A Decidable Timed mu-calculus for Event-Recording Automata

Omer Landry Nguena Timo

A Decidable Timed μ-calculus for Event-Recording Automata

Omer Landry Nguena Timo[*]
Université de Bordeaux-LaBRI-CNRS,
351 Cours de la Libération, Talence, France

omer-landry.nguena-timo@labri.fr

Abstract

The Logic WT_μ is considered. We show decidability of the satisfiability checking problem for a fragment of WT_μ called C-WT_μ. WT_μ is an extension of the modal μ-calculus with event-recording clocks. C-WT_μ is more expressive than Event-Recording logic, another EXPTIME complete decidable extension of the modal μ-calculus with event-recording clocks. Based on the techniques for deciding untimed μ-calculus, we present a set of rules for constructing tableaux for formulas of C-WT_μ. The decidability problem is shown to be EXPTIME complete. We construct a witness event-recording automaton that satisfies a given C-WT_μ formula.

1 Introduction

Formal methods need "good" and efficient formalisms to describe and specify models of systems. In this context, temporal and modal logics allow elegant and succinct representations for specifications. Decidability of satisfiability checking problem together with construction of witness models are important for logics. The satisfiability checking problem for a logic is to check whether a given formula has a model. In this paper, we consider the satisfiability checking problem for an interesting fragment of WT_μ with respect to event-recording automata.

WT_μ [12] is a "weak" timed extension of the standard μ-calculus [10] for Event-Recording Automata [3]. The syntax of WT_μ extends the syntax of the μ-calculus with two modal operators indexed with clock constraints. The existential modal operator indexed with a clock constraint ($\langle g\rangle$) allows to describe a behaviour in a future time instant at which the clock constraint g is satisfied. The universal modal operator indexed with a clock constraint ($[g]$) allows to describe all behaviours in all future time instants at which the constraint g is satisfied. Formulas of $\mathrm{WT}_\mu$are boolean combinations of fixpoint formulas and

[*]The author thanks his Phd thesis supervisor for his fruitful comments and advices.

formulas starting with modal operators. C-WT$_\mu$ is a fragment of WT$_\mu$. When formulas are viewed as trees, C-WT$_\mu$ requires that every modal quantification over events to be directly preceded by a modal quantification over delays; Additionally universal quantification over events must not be directly preceded by an existential quantification over delays. Models for WT$_\mu$ and C-WT$_\mu$ are Event-Recording Automata (ERA) [3]. ERA are finite graphs. Transitions in ERA are labelled with couples made of a clock constraint and an event. A *clock constraint* is just a conjunction of comparisons of clocks with constants. A transition can be fired when values of clocks satisfy its constraints; and when a transition is fired the unique clock associated to the event of the transition is reset. In ERA, Each clock refers to a single event. ERA are closed under boolean operations [2].

The main contribution of this paper is an EXPTIME-Complete decision procedure for the satisfiability checking problem of C-WT$_\mu$. The procedure does not require limit on constants of models; it also constructs witness models. We briefly address relations between WT$_\mu$, C-WT$_\mu$and other timed extensions of the standard μ-calculus like the logic L_ν [11] and Event-Recording Logic [17] (ERL). Modal operators ERL can be translated into equivalent modal operator of WT$_\mu$and modal operators of WT$_\mu$can be translated into equivalent modal operator of L_ν. C-WT$_\mu$ is more expressive than Event-Recording logic, another EXPTIME-Complete timed μ-calculus for event-recording automata. C-WT$_\mu$ allows to characterise event-recording automata up to timed bisimulation and timed simulation [13]. We also present a method to reduce the model-checking problem of WT$_\mu$ to the model-checking problem of the standard μ-calculus which is the same as the problem of checking winning strategies in two player parity game [20]. The reduction leads to and EXPTIME algorithm.

For the satisfiability problem of C-WT$_\mu$, we use similar techniques to the satisfiability problem for the μ-calculus [18, 14] and ERL [17]. We present a set of reduction rules for constructing tableaux for WT$_\mu$ formulas. We show that a formula is satisfiable if and only if its tableau contains a *pre-model*. From pre-models we construct witness models. Pre-models are parts of tableaux. To keep track of the actual values of the clocks, *premises* and *conclusions* in our tableaux rules uses timing contexts. Timing contexts are well known equivalence classes of valuations of clocks called regions [2]. Applying a rule not only reduces the formula but also changes the values of the clocks, by performing operations such as *time elapse* and *clock reset*. There will be finitely many labels for nodes of tableaux (this is because there is finitely many regions), and nodes of tableaux will be the states of witness models. Our tableau rules can not handle the satisfiability checking problem for formulas like $\langle h_a > 1\rangle\langle a\rangle\mathrm{tt} \wedge \langle h_a > 1\rangle[a]\mathrm{ff}$ as we will need to introduce new constants in models. The latter formula is not a C-WT$_\mu$ formula; it requires that after one time unit delay, there exists a future time instant at which a can be fired and there exists another future time instant at which a can not be fired.

Related results. Event-Recording Automata (ERA) [3] is a restricted class of timed automata [2]. TML [9], L^t_μ [16], T_μ [19] and L$_\nu$ [11] are timed logics for for timed automata. T_μ is a timed extension of the μ-calculus. L$_\nu$ is a fragment of T_μ and it does not have the least fixpoint operator. L_ν allows to characterise timed simulation and timed bisimulation between timed automata [1]. The model-checking of L_ν is EXPTIME Complete and the satisfiability checking problem for L_ν have been left open [11]. The number of clocks in the models cannot be deduced from the number of clocks in a L_ν formula. But, the satisfiability problem for L_ν , with a restrictive bound assumption on clocks and constants in models, is decidable [11]. The satisfiability problem for ERL, without any bound assumption on constants in models, is EXPTIME Complete [17] and the construction of witness model is effective. ERL is more expressive than the event-recording part of timed linear-time logic EventClockTL [15]. WT_μ is expressive enough for characterising timed simulation and timed bisimulation relations between ERA [13]. One can check that characteristic formulas in [13] belong to C-WT_μ. They have also shown that ERL is a fragment of WT_μ. D'Sousa [6] gives a logical characterisation of ERA with a monadic second order logic, MSO_{er}, interpreted over linear models.

Organisation of the paper. in the next section, we present Event-Recording Automata. In Section 3, we define WT_μ, and C-WT_μ; we briefly compare WT_μ, C-WT_μ with ERL and L_ν . We also present a μ-calculus based algorithm for the model-checking problem of WT_μ. In Section 4, we consider the satisfiability problem of C-WT_μ. We show that a formula is satisfiable if and only if its tableau contains a pre-model. At the end of Section 4, we discuss about the satisfiability checking problem of WT_μ.

2 Event-Recording automata

The sets $\mathbb{N}$, $\mathbb{R}^+$, are respectively the sets of natural and non-negative real numbers. We consider a finite set $\mathcal{H}$ of variables, called *clocks*. A *clock valuation* over $\mathcal{H}$ is a mapping $v : \mathcal{H} \to \mathbb{R}^+$ that assigns to each clock a time value. The set of all clock valuations over $\mathcal{H}$ is denoted $\mathcal{V}_\mathcal{H}$. Let $t \in \mathbb{R}^+$, the valuation $v+t$ is defined by $(v+t)(h) = v(h) + t$, $\forall h \in \mathcal{H}$. For a clock $h \in \mathcal{H}$, we denote by $v[h := 0]$ the valuation such that $(v[h := 0])(h) = 0$ and for every $h \neq h'$ $(v[h := 0])(h') = v(h')$. Finally, v^0 is the valuation that maps every clock on 0.

Given a set of clocks $\mathcal{H}$, the set of clock constraints over $\mathcal{H}$ is defined by the grammar "$g ::= h \sim c \mid g \wedge g \mid \text{true}$" where $h \in \mathcal{H}$, $c \in \mathbb{N}$, $\sim \in \{\leq, <, >, \geq\}$ and true stands for true. We write $v \models g$ (or $v \in [\![g]\!]$) when the clock valuation v satisfies the clock constraint g. A *M-rectangular constraint* is a constraint of the form $\bigwedge_{h\in\mathcal{H}} e_h$ where e_h is an equation of the form $c \leq h \leq c$, $c < h < c+1$, $M \leq h \leq M$, or $h > M$ with $c < M$. The bound M_g of a constraint g is the maximal constant that occurs in g. It should be obvious that for every $M \geq M_g$, a constraint g can be decomposed into a finite set $Rect_M(g)$ of M-rectangular constraints such that $v \models g$ iff $\exists g' \in Rect_M(g)\, s.t\, v \models g'$.

Given a constant $M \in \mathbb{N}$, a set of clock $\mathcal{H}$, the set $\mathcal{V}_\mathcal{H}$ can be partitioned into a finite set, denoted by $Reg(M)$, of equivalence classes called *regions* [2]. For a given M, the number of regions is bounded by $|\mathcal{H}|!.2^{|\mathcal{H}|}.(2.M+2)^{|\mathcal{H}|}$. It is standard that any region r from $Reg(M)$ can be represented by an atomic clock constraint that additionally uses equations of the form $h - h' \sim c$. For a valuation v, $[v]_M$ (or $[v]$ when it is clear from the context) denotes the region that contains v. We also define $\mathrm{r}^0 = [v^0]$, $v\!\uparrow = \{v + t \mid t \in \mathbb{R}^+\}$; and given a region r, we define $\mathrm{r}\!\uparrow = \{[v + t] \mid t \in \mathbb{R}^+ \text{ and } v \in \mathrm{r}\}$.

In the context of event-recording automata, each clock refers to a specific event. Then, we associate clocks with letters of an alphabet. Given an alphabet Σ, we then denote by $\mathcal{H}_\Sigma$ the set of clocks $\{h_a \mid a \in \Sigma\}$. Sets of valuations, constraints, and M-rectangular constraints are denoted by $\mathcal{V}_\Sigma$, Gds_Σ, and $Agds_\Sigma(M)$.

Definition 1 *An* event-recording automaton (ERA) *is a tuple* $\mathcal{P} = \langle P, \Sigma, p^0, \Delta_\mathcal{P}\rangle$ *where, P is a finite set of locations, $p^0 \in P$ is the initial location, $\Delta_\mathcal{P} \subseteq P \times Gds_\Sigma \times \Sigma \times P$ is a transition relation. The ERA is* deterministic *if* $[\![g_1 \wedge g_2]\!] = \emptyset$ *whenever* $(p, g_1, a, p_1) \in \Delta_\mathcal{P}$ *and* $(p, g_2, a, p_2) \in \Delta_\mathcal{P}$.

The *bound* of an ERA $\mathcal{P}$, $M_\mathcal{P}$ is the maximal constant that occurs in its constraints. Given a constant M, we say an ERA is M-bounded iff its bound is smaller than M. The semantics of ERA are labelled transition systems.

A state of an event-recording automaton $\mathcal{P}$ as above defined, is a pair (p, v) where p is a location and v is a valuation over $\mathcal{H}_\Sigma$. The initial state of $\mathcal{P}$ is (p^0, v^0) where v^0 maps all clocks in $\mathcal{H}_\Sigma$ to 0. The semantics of an event-recording automaton $\mathcal{P}$ as above is the $(\Sigma \cup \mathcal{V}_\Sigma)$-labelled transition system (LTS) $[\![\mathcal{P}]\!] = \langle P \times \mathcal{V}_\Sigma, \Sigma \cup \mathcal{V}_\Sigma, (p^0, v^0), \rightarrow\rangle$ where $\rightarrow \subseteq (P \times \mathcal{V}_\Sigma) \times (\Sigma \cup \mathbb{R}^+) \times (P \times \mathcal{V}_\Sigma)$ is defined by: $(p, v) \xrightarrow{v+t} (p, v + t)$ for every $t \geq 0$; and $(p, v) \xrightarrow{a} (p', v[h_a := 0])$ if and only if there is $p \xrightarrow{g,a} p' \in \Delta_\mathcal{P}$ such that $v \models g$. Each clock of an event-recording automata records the amount of time elapsed since the last occurrence of its corresponding event. It is common to label continuous transition with delays in $\mathbb{R}^+$ instead of valuations as above. The two representations are equivalent; the benefit of our representation will appear later in Sect. 3.

Let us present two abstractions for the semantics. The M-action abstraction is obtained from the semantics by replacing valuations on delay-transitions with M-rectangular constraints they satisfy.

Definition 2 *The M-*action abstraction *of an ERA $\mathcal{P}$, as above defined, is the* $(\Sigma \cup Agds_\Sigma(M))$*-LTS* $\langle\!|\mathcal{P}|\!\rangle^M = \langle P \times \mathcal{V}_\Sigma, \Sigma \cup Agds_\Sigma(M), (s^0, v^0), \Delta_v\rangle$ *where,* $\Delta_v \subseteq (P \times \mathcal{V}_\Sigma) \times (\Sigma \cup Agds_\Sigma(M)) \times (P \times \mathcal{V}_\Sigma)$ *is defined by:* $(p, v) \xrightarrow{g'} (p, v + t)$ *for any* $t \in \mathbb{R}^+$ *s.t* $v + t \models g'$ *with* $g' \in Agds(M)$*; and* $(p, v) \xrightarrow{a} (p', v[h_a := 0])$ *if there is* $(p, g, a, p') \in \Delta_\mathcal{P}$ *with* $v \models g$.

One can check that for every $M \in \mathbb{N}$, $[\![\mathcal{P}]\!]$ and $\langle\!|\mathcal{P}|\!\rangle^M$ are isomorphic.

Definition 3 *The M-*region abstraction *of $\mathcal{P}$ is* $\langle\!|\mathcal{P}|\!\rangle_r^M = \langle P \times Reg(M), \Sigma \cup Agds(M), (p^0, r^0), \Delta_r\rangle$ *where,* $v^0 \in r^0$, $\Delta_r \subseteq (P \times Reg(M)) \times (\Sigma \cup Agds_\Sigma(M)) \times (P \times Reg(M))$ *is given by:* $(p, r) \xrightarrow{g'} (p, r')$ *with* $r' \subseteq r\!\uparrow$ *and* $r' \subseteq g'$*; and* $(p, r) \xrightarrow{a} (p', r[h_a := 0])$ *if there is* $(p, g, a, p') \in \Delta_\mathcal{P}$ *with* $r \subseteq g$.

Proposition 4 *Let $\mathcal{P}$ be an ERA. $\forall M \geq M_{\mathcal{P}}$: $\langle\!\langle \mathcal{P} \rangle\!\rangle^M$ is bisimilar*[1] *to $\langle\!\langle \mathcal{P} \rangle\!\rangle^M_r$.*

3 The Logic WT$_\mu$

We want a modal logic for semantics of ERA. Modalities of that logics should allow to quantify delay or discrete successors of semantics of ERA. The logic in [17] allows quantification on successions of a delay with an event. That kind of quantification is "weak" as for example they do not allows to characterise event-recording automata up to some behavioural relations [13]. We introduced the logic WT$_\mu$ [12].

3.1 Syntax and Semantics of WT$_\mu$

The syntax of WT$_\mu$[12] is the syntax of the standard μ-calculus [10] augmented with two dual modalities indexed with constraints: the existential modality $\langle g \rangle$ and the universal modality $[g]$.

Definition 5 *Formulas of WT$_\mu$ over an alphabet Σ, a finite set of variables $Var = \{X, Y, \ldots\}$ and Gds_Σ are defined by the following grammar:*

$$\varphi ::= tt \mid ff \mid X \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle a \rangle \varphi \mid \langle g \rangle \varphi \mid [a]\varphi \mid [g]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $g \in Gds_\Sigma$; the symbols tt and ff denote "true" and "false" formulas.

The following definitions are taken from [14] and they are used in the context of WT$_\mu$. A variable X is *bound* in a formula φ if there is a sub formula $\sigma X.\psi(X)$ of φ with $\sigma \in \{\mu, \nu\}$. A *well named* formula is such that an expression $\mu X.\varphi(X)$ (or $\nu X.\varphi(X)$) occurs at most once for each variable X. By renaming variables if necessary, every formula can be translated into an equivalent well named formula. In what follows, w.l.o.g we assume that formulas are well named. The *binding definition* of a bound variable X in a well named formula φ, $Bd_\varphi(X)$ is the unique sub formula of φ of the form $\sigma X.\psi(X)$; we call X a *μ-variable* when $\sigma = \mu$, otherwise we call X a *ν-variable*. We will omit subscript φ when it causes no ambiguity. The *dependency order* over the bound variables of a formula φ, is the least partial order such that X is *older than* Y (and we write $X \leq_\varphi Y$) if X occurs in $Bd_\varphi(Y)$ (and $Bd_\varphi(Y)$ is a sub formula of $Bd_\varphi(X)$). Given φ, Bd_φ, and ψ a sub formula of φ, the *expansion* $\mathrm{Exp}_{Bd_\varphi}(\psi)$ of ψ w.r.t Bd_φ is defined by $\mathrm{Exp}_{Bd_\varphi}(\psi) = \psi[Bd_\varphi(X_n)/X_n]\cdots[Bd_\varphi(X_1)/X_1]$, where $\{X_1, X_2, \cdots, X_n\}$ is an increasing chain of bound variables of φ with respect to $\leq_\varphi$.

WT$_\mu$ is interpreted over $(\Sigma \cup \mathcal{V}_\Sigma)$-LTS that may represent semantics of ERA.

Definition 6 *For a $(\Sigma \cup \mathcal{V}_\Sigma)$-LTS $\mathcal{S} = \langle S, \Sigma \cup \mathcal{V}_\Sigma, s^0, \Delta_\mathcal{S} \rangle$ and an assignment $Val : \mathrm{Var} \rightarrow 2^S$, the set of states in which a formula φ is true, $[\![\varphi]\!]^S_{Val}$ is defined*

[1]One considers the timed-abstract bisimilarity relation between $[\![\mathcal{P}]\!]$ and $\langle\!\langle \mathcal{P} \rangle\!\rangle^M_r$ [2].

inductively as follows:

$$\begin{array}{rcl}
[\![X]\!]^{\mathcal{S}}_{Val} & := & Val(X) \\
[\![\varphi_1 \vee \varphi_2]\!]^{\mathcal{S}}_{Val} & := & [\![\varphi_1]\!]^{\mathcal{S}}_{Val} \cup [\![\varphi_2]\!]^{\mathcal{S}}_{Val} \\
[\![\varphi_1 \wedge \varphi_2]\!]^{\mathcal{S}}_{Val} & := & [\![\varphi_1]\!]^{\mathcal{S}}_{Val} \cap [\![\varphi_2]\!]^{\mathcal{S}}_{Val} \\
[\![\langle a \rangle \varphi]\!]^{\mathcal{S}}_{Val} & := & \{s \mid \exists s \xrightarrow{a} s' \textit{ and } s' \in [\![\varphi]\!]^{\mathcal{S}}_{Val}\} \\
[\![\langle g \rangle \varphi]\!]^{\mathcal{S}}_{Val} & := & \{s \mid \exists s \xrightarrow{v} s' \textit{ s.t } v \in [\![g]\!] \textit{ and } s' \in [\![\varphi]\!]^{\mathcal{S}}_{Val}\} \\
[\![[a]\varphi]\!]^{\mathcal{S}}_{Val} & := & \{s \mid \forall s \xrightarrow{a} s'.\ s' \in [\![\varphi]\!]^{\mathcal{S}}_{Val}\} \\
[\![[g]\varphi]\!]^{\mathcal{S}}_{Val} & := & \{s \mid \forall s \xrightarrow{v} s' \textit{ s.t } v \in [\![g]\!].\ s' \in [\![\varphi]\!]^{\mathcal{S}}_{Val}\} \\
[\![\mu X.\varphi(X)]\!]^{\mathcal{S}}_{Val} & := & \bigcap\{T \subseteq S \mid [\![\varphi(X)]\!]^{\mathcal{S}}_{Val[X/T]} \subseteq T\} \\
[\![\nu X.\varphi(X)]\!]^{\mathcal{S}}_{Val} & := & \bigcup\{T \subseteq S \mid T \subseteq [\![\varphi(X)]\!]^{\mathcal{S}}_{Val[X/T]}\}
\end{array}$$

where for a set of states $T \subseteq S$, the assignment $Val[X/T]$ is such that $Y \in \mathrm{Var}$, $Val[X/T](Y) = T$ if $Y = X$ and $Val[X/T](Y) = Val(Y)$ otherwise. We will omit Val when φ is a sentence.

Definition 7 *Let $\mathcal{P} = \langle P, \Sigma, p^0, \Delta_{\mathcal{P}} \rangle$ be an ERA and $Val : \mathrm{Var} \to 2^{P \times \mathcal{V}_\Sigma}$ be an assignment. $\mathcal{P}$ is a* model *of φ with respect to Val iff $(p^0, v^0) \in [\![\varphi]\!]^{[\![\mathcal{P}]\!]}_{Val}$.*

In example, the formula $\varphi = \langle 0 < h_a < 1 \rangle((\langle b \rangle \mathrm{tt} \wedge [a]\mathrm{ff}) \vee \langle c \rangle \mathrm{tt})$ is such that events a, b and c are in the scope of the modality $\langle 0 < h_a < 1 \rangle$. The formula φ says that there is a time at which $0 < h_a < 1$ holds and at that time, the event c or b can be completed and the event a can not be completed.

M-rectangular formulas are formulas that only use M-rectangular constraints from $Agds(M)$. The *M-rectangular formula* associated to a formula φ is the formula $Rect_M(\varphi)$ defined inductively as follows:

- $Rect_M(\mathrm{ff}) = \mathrm{ff}$, $\quad Rect_M(\mathrm{tt}) = \mathrm{tt}$, $\quad Rect_M(X) = X$,
- $Rect_M(\varphi \wedge \psi) = Rect_M(\varphi) \wedge Rect_M(\psi)$,
- $Rect_M(\varphi \vee \psi) = Rect_M(\varphi) \vee Rect_M(\psi)$,
- $Rect_M(\langle a \rangle \varphi) = \langle a \rangle Rect_M(\varphi)$, $\quad Rect_M([a]\varphi) = [a]Rect_M(\varphi)$,
- $Rect_M(\langle g \rangle \varphi) = \bigvee_{g' \in Rect_M(g)} \langle g' \rangle Rect_M(\varphi)$,
- $Rect_M([g]\varphi) = \bigwedge_{g' \in Rect_M(g)} [g'] Rect_M(\varphi)$,
- $Rect_M(\sigma X.\varphi(X)) = \sigma X.Rect_M(\varphi(X))$ where σ is one of $\{\mu, \nu\}$.

Let $g, g_1, g_2, \ldots, g_n$ such that $[\![g]\!] = \bigcup_{i=1..n} [\![g_i]\!]$. One can show that $\langle g \rangle \varphi$ is equivalent to $\bigvee_{i=1..n} \langle g_i \rangle \varphi$, and $[g]\varphi$ is equivalent to $\bigwedge_{i=1..n} [g_i]\varphi$; and then one can show the following result.

Proposition 8 *Given a formula φ, a $(\Sigma \cup \mathcal{V}_\Sigma)$-LTS $\mathcal{S}$, for every $M \geq M_\varphi$, $s \in [\![\varphi]\!]^{\mathcal{S}}_{Val}$ iff $s \in [\![Rect_M(\varphi)]\!]^{\mathcal{S}}_{Val}$.*

Modal operators are monotonic. As usual the Knaster-Tarski theorem allows to approximate fixpoints by iterative computations as follows:

- $s \in [\![\mu X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ iff $s \in \bigcup_{\lambda} [\![\mu^{\lambda} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ and $s \in [\![\nu X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ iff $s \in \bigcap_{\lambda} [\![\mu^{\lambda} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ where λ is an ordinal and:
  - $s \notin [\![\mu^{0} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ and $s \in [\![\nu^{0} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$
  - $s \in [\![\sigma^{\lambda+1} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ iff $s \in [\![\varphi(X)]\!]^{\mathcal{S}}_{Val[X/\sigma^{\lambda} X.\varphi(X)]}$ (regeneration step).
  - When β is a limit ordinal, $s \in [\![\mu^{\beta} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ iff $s \in \bigcup_{\lambda<\beta} [\![\mu^{\lambda} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ and $s \in [\![\nu^{\beta} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$ iff $s \in \bigcap_{\lambda<\beta} [\![\mu^{\lambda} X.\varphi(X)]\!]^{\mathcal{S}}_{Val}$.

Expressiveness and relations with L_ν and ERL. The fragment of WT$_\mu$ without the least fixpoint operator is a fragment of L_ν for ERA. The inclusion is a consequence of that the modal operators $[g]\varphi$, $\langle g\rangle\varphi$, $[a]\varphi$ and $\langle a\rangle\varphi$ of WT$_\mu$ are respectively equivalent to $[\delta](g \rightarrow \varphi)$, $\langle\delta\rangle(g \wedge \varphi)$, $[a](h_a \; \underline{in} \; \varphi)$ and $\langle a\rangle(h_a \; \underline{in} \; \varphi)$ of L_ν. As L_ν is a fragment of T$_\mu$ without the least fixpoint operator, we get that WT$_\mu$ is a fragment of T$_\mu$.

ERL is a fragment of C-WT$_\mu$. One can check that the proof in [13] for the inclusion of ERLto WT$_\mu$works for C-WT$_\mu$. Moreover characteristic formulas in [13] also belong to C-WT$_\mu$ .The time-liveliness property requires an ERA to be able to fire an event e, in all states and in all future time instants (this is an important property in control theory[4, 5]). The property can be describe with the C-WT$_\mu$ formula $\varphi := \nu X.[\mathrm{tt}]\langle e\rangle X$.

3.2 Model-Checking

The model-checking problem for WT$_\mu$ is to check whether a given ERA is a model of a given formula. This section provides a new proof for the EXPTIME-Complete membership of the model-checking problem of WT$_\mu$. Another proof of this result appeared in [13]. The original idea is to reduce the model-checking problem of WT$_\mu$to the model-checking problem of the standard μ-calculus.

First, we present an inductive procedure for checking whether a formula φ is true in a state s of a $(\Sigma \cup \mathcal{V}_\Sigma)$-LTS:

- If φ is ff then return "no"
- If φ is tt then return "yes"
- If φ is $\varphi_1 \vee \varphi_2$ then check if φ_1 is true in s or check if φ_2 is true in s
- If φ is $\varphi_1 \wedge \varphi_2$ then check if φ_1 is true in s and check if φ_2 is true in s
- If φ is $\langle a\rangle\varphi_1$ then check whether φ_1 is true in some a successor of s or return "no" if s does not have an a-successor

- If φ is $[a]\varphi_1$ then check whether φ_1 is true in every a-successor of s or return "yes" if s does not have an a-successor
- If φ is $\langle g\rangle\varphi_1$ then check whether φ_1 is true in some v-successor of s such that $v \in [\![g]\!]$ or return "no" if s does not have an v-successor such that $v \in [\![g]\!]$
- If φ is $[g]\varphi_1$ then check whether φ_1 is true in every v-successor of s such that $v \in [\![g]\!]$ or return "yes" if s does not have an v-successor such that $v \in [\![g]\!]$
- If φ is $\mu X.\varphi_1(X)$ then check if $\mu^\lambda X.\varphi_1(X)$ is true at s for some ordinal λ
- If φ is $\nu X.\varphi_1(X)$ then check if $\nu^\lambda X.\varphi_1(X)$ is true at s for every ordinal λ

The last two rules follows the Knaster-Tarski fixpoint computation rules. In the rule for the least fixpoint, one guesses a number for computation steps for showing that $s \in [\![\mu X.\varphi_1(X)]\!]$. In the rule for the greatest fixpoint operator, there is no guess and one must ensure that $\nu X.\varphi_1(X)$ is true in s at every computation step. In the rules for the operators $\vee$, $\langle a\rangle$, $\langle g\rangle$ and $\mu X.$, one chooses (or guesses) sub formulas, successors or ordinals.

A formula φ is true in a state s if some choices can be made in the above procedure to return "yes". If the procedure return "yes", it also assigns a set of true formulas to every state of the LTS meaning that every state of the LTS is an abstraction of the set of formulas assigned to it. Every step of the procedure makes a relation between a formula to verify and a next formula. A path in the procedure shows a trace of that relation. It is worth noticing that, according to the Knaster-Tarski fixpoint computation method, a trace that starts with a least fixpoint formula always ends while a trace that starts with a greatest fixpoint formulas may not. Additionally, traces that do not contains fixpoint formulas are necessarily finite, as each step reduces the size of the formula. Finite traces may ends with tt, ff or a formula of one of the forms $\langle a\rangle\varphi_1$, $[a]\varphi_1$, $\langle g\rangle\varphi_1$, $[g]\varphi_1$.

For the model-checking of WT$_\mu$with respect to ERA, an effective procedure may work on finite structures. But semantics of ERA are infinite structures. We use the region representation of semantics and we reduce the model-checking problem for WT$_\mu$with respect to ERA to the model-checking problem of the standard μ-calculus. For that purpose, we define the *abstract semantics* $\langle\!|\varphi|\!\rangle^{\mathcal{S}}_{Val}$ for a WT$_\mu$ formula φ over a $(\Sigma \cup Gds_\Sigma)$-LTS $\mathcal{S}$ and a valuation $Val : \text{Var} \to 2^{\mathcal{S}}$. The abstract semantics for all the operators but the modal operators indexed with constraints are obvious:

$$\begin{aligned}
\langle\!|\langle g\rangle\varphi|\!\rangle^{\mathcal{P}}_{Val} &:= \{p \mid \exists p \xrightarrow{g} p' \text{ s.t } p' \in \langle\!|\varphi|\!\rangle^{\mathcal{P}}_{Val}\} \\
\langle\!|[g]\varphi|\!\rangle^{\mathcal{P}}_{Val} &:= \{p \mid \forall p \xrightarrow{g} p'. \, p' \in \langle\!|\varphi|\!\rangle^{\mathcal{P}}_{Val}\}
\end{aligned}$$

The abstract semantics of WT$_\mu$is very close to the semantics of the μ-calculus; equality tests are performed between labels of transitions and indexes in modal operators. The two non standard steps of an abstract model-checking procedure work as follows: if φ is $\langle g\rangle\varphi_1$ then check whether φ_1 is true in some g-successor of s or return "no" if s does not have a g-successor such that $v \in [\![g]\!]$. If φ is $[g]\varphi_1$ then check whether φ_1 is true in every g-successor of s or return "yes" if s does not have an v-successor such that $v \in [\![g]\!]$.

Proposition 9 *For any ERA $\mathcal{P}$, any sentence φ, for every $M \geq \max(M_\varphi, M_\mathcal{P})$, $(p, v) \in [\![\varphi]\!]^{[\![\mathcal{P}]\!]}$ iff $(p, [v]_M) \in [\![Rect_M(\varphi)]\!]^{\langle\!\langle \mathcal{P} \rangle\!\rangle_r^M}$.*

Our proof for Proposition 9 will use Proposition 4 and Proposition 8. It follows that model-checking algorithms for the μ-calculus [10] can be easily adapted for the model-checking of WT$_\mu$.

Theorem 10 *The model-checking problem for WT$_\mu$ is EXPTIME Complete.*

The lower bound comes from that ERL is a fragment of WT$_\mu$and the model-checking of ERL is EXPTIME Complete [17]. The EXPTIME membership (see [12] for more details) is a consequence of the EXPTIME membership of the model-checking problem of the standard μ-calculus [8] in relation with the problem of checking the existence of winning strategy in a two player parity game [20]. The use of regions does not change the complexity.

4 Satisfiability of C-WT$_\mu$

The satisfiability problem is to check whether a given formula has a model. Model for formulas are event-recording automata or more precisely $(\mathbb{R}^+ \cup \Sigma)$-LTS that may represent the semantics of event-recording automata. It is worth noticing that the satisfiability problem is somehow related to the model-checking problem. Here, the exercise is to guess ERA from formulas.

Consider the WT$_\mu$ formula $\varphi := \langle h_a > 1\rangle[a]\mathrm{ff} \wedge \langle h_a > 1\rangle\langle a\rangle\mathrm{tt}$. The formula φ requires that there are one time instant satisfying $h_a > 1$ at which event a must occur and one time instant satisfying $h_a > 1$ at which event a never occurs. The union of the semantics of constraints on the outgoing transitions from the initial state of a model of φ should be strictly contained in the semantics of $h_a > 1$. For instance the constraint on the transition of a model with a single must have one of the forms $n < h_a$ or $m_1 < h_a < m_2$ where $n > 1$ and m_2 are new constants. Procedure for the satisfiability checking problem of WT$_\mu$ must be able to introduce constants that do not appears in formulas. Such a procedure (if it exists) may become very sophisticated for formulas with fixpoints operators.

In this section we consider the satisfiability checking problem of C-WT$_\mu$. C-WT$_\mu$ is a fragment of WT$_\mu$. Formulas of C-WT$_\mu$ are such that if we look at a formula as a tree, then the modalities indexed with constraints and with events must alternate on each path. C-WT$_\mu$ also requires that in the tree representation of a modality $[a]$ must not follow a modality $\langle g\rangle$.

Definition 11 *Formulas φ of C-WT$_\mu$ (WT$_\mu$ for control) over an alphabet Σ, a finite set of variables Var and Gds_Σ are defined by the following grammar:*

$$\varphi ::= tt \mid ff \mid X \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle g\rangle\psi_e \mid [g]\varphi_e \mid \nu X.\varphi \mid \mu X.\varphi$$

where $X \in Var$, $g \in Gds_\Sigma$, φ_e is a boolean combination of formula of the forms $\langle a\rangle\varphi$ or $[a]\varphi$ and ψ_e is a boolean combination of formulas of the form $\langle a\rangle\varphi$.

The formula φ just above is not from C-WT$_\mu$because the formula $[a]$ff appears in the scope of $\langle 0 < h_a < 1\rangle$. We stress that C-WT$_\mu$ is an interesting fragment of WT$_\mu$. It allows to characterise ERA up to behavioural relation such as timed bisimulation and timed simulation. Indeed one can check that characteristic formulas in [13] conform to the syntax C-WT$_\mu$. C-WT$_\mu$allows to describe basic requirement for controllers of event-recording automata.

Our method for the satisfiability checking problem use tableaux-based methods like in [18, 14, 17]. W.l.o.g we consider rectangular formulas.

4.1 Tableau, Trace and Pre-model

Tableaux are built using the system of tableau rules. A *rule* is of the form $\frac{T_1\ T_2\ \ldots T_n}{T}$ where T and T_i for every $i = 1..n$ are sequent made of a set of formulas (satisfiability objectives) and a region (timing context). The tuples over the lines of a rules (T_i for every $i = 1..n$) are the *premises* and the tuple T below the line of a rule is the *conclusion*. A rule as above is interpreted as follows: verifying whether the formulas in the conclusion are satisfiable from the timing context is reduced to checking if the formulas in all (some) hypothesis are satisfiable in their corresponding timing context.

Definition 12 *Let a φ be a C-WT$_\mu$formula and let Bd_φ be its binding function. We define the system of tableau rules $\mathcal{SR}_c^\varphi$ parametrised by φ, its binding function and the set of regions $Reg(M)$:*

$$\frac{\{\mathit{ff}\};\emptyset}{\{\varphi,\Gamma\};\emptyset}(\mathit{ff}_r) \qquad \frac{\{\varphi_1,\Gamma\};r \qquad \{\varphi_2,\Gamma\};r}{\{\varphi_1\vee\varphi_2,\Gamma\};r}(\vee) \qquad \frac{\{\varphi_1,\varphi_2,\Gamma\};r}{\{\varphi_1\wedge\varphi_2,\Gamma\};r}(\wedge)$$

$$\frac{\{\mathit{ff}\};\emptyset}{\{\langle g\rangle\varphi,\Gamma\};r\ \ s.t\ [\![g]\!]\cap r\!\uparrow=\emptyset}(\mathit{fte}) \qquad \frac{\{\Gamma\};r}{\{[g]\varphi,\Gamma\};r\ s.t\ [\![g]\!]\cap r\!\uparrow=\emptyset}(\mathit{wtt})$$

$$\frac{\{\varphi(X),\Gamma\};r}{\{\nu X.\varphi(X),\Gamma\};r}(\nu) \qquad \frac{\{\varphi(X),\Gamma\};r}{\{X,\Gamma\};r}(reg) \quad Bd_\varphi(X)=\sigma X.\varphi(X)$$

$$\frac{\{\varphi(X),\Gamma\};r}{\{\mu X.\varphi(X),\Gamma\};r}(\mu) \qquad \frac{\{\varphi \mid (g)\varphi\in\Gamma_{r_i}\};r_i \quad \forall r_i\in r\!\uparrow}{\Gamma;r}(delay)$$

$$\frac{\varphi\cup\{\psi \mid [a]\psi\in\Gamma\};r[h_a:=0]\ \mathit{for\ each}\ \langle a\rangle\varphi\in\Gamma}{\Gamma;r}(mod)$$

where Γ_{r_i} is the set of formulas in Γ of the of form $\langle g\rangle\varphi$ or $[g]\varphi$ s.t $r_i \subseteq [\![g]\!]$.

The system of rules above has ten rules; almost as many rules as in the model checking procedure in the previous subsection. Formulas can not be satisfied from an empty timing context and then the rule ff_r replaces formulas in the conclusion by the formula ff. The rule fte is applied when a formula in the conclusion ($\langle g\rangle\varphi$) requires that the future of the timing context in conclusion satisfies a constraint whereas it is not possible. The rule *wtt* discards formulas that start with universal modality indexed with a constraint which can not be satisfied by the future of the timing context in the conclusion. When applying the rule (*delay*) we require that every formula in the conclusion should be one of the forms ff, tt, $\langle g\rangle\psi$ or $[g]\psi$ with $g \cap \mathrm{r}\uparrow \neq \emptyset$; in this case the satisfiability checking of the conclusion is reduced to the satisfiability checking of formulas behind the modal operators in the appropriate timing context. When applying the rule (*mod*) we require the form of every formula in conclusions to be one of ff, tt, $\langle a\rangle\psi$ or $[a]\psi$; as we are trying to build a model, a reduction with the rule (*mod*) is done as soon as conclusions contain a formula of the form $\langle a\rangle\psi$.

Definition 13 *A tableau for a formula φ from a region r^0 is a tree $\mathcal{T}^\varphi = \langle N, E, \mathcal{L}\rangle$ where N is the set of nodes, E is the set of edges, and $\mathcal{L}$ is a labeling function such that:*

1. *The root n^0 of $\mathcal{T}^\varphi$ is labeled by $\{\varphi\}; r^0$*
2. *The sons of any node n are created and labeled according to the rules of systems $\mathcal{SR}^\varphi$. It is required the rules* (*mod*) *and* (*delay*) *are applied only when no other rule is applicable.*

Remark that a conclusion in a tableau never contains at the same time a formula starting with a modality indexed with a guard and a formula starting with a modality indexed with an event. The formula part of timed sequents on which no rule is applicable never contain formulas of the forms $\langle a\rangle\psi$, $\langle g\rangle\psi$ and $[g]\psi$. Given a node n of a tableau with $\mathcal{L}(n) = \Gamma; \mathrm{r}$, $\mathcal{L}_1(n) = \Gamma$ and $\mathcal{L}_2(n) = \mathrm{r}$ denotes respectively the formula and the timing part of $\mathcal{L}(n)$. A $\mathcal{L}(n)$ is satisfiable iff every formula φ in $\mathcal{L}_1(n)$ is satisfiable from the timing context $\mathcal{L}_2(n)$. The construction of a tableau for a formula can be seen as a procedure for checking, for every node n whether every formula $\varphi \in \mathcal{L}_1(n)$ is satisfiable from $\mathcal{L}_2(n)$. This procedure is inductive; therefore there are links between formulas in the conclusion and formulas in premises. How we get a path in the (abstract) model-checking procedure is very similar to how we get a path in the tableau. But paths of the (abstract) model-checking procedure contains links between formulas while paths in tableaux do not. This is because formula part of sequents are sets of formulas.

Definition 14 *Given a path π of $\mathcal{T}^\varphi = \langle N, E, \mathcal{L}\rangle$, a* trace *on π will be a function Tr that assigns a tuple made of a formula and a region to each node in some initial segment of π, according to the rules applied for the construction of π. Tr_1 and Tr_2 denotes the* formula part *and the* timing part *of $Tr(n)$. Tr satisfies the following conditions:*

1. *If $Tr(n)$ is defined then $Tr_1(n) \in \mathcal{L}_1(n)$ and $Tr_1(n) = \mathcal{L}_1(n)$.*
2. *If the rule applied at the node m is not directed (ordered) by $Tr(m)$ then the son $n \in \pi$ of m is such that $Tr(m) = Tr(n)$.*
3. *If the rule applied at the node m differs from (mod), then the tuple $Tr(n)$ of the son $n \in \pi$ of m is one of the results of the application.*
4. *If the rule $(delay)$ is applied at the node m and the son $n \in \pi$ of m is labeled by $\{\varphi \mid (g)\varphi \in \Gamma_{r_i}\}; r_i$ then:*
   - *$Tr(n)$ is equal to $\varphi; r_i$ if $Tr(m) = (g)\varphi; r$*
   - *Otherwise $Tr(n)$ is undefined.*
5. *If the rule (mod) is applied at m and the son $n \in \pi$ of m is labeled by $\varphi \cup \{\psi \mid [a]\varphi \in \Gamma\}; r[h_a := 0]$ for some $\langle a\rangle\varphi \in \Gamma$ then:*
   - *$Tr(n) = \varphi; r[h_a := 0]$ if $Tr(m) = \langle a\rangle\varphi; r$*
   - *$Tr(n) = \psi; r[h_a := 0]$ if $Tr(m) = [a]\psi; r$*
   - *Otherwise $Tr(n)$ is undefined and $Tr(m)$ is the last element of the trace.*

The notion of traces in paths of tableaux links formulas in premises with formulas in conclusion. By this way, we are able to handle computational steps for fixpoint formulas. We recall that in the rules above computational steps are given by the rules (reg) and (σ).

A variable X is *regenerated* on a trace Tr of some path if and only if for some m and its son n on the path $\mathrm{Tr}_1(m) = X$ and $\mathrm{Tr}_1(n) = \psi(X)$ with $Bd_\varphi(X) = \sigma X.\psi(X)$. A *$\mu$-trace* is a infinite trace on which the oldest variable regenerated infinitely often is a *μ-variable*; or a maximal finite trace, ending with a tuple the formula part of which contains ff. A *pre-model* Prem is a part of a tableau $\mathcal{T}^\varphi$ satisfying the four conditions: the root of $\mathcal{T}^\varphi$ belongs to Prem; if a disjunctive node belongs to Prem, then only one of its son belongs to Prem; for all other kinds of nodes, if a node belongs to Prem then all its successors too; there is no path with a μ-trace in Prem.

Formulas part of sequents are set of formulas and we need to count computational steps for every fixpoint formulas in sequents. A *signature* sig $= (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a sequence of ordinals value of which depends on a state. We distinguish μ-signature from ν-signature that we simply call signature when it is clear from the context.

Let $\mathcal{S} = \langle S, \Sigma \cup \mathcal{V}_\Sigma, s^0, \Delta_\mathcal{S}\rangle$ be a $(\Sigma \cup \mathcal{V}_\Sigma)$-LTS. Let ψ be a sentence. If $s \in [\![\mathrm{Exp}_{Bd_\varphi}(\psi)]\!]^\mathcal{S}$ then, ψ has the *μ-signature* ${}^\mu\mathrm{sig}(\psi, s) = (\alpha_1, \ldots \alpha_{d^\mu})$ in s if ${}^\mu\mathrm{sig}(s, \psi)$ is the least (in lexicographical order) sequence of ordinals such that $s \in [\![\mathrm{Exp}_{Bd'_\varphi}(\psi)]\!]^\mathcal{S}$ where Bd'_φ is obtained from the binding function Bd_φ by changing definitions of X_i (for $i = 1, \ldots, d^\mu$) from $Bd_\varphi(X_i) = \mu X_i.\varphi_i(X_i)$ to $Bd'_\varphi(X_i) = \mu^{\alpha_i} X_i.\varphi_i(X_i)$.
If $s \notin [\![\mathrm{Exp}_{Bd_\varphi}(\psi)]\!]^\mathcal{S}$ then, ψ has the *ν-signature* ${}^\nu\mathrm{sig}(\psi, s) = (\alpha_1, \ldots \alpha_{d^\nu})$ in

s if ${}^{\nu}\mathrm{sig}(s,\psi)$ is the least (in lexicographical order) sequence of ordinals such that $s \notin [\![\mathrm{Exp}_{Bd'_\varphi}(\psi)]\!]^{\mathcal{S}}$ where Bd'_φ is obtained from the binding function Bd_φ by changing definitions of Y_i (for $i = 1,\dots,d^\nu$) from $Bd_\varphi(Y_i) = \nu Y_i.\varphi_i(Y_i)$ to $Bd'_\varphi(Y_i) = \nu^{\alpha_i} Y_i.\varphi_i(Y_i)$.

The proof of Lemma 15 is standard from the proof a similar Lemma in [14].

Lemma 15 *Let ${}^{\mu}sig(\varphi, s)$ the signature of φ at s, it is true that:*

- ${}^{\mu}sig(\varphi_1 \wedge \varphi_2, s) = \max\{{}^{\mu}sig(\varphi_1, s), {}^{\mu} sig(\varphi_2, s)\}$
- ${}^{\mu}sig(\varphi_1 \vee \varphi_2, s) ={}^{\mu} sig(\varphi_1, s)$ *or* ${}^{\mu}sig(\varphi_1 \vee \varphi_2, s) ={}^{\mu} sig(\varphi_2, s)$
- ${}^{\mu}sig(\langle a\rangle\varphi, s) ={}^{\mu} sig(\varphi, s')$ *for some s' such that* $s \xrightarrow{a} s'$
- ${}^{\mu}sig([a]\varphi, s) = \max\{{}^{\mu}sig(\varphi, s')$ *for all s' such that* $s \xrightarrow{a} s'\}$
- ${}^{\mu}sig(\langle g\rangle\varphi, s) ={}^{\mu} sig(\varphi, s')$ *some s' such that* $s \xrightarrow{g} s'$
- ${}^{\mu}sig([g]\varphi, s) = \max\{{}^{\mu}sig(\varphi, s')$ *for all s' such that* $s \xrightarrow{g} s'\}$
- *If X_i is the $i-th$ variable of Bd_φ and $Bd_\varphi(X_i) = \mu X_i \varphi(X_i)$, then the prefix of length $i-1$ of ${}^{\mu}sig(\mu X_i.\varphi(X_i), s)$ and ${}^{\mu}sig(\varphi(X), s)$ are equal*
- ${}^{\mu}sig(\nu X.\varphi(X), s) ={}^{\mu} sig(\varphi(X), s)$ *where* $Bd_\varphi(X) = \nu X.\varphi(X)$
- *If $Bd_\varphi(Y) = \mu Y.\varphi(Y)$, then ${}^{\mu}sig(Y, s) >{}^{\mu} sig(\varphi(Y), s)$*
- *If $Bd_\varphi(Y) = \nu Y.\varphi(Y)$, then ${}^{\mu}sig(Y, s) ={}^{\mu} sig(\varphi(Y), s)$*

Proof
Considering the last cases, we suppose that $\mathcal{S}, s \models \mathrm{Exp}_{Bd_\varphi}(X_i)$ with $Bd_\varphi(X_i) = \mu X_i.\psi_i(X_i)$. X_j occurs in $\psi_i(X_i)$ implies that $X_i \leq_\varphi X_j$ and X_j is free $\psi_i(X_i)$. Let ${}^{\mu}\mathrm{sig}(X_i, s) = (\alpha_1, \alpha_2, \dots, \alpha_n)$ and Bd' obtained from Bd_φ by changing definitions of X_i (for $i = 1,\dots,d^\mu$) from $Bd_\varphi(X_i) = \mu X_i.\psi_i(X_i)$ to $Bd'_\varphi(X_i) = \mu^{\alpha_i} X_i.\psi_i(X_i)$.

It follows from the definition of the signature that $\mathcal{S}, s \models \mu^{\alpha_i} X_i.\psi(X_i)$. Since α_i should be a finite ordinal, it follows that $\mathcal{S}, s \models \psi(\mu^{\alpha_i - 1} X.\psi(X_i))$, which means that the signature of $\psi(\mu^{\alpha_i-1}X_i.\psi(X_i))$ at s is $(\alpha_1, \dots, \alpha_{i-1}, \alpha_i - 1, \alpha'_{i+1}, \dots, \alpha'_{d^\mu})$ and is lower than $\mathrm{sig}(W_i, s)$. The difference occurs at the position i.

Observe that Lemma 15 translates to ν-signature after interchanging μ with ν, $\langle\rangle$ with $[]$, $\vee$ with $\wedge$.

4.2 Satisfiability Results

Now we will sketch a proof for the following theorem.

Theorem 16 *There is an EXPTIME Complete decision procedure in the size of the formula that checks if a formula φ is satisfiable. The construction of a witness model is effective.*

We need some definition for the construction of witness model. From the definition of the tableau system of rules, applying a rule different from (mod), $(delay)$ and $(\vee)$ to a node of a tableau generates a unique successor. In a pre-model we choose only one son of a disjunctive node and all the sons of a modal or a delay node. It follows that in a symbolic pre-model, the nodes with more that one successor are modal or delay nodes. Given a node n of Prem we denote $des^{\alpha}(n)$ the closest descendant of n or n itself in Prem that is either a delay node, a modal node, or a leaf. Observe that, if n is the root of Prem or n is a successor of a modal node of Prem, then $des^{\alpha}(n)$ is a delay node or a leaf; if n is a successor of a delay node of Prem, then $des^{\alpha}(n)$ is a modal node.
Given a pre-model Prem for a formula φ, the *model* based on Prem is the ERA $\mathcal{P} = \langle P, \Sigma, p^0, \Delta_{\mathcal{P}} \rangle$ such that: $p^0 = des^{\alpha}(n^0)$ where n^0 is the root of Prem; P consists of all the leaves and delay nodes of Prem; $(p, g, a, des^{\alpha}(n')) \in \Delta_{\mathcal{P}}$ if there is in Prem a successor n of p obtained by reducing a region $\mathrm{r}_i \subseteq g$ with $g \in Agds(M_{\varphi})$ and a successor n' of $des^{\alpha}(n)$ obtained by reducing an action a. In Consequence, the maximal constant in the model constructed from a pre-model and the maximal constant in the formula are equal; constraints in the model are rectangular.

A detailed proof for Proposition 17 can be found in Appendix.

Proposition 17 *A formula φ is satisfiable iff $\mathcal{T}^{\varphi}$ contains a pre-model.*

The Proof of Proposition 17 is decomposed into the proof of Lemma 18 for the "if" part and the proof of Lemma 19 for the "only if" part.

Lemma 18 *Given a formula φ, φ is satisfiable if there is a pre-model in $\mathcal{T}^{\varphi}$*

Proof
If φ is satisfiable then there exists an ERA model of φ, and by definition $(p^0, v^0) \in [\![\varphi]\!]^{[\![\mathcal{P}]\!]}$. Without the loss of generality we can assume that φ is rectangular. By Proposition 9, for every $M \geq \max(M_{\mathcal{P}}, M_{\varphi})$, we have that $(p^0, [v^0]_M) \in \langle\!\langle \varphi \rangle\!\rangle^{\mathcal{P}^M_{reg}}$.

Consider $\mathcal{T}^{\varphi}$ the tableau for φ; then we choose the nodes of $\mathcal{T}^{\varphi}$ that we include in the pre-model Prem accordingly to a *marking relation* $M : N \rightarrow 2^{P \times Reg(M)} S$. It will be defined in such a way that (1): $(p, \mathrm{r}) \in M(n)$ if and only if $(p, \mathrm{r}) \in \langle\!\langle \varphi \rangle\!\rangle^{\mathcal{P}^M_{reg}}$ for every $\varphi \in \mathcal{L}_1(n)$. First, we put $(p^0, [v^0]_M)$ in $M(n^0)$ with n^0 being the root of $\mathcal{T}_{\varphi}$. This is consistent as $\mathcal{P}$ is a model of φ and then $(p^0, [v^0]_M) \in \langle\!\langle \varphi \rangle\!\rangle^{\langle\!\langle \mathcal{P} \rangle\!\rangle^M_{reg}}$. Let us denote by s_n a state (p, r) of $\langle\!\langle \mathcal{P} \rangle\!\rangle^M_{reg}$ assigned to a node n of the tableau.

Then, if we assume that the node n has been included in the pre-model Prem with $s_n \in M(n)$, we choose the next node to include in the tableau using the following rules:

- The only son n' of some node n, marked with s_n, on which an unary rule $(\mathrm{ff_r}, fte, wtt, \wedge, reg, \mu,$ or $\nu)$ was applied is included in Prem and we set $s_n \in M(n')$.

- If n is a disjunctive node, then s_n is put into the marking of the son for which it has the least μ-signature. By Lemma 15, such a son exists.
- If n is a delay node, then we add all the sons of n in Prem. Each son n' of n is the result of the reduction of a set of formulas of the form $\langle g\rangle\psi$ or $[g]\psi$ with respect to a region r_i. Then, we set $s_{n'} \in M(n')$ where $s_{n'}$ is the unique state such that $s_n \xrightarrow{g} s_{n'}$.
- If n is a modal node, then we add all the sons of n in Prem. Each son n' of n is the result of the reduction of a formula of the form $\langle a\rangle\psi$. Then, we set $s_{n'} \in M(n')$ where $s_{n'}$ is a state such that $s_n \xrightarrow{a} s_{n'}$ and ${}^{\mu}\mathrm{sig}(\langle a\rangle\varphi, s_n) \geq^{\mu} \mathrm{sig}(\varphi, s_{n'})$. By Lemma 15, such a son exists.

By Property (1) above, it is obvious that every leaf of Prem does not contain ff. It remains to show that the tree we have constructed does not have an infinite with a μ-trace.

Now, assume that there is an infinite path π that has a μ-trace on it. Then, there is an oldest μ-variable X_i infinitely often regenerated along the trace. According to Lemma 15, from the point when no variable older that X_i is regenerated μ-signatures of formulas on that trace never increase on position $1, \ldots, i-1$. Then maximal signature of formulas on the trace considered up to position i never increases and decreases every time X_i is regenerated. In the other words, μ-signature decreases infinitely often in position i. This is a contradiction because sequences of ordinals of bounded length are well-ordered.

Lemma 19 *Given a formula φ, if there is a pre-model in $\mathcal{T}^{\varphi}$, then φ is satisfiable.*

Proof
The proof is dual to the proof of Lemma 18. Assume that φ has a symbolic pre-model Prem and φ is not satisfiable. Let $M = M_{\varphi}$. Consider $\mathcal{P}$, the model associated to Prem. From the remark above, $M_{\mathcal{P}} = M_{\varphi}$. If $(p^0, v) \notin [\![\varphi]\!]^{[\![\mathcal{P}]\!]}$, then by Proposition 9 we get that, $(p^0, [v]_M) \notin \langle\!|\varphi|\!\rangle^{\langle\!|\mathcal{P}|\!\rangle^{M}_{reg}}$. Now, we show that Prem contains a path π with a μ-trace $\mathrm{Tr} = \{\varphi_m; \mathrm{r}_m\}_{m\in\pi}$. The path π and the trace Tr are built in the following way:

- π starts at m^0 and $\varphi_{m_0} = \varphi$.
- Assume that, we built Tr up to the tuple $\varphi_m; \mathrm{r}_m$ with, $\varphi_m \in \mathcal{L}_1(m)$ and $\mathrm{r}_m \in \mathcal{L}_2(m)$, such that $(des^{\alpha}(m), \mathrm{r}_m) \notin \langle\!|\mathrm{Exp}_{bd_{\varphi}}(\varphi_m)|\!\rangle^{\langle\!|\mathcal{P}|\!\rangle^{M}_{reg}}$. The formula of the next tuple (the timing part is obvious) is selected as follows:
    1. If m is not a delay nor a modal node, then the only son m' of m is such that
        - $\mathcal{L}_2(m) = \mathcal{L}_2(m')$ and there are equal to r_m.
        - $\varphi_{m'} = \varphi_m$ if φ_m was not reduced by the rule.

  - $\varphi_{m'} = \varphi_1$ with $\varphi_m = \varphi_1 \wedge \varphi_2$, if ${}^{\nu}\mathrm{sig}(\varphi_m, (des^{\alpha}(m), \mathrm{r})) \geq^{\nu} \mathrm{sig}(\varphi_1, (des^{\alpha}(m), \mathrm{r}))$; otherwise $\varphi_{m'} = \varphi_2$.
  - $\varphi_{m'}$ is the formula that occurs in $\mathcal{L}_1(m')$ if $\varphi_m = \varphi_1 \vee \varphi_2$. We remark that, the choice in this case is directed by Prem.
  - In the other sub cases *i.e* (ff, fte, wtt, μ, ν or reg), we just take the resulting formula as the one for the next tuple of the trace.

2. If m is a delay node and φ_m is of the form $\langle g \rangle \psi$ or $[g]\psi$ and there is a son m' of m the formula part of which contains ψ, then we take $\varphi_{m'} = \psi$. The region part of m' is a region $\mathrm{r}' \in \mathrm{r}_m \uparrow$. Of course, $des^{\alpha}(m')$ is a modal node.
3. If m is a modal node, it is necessarily the closest descendant of a successor n' (with respect to some region r_m) of some delay node n; then,
  - if $\varphi_m = \langle a \rangle \psi$, there is a son m' of m the formula part of which was obtained by reducing φ_m, and the timing part of which is $\mathrm{r}_m[h_a := 0]$. We take $\varphi_{m'} = \psi$.
  - if $\varphi_m = [a]\psi$, then because $(des^{\alpha}(m), \mathrm{r}_m) \notin \langle\!\langle \varphi_m \rangle\!\rangle^{\langle\!\langle \mathcal{P} \rangle\!\rangle^{M}_{reg}}$, there exists a state p', g such that $des^{\alpha}_{n} \xrightarrow{g,a} p'$ is in $\mathcal{P}$ and ${}^{\nu}\mathrm{sig}([a]\psi, (des^{\alpha}(n), \mathrm{r}_m)) =^{\nu} \mathrm{sig}(\psi, (p', \mathrm{r}_m[h_a := 0])$ with $\mathrm{r}_m \in g$ and $g \in Agds(M_{\varphi})$. We take $\varphi_{m'} = \psi$ and $\mathrm{r}_{m'} = \mathrm{r}_m[h_a := 0]$

We remark that Tr is a valid trace of Prem and we distinguish two cases:

1. The trace is finite;
   - If the trace ends with the formula ff, then we get a contradiction with that $\mathcal{P}$ derived from Prem; Indeed a trace of Prem never ends with ff.
   - If the trace ends at the node m with the formula $\varphi_m = \mathrm{tt}$, then m is a leaf or delay node and obviously, $(des^{\alpha}(m), \mathrm{r}_m) \in \langle\!\langle \mathrm{tt} \rangle\!\rangle^{\langle\!\langle \mathcal{P} \rangle\!\rangle^{M}_{reg}}$, leading to a contradiction with the hypothesis.
   - If the trace ends with a formula of form $[g]\varphi$, then the region at node m can never reached g meaning that $[g]\varphi$ is satisfied at m. We also get a contradiction with our hypothesis.
   - Assume that the trace ends at the node m with a formula of the form $[a]\varphi$. There is a unique ancestor n of node m which is a delay node such that no delay node occurs between n and m. The selected formula at the node n that occurs in the trace is of the form $(g)\psi$ and ψ is a boolean combination of formulas containing $[a]\varphi$.
     Let p be the state in $\mathcal{P}$ that corresponds to the node n. Such a state exits because n is a delay node. Let r be the region at the node m and r$'$ be region at the node n. Because m is a son of n, we have that $\mathrm{r} \in \mathrm{r}' \uparrow$. Additionally, by hypothesis, $(p, \mathrm{r}') \notin \langle\!\langle (g)\psi \rangle\!\rangle^{\langle\!\langle \mathcal{P} \rangle\!\rangle^{M}_{reg}}$. Because the trace is maximal, there is no transition from p labelled with (g, a)

for the unique constraint $g \in Agds(M)$ such that $\mathrm{r} \subseteq g$. It follows that in $\langle\!\langle \mathcal{P} \rangle\!\rangle^M_{reg}$ there a unique outgoing transition $(p, \mathrm{r}') \xrightarrow{g} (p, \mathrm{r})$ and there is no outgoing transition from (p, r) labelled with a, involving that $(p, \mathrm{r}) \in \langle\!\langle [a]\varphi \rangle\!\rangle^{\langle\!\langle \mathcal{P} \rangle\!\rangle^M_{reg}}$. This leads to a contradiction with that in the trace ${}^{\nu}\mathrm{sig}((g)\psi, (p, \mathrm{r}')) ={}^{\nu} \mathrm{sig}([a]\varphi, (p, \mathrm{r}))$. Indeed, remember that the trace has been built by choosing at every node, the formula and the configuration with the least ν-signature.

2. If the trace is infinite, then because the ν-signature decreases along the trace and the formula is of finite length, there is necessarily a μ-variable X that is infinitely often regenerated and no older variable than X is infinitely often regenerated. This is a contradiction with that Prem does not contain such a trace.

Let us discuss the complexity of the satisfiability problem for C-WT$_\mu$. The satisfiability problem for C-WT$_\mu$ is EXPTIME Complete in the size of the formula, in the size of the alphabet, and in the size of the encoding of the maximal constant appearing in constraints of the formula. The EXPTIME hardness follows from ERL is included in C-WT$_\mu$ and the satisfiability problem of ERL is EXPTIME hard [17]. Arguments for the EXPTIME membership are the same as arguments for the EXPTIME membership for the satisfiability procedure for ERL [17] and the μ-calculus [18, 14]. Indeed, One can construct is a finite Rabin tree-automaton over a single alphabet, with $n = 0(2^{|\varphi|} \times |Reg(M_\varphi)|)$ states and $m = |\varphi|$ pairs[2], which recognises pre-models for a formula φ. We recall that $|Reg(M_\varphi)| \in O(2^{\mathcal{H}_\Sigma} \times 2^{C_{M_\varphi}})$ where C_{M_φ} is the length of the binary encoding of M_φ. It have been shown [7] that for a Rabin tree-automaton over a single alphabet that have an accepting run, there is a graph with states of the automaton as nodes which unwinds to an accepting run of the automaton. The test of the emptiness of the graph obtained from the pre-model and the Rabin-tree automaton is in $O((mn)^{3m})$. In consequence we get the EXPTIME membership.

4.3 On the Satisfiability of WT$_\mu$

The procedure above can not work for the satisfiability problems of WT$_\mu$. This is because models may need constants that are strictly greater than the maximal constant occurring in formulas. We can also consider a situation when we impose a maximal constant with which the clocks can be compared in the models. Such a constant can be greater than the maximal constant occurring in the formula. Under such an assumption, the satisfiability problems (non deterministic and deterministic model) for WT$_\mu$ are decidable and the construction of witness model is also effective.

[2]The construction of pairs follows the relation order between fixpoint variables in formulas.

5 Conclusion and Perspectives

We presented WT_μ as a timed μ-calculus for Event-Recording Automata. We presented an EXPTIME Complete decidable fragment of WT_μ called C-WT_μ. The decision procedure for the satisfiability problem of C-WT_μ does not need limit assumption on constants the models and it constructs witness models. C-WT_μ is more expressive than Event-Recording Logic and it allows to characterise event-recording automata up to timed simulation and timed bisimulation relations whereas ERL does not. The complexity for the satisfiability checking problems for C-WT_μ and ERL are the same. Relation between modalities of C-WT_μ and L_ν, and results in [5] let us believe the existence of an interesting decidable fragment of L_ν(with timed automata) without limit assumption on constants of models, but with a bound assumption on the clocks in the models. Work in progress apply the decidability result for the satisfiability checking problem of C-WT_μ to the synthesis of event-recording automata based controller. Future works include the satisfiability checking problem of full WT_μ.

References

[1] Luca Aceto, Anna Ingólfsdóttir, Mikkel Lykke Pedersen, and Jan Poulsen. Characteristic formulae for timed automata. *Theo. Informatics and Appl.*, 34(6):565–584, 2000.

[2] Rajeev Alur and David L. Dill. A theory of timed automata. *Theo. Comput. Sci.*, 126(2):183–235, 1994.

[3] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theo. Comput. Sci.*, 211(1-2):253–273, 1999.

[4] André Arnold and Igor Walukiewicz. Nondeterministic controllers of nondeterministic processes. In *Logic and Automata*, Texts in Logic and Games, pages 29–52. Amsterdam University Press, 2007.

[5] Patricia Bouyer, Franck Cassez, and François Laroussinie. Modal logics for timed control. In *Proc. of the 16th Inter. Conf. on Concurrency Theory (CONCUR'05)*, volume 3653 of *Lect. Notes in Comput. Sci.*, pages 81–94, San Francisco, CA, USA, august 2005. Springer.

[6] Deepak D'Souza. A logical characterisation of event recording automata. In *Proc. of the 6th Inter. Symp. on Formal Techniq. in Real-Time and Fault-Tolerant Systems (FTRTFT '00)*, pages 240–251, London, UK, 2000. Springer-Verlag.

[7] E. Allen Emerson. Automata, tableaux and temporal logics (extended abstract). In *Proc. of the Conf. on Logic of Programs*, pages 79–88, London, UK, 1985. Springer-Verlag.

[8] E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. of the 32nd annual Symp. on Found. of Computer Sci. (SFCS '91)*, pages 368–377, Washington, DC, USA, 1991. IEEE Computer Society.

[9] Uno Holmer, Kim Larsen, and Wang Yi. Deciding properties of regular real timed processes. In *Proc. of the 3th Inter. Conf. on Comput. Aided Verif. (CAV '91)*, volume 575 of *Lect. Notes in Comput. Sci.*, pages 432–442. Springer-Verlag, 1991.

[10] Dexter Kozen. Results on the propositional μ-calculus. In *Proc. of the 9th Inter. Colloq. on Aut., Lang. and Prog. (ICALP'82)*, pages 348–359, 1982.

[11] François Laroussinie and Kim Guldstrand Larsen. Compositional model checking of real time systems. In *Proc. of the 6th Inter. Conference on Concurrency Theory (CONCUR '95)*, pages 27–41, London, UK, 1995. Springer-Verlag.

[12] Omer Landry Nguena Timo. The logic wt_μ. Research Report RR-1460-09, LaBRI, CNRS, France, 2009.

[13] Omer Landry Nguena Timo and Pierre-Alain Reynier. On Characteristic Formulae for Event-Recording Automata. In *Proc. of the 6th Workshop on Fixpoints In Computer Science (FICS'09).*, pages 70–78, september 2009.

[14] Damian Niwiński and Igor Walukiewicz. Games for the μ-calculus. *Theo. Comput. Sci.*, 163(1-2):99–116, 1996.

[15] Jean-François Raskin and Pierre-Yves Schobbens. The logic of event clocks - decidability, complexity and expressiveness. *Journal of Automata, Languages and Combinatorics*, 4(3):247–286, 1999.

[16] Oleg Sokolsky and Scott A. Smolka. Local model checking for real-time systems (extended abstract). In *Proc. of the 7th Inter. Conf. on Comput. Aided Verif. (CAV'95)*, pages 211–224, London, UK, 1995. Springer-Verlag.

[17] Maria Sorea. A decidable fixpoint logic for time-outs. In *Proc. of the 13th Inter. Conf. on Concurrency Theory (CONCUR '02)*, pages 255–271, London, UK, 2002. Springer-Verlag.

[18] Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inform. and Comput.*, 81(3):249–264, 1989.

[19] T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine . Symbolic Model Checking for Real-Time Systems. In *Proc. of the 7th. Symp. of Logics in Comput. Sci. (LICS'92)*, pages 394–406, Santa-Cruz, California, 1992. IEEE Computer Scienty Press.

[20] Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theo. Comput. Sci.*, 200(1-2):135–183, 1998.