedure for parametrised DSAT. Due to the paper format, intuitions are preferred to the long proofs. Section 5 concludes the paper.

2. Definitions

In the sequel $\Sigma = \{a, b, \ldots\}$ denotes a set of actions, $\mathsf{Var} = \{X, Y, \ldots\}$ denotes a finite set of variables, and the time domain $\mathbb{T}$ is the set $\mathbb{R}_{\geq 0}$ of non negative real numbers.

Clocks, Constraints. In the context of ERA and ERL, we consider $\mathcal{X}_\Sigma = \{x_a, x_b, \ldots\}$ the set of clocks. A clock x_a refers to the action a. A *clock valuation* $v : \mathcal{X}_\Sigma \to \mathbb{T}$ assigns to each clock a time value. The set of clock valuations is denoted by $\mathbb{T}^\Sigma$. Given $\delta \in \mathbb{T}$, the valuation $(v + \delta)$ is defined by: $(v + \delta)(x) = v(x) + \delta$ for every $x \in \mathcal{X}_\Sigma$. For $x' \in \mathcal{X}_\Sigma$, $v[x' := 0]$ denotes the valuation such that, $(v[x' := 0])(x') = 0$ and for

each $x \in \mathcal{X}_\Sigma \setminus \{x'\}$, $(v[x' := 0])(x) = v(x)$. The valuation denoted by $\mathbf{0}$ maps every clock to zero and $v\uparrow = \{v + \delta \mid \delta \in \mathbb{T}\}$. The set of *constraints* over $\mathcal{X}_\Sigma$, denoted by $\mathcal{C}(\Sigma)$, is defined by the grammar "$g ::= x \bowtie c \mid g \wedge g \mid \mathtt{tt}$" where $x \in \mathcal{X}_\Sigma$, $c \in \mathbb{Q}_{\geq 0}$ is a non negative rational, $\bowtie \in \{<, >, \geq, \leq, =\}$ and $\mathtt{tt}$ stands for true. We write $v \models g$ (or $v \in [\![g]\!]$) when the valuation v satisfies g, using the standard semantics.

History clock timed transition system (HCTTS). A HCTTS is a tuple $\mathcal{S} = \langle Q, q^0, \mathbb{T}^\Sigma \times \Sigma, \pi, \rightarrow \rangle$ where Q is the set of states, q^0 is the initial state, the function $\pi : Q \rightarrow \mathbb{T}^\Sigma$ assigns a valuation to every state and the transition relation $\rightarrow \subseteq Q \times (\mathbb{T}^\Sigma \times \Sigma) \times Q$ is such that: $q \xrightarrow{v,a} q'$ if $v \in \pi(q)\uparrow$ and $\pi(q') = \pi(q)[x_a := 0]$. We say that $\mathcal{S}$ is *deterministic* (DHCTTS) iff for every $(q, v, a) \in Q \times \mathbb{T}^\Sigma \times \Sigma$ there is at most one outgoing transition from q labelled with (v, a).

Event-recording logic (ERL). The formulae of ERL [8] are defined by the following grammar : $\varphi ::= \mathtt{tt} \mid \mathtt{ff} \mid X \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [g, a]\varphi \mid \langle g, a \rangle \varphi \mid \nu X.\varphi \mid \mu X.\varphi$ where $g \in \mathcal{C}(\Sigma)$.
For a HCTTS $\mathcal{S} = \langle Q, q^0, \mathbb{T}^\Sigma \times E, \pi, \rightarrow \rangle$ an assignment $\mathcal{V} : \mathsf{Var} \rightarrow \mathcal{P}(Q)$, the semantics of a formula φ under $\mathcal{S}$, $[\![\varphi]\!]^{\mathcal{S}}_{\mathcal{V}}$ is the set of states for which the formula holds:

– $[\![[g, a]\varphi]\!]^{\mathcal{S}}_{\mathcal{V}} := \{q \in Q \mid \forall q \xrightarrow{v,a} q', v \models g, \text{implies } q' \in [\![\varphi]\!]^{\mathcal{S}}_{\mathcal{V}}\}$

– $[\![\langle g, a \rangle \varphi]\!]^{\mathcal{S}}_{\mathcal{V}} := \{q \in Q \mid \exists q \xrightarrow{v,a} q' \text{ s.t } v \models g \wedge \text{and } q' \in [\![\varphi]\!]^{\mathcal{S}}_{\mathcal{V}}\}$

– The semantics of the other operators is standard.

We say that $\mathcal{S}$ *models* φ, denoted by $\mathcal{S} \models \varphi$ iff $q^0 \in [\![\varphi]\!]^{\mathcal{S}}_{\mathcal{V}}$.

We consider standard notions of sentences, binding definitions $Bd_\varphi(X)$ and the older relation order between variables [7, 6].

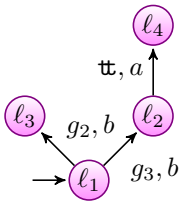
Deterministic event-recording automata. An ERA is a tuple $\mathcal{A} = \langle L_\mathcal{A}, \ell^0_\mathcal{A}, \mathcal{X}_\Sigma, \Sigma, E_\mathcal{A} \rangle$ where $L_\mathcal{A}$ and $E_\mathcal{A} \subseteq L_\mathcal{A} \times \mathcal{C}(\Sigma) \times \Sigma \times L_\mathcal{A}$ are finite sets of *locations* and *edges*, respectively. The initial location is $\ell^0_\mathcal{A}$. For an edge $e = \ell \xrightarrow{g,a} \ell'$, we define $src(e) = \ell$, $tgt(e) = \ell'$, $g_e = g$, $\sigma(e) = a$. The semantics of $\mathcal{A}$ is the HCTTS $\mathcal{S}_\mathcal{A} = \langle Q_\mathcal{A}, q^0_\mathcal{A}, \mathbb{T}^\Sigma \times E, \pi_\mathcal{A}, \rightarrow \rangle$ where $Q_\mathcal{A} = (L_\mathcal{A} \times \mathbb{T}^\Sigma)$, $q^0_\mathcal{A} = (\ell^0_\mathcal{A}, \mathbf{0})$, $\pi_\mathcal{A}(\ell, v) = v$, and the transitions in $\rightarrow$ are such that: $(\ell, v) \xrightarrow{v',a} (\ell', v'[x_a := 0])$ iff there exist $\ell \xrightarrow{g,a} \ell' \in E_\mathcal{A}$, $v' \in v\uparrow$ such that $v' \models g$. We say that $\mathcal{A}$ is *deterministic* (DERA) if $\mathcal{S}_\mathcal{A}$ is deterministic. The semantics defines the crossing of the edges. Each transition corresponds to an elapse of the time followed by an the crossing of an edge e and the firing of the action $\sigma(e)$, provided that $\sigma(e) \in \Sigma$ occurs when the constraint g_e is satisfied. The history clock x_a associated to $a \in \Sigma$ measures the time elapsed since the last occurrence of a.

Satisfiability (SAT and DSAT). A ERL formula φ is *satisfiable* (resp. *d-satisfiable*) is there exists an ERA (resp. DERA) $\mathcal{A}$ s.t $\mathcal{S}_\mathcal{A} \models \varphi$. The *satisfiability (resp. d-satiafiablity) problem*, SAT (resp. DSAT) amounts to decide whether ERL formulae are satiafiable (resp. d-satiafiable) and eventually to construct witness models.

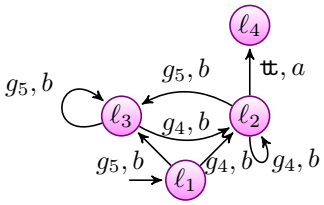
Examples. Examples of constraints, ERA ($\mathcal{A}_1$ and $\mathcal{A}_3$), DERA ($\mathcal{A}_2$ and $\mathcal{A}_4$) and ERL formulae appear in Figure 1. The formula φ_3 requires to fire a after b and sometimes forbid any firing of a after b. The formulae φ_4, φ_5 are greatest fixpoint formulae describing liveliness properties or "infinite repetitions" of φ_3. In particular, φ_4 states that the requirements of φ_3 must be satisfied after each firing of b. Observe that $\mathcal{A}_1 \models \varphi_1$. Besides, $\mathcal{A}_1 \models \varphi_2$ even if its initial location ℓ_1 has no outgoing edge labelled with a. We also observe that the ERA $\mathcal{A}_1 \models \varphi_3$, $\mathcal{A}_2 \models \varphi_3$, $\mathcal{A}_3$ models φ_4 and φ_5. Finally, we observe that the DERA $\mathcal{A}_4$ models φ_5, but it does not models φ_4. Later we discuss why φ_4 is satisfiable, but not d-satisfiable.

$g_1 := (0 < x_a < 1 \wedge g_6)$
$g_2 := (0 < x_a < \frac{1}{2} \wedge g_6)$
$g_3 := (\frac{1}{2} < x_a < 1 \wedge g_6)$
$g_4 := 0 < x_b < \frac{1}{2}$
$g_5 := \frac{1}{2} < x_b < 1$
$g_6 := 0 < x_b < 1$

(a) Examples of constraints



(b) DERA $\mathcal{A}_2$



(c) DERA $\mathcal{A}_4$

$\mathcal{A}_1$ (resp. $\mathcal{A}_3$) := "replace both g_2 and g_3 (resp. g_4 and g_5) by g_1 in $\mathcal{A}_2$ (resp. $\mathcal{A}_4$)"

$\varphi_1 := \langle g_1, b\rangle \mathtt{tt} \quad \varphi_2 := [\mathtt{tt}, a]\mathtt{ff} \quad \varphi_4 := \nu X.\langle g_1, b\rangle\langle \mathtt{tt}, a\rangle \mathtt{tt} \wedge \langle g_2, b\rangle[\mathtt{tt}, a]\mathtt{ff} \wedge [\mathtt{tt}, b]X$
$\varphi_3 := \langle g_1, b\rangle\langle \mathtt{tt}, a\rangle \mathtt{tt} \wedge \langle g_1, b\rangle[\mathtt{tt}, a]\mathtt{ff} \quad \varphi_5 :=$ "replace both g_1 and g_2 with g_6 in φ_4"

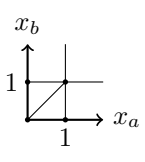
(e) Examples of ERL formulae

FIGURE 1: Examples of ERA, DERA and ERL formulae

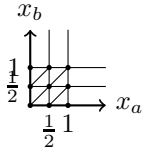
3. Region-based Normalised representations and semantics

Granularity. A granularity is a measure of rational constants used in constraints. A *granularity* is a pair $(d, M) \in \mathbb{N} \times \mathbb{N}$. Let $\xi_1 = (d_1, M_1)$ and $\xi_2 = (d_2, M_2)$ be two granularities: ξ_1 is *finer* than ξ_2 and we write $\xi_1 \preceq \xi_2$ if $\exists k \in \mathbb{N}^*$ s.t $d_2 = k \times d_1$ and $M_1 \geq M_2$. The *sum* $\xi_1 \oplus \xi_2$ is the granularity $(lcm(d_1, d_2), Max(M_1, M_2))$ where lcm stands for the least common multiple. A rational $r \in \mathbb{Q}_{\geq 0}$ can be produced by granularity (d, M) iff $r \leq M$ and there exist $n \in \mathbb{N}$ such that $r = \frac{n}{d}$. The granularity of a constrained object O, ξ_O is the less fine granularity used for producing the constants occurring in the constraints. We denote by $\mathcal{C}_\xi(\Sigma)$, the set of constraints of granularity ξ. ERA$_\xi$ and DERA $_\xi$ denotes the set of ERA and DERA with the granularity ξ.

Regions Given a granularity $\xi = (d, M)$, two clock valuations are equivalent of they satisfy the same constraints in $\mathcal{C}_\xi(\Sigma)$, when the time elapses or when clocks are reset. The region [1] of a valuation v, $[v]_\xi$ is the set of valuations equivalent to v. The set of regions is denoted by $Reg_\xi(\Sigma)$. Note that the size of $Reg_\xi(\Sigma)$ is in $O(2^{|\Sigma|})$. Given $[v]_\xi \in Reg_\xi(\Sigma)$, we define the $[v]_\xi\uparrow = \{[v'+\delta]_\xi \mid v' \in [v]_\xi, \delta \in \mathbb{T}\}$ and $([v]_\xi)[x_a := 0] = \{[v'[x_a := 0]]_\xi \mid v' \in [v]_\xi\}$, the regions reachable from $[v]_\xi$ after the time elapsing and the reset of x_a. Figure 2(a) presents the regions for $\xi = (1, 1)$ and $\Sigma = \{a, b\}$. A region is a black point, a triangle, a half line or an open space. A region is definable with constraints involving comparisons between two clocks. Let the region $r_0 := (x_a = 0 \wedge x_b = 0)$. The region $r_1 := (0 < x_a < 1 \wedge 0 < x_b < 1 \wedge x_a - x_b = 0)$ is the immediate time successor of r_0 and $r_2 := (0 < x_a < 1 \wedge x_b = 0 \wedge 0 < x_a - x_b < 1)$ equals $r_1[x_a := 0]$.



(a) $\xi = (1, 1)$



(b) $\xi = (2, 1)$

FIGURE 2: Regions with $\Sigma = \{a, b\}$

ξ-atomic constraints. A ξ-atomic constraint is a smallest constraint in $\mathcal{C}_\xi(\Sigma)$; it is a constraint of the form $\bigwedge_{x \in \mathcal{X}_\Sigma} e_x$ where e_x is of the form $x > M$, $x = M$, $x = c$, $c < x < c + \frac{1}{d}$ with $c < M$. Note that any region $[v]_\xi$ is included in a unique ξ-atomic constraint g such that $v \models g$. We can show that [6], for any constraint g, any granularity $\xi \preceq \xi_g$, g can be decomposed into a set $R_\xi(g)$ of disjoints ξ_g-atomic constraints.

Abstract representations for ERA. A ξ-abstract representation for an ERA $\mathcal{A}$, is the kripke structure $\mathcal{R}^\xi_\mathcal{A} = \langle L \times Reg_\xi(\Sigma), (\ell^0, [\mathbf{0}]_\xi), \mathcal{C}_\xi(\Sigma) \times \Sigma, \rightarrow\rangle$ where the transitions

between *abstract* states are such that: $(\ell, [v]_\xi) \xrightarrow{g',a} (\ell, ([v']_\xi)[x(e) := 0])$ whenever there exist $[v']_\xi \in ([v]_\xi\uparrow)$, $e = \ell \xrightarrow{g,a} \ell'$ such that $[v']_\xi \subseteq [\![g']\!]$ and g' is atomic. As in [6], we can show that for any ERA $\mathcal{A}$, any granularity $\xi \preceq \xi_\mathcal{A}$, $\mathcal{S}_\mathcal{A}$ is isomorphic to $\mathcal{S}_{\mathcal{R}^\xi_\mathcal{A}}$.

Normalised formula $N_\xi(\varphi)$. $N_\xi(\varphi)$ is constructed from φ by replacing each subformula of the form $\langle g, a\rangle\varphi$ (resp. $[g, a]N_\xi(\varphi)$) with $\bigvee_{g' \in R_\xi(g)} \langle g', a\rangle N_\xi(\varphi)$ (resp. $\bigwedge_{g' \in R_\xi(g)} [g', a]\varphi$). As in [6], we can show that for any sentence φ, any ERA $\mathcal{A}$, any granularity $\xi \preceq \xi_\varphi$, $(\ell, v) \in [\![\varphi]\!]^\mathcal{A}$ iff $(\ell, v) \in [\![N_\xi(\varphi)]\!]^\mathcal{A}$.

Abstract semantics for ERL. The abstract semantics of φ, $\langle\!|\varphi|\!\rangle$ is defined by performing an equality test between the constraints and the action in formulae and the constraints and action labelling the transitions of ERA: $\langle\!|\langle g, a\rangle\varphi|\!\rangle^\mathcal{A} = \{\ell \mid \exists(\ell, g, a, \ell') \in E_\mathcal{A}$ s.t $\ell' \in \langle\!|\varphi|\!\rangle^\mathcal{A}\}$ and $\langle\!|[g, a]\varphi|\!\rangle^\mathcal{A} = \{\ell \mid \forall(\ell, g, a, \ell') \in E_\mathcal{A}$ it holds that $\ell' \in \langle\!|\varphi|\!\rangle^\mathcal{A}\}$. Observe that this semantics is similar to the Kozen's μ-calculus semantics [7], where ERL and ERA are considered as μ-calculus and Kripke structure over the alphabet $(\mathcal{C}_\xi(\Sigma) \times \Sigma)$. According to Proposition 3.1, one can adapt the model-checking result on the μ-calculus to ERL by choosing a granularity finer than those of the formula and the ERA.

Proposition 3.1. *For every $\xi \preceq (\xi_\varphi \oplus \xi_\mathcal{A})$, $\mathcal{A} \models \varphi$ iff $(\ell^0, [\mathbf{0}]) \in \langle\!|N_\xi(\varphi)|\!\rangle^{\mathcal{R}^\xi_\mathcal{A}}$*

In the sequel, we consider normalised formulae only.

4. Tableau for SAT and DSAT with ERL

We adapt the rules for the μ-calculus [7] to ERL by adding timing information. ERL is a kind of Kozen's μ-calculus augmented with clock constraints. We propose valuation-based and region-based rules for SAT and DSAT.

Clock valuation-based systems of rules for ERL (NR and DR). In Figure 3, we propose two clock valuation-based systems of rules for ERL: NR and DR. The first system adapted for SAT, SR is composed of five standard rules ($\vee$, $\wedge$, ν, μ, reg), the rules $time$ and $tmod$. The second system adapted for DSAT, DR is composed of the standard rules, the rules $time$ and $tdmod$. Each rule reduces the satisfiability checking of formulae in its conclusion (below the line of the rules) to the satisfiability checking of formulae in its premises (above the line). The rule $\vee$, $time$, $tmod$ and $dtmod$ have more than one premise. Except the rule reg, which abstracts the computation of fixpoint formulae, all the other rules reduce the size of formulae in their conclusion.

$$\frac{\{\varphi_1, \varphi_2, \Gamma\}; v}{\{\varphi_1 \wedge \varphi_2, \Gamma\}; v}(\wedge) \quad \frac{\{\varphi(X), \Gamma\}; v}{\{\mu X.\varphi(X), \Gamma\}; v}(\mu) \quad \frac{\{\varphi(X), \Gamma\}; v}{\{X, \Gamma\}; v}(reg)\, Bd_\varphi(X) = \sigma X.\varphi(X)$$

$$\frac{\{\varphi(X), \Gamma\}; v}{\{\nu X.\varphi(X), \Gamma\}; v}(\nu) \quad \frac{\{\varphi_1, \Gamma\}; v \quad \{\varphi_2, \Gamma\}; v}{\{\varphi_1 \vee \varphi_2, \Gamma\}; v}(\vee) \quad \frac{\Gamma; v; \tau_v \quad \text{for each } \tau_v \in \mathcal{F}_v(\Gamma_{\langle\rangle})}{\Gamma; v}(time)$$

$$\frac{\varphi \cup \{\psi \mid [g, a]\psi \in \Gamma_{v_i}\}; v_i[x_a := 0] \left\{ \begin{array}{l} \text{for each } v_i \in v\uparrow \\ \quad \text{for each} \langle g, a\rangle\varphi \in \tau^{-1}(v_i) \end{array} \right.}{\Gamma; v; \tau}(tmod)$$

$$\frac{\{\psi \mid \begin{array}{l} \langle g, a\rangle\psi \in \tau^{-1}(v_i) \\ \text{or } [g, a]\psi \in \Gamma_{v_i} \end{array} \}; (v + \delta_i)[x_a := 0] \left\{ \begin{array}{l} \text{for each } v_i \in v\uparrow \\ \quad \text{for each } a \in \Sigma \text{ s.t} \\ \quad\quad \langle g, a\rangle\varphi \in \tau^{-1}(v_i) \end{array} \right.}{\Gamma; v; \tau}(tdmod)$$

FIGURE 3: Valuations-based tableau rules for ERL

Let us briefly comment the non standard rules. The rule $time$ allows to choose the reduction time for existential modalities of the form $\langle g, a\rangle$. It considers the set of functions $\mathcal{F}_v(\Gamma_{\langle\rangle})$ of the form $\tau_v : \Gamma_{\langle\rangle} \to \mathbb{T}^{\Sigma}$ assigning a reduction time to each $\langle g, a\rangle\varphi \in \Gamma_{\langle\rangle}$ such that $v' = \tau_v(\langle g, a\rangle\varphi)$ implies that $v' \in v{\uparrow}$ and $v' \models g$. We define $\Gamma_{\langle\rangle} = \{\langle g, a\rangle\varphi \mid \langle g, a\rangle\varphi \in \Gamma\}$ and $\Gamma_v = \{[g, a]\varphi \mid [g, a]\varphi \in \Gamma \wedge v \models g\}$. The rule $tmod$ applies the reduction and adds a premise for each subformula $\langle g, a\rangle\varphi$ in $\Gamma_{\langle\rangle}$ according to its reduction time given by $\tau(\langle g, a\rangle\varphi)$. The reduction is "non deterministic" since two premises are created for two subformulae with equal reduction times. The rule $tdmod$ performs a "deterministic reduction" and adds one premise for each subset of subformulae reduced in equal times and with a same action. Observe that $tmod$, $tdmod$ are timed extensions of the following rules defined for the μ-calculus:

$$\frac{\varphi \cup \{\psi \mid [a]\psi \in \Gamma\}, \text{for each} \langle a\rangle\varphi \in \Gamma}{\Gamma}(mod)$$

$$\frac{\{\psi \mid \langle a\rangle\psi \in \Gamma \text{ or } [a]\psi \in \Gamma\}, \text{for each } a \in \Sigma \text{ s.t } \exists\langle a\rangle\varphi \in \Gamma}{\Gamma}(dmod)$$

Tableaux, trace and pre-models. These notions are adapted from [7, 6]. Given SR or DR, a *valuation-based tableau* for an ERL formula φ is a tree the root of which is labelled with $\{\varphi\}; \mathbf{0}$ and the nodes is obtained by applying the rules. We require that the rule $time$ is applicable when no standard rule is applicable; the rules $tmod$, $tdmod$ are applied just after the rule $time$. Given a rule (for example $\vee$), the reduced formula with the valuation (for example $\varphi_1 \vee \varphi_2$) in conclusion is linked to its subformula (for example φ_1 or φ_2) with the valuation in a premise: a succession of such links defines a *trace* of the reduction of a formula along a path of a tableau. A variable X regenerates on a trace iff some node and its successor are labelled with X and $\varphi(X)$ respectively. A ν-trace (resp. a μ-trace) is either finite and neither (resp. either) ends with $\mathtt{ff}$, nor (resp. or) with a formula of the form $\langle g, a\rangle\varphi$, or is infinite and the oldest variable that is regenerated infinitely often is a ν-variable (resp. μ-variable). A *pre-model* is a sub-tableau obtained by selecting exactly one premise for each node at which the rule $\vee$ or $time$ is applied, so that there is no μ-trace in the remaining paths.

Given a formula φ, let $\mathcal{T}(\varphi)$ and $\mathcal{DT}(\varphi)$ denote the tableaux for φ constructed using NR and DR respectively. Guided by the proofs of similar results in [6] and after we have adapted the concept of *signature*, we can prove the following propositions.

Lemma 4.1. *A HCTTS models an* ERL *formula φ iff there is a pre-model in $\mathcal{T}(\varphi)$.*

Let us sketch the proof. For ($\Rightarrow$), we assume that here is a HCTTS $\mathcal{S}$ which models φ. We construct $\mathcal{T}(\varphi)$, then we mark the nodes with the states of $\mathcal{S}$ and we select the premises of the nodes at which the rule $\vee$ or $time$ is applied according to the signature property. Since $\mathcal{S}$ models φ, the subtree composed of the selected nodes is a pre-model. For ($\Leftarrow$), we construct a HCTTS $\mathcal{S}$ from the pre-model and we show that $\mathcal{T}(\varphi)$ does not contain a pre-model if $\mathcal{S}$ does not model φ, leading to a contradiction.

Lemma 4.2. *A DHCTTS models an* ERL *formula φ iff there is a pre-model in $\mathcal{DT}(\varphi)$.*

Note that checking the existence of a pre-model in $\mathcal{T}(\varphi)$ or $\mathcal{DT}(\varphi)$ may not be decidable since, contrarily to the case of the μ-calculus, the labels of the nodes range over an infinite set (valuations occur labels). Moreover, the sketch of the proof above construct witness HCTTS which models φ. So, it is not immediate that the existence of pre-models in $\mathcal{T}(\varphi)$ or $\mathcal{DT}(\varphi)$ leads to a decision procedure for SAT and DSAT since we want ERA or DERA model. For these reasons, we consider abstract rules.

$$\frac{\Gamma; r; \theta \quad \text{for each } \theta \in \Theta_r(\Gamma_{\langle\rangle})}{\Gamma; r}(time_\xi)$$

$$\frac{\varphi \cup \{\psi \mid [g,a]\psi \in \Gamma\}; r_i[x_a := 0] \left\{ \begin{array}{l} \text{for each } r_i \in r\uparrow \\ \quad \text{for each} \langle g,a\rangle\varphi \in \theta^{-1}(r_i) \end{array} \right.}{\Gamma; r; \theta}(tmod_\xi)$$

$$\frac{\{\psi \mid \langle g,a\rangle\psi \in \theta^{-1}(r_i) \text{ or } [g,a]\psi \in \Gamma\}; r_i[x_a := 0] \left\{ \begin{array}{l} \text{for each } r_i \in r\uparrow \\ \quad \text{for each } a \in \Sigma \text{ s.t} \\ \quad\quad \langle g,a\rangle\varphi \in \theta^{-1}(r_i) \end{array} \right.}{\Gamma; r; \theta}(tdmod_\xi)$$

FIGURE 4: Regions based tableau rules for ERL

Region-based tableaux for ERL ($NR(\xi)$ and $DR(\xi)$). Timing context in regions-based rules are regions parametrised with a granularity ξ. The rules for $NR(\xi)$ and $DR(\xi)$ are composed of standard rules $\vee$, $\wedge$, ν, μ, reg and the rules in Figure 4 inspired by the rules in Figure 3. A function $\theta_r \in \Theta_r(\Gamma_{\langle\rangle})$ in the rule $time_\xi$ is such that $r' = \theta_r(\langle g,a\rangle\varphi)$ implies and $r' \subseteq [\![g]\!]$. Then, we define region-based tableaux similarly to valuation-based tableaux and we consider a similar notion of well-founded path.

Given a granularity ξ and a formula φ, $\mathcal{T}(\varphi,\xi)$ and $\mathcal{DT}(\varphi,\xi)$ denote the tableaux for φ constructed with the rules of $NR(\xi)$ and $DR(\xi)$ respectively.

Lemma 4.3.
- It is decidable and EXPTIME *whether $\mathcal{T}(\varphi,\xi)$ or $\mathcal{DT}(\varphi,\xi)$ contains a pre-model.*
- If $\mathcal{T}(\varphi,\xi)$ contains a pre-model then $\mathcal{T}(\varphi,\xi')$ contains a pre-model for every $\xi' \preceq \xi$.
- If $\mathcal{DT}(\varphi,\xi)$ contains a pre-model then $\mathcal{DT}(\varphi,\xi')$ contains a pre-model for every $\xi' \preceq \xi$.

For example, let us consider φ_3 in Figure 1(e). Clearly, $\xi_{\varphi_3} = (1,1)$. Figure 5 shows fragments of $\mathcal{T}(\varphi_3,\xi_{\varphi_3})$ and $\mathcal{DT}(\varphi_3,\xi_{\varphi_3})$ (The rule time is hidden). Observe that $\mathcal{T}(\varphi_3,\xi_{\varphi_3})$ has a pre-model. But, $\mathcal{DT}(\varphi_3,\xi_{\varphi_3})$ has no pre-model, since all the traces end with $\mathtt{ff}$. However, using the region map in Figure 2(b) one can easily check that $\mathcal{DT}(\varphi_3,(2,1))$ contains a pre-model. This example provide a proof of Lemma 4.4 below.

Lemma 4.4. *There is a formula φ such that $\mathcal{DT}(\varphi,\xi_\varphi)$ does not contains a pre-model and for some $\xi \preceq \xi_\varphi$, $\mathcal{DT}(\varphi,\xi)$ contains a pre-model.*

Decidability results. We can provide, following [6], a proof of Proposition 4.1. The proposition reduces SAT to the pre-model checking problem in $\mathcal{T}(\varphi,\xi_\varphi)$.

Proposition 4.1. *A formula φ is satisfiable iff $\mathcal{T}(\varphi,\xi_\varphi)$ contains a pre-model.*

Proposition 4.1 asserts that the granularity of a formula is enough to solve SAT. But, this is not true for DSAT (see Lemma 4.4). However, we get the partial result of Proposition 4.2 by adapting our proof of Proposition 4.1.

Proposition 4.2. *A formula φ is d-satisfiable with a* DERA$_\xi$ *iff $\mathcal{DT}(\varphi,\xi)$ contains a pre-model.*

$$\ell_4\colon \{\mathtt{tt}\}; r_4$$
$$\ell_2\colon \{\langle\mathtt{tt},a\rangle\mathtt{tt}\}; r_2 \qquad \ell_3\colon \{[\mathtt{tt},a]\mathtt{ff}\}; r_2$$
$$\ell_1'\colon \{\langle g_1,b\rangle\langle\mathtt{tt},a\rangle\mathtt{tt}, \langle g_1,b\rangle[\mathtt{tt},a]\mathtt{ff}\}; r_0$$
$$\ell_1\colon \{\langle g_1,b\rangle\langle\mathtt{tt},a\rangle\mathtt{tt} \wedge \langle g_1,b\rangle[\mathtt{tt},a]\mathtt{ff}\}; r_0$$

(a) $\mathcal{T}(\varphi_3,\xi_{\varphi_3})$

$$\ell_4\colon \{\mathtt{ff}\}; r_4$$
$$\ell_3\colon \{\mathtt{tt},\mathtt{ff}\}; r_4$$
$$\ell_2\colon \{\langle\mathtt{tt},a\rangle\mathtt{tt}, [\mathtt{tt},a]\mathtt{ff}\}; r_2$$
$$\ell_1'\colon \{\langle g_1,b\rangle\langle\mathtt{tt},a\rangle\mathtt{tt}, \langle g_1,b\rangle[\mathtt{tt},a]\mathtt{ff}\}; r_0$$
$$\ell_1\colon \{\langle g_1,b\rangle\langle\mathtt{tt},a\rangle\mathtt{tt} \wedge \langle g_1,b\rangle[\mathtt{tt},a]\mathtt{ff}\}; r_0$$

(b) $\mathcal{DT}(\varphi_3,\xi_{\varphi_3})$

FIGURE 5: Tableaux for φ_3

Using, Lemma 4.3, we get a partial decidability result for DSAT when the granularity of the model is known in advance.

Theorem 4.1. *Given an* ERL *formula φ and a granularity ξ, there is a procedure that checks whether there exists a* DERA $_\xi$ *that satisfies φ.*

Let us end the section by giving some links between tableau and abstract tableau.

Corollary 4.1. *$\mathcal{T}(\varphi)$ contains a pre-model iff $\mathcal{T}(\varphi, \xi_\varphi)$ contains a pre-model*

Corollary 4.1 is a consequence of Proposition 4.1. But, $\mathcal{DT}(\varphi, \xi)$ can not be used to decide whether $\mathcal{DT}(\varphi)$ contains a model (see Lemma 4.5). Moreover, deciding the existence of pre-models in valuation-based tableaux is not a useful result for DSAT.

Lemma 4.5. *There exists a formula φ such that $\mathcal{DT}(\varphi)$ contains a pre-model, and for every $\xi \preceq \xi_\varphi$ $\mathcal{DT}(\varphi, \xi)$ does not contains a pre-model.*

Formula φ_4 (see Figure 1(e)) is a witness formula for Lemma 4.5. Indeed after each firing of b we need an additional constant to separate the instants at which a should be fired from those at which it should not. Since the process repeats infinitely often, there number of required constants is infinite and it can not be generated by a granularity.

5. Concluding remarks

We have considered the DSAT problem for ERL. We have proposed two systems of rules: the valuation-based and regions-based systems. We have established relations between the two systems. We have shown that the valuation based system can not be used to decide DSAT and the region-based system allow to decide DSAT when the constants of the models is known in advance. In case of unknown constants, DSAT is left open. Future work in this direction consider abstract rules only and the rules should handle the introduction of new constants in deterministic models.

6. References

[1] R. ALUR , D. DILL, " A theory of timed automata ", *Theoret. Comput. Sci.*, num. 126 vol. 2:183–235, 1994.

[2] R. ALUR , L. FIX , T. A. HENZINGER, " Event-clock automata: A determinizable class of timed automata", *Theor. Comput. Sci.*, num. 211 vol. 1-2:253–273, 1999.

[3] N. BERTRAND, A. LEGAY, S. PINCHINAT, J.-B. RACLET, " Modal event-clock specifications for timed component-based design", *Science of Computer Programming*, 2011.

[4] P. BOUYER, F. CASSEZ, andF. LAROUSSINIE, " Timed modal logics for real-time systems: Specification, verification and control", *J. Logic Lang. Inform.*, num. 20 vol. 2:169–203, 2011.

[5] D. KOZEN, " Results on the propositional μ-calculus", In *ICALP*, pages 348–359, 1982.

[6] O. NGUENA-TIMO, " Synthesis for a Weak Real-Time Logic", *PhD thesis*, University of Bordeaux, 2009.

[7] D. NIWINSKI, I. WALUKIEWICZ, " Games for the mu-calculus", *Theor. Comput. Sci.*, num. 163 vol. 1&2:99–116, 1996.

[8] M. SOREA, " A decidable fixpoint logic for time-outs", In *Proc. of 13th Int. Conf. on Concurrency Theory*, vol. 2421:255–271 of *LNCS*. Springer, 2002.