Timed temporal logics for abstracting transient states

Houda Bel Mokadem[1], Béatrice Bérard[2], Patricia Bouyer[1], François Laroussinie[1]

[1] LSV, CNRS & ENS de Cachan, France
Emails: {mokadem,bouyer,fl}@lsv.ens-cachan.fr
[2] LAMSADE, CNRS & Université Paris-Dauphine, France
Email: berard@lamsade.dauphine.fr

Abstract. In previous work, the timed logic TCTL was extended with an "almost everywhere" Until modality which abstracts *negligible* sets of positions (*i.e.* with a null duration) along a run of a timed automaton. We propose here an extension of this logic with more powerful modalities, in order to specify properties abstracting *transient* states, which are events that last for less than k time units. Our main result is that model-checking is still decidable and PSPACE-complete for this extension. On the other hand, a second semantics is defined, in which we consider the *total* duration where the property does not hold along a run. In this case, we prove that model-checking is undecidable.

1 Introduction

Timed verification. Temporal logic is a convenient formalism for specifying systems and reasoning about them. Furthermore, model-cheking techniques lead to the automatic verification that a model of a system satisfies some temporal logic specification. These methods have been extended to real-time verification: systems are modeled with timed automata [4] and timed logics like TCTL [1] are used to express timed specification like "any problem is followed by an alarm within 3 seconds". Analysis tools have been developped [22, 15, 20] and successfully applied to numerous case studies.

Timed temporal logics and duration properties. Along with the study of timed automata, various timed logics have been defined to extend the classical temporal logics with quantitative modalities. For example, this was done with MTL [19, 5, 21], an extension of LTL, and TCTL [6, 1, 17], where CTL modalities are augmented with time comparisons of the form $\sim c$, where $\sim$ is a comparison operator. Another related logic is the Parametrized TCTL [13] where TCTL and the timed model are in turn extended with parameters.
In another direction, since the introduction of the *duration calculus* [14] in order to express duration properties, numerous works have been devoted to the algorithmic computation of such properties for timed systems. Since *clocks*, which

evolve at the rate of time (as in timed automata), are sometimes not expressive enough, hybrid variables (with multiple slopes) have been considered. The resulting model of hybrid automata has been largely studied in the subsequent years [16]. However, while some decidability results could be obtained [3, 18], using stopwatches (*i.e.* variables with slopes 0 and 1) already leads to undecidability for the reachability problem [2].

Further research has thus been devoted to weaker models where hybrid variables are only used as *observers*, *i.e.* are not tested in the automaton and thus play no role during a computation. These variables, sometimes called costs or prices in this context can be used in an optimization criterium [3, 7, 8, 11] or as constraints in temporal logic formulas. For instance, the logic WCTL [12, 10], interpreted over timed automata extended with costs, adds cost contraints on modalities: it is possible to express that a given state is reachable within a fixed cost bound.

Abstracting transient states. When practical examples are considered, the need for abstracting transient states often happens. For example, modeling the instantaneous changes of a variable may introduce artificial (and thus non pertinent) transient states in the model. This motivated the work in [9], where configurations with zero duration could be abstracted by introducing into TCTL the *almost everywhere* U^{a} modality. However, this is not sufficient in some cases.

Contribution. In this paper, we propose an extension of TCTL called TCTL^{Δ}, which brings out a powerful generalization of the results in [9]. We introduce a new modality U^k, where $k \in \mathbb{N}$ is a parameter, in order to abstract events that do not last continuously for at least k time units (t.u). For example, $\mathsf{AF}^2_{\leq 100}\,alarm$ expresses that for any execution, the atomic proposition *alarm* becomes true before 100 t.u and will hold for at least 2 time units. One also could express the fact that an event a precedes an event b along any run, an event being actually considered iff it lasts for at least k time units: the formula $\mathsf{A}\,request\mathsf{P}^3\,grant$ states that along any run where *grant* has occurred for a duration greater than 3, a *request* has been emitted continusously for a duration greater than 3. We prove that model-checking for TCTL^{Δ} is still PSPACE-complete. While the analogous result for TCTL or the extended version of [9] relies on the standard notion of equivalent runs, we have to define a stronger form for this equivalence, in order to obtain the consistency of TCTL^{Δ}-formulae on the regions of the timed automaton.

Finally, we also consider a *global* semantics, called $\mathsf{TCTL}^{\Delta}_{\Sigma}$, for which the global duration during which a property does not hold, is bounded by a fixed constant k. Although this semantics is more natural and uses only observer hybrid variables in the model, we prove that model-checking $\mathsf{TCTL}^{\Delta}_{\Sigma}$ is undecidable.

Outline. Section 2 recalls the main features of timed automata model and gives definitions for the syntax and semantics of our extended logics. Sections 3 and 4 are devoted to the model-checking of TCTL^{Δ} and, in the last section, we show that model-checking the extended logic $\mathsf{TCTL}^{\Delta}_{\Sigma}$ is undecidable.

2 Logic TCTL^{Δ}

Let $\mathbb{N}$ and $\mathbb{R}$ denote the sets of natural and non-negative real numbers, respectively. Let X be a set of real valued clocks. We write $\mathcal{C}(X)$ for the set of boolean expressions over atomic formulae of the form $x \sim k$ with $x \in X$, $k \in \mathbb{N}$, and $\sim \in \{<, \leq, =, \geq, >\}$. Constraints of $\mathcal{C}(X)$ are interpreted over *valuations* for clocks, i.e. mappings from X to $\mathbb{R}$. The set of valuations is denoted by $\mathbb{R}^X$. For every $v \in \mathbb{R}^X$ and $d \in \mathbb{R}$, we use $v+d$ to denote the time assignment which maps each clock $x \in X$ to the value $v(x)+d$. For every $r \subseteq X$, we write $v[r \leftarrow 0]$ for the valuation which maps each clock in r to the value 0 and agrees with v over $X \setminus r$. Let AP be a set of atomic propositions.

2.1 Timed Automata

Definition 1. *A* timed automaton *(TA) is a tuple* $A = \langle X, Q_A, q_{init}, \rightarrow_A, \mathsf{Inv}_A, l_A \rangle$ *where* X *is a finite set of clocks,* Q_A *is a finite set of* locations or control states *and* $q_{init} \in Q_A$ *is the* initial location*. The set* $\rightarrow_A \subseteq Q_A \times \mathcal{C}(X) \times 2^X \times Q_A$ *is a finite set of* action transitions*: for* $(q, g, r, q') \in \rightarrow_A$*,* g *is the enabling condition and* r *is a set of clocks to be reset with the transition (we write* $q \xrightarrow{g,r}_A q'$*).* $\mathsf{Inv}_A \colon Q_A \rightarrow \mathcal{C}(X)$ *assigns an* invariant *to each control state. Finally* $l_A \colon Q_A \rightarrow 2^{\mathsf{AP}}$ *labels every location with a subset of* AP*.*

A *state* (or *configuration*) of a TA A is a pair (q, v), where $q \in Q_A$ is the current location and $v \in \mathbb{R}^X$ is the current clock valuation. The initial state of A is (q_{init}, v_0) with $v_0(x) = 0$ for any x in X. There are two kinds of transition. From (q, v), it is possible to perform the *action transition* $q \xrightarrow{g,r}_A q'$ if $v \models g$ and $v[r \leftarrow 0] \models \mathsf{Inv}_A(q')$ and then the new configuration is $(q', v[r \leftarrow 0])$. It is also possible to let time elapse, and reach $(q, v+d)$ for some $d \in \mathbb{R}$ whenever the invariant is satisfied along the delay. Formally the semantics of a TA A is given by a Timed Transition System (TTS) $\mathcal{T}_A = (S, s_{\text{init}}, \rightarrow_{\mathcal{T}_A}, l)$ where:

- $S = \{(q, v) \mid q \in Q_A \text{ and } v \in \mathbb{R}^X \text{ s.t. } v \models \mathsf{Inv}_A(q)\}$ and $s_{\text{init}} = (q_{\text{init}}, v_0)$.
- $\rightarrow_{\mathcal{T}_A} \subseteq S \times S$ and we have $(q, v) \rightarrow_{\mathcal{T}_A} (q', v')$ iff
  - either $q' = q$, $v' = v + d$ and $v + d' \models \mathsf{Inv}_A(q)$ for any $d' \leq d$. This is a delay transition — we write $(q, v) \xrightarrow{d} (q, v+d)$ —,
  - or there exists $q \xrightarrow{g,r}_A q'$ s.t $v \models g$, $v' = v[r \leftarrow 0]$ and $v' \models \mathsf{Inv}_A(q')$. This is an action transition — we write $(q, v) \rightarrow_a (q', v')$.
- $l \colon S \rightarrow 2^{\mathsf{AP}}$ labels every state (q, v) with the subset $l_A(q)$ of AP .

An execution (or run) of A is an infinite path $s_0 \rightarrow_{\mathcal{T}_A} s_1 \rightarrow_{\mathcal{T}_A} s_2 \ldots$ in $\mathcal{T}_A$ such that (1) time diverges and (2) there are infinitely many action transitions. Note that an execution can be described as an alternating infinite sequence $s_0 \xrightarrow{d_1}\rightarrow_a s_1 \xrightarrow{d_2}\rightarrow_a \cdots$ for some $d_i \in \mathbb{R}$. Such an execution ρ goes through any configuration s' reachable from some s_i by a delay transition of duration $d \in [0, d_i]$. Let Exec(s) be the set of all executions from s. With a run $\rho : (q_0, v_0) \xrightarrow{d_1}\rightarrow_a (q_1, v_1) \xrightarrow{d_2}\rightarrow_a \ldots$ of A, we associate the sequence of absolute dates

defined by $t_0 = 0$ and $t_i = \sum_{j \leq i} d_j$ for $i \geq 1$, and in the sequel, we often write ρ as the sequence $((q_i, v_i, t_i))_{i \geq 0}$.
A state (q, v) can occur several times along a run ρ, the notion of *position* [3] allows us to distinguish them: every occurrence of a state is associated with a unique position. Given a position p, the corresponding state is denoted by s_p.
The standard notions of prefix, suffix and subrun apply to paths in TTS: given a position $p \in \rho$, $\rho^{\leq p}$ is the prefix leading to p, $\rho^{\geq p}$ is the suffix issued from p. Finally a subrun σ from p to p' is denoted by $p \overset{\sigma}{\mapsto} p'$.
Note that the set of positions along ρ is totally ordered by $<_\rho$. Given two positions p and p', we say that p *precedes strictly* p' along ρ (written $p <_\rho p'$) iff there exists a finite subrun σ of ρ s.t. $p \overset{\sigma}{\mapsto} p'$ and σ contains at least one non null delay transition or one action transition (*i.e.* σ is not reduced to $\xrightarrow{0}$). We write $\sigma <_\rho p$ when for any position p' in the subrun σ, we have $p' <_\rho p$.
Given a position $p \in \rho$, the prefix $\rho^{\leq p}$ has a *duration*, $\mathsf{Time}(\rho^{\leq p})$, defined as the sum of all delays along $\rho^{\leq p}$. Since time diverges along an execution, we have: for any $t \in \mathbb{R}$, there exists $p \in \rho$ such that $\mathsf{Time}(\rho^{\leq p}) > t$.
For a subset $P \subseteq \rho$ of positions in ρ, we define a natural measure $\hat{\mu}(P) = \mu\{\mathsf{Time}(\rho^{\leq p}) \mid p \in P\}$, where μ is Lebesgue measure on the set of real numbers. In the sequel, we only use this measure when P is a subrun of ρ: in this case, for a subrun σ such that $p \overset{\sigma}{\mapsto} p'$, we simply have $\hat{\mu}(\sigma) = \mathsf{Time}(\rho^{\leq p'}) - \mathsf{Time}(\rho^{\leq p})$.

2.2 Definition of TCTL^Δ.

TCTL^Δ is obtained by adding to TCTL the modalities $\mathsf{E\_U}^k_{\sim c}\mathsf{\_}$ and $\mathsf{A\_U}^k_{\sim c}\mathsf{\_}$ with $k \in \mathbb{N}$:

Definition 2 (Syntax of TCTL^Δ). *TCTL^Δ formulae are given by the following grammar:*

$$\varphi, \psi ::= P_1 \mid P_2 \mid \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\sim c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\sim c}\psi \mid \mathsf{E}\varphi\mathsf{U}^k_{\sim c}\psi \mid \mathsf{A}\varphi\mathsf{U}^k_{\sim c}\psi$$

where $P_i \in \mathsf{AP}$, $\sim$ belongs to the set $\{<, >, \leq, \geq, =\}$ and $c, k \in \mathbb{N}$.

Standard abbreviations include $\top, \bot, \varphi \vee \psi, \varphi \Rightarrow \psi, \ldots$ as well as :

$$\begin{array}{ll} \mathsf{EF}^k_{\sim c}\,\varphi \overset{def}{=} \mathsf{E}(\top\,\mathsf{U}^k_{\sim c}\,\varphi) & \mathsf{AF}^k_{\sim c}\,\varphi \overset{def}{=} \mathsf{A}(\top\,\mathsf{U}^k_{\sim c}\,\varphi) \\ \mathsf{EG}^k_{\sim c}\,\varphi \overset{def}{=} \neg\mathsf{AF}^k_{\sim c}\neg\varphi & \mathsf{AG}^k_{\sim c}\,\varphi \overset{def}{=} \neg\mathsf{EF}^k_{\sim c}\neg\varphi \end{array}$$

Moreover U^k stands for $\mathsf{U}^k_{\geq 0}$.

Definition 3 (Semantics of TCTL^Δ). *The following clauses define when a state s of some TTS $\mathcal{T} = \langle S, s_{init}, \rightarrow, l\rangle$ satisfies a TCTL^Δ formula φ, written*

[3] Note that as it is possible to perform a sequence of action transitions in 0 t.u., we cannot replace the notion of positions by a function from f_ρ from $\mathbb{R}$ to S.

$s \models \varphi$, by induction over the structure of φ (the semantics of boolean operators is omitted).

$$\begin{array}{ll}
s \models \mathsf{E}\varphi\mathsf{U}_{\sim c}\psi & \textit{iff } \exists\, \rho \in \textit{Exec}(s) \textit{ s.t. } \rho \models \varphi\mathsf{U}_{\sim c}\psi \\
s \models \mathsf{A}\varphi\mathsf{U}_{\sim c}\psi & \textit{iff } \forall\, \rho \in \textit{Exec}(s) \textit{ we have } \rho \models \varphi\mathsf{U}_{\sim c}\psi \\
s \models \mathsf{E}\varphi\mathsf{U}^k_{\sim c}\psi & \textit{iff } \exists\, \rho \in \textit{Exec}(s) \textit{ s.t. } \rho \models \varphi\mathsf{U}^k_{\sim c}\psi \\
s \models \mathsf{A}\varphi\mathsf{U}^k_{\sim c}\psi & \textit{iff } \forall\, \rho \in \textit{Exec}(s) \textit{ we have } \rho \models \varphi\mathsf{U}^k_{\sim c}\psi
\end{array}$$

$$\begin{array}{ll}
\rho \models \varphi\mathsf{U}_{\sim c}\psi & \textit{iff } \exists p \in \rho \textit{ s.t. } \mathsf{Time}(\rho^{\leq p}) \sim c \wedge s_p \models \psi \wedge \forall p' <_\rho p,\ s_{p'} \models \varphi \\
\rho \models \varphi\mathsf{U}^k_{\sim c}\psi & \textit{iff there exists a subrun } \sigma \textit{ along } \rho, \textit{ a position } p \in \sigma \textit{ s.t.} \\
 & \mathsf{Time}(\rho^{\leq p}) \sim c \wedge \hat{\mu}(\sigma) > k \wedge \forall p' \in \sigma,\ s_{p'} \models \psi \\
 & \textit{and for all subrun } \sigma' \textit{ s.t. } \sigma' <_\rho p \wedge \forall p' \in \sigma',\ s_{p'} \models \neg\varphi \\
 & \textit{we have } \hat{\mu}(\sigma') \leq k
\end{array}$$

The modality U^k allows us to abstract intervals with duration less than k t.u. where φ does not hold. Thus $\mathsf{AF}^2_{\leq 100}\,\textit{alarm}$ states that along every run, there is an event *alarm* of duration greater than 2 t.u. that occurs before 100 t.u.

The precedence operator [4] P can be written as follows: $\mathsf{A}\varphi\mathsf{P}^k\psi \overset{\text{def}}{=} \neg\mathsf{E}(\neg\varphi)\mathsf{U}^k\psi$. For example, A *request* P^3*grant* states that a *request* of duration greater than 3 has to occur before an event *grant* (which must also last more than 3 t.u.).
Note that the semantics has to be handled carefully: $\Phi = \mathsf{AG}^k\varphi$ expresses that no event $\neg\varphi$ occurs, *i.e.* it is not possible to have $\neg\varphi$ continuously for more than k t.u. An execution where $\neg\varphi$ holds for everywhere except every k t.u. would satisfy Φ. This choice of semantics is also motivated by negation closure of the Until modality.
Note that the logic $\mathsf{TCTL}^{\mathsf{ext}}$ defined in [9] is the restriction of TCTL^{Δ} where the parameter k is always 0. As the modality $\mathsf{E\_U^0\_}$ cannot be expressed in TCTL[9], TCTL^{Δ} is clearly more expressive then TCTL.

The size of a timed automaton and the size of a TCTL^{Δ} formula are defined in the standard way with constants written in binary notation.

3 Equivalence of runs

In this section, we show that the classical notion of region proposed by Alur, Courcoubetis and Dill [1] for TCTL is also correct for TCTL^{Δ}. Nevertheless we need a stronger notion of equivalence for the runs in order to preserve the truth value of TCTL^{Δ} formulae.
First let us recall the standard equivalence over valuations:

Definition 4 (Equivalence on valuations [1]). *Given a set X of clocks and $M \in \mathbb{N}$, two valuations $v, v' \in \mathbb{R}^X$ are* M-equivalent *(written $v \cong_M v'$) if:*

1. *for any $x \in X$ $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$ or $(v(x) > M \wedge v'(x) > M)$,*

[4] This is a kind of release operator.

2. for any $x, y \in X$ s.t. $v(x) \leq M$ and $v(y) \leq M$, we have:
$\mathsf{frac}(v(x)) \leq \mathsf{frac}(v(y)) \Leftrightarrow \mathsf{frac}(v'(x)) \leq \mathsf{frac}(v'(y))$ *and*
$\mathsf{frac}(v(x)) = 0 \Leftrightarrow \mathsf{frac}(v'(x)) = 0$.

An equivalence class of $\cong$ is called a *region*; and a region is called a *boundary region* if it contains valuations v s.t. the fractional part of $v(x)$ is 0, for some clock x. Given a TA A, we use M_A to denote the maximal constant occurring in A (in its guards or invariants). We write simply $\cong$ instead of $\cong_M$ when M is clear from the context. The equivalence $\cong_{M_A}$ is consistent w.r.t. TCTL^{Δ} formulae:

Theorem 1 (Consistency of $\cong$). *Given a TA A, $\Phi \in \mathsf{TCTL}^{\Delta}$ and $v, v' \in \mathbb{R}^X$ s.t. $v \cong_{M_A} v'$, we have: $(q, v) \models \Phi \;\Leftrightarrow\; (q, v') \models \Phi$.*

Consider the formula $\Phi = \mathsf{E}\varphi\mathsf{U}^k_{\sim c}\psi$ and assume $(q, v) \models \Phi$, *i.e.* there exists a run $\rho = ((q_i, v_i, t_i))_{i\geq 0}$ from (q, v) satisfying $\varphi\mathsf{U}^k_{\sim c}\psi$. In order to prove the theorem, we need to show that there exists an *equivalent* run ρ' from (q, v') which also satisfies $\varphi\mathsf{U}^k_{\sim c}\psi$.

For this, we first extend $\cong$ to pairs (v_i, t_i) as follows: $(v_i, t_i) \cong (v'_i, t'_i)$ iff (1) $v_i \cong v'_i$, (2) $\lfloor t_i \rfloor = \lfloor t'_i \rfloor$ and $\mathsf{frac}(t_i) = 0$ iff $\mathsf{frac}(t'_i) = 0$ and (3) for each clock $x \in X$, (i) $\mathsf{frac}(v_i(x)) < \mathsf{frac}(t_i)$ iff $\mathsf{frac}(v'_i(x)) < \mathsf{frac}(t'_i)$ and (ii) $\mathsf{frac}(v_i(x)) = \mathsf{frac}(t_i)$ iff $\mathsf{frac}(v'_i(x)) = \mathsf{frac}(t'_i)$.

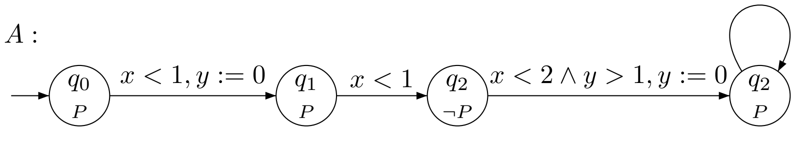
Now we define the equivalence over runs as follows:

Definition 5 (Equivalence on runs). *Given a TA A, two runs $\rho = ((q_i, v_i, t_i))_{i\geq 0}$ and $\rho' = ((q'_i, v'_i, t'_i))_{i\geq 0}$ are equivalent (written $\rho \cong^* \rho'$) if*

(ER a.) for all $i \geq 0$, $q_i = q'_i$,
(ER b.) for all $i \geq 0$, $(v_i, t_i) \cong_{M_A} (v'_i, t'_i)$,
(ER c.) for all $0 \leq j < i$, (i) $\mathsf{frac}(t_j) < \mathsf{frac}(t_i)$ iff $\mathsf{frac}(t'_j) < \mathsf{frac}(t'_i)$
and (ii) $\mathsf{frac}(t_j) = \mathsf{frac}(t_i)$ iff $\mathsf{frac}(t'_j) = \mathsf{frac}(t'_i)$.

The equivalence on runs used in [1] to prove that regions are compatible with TCTL formulae only requires conditions *(ER a)* and *(ER b)*. This is however not sufficient for proving Theorem 1. Indeed, let A be the automaton depicted below, with atomic proposition P and two clocks x and y, and consider the two following runs, which are equivalent in [1]:

$\rho : (q_0, (0, 0)) \xrightarrow{0.1}\rightarrow_a (q_1, (0.1, 0)) \xrightarrow{0.8}\rightarrow_a (q_2, (0.9, 0.8)) \xrightarrow{0.3}\rightarrow_a (q_3, (1.2, 0)) \ldots$
$\rho' : (q_0, (0, 0)) \xrightarrow{0.8}\rightarrow_a (q_1, (0.8, 0)) \xrightarrow{0.1}\rightarrow_a (q_2, (0.9, 0.1)) \xrightarrow{1.05}\rightarrow_a (q_3, (1.95, 0)) \ldots$



The runs ρ and ρ' satisfy conditions *(ER a)* and *(ER b)* but the delays spent in state q_2 where P does not hold are respectively 0.3 and 1.05, so that $\rho \models G^1 P$ whereas $\rho' \not\models G^1 P$.

This is why we need the stronger equivalence above which also requires condition *(ER c)*. Note that this condition *(ER c)* does not correspond to a splitting of the regions. Moreover, we will not prove that all equivalent paths satisfy the same until-formulae but rather that given a path ρ leaving from a configuration (q,v), we can build a path ρ', equivalent to ρ and which satisfies the formula we consider. The following proposition [5] then ensures the existence of equivalent runs:

Proposition 1. *Given a TA A, $q \in Q_A$, and $v, v' \in \mathbb{R}^X$ s.t. $v \cong_{M_A} v'$, then $\forall \rho \in Exec((q,v))$, there exists a run $\rho' \in Exec((q,v'))$ s.t. $\rho \cong^* \rho'$.*

We can now prove Theorem 1.

Proof (Theorem 1 – sketch). The proof is done by structural induction on Φ. We omit the basic cases and the TCTL operators (similar to [1]). Assume $(q,v) \models \mathsf{E}\varphi\mathsf{U}^k_{\sim c}\psi$. Let $\rho = ((q_i,v_i,t_i))_{i\geq 0}$ be a run from (q,v) s.t. $\rho \models \varphi\mathsf{U}^k_{\sim c}\psi$. Consider a run ρ' from (q,v') equivalent to ρ (its existence is ensured by Proposition 1). Along ρ the truth value of φ and ψ depends on the current *region*. We know that ρ' goes through the same sequence of regions (as for TCTL) but we have also to show that the amounts of time spent in every sequence of consecutive regions in ρ and ρ' have the same integral part (less than or equal to k for $\neg\varphi$ and greater than k for ψ). Let σ be a subrun of ρ corresponding to an arrival in some region at time δ_1 until a departure from another region at time δ_2. Let δ'_1 and δ'_2 be the corresponding dates in ρ'. We want to prove that $\lfloor\delta_2-\delta_1\rfloor = \lfloor\delta'_2-\delta'_1\rfloor$.
A sufficient condition for this would be (1) $\lfloor\delta_i\rfloor = \lfloor\delta'_i\rfloor$ for $i=1,2$, (2) $\mathsf{frac}(\delta_1) < \mathsf{frac}(\delta_2)$ iff $\mathsf{frac}(\delta'_1) < \mathsf{frac}(\delta'_2)$ and (3) $\mathsf{frac}(\delta_1) = \mathsf{frac}(\delta_2)$ iff $\mathsf{frac}(\delta'_1) = \mathsf{frac}(\delta'_2)$. Such a property would be ensured if the dates δ_i (and δ'_i) occurred as some t_j in ρ (and ρ'). But the t_js are the dates of *action* transitions. Consider the new TA $\bar{A}$ that extends A with loops on every control states, with no guard and no reset. In $\bar{A}$, there are additional runs (compared to A) but they induce no problem for checking $\mathsf{E}\_\mathsf{U}^k_{\sim c}\_$ formulae.
Consider the run $\bar{\rho}$ in $\bar{A}$ that mimics ρ except that it performs a loop before entering/exiting a region [6]. Clearly $\bar{\rho}$ satisfies also $\varphi\mathsf{U}^k_{\sim c}\psi$. Now we consider a run $\bar{\rho}'$ from (q,v') equivalent to $\bar{\rho}$; then the property above over the δ_i is ensured by the definition of $\cong^*$. Clearly $\bar{\rho}' \models \varphi\mathsf{U}^k_{\sim c}\psi$. We can consider in A the run ρ' similar to $\bar{\rho}'$ without using the loops: ρ' satisfies $\varphi\mathsf{U}^k_{\sim c}\psi$. Then $(q,v') \models \mathsf{E}\varphi\mathsf{U}^k_{\sim c}\psi$.

Now consider the case of $\Phi = \mathsf{A}\varphi\mathsf{U}^k_{\sim c}\psi$. Assume $(q,v) \not\models \Phi$ and let ρ be a run from (q,v) s.t. $\rho \models \neg(\varphi\mathsf{U}^k_{\sim c}\psi)$. Thus we have either (1) there is no subrun σ of duration greater than k satisfying ψ and containing a position p located at duration $\sim c$, or (2) for any such σ and p, there exists a subrun $\sigma' <_\rho p$ s.t. σ' satisfies $\neg\varphi$ and $\hat{\mu}(\sigma') > k$. In both case, we can build a corresponding run from (q,v') witnessing $\neg(\varphi\mathsf{U}^k_{\sim c}\psi)$. □

[5] The omitted proofs are given in the long version of the paper.

[6] NB: when going into/out a non-boundary region, we consider the date corresponding to the previous/next boundary region.

4 Model-Checking algorithm

In this section we show how to reduce the model-checking problem $A \models \Phi$ with a TA $A = \langle X, Q_A, q_{\mathrm{init}}, \rightarrow_A, \mathsf{Inv}_A, l_A \rangle$ and $\Phi \in \mathsf{TCTL}^{\Delta}$, to a model-checking problem $A' \models \Phi'$ where A' is a *region graph* (*i.e.* a finite Kripke structure) and Φ' is a CTL-like formula.

Let X^* be the set of clocks $X \cup \{z, z_r, z_{\bar{l}}\}$. The three extra clocks are used to verify timing constraints in the formula: z is used to handle subscripts $\sim c$ in U modalities (as in TCTL model checking) and the clock $z_{\bar{l}}$ (resp z_r) is used to measure time elapsing when the left (resp. right) part in U^k modalities is false (resp true). Thus these new clocks are used as observers and do not modify the behavior of A.

Let M_Φ be the maximal constant occurring in the timing constraints in Φ and k_m be the maximal k occurring in a modality U^k in Φ. Let M be $\max(M_A, M_\Phi + k_m)$. The region graph $\mathcal{R}_{A,\Phi} = (V, \rightarrow, l, F)$ for A and Φ is defined as usual over X^* and M [1]: its set of states V is $\{(q,\gamma) \,|\, q \in Q_A \,\mathrm{and}\, \gamma \in \mathbb{R}^{X^*}/\cong_M\}$, the transitions correspond to action transitions ($\rightarrow_a$) in A or delay transitions ($\rightarrow_t$, leading to the *successor region* denoted by $succ(\gamma)$). The states are labeled with atomic propositions AP and we also use additional propositions for the extra clocks: a state (q,γ) is labeled with the proposition $(\!|y \sim a|\!)$ with $y \in \{z, z_{\bar{l}}, z_r\}$ and $0 \le a \le M$, when $\gamma \models y \sim a$. Moreover we use the proposition P_b to mark boundary regions. And F is a fairness constraint to enforce time divergence (see [1, 9] for the detailed construction of $\mathcal{R}_{A,\Phi}$).

Labeling algorithm. We label the vertices of $\mathcal{R}_{A,\Phi}$ with the subformulae of Φ they satisfy, starting from the subformulae of length 1 and length 2 and so on. Here we only consider the U^k modalities.

Consider a formula Ψ of the form $\mathsf{E}\varphi_l \mathsf{U}^k_{\sim c} \varphi_r$ or $\mathsf{A}\varphi_l \mathsf{U}^k_{\sim c} \varphi_r$. At this step we know for every state (q,γ) of $\mathcal{R}_{A,\Phi}$ whether it satisfies (or not) φ_l and φ_r (*i.e.* whether any (q,v) with $v \in \gamma$ satisfies φ_l or/and φ_r). First we define a variant of $\mathcal{R}_{A,\Phi}$, called $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$, where some transitions are modified according to the truth value of φ_l and φ_r:

1. we replace the transitions $(q,\gamma) \rightarrow_t (q, succ(\gamma))$ by $(q,\gamma) \rightarrow_a (q, \gamma[z_{\bar{l}} \leftarrow 0])$ when $(q,\gamma) \models \varphi_l$, $(q, succ(\gamma)) \models \neg\varphi_l$ and $\gamma \not\models z_{\bar{l}} = 0$.
2. we replace the transitions $(q,\gamma) \rightarrow_a (q',\gamma')$ by $(q,\gamma) \rightarrow_a (q', \gamma'[z_{\bar{l}} \leftarrow 0])$ when $(q,\gamma) \models \varphi_l$, $(q',\gamma') \models \neg\varphi_l$.
3. we replace the transitions $(q,\gamma) \rightarrow_t (q, succ(\gamma))$ by $(q,\gamma) \rightarrow_a (q, \gamma[z_r \leftarrow 0])$ when $(q,\gamma) \models \neg\varphi_r$, $(q, succ(\gamma)) \models \varphi_r$ and $\gamma \not\models z_r = 0$.
4. we replace the transitions $(q,\gamma) \rightarrow_a (q',\gamma')$ by $(q,\gamma) \rightarrow_a (q', \gamma'[z_r \leftarrow 0])$ when $(q,\gamma) \models \neg\varphi_r$, $(q,\gamma') \models \varphi_r$.

Due to these changes, in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$, the clock $z_{\bar{l}}$ (resp. z_r) measures the time elapsed since $\neg\varphi_l$ (resp. φ_r) is true : they are reset when the truth value of the corresponding formula changes. Thus given a path ρ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ and a state (q,γ) along ρ, we have $(q,\gamma) \models \neg\varphi_l \wedge (\!|z_{\bar{l}} \le k|\!)$ iff there was (along ρ) a region satisfying φ_l "just before" (q,γ) where "just before" means "in less than k time units".

In the following we will use two abbreviations:

$$\overset{\dashleftarrow}{\varphi_l} \overset{\text{def}}{=} \varphi_l \vee \llparenthesis z_{\bar{l}} \leq k \rrparenthesis \qquad\qquad \overleftarrow{\varphi_r} \overset{\text{def}}{=} \varphi_r \wedge \llparenthesis z_r > k \rrparenthesis$$

The first one states that φ_l holds or did hold less than k t.u. ago. And the second one states that φ_r lasts for more than k t.u. We will also use the abbreviation to $\overleftarrow{\neg\varphi_l}$ to denote $\neg\varphi_l \wedge \llparenthesis z_{\bar{l}} > k \rrparenthesis$: the formula $\neg\varphi_l$ has held for more than k t.u. And we use $\overset{\dashleftarrow}{\neg\varphi_r}$ for $\neg\varphi_r \vee \llparenthesis z_r \leq k \rrparenthesis$. In this context, we have: $\overset{\dashleftarrow}{\varphi} \equiv \neg\overleftarrow{(\neg\varphi)}$. Thus the region graph $\mathcal{R}_{A,\Phi}^{\varphi_l,\varphi_r}$ allows us to decide $\overset{\dashleftarrow}{\varphi_l}$, $\overleftarrow{(\neg\varphi_l)}$, $\overleftarrow{\varphi_r}$ and $\overset{\dashleftarrow}{(\neg\varphi_r)}$.
Now we distinguish different cases depending on the modality rooted in Ψ:

- $\Psi \overset{\text{def}}{=} \mathsf{E}\varphi_l \mathsf{U}^k_{\sim c}\varphi_r$. We label a state (q,γ) of $\mathcal{R}_{A,\Phi}$ by Ψ iff $(q,\gamma[z,z_{\bar{l}},z_r \leftarrow 0])$ satisfies in $\mathcal{R}_{A,\Phi}^{\varphi_l,\varphi_r}$ the following CTL-formula:

$$\Psi_1 \overset{\text{def}}{=} \mathsf{E}\,\overset{\dashleftarrow}{\varphi_l}\,\mathsf{U}\Big(\llparenthesis z \sim c \rrparenthesis \wedge (\textsf{after-a} \vee P_b \vee \overset{\dashleftarrow}{\varphi_l}) \wedge \mathsf{E}\,\varphi_r\,\mathsf{U}\overleftarrow{\varphi_r}\Big)$$

  where $\textsf{after-a}$ holds for a state s along a path when the last transition performed (before reaching s) is an action transition. This is not, properly speaking, an atomic proposition since it depends on the way used to reach the state but it can easily be obtained either by using an EX modality or by changing $\mathcal{R}_{A,\Phi}$ in order to use an atomic proposition.
  Note that for labeling the TCTL formula $\mathsf{E}\varphi_l\mathsf{U}_{\sim c}\varphi_r$, one use the following formula: $\mathsf{E}\,\varphi_l\,\mathsf{U}\Big(\llparenthesis z \sim c \rrparenthesis \wedge \varphi_r \wedge (\textsf{after-a} \vee P_b \vee \varphi_l)\Big)$. This formula states that there exists a path leading to a state s satisfying $\llparenthesis z \sim c \rrparenthesis$ (*i.e.* the amount of elapsed time since $(q,\gamma[z,z_{\bar{l}},z_r \leftarrow 0])$ satisfies $\sim c$), φ_r and either $\textsf{after-a}$, P_b or φ_l: this last requirement is necessary because when s is not a boundary region and it has been reached via a delay transition, the formula φ_l has to hold also for this state [1].
  The formula Ψ_1 used for $\mathsf{E}\varphi_l\mathsf{U}^k_{\sim c}\varphi_r$ is based on the same structure, except that φ_l is replaced by $\varphi_l \vee \llparenthesis z_{\bar{l}} \leq k \rrparenthesis$ (we allow short periods –of duration less than k – where $\neg\varphi_l$ holds) and we also specify that φ_r has to hold during more than k time units (*i.e.* $\varphi_r \wedge \llparenthesis z_r > k \rrparenthesis$ has to hold).
  The notion of fair runs (used to ensure time divergence) is handled in the same manner as for TCTL.

- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k\varphi_r$. We label a state (q,γ) by Ψ iff $(q,\gamma[z,z_{\bar{l}},z_r \leftarrow 0])$ satisfies in $\mathcal{R}_{A,\Phi}^{\varphi_l,\varphi_r}$ the CTL-formula $\Psi_2 \overset{\text{def}}{=} \Psi_2' \wedge \Psi_2'' \wedge \Psi_3'''$ with:

$$\begin{aligned}
\Psi_2' &\overset{\text{def}}{=} \mathsf{AF}(\overleftarrow{\varphi_r}) \\
\Psi_2'' &\overset{\text{def}}{=} \neg\mathsf{E}(\neg\,\overleftarrow{\varphi_r})\mathsf{U}\Big(\overleftarrow{(\neg\varphi_l)} \wedge \neg\mathsf{A}(\varphi_r\mathsf{U}\overleftarrow{\varphi_r})\Big) \\
\Psi_2''' &\overset{\text{def}}{=} \neg\mathsf{E}(\neg\,\overleftarrow{\varphi_r})\mathsf{U}\Big(P_b \wedge \neg\varphi_r \wedge \mathsf{EX}(\neg P_b \wedge \neg\overset{\dashleftarrow}{(\varphi_l)})\Big)
\end{aligned}$$

  Ψ_2' states that along any path, eventually φ_r holds for at least k t.u. Ψ_2'' expresses that it is not possible to have $\neg\varphi_l$ for more than k t.u. unless either φ_r has already been verified for k t.u. before, or the current state belongs to the interval σ witnessing $\overleftarrow{\varphi_r}$. Finally Ψ_2''' is used to specify that, in the last case, if the first region of σ is a not a boundary region and if it has been reached via a delay transition, then it also has to satisfy $\overset{\leftarrow\cdot\cdot\shortmid}{\varphi_l}$ (for the same reason as for the $\mathsf{E\_U\_}$ modality).

- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k_{<c}\varphi_r$. For dealing with this case, we first consider the formula $\mathsf{AF}^k_{<c}\varphi_r$ and more precisely we consider the dual modality $\mathsf{EG}^k_{<c}$.

  The formula $\mathsf{EG}^k_{<c}\psi$ expresses that there exists an execution (from the current state s) where any subrun σ s.t. (1) $\hat{\mu}(\sigma) > k$ and (2) σ contains states located before c t.u. from s, contains a state satisfying ψ. Thus states satisfying ψ have to occur "often" (at least every k t.u.) during $c + k$ t.u. Therefore we label states (q, γ) by $\mathsf{EG}^k_{<c}\psi$ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies the CTL-formula $\mathsf{E}(\overset{\leftarrow\cdot\cdot\shortmid}{\psi})\mathsf{U}\llparenthesis z = c + k\rrparenthesis$.

  For labeling $\mathsf{AF}^k_{<c}\varphi_r$, we can then use: $\Psi_3 \overset{\text{def}}{=} \neg\mathsf{E}\overset{\leftarrow\cdots\cdots\shortmid}{(\neg\varphi_r)}\ \mathsf{U}\ \llparenthesis z = c+k\rrparenthesis$ for $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$.

  Therefore we label states (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies the CTL-formula $\Psi_3 \wedge \Psi_2'' \wedge \Psi_2'''$: compared with Ψ_2, we just have to require that φ_r holds before c t.u. (for more than k t.u.).
- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k_{\leq c}\varphi_r$. One just has to consider the following formula: $\Psi_4 \overset{\text{def}}{=} \neg\mathsf{E}\overset{\leftarrow\cdots\cdots\shortmid}{(\neg\varphi_r)}\ \mathsf{U}\ \llparenthesis z > c+k\rrparenthesis$. And we label states (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies the CTL-formula $\Psi_4 \wedge \Psi_2'' \wedge \Psi_2'''$ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$.
- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k_{\geq c}\varphi_r$. We label a state (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ the formula: $\Psi_5 \overset{\text{def}}{=} \mathsf{A}\overset{\leftarrow\cdot\cdot\shortmid}{\varphi_l}\mathsf{U}\big(\llparenthesis z = c\rrparenthesis \wedge \mathsf{AF}(\overleftarrow{\varphi_r}) \wedge \Psi_2'' \wedge \Psi_2'''\big)$. Ψ_5 states that along any run, φ_r will hold for more than k t.u. beyond a position where $z = c$, and that $\neg\varphi_l$ does not hold for more than k t.u. except after or in the interval witnessing $\overleftarrow{\varphi_r}$ *etc.*
- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k_{>c}\varphi_r$. We label a state (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies the formula: $\Psi_6 \overset{\text{def}}{=} \mathsf{A}(\llparenthesis z \leq c\rrparenthesis \wedge \overset{\leftarrow\cdot\cdot\shortmid}{\varphi_l})\mathsf{U}\big(\llparenthesis z > c\rrparenthesis \wedge \mathsf{AF}(\overleftarrow{\varphi_r}) \wedge \Psi_2'' \wedge \Psi_2'''\big)$
- $\Psi \overset{\text{def}}{=} \mathsf{A}\varphi_l\mathsf{U}^k_{=c}\varphi_r$. If $c \geq k$, we label a state (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ the following CTL-formula:

$$\Psi_7 \overset{\text{def}}{=} \mathsf{A}\overset{\leftarrow\cdot\cdot\shortmid}{\varphi_l}\mathsf{U}\llparenthesis z = c\rrparenthesis \ \wedge\ \underbrace{\neg\mathsf{E}\ \llparenthesis z < c\rrparenthesis\ \mathsf{U}\ \Big(\llparenthesis z = c\rrparenthesis \wedge \mathsf{E}\ \neg\overleftarrow{\varphi_r}\ \mathsf{U}\ \llparenthesis z > c+k\rrparenthesis\Big)}_{\Psi_8}$$

  The first term ensures that $\neg\varphi_l$ does not hold for a duration greater than k before the position $z = c$. And the formula Ψ_8 states that it is not possible to avoid $\overleftarrow{\varphi_r}$ between the position $z = c$ and the position $z > c + k$: thus any run has some interval (of duration greater than k) satisfying φ_r and containing a position located at duration c from the initial state.

  If $c < k$, then we label (q, γ) by Ψ iff $(q, \gamma[z, z_{\bar{l}}, z_r \leftarrow 0])$ satisfies Ψ_8.

This algorithm is correct:

Lemma 1 (Correctness of the labeling algorithm). *Given a TA A, a TCTL^{Δ} formula Φ and Ψ a subformula of Φ, the labeling algorithm labels (q,γ) with Ψ in $\mathcal{R}_{A,\Phi}$ iff $(q,v)\models\Psi$ for any $v\in\gamma$.*

Proof (sketch). The proof is done by induction over the formulae. We only deal with the modalities $\mathsf{E}\_\mathsf{U}^k_{\sim c}\_$ and $\mathsf{A}\_\mathsf{U}^k\_$.
First consider $\Psi=\mathsf{E}\varphi_l\mathsf{U}^k_{\sim c}\varphi_r$.
$\Rightarrow$ Assume that the procedure labels (q,γ) with Ψ. Then in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$, $(q,\gamma[z,z_l,z_r\leftarrow 0])$ satisfies $\Psi_1 \overset{\text{def}}{=} \mathsf{E}\,\overset{\leftarrow\cdot\cdot\dashv}{\varphi_l}\,\mathsf{U}\Big((\!|z\sim c|\!)\wedge(\textsf{after-a}\vee P_b\vee\overset{\leftarrow\cdot\cdot\dashv}{\varphi_l})\wedge\mathsf{E}\,\varphi_r\,\mathsf{U}\overleftarrow{\varphi_r}\Big)$ Thus there exists a path $\bar\rho$ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ leading to some (q',γ') satisfying the right-hand side of Ψ_1. From $\bar\rho$ one can build a run ρ in A from any (q,v) with $v\in\gamma$ (as it is done in the TCTL case). Before (q',γ'), the states along $\bar\rho$ verify $\overset{\leftarrow\cdot\cdot\dashv}{\varphi_l}$, that is $\varphi_l\vee(\!|z_{\bar l}\le k|\!)$: given the definition of $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ this means that the durations of the corresponding $(\neg\varphi_l)$-subruns in ρ are less than k. Finally the state (q',γ') is located at duration $\sim c$ from the initial state ($\bar\rho$ starts from a region where z is equal to 0) and from this point, it is possible to verify φ_r for some time ensuring that (q',γ') belongs to an φ_r-subrun of duration greater than k. This ensures that the corresponding run in A satisfies $\varphi_l\mathsf{U}^k_{\sim c}\varphi_r$.
$\Leftarrow$ Assume $(q,v)\models\Psi$. From the run ρ witnessing $\varphi_l\mathsf{U}^k_{\sim c}\varphi_r$, on can build in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ a path $\bar\rho$ from $(q,\gamma[z,z^l,z^r\leftarrow 0])$ leading to a position located at duration $\sim c$ (then $(\!|z\sim c|\!)$ holds) and belonging to a φ_r-subrun of duration greater than k: then $\mathsf{E}\varphi_r\mathsf{U}(\varphi_r\wedge(\!|z_r>k|\!))$ holds. Moreover since the run ρ contains no $(\neg\varphi_l)$-subrun of duration greater than k, the path $\bar\rho$ never goes through a region where $\neg\varphi_l\wedge(\!|z_{\bar l}>c|\!)$ is true. This gives the result.

Now consider the case $\Psi=\mathsf{A}\varphi_l\mathsf{U}^k\varphi_r$.
$\Rightarrow$ Assume (q,γ) is labeled by Ψ. Thus $(q,\gamma[z,z_l,z_r\leftarrow 0])$ satisfies $\Psi_2'\wedge\Psi_2''\wedge\Psi_2'''$. Let v be a valuation in γ. Any run ρ from (q,v) has a corresponding run $\bar\rho$ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$. From Ψ_2', we know that ρ has to contain an interval σ of duration greater than k satisfying φ_r.
Now Ψ_2'' states that before reaching σ, it is not possible to verify $\neg\varphi_l$ for a duration greater than k except if we have entered the interval σ witnessing $\overleftarrow{\varphi_r}$. In this last case, we also have to ensure that if the first region of σ is not a boundary region and if it has been reached via a delay transition, then it also has to satisfy $\overset{\leftarrow\cdot\cdot\dashv}{\varphi_l}$: this is done by the formula Ψ_2'''.
$\Leftarrow$ Assume $(q,v)\models\mathsf{A}\varphi_l\mathsf{U}^k\varphi_r$. We clearly have $(q,\gamma[z,z_l,z_r\leftarrow 0])\models\mathsf{AF}\overleftarrow{\varphi_r}$. Now assume $\neg\Psi_2''$ holds for $(q,\gamma[z,z_l,z_r\leftarrow 0])$. Then there exists a path $\bar\rho$ in $\mathcal{R}^{\varphi_l,\varphi_r}_{A,\Phi}$ satisfying $(\neg\overleftarrow{\varphi_r})\mathsf{U}(\overleftarrow{(\neg\varphi_l)}\wedge\neg\mathsf{A}\varphi_r\mathsf{U}\overleftarrow{\varphi_r})$. Thus the corresponding path ρ from (q,v) contains an interval σ' of duration greater than k where $\neg\varphi_l$ holds, and from σ' there is a run ρ' leading to some state satisfying $\neg\varphi_r$ before reaching the interval σ witnessing φ_r: the run $\rho\cdot\rho'$ does not satisfy $\varphi_l\mathsf{U}^k\varphi_r$ (σ' precedes strictly σ).
If $\neg\Psi_2'''$ holds for $(q,\gamma[z,z_l,z_r\leftarrow 0])$. Let (q',γ') be the region satisfying the right-hand side of the U, and let (q'',γ'') be its successor satisfying $\neg P_b\wedge\neg\overset{\leftarrow\cdots\cdot\dashv}{(\varphi_l)}$ along a path $\bar\rho$. The transition from (q',γ') to (q'',γ'') is a delay transition (the truth value of P_b goes from $\top$ to $\bot$). Moreover the corresponding run ρ from

(q, v) has to contain an interval σ witnessing $\overset{\longmapsfrom}{\varphi_r}$; in $\bar{\rho}$ this interval cannot be before (q', γ'), it is either after (q'', γ'') or it starts from (q'', γ''). Thus for any position p in σ along ρ, there will be states preceding p in the non-boundary region (q'', γ'') and since $\neg\overset{\leftarrow\cdot\cdot\shortmid}{\varphi_l}$ holds for this region, the formula $\varphi_l \mathsf{U}^k \varphi_r$ cannot hold for ρ. □

Finally we have:

Theorem 2 (Complexity of model checking). *Given a TA A and a TCTL^{Δ} formula Φ, deciding whether Φ holds for A is a PSPACE-complete problem.*

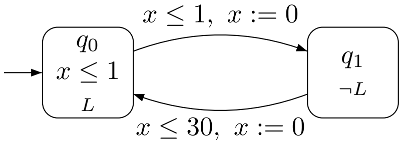
PSPACE-hardness comes from TCTL, and the PSPACE-membership can be obtained by using an on-the-fly algorithm over the region graph.

5 Undecidability Result for the Global Semantics

In this section we propose an alternative semantics for the logic, denoted by $\mathsf{TCTL}^{\Delta}_{\Sigma}$, which can also be viewed as an extension of $\mathsf{TCTL}^{\mathsf{ext}}$ [9]. Now we require that the sum of all delays during which the property does not hold is bounded by some constant. The syntax of $\mathsf{TCTL}^{\Delta}_{\Sigma}$ is the same as for TCTL^{Δ} but $\varphi \mathsf{U}^k_{\sim c} \psi$ is now interpreted as follows:

$$\begin{array}{rl} \rho \models \varphi \mathsf{U}^k_{\sim c} \psi \text{ iff } & \text{there exists a subrun } \sigma \text{, a position p} \in \sigma \text{ s.t} \\ & \mathsf{Time}(\rho^{\leq p}) \sim c \ \wedge \ \hat{\mu}(\sigma) > k \ \wedge \ \forall p' \in \sigma \ s_{p'} \models \psi \\ & \text{and } \ \hat{\mu}(\{p' \mid p' <_\rho p \ \wedge \ s_{p'} \not\models \varphi\}) \leq k \end{array}$$

Consider the "leaking gas burner" example, often used for verification with hybrid automata. As depicted by the TA below, the system can be in one of two modes, either leaking or not leaking, and it is initially leaking. Leakages are detected and stopped within 1 second and, once a leakage has been stopped, the burner is guaranteed not to leak again until at least 30 seconds later. The usual requirement for the gas burner states that, if at least 60 seconds have passed, then the gas burner has been leaking for less than one fifth of the total elapsed time. Using the atomic proposition L for the leaking mode, we can express this property in $\mathsf{TCTL}^{\Delta}_{\Sigma}$ by the formula: $\mathsf{AG}(\mathsf{A}(\neg L)\mathsf{U}^{12}_{\geq 60}\top)$: any period of duration greater than 60s has to include less than 12s of leaking.



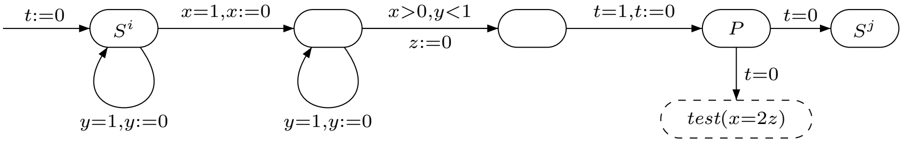
This problem is usually modeled with a stopwatch with respective slopes 1 in state q_0 and 0 in state q_1, in order to compute the leaking duration. But recall that model-checking is undecidable for hybrid automata. Moreover considering costs also makes verification undecidable (see for example the case of

WCTL [10]). However, we need to be careful because of some positive results: for instance in [3, 18, 7, 8, 11], some duration-bounded reachability problems are proved to be decidable. Indeed, this kind of results can be obtained when the cost variables are only used as observers. Our case is even simpler because there is only one slope which is equal to the rate of time and $\mathsf{TCTL}^{\Delta}_{\Sigma}$ is clearly less expressive than a logic like WCTL. For example, deciding the formula $\mathsf{E}P_1\mathsf{U}^k P_2$ – with $P_1, P_2 \in \mathsf{AP}$ – interpreted with the global semantics can easily be done by using the procedure to check the *duration bounded reachability* proposed in [3]; the technique can also be adapted to handle formulas like $\mathsf{E}P_1\mathsf{U}^k_{\leq c}P_2$. Unfortunately, we still have the following result:

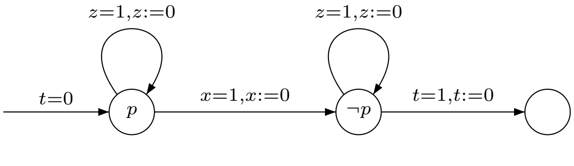
Theorem 3. *Model-checking* $\mathsf{TCTL}^{\Delta}_{\Sigma}$ *over timed automata is undecidable.*

The proof of this theorem consists in a reduction from the halting problem of a two-counter machine. The construction we present here is adapted from [10]. Let $\mathcal{M}$ be a two-counter machine. We build a timed automaton $\mathcal{A}_{\mathcal{M}}$ with initial location q_{init} and a $\mathsf{TCTL}^{\Delta}_{\Sigma}$ formula φ such that $\mathcal{M}$ halts iff $(q_{\text{init}}, v_{\text{init}}) \models \varphi$. The two counters c_1 and c_2 will be alternatively encoded by three clocks x, y and z. The value of c_1 and c_2 are encoded respectively by $h_1 = 1/2^{c_1}$ and $h_2 = 1/2^{c_2}$ with $h_1, h_2 \in \{x, y, z\}$. We use an extra clock t as a "tick".

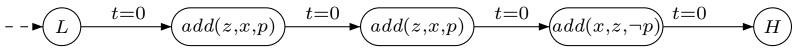
We first explain how to encode the incrementation of counter c_1 with the module on the next figure (it corresponds to instruction i, going to instruction j after the counter operation). We assume that $x = 1/2^{c_1}$ and $y = 1/2^{c_2}$ when this module is entered (which means that counter c_1 is encoded by x and counter c_2 by y).



In this module, because of the constraints, it is easy to check that the values of the clocks when arriving in state labeled by P (or similarly S^j) are $x = 1/2^{c_1}$, $y = 1/2^{c_2}$, and $z = \gamma$ where $\gamma \in [0, 1)$ depends on the time at which the transition labeled by "$x > 0, y < 1, z := 0$" is taken. The test module "*test*$(x = 2z)$" (described later) checks that γ is half the value of x, i.e. $\gamma = 1/2^{c_1+1}$ which will ensure that z correctly encodes the value of the first counter at the end of the incrementation instruction (whereas counter c_2 is correctly encoded by clock y). Before describing the test module *test*$(x = 2z)$, we present the timed automaton *add*(x, z, p) below:



In this automaton, if $\alpha \in [0,1]$ is the initial value of x when entering the module, then we stay $(1-\alpha)$ time units in the location labeled by atomic proposition p and α time units in the location labeled by $\neg p$.
Finally the test module "*test*$(x = 2z)$" is depicted below:



This test module has only one path which reaches the location labeled by H. If α and γ are the respective values of x and z on entering the module, this path will stay $2\cdot(1-\gamma)+\alpha$ time units in locations labeled by p and $2\cdot\gamma+(1-\alpha)$ time units in locations labeled by $\neg p$. Moreover the global time elapsed between L and H is exactly 3 time units. Thus, if formula $L \wedge \mathsf{E}(p\mathsf{U}^1 H) \wedge \mathsf{E}(\neg p\mathsf{U}^2 H)$ holds in state L, this will ensure that $2\cdot(1-\gamma)+\alpha \leq 2$ and $2\cdot\gamma+(1-\alpha) \leq 1$, which implies $2\cdot(1-\gamma)+\alpha = 2$ and $2\cdot\gamma+(1-\alpha) = 1$, thus $\gamma = \alpha/2$.
The simulation of a decrementation for a counter is very similar to the simulation of the incrementation, and we assume that we have constructed a module for every instruction (with the correct test module attached to state P, depending on what constraint we want to check) and that we have correctly glued the modules together. Then the formula that we want to check on the global automaton is $\mathsf{E}(\psi\mathsf{U}S^{\mathsf{Halt}})$ where ψ is equal to $P \Rightarrow \mathsf{E}\Big[(P \vee L)\mathsf{U}(L \wedge \mathsf{E}(p\mathsf{U}^1 H) \wedge \mathsf{E}(\neg p\mathsf{U}^2 H))\Big]$, which ensures that for each instruction we correctly store the value of the counter in the clocks. The correctness of the global reduction is a consequence of the previous discussion.

6 Conclusion

We have proposed an extension of TCTL in order to abstract transient events, where the notion of transient properties is parameterized by an integer k. We proved that model-checking for the new logic TCTL^{Δ} is still PSPACE-complete. We also proposed to interpret k-modalities with a global semantics but then we showed that model checking becomes undecidable. As future work, we plan to look for decidable fragments of $\mathsf{TCTL}^{\Delta}_{\Sigma}$, beyond the simple $\mathsf{E}P_1\mathsf{U}^{k}_{\leq c}P_2$.

References

1. R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
3. R. Alur, C. Courcoubetis, and T. A. Henzinger. Computing accumulated delays in real-time systems. *Formal Methods in System Design*, 11(2):137–156, 1997.
4. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

5. R. Alur, T. Feder, and Th. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.
6. R. Alur and Th. A. Henzinger. Logics and models of real-time: a survey. In *Real-Time: Theory in Practice, Proc. REX Workshop, Mook, NL, June 1991*, vol. 600 of *LNCS*, p. 74–106. Springer, 1992.
7. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *Proc. 4th Int. Workshop Hybrid Systems: Computation and Control (HSCC 2001), Roma, Italy, Mar. 2001*, vol. 2034 of *LNCS*, p. 49–62. Springer, 2001.
8. G. Behrmann, A. Fehnker, Th. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In *Proc. 4th Int. Workshop Hybrid Systems: Computation and Control (HSCC 2001), Roma, Italy, Mar. 2001*, vol. 2034 of *LNCS*, p. 147–161. Springer, 2001.
9. H. Belmokadem, B. Bérard, P. Bouyer, and F. Laroussinie. A new modality for almost everywhere propeties in timed automata. In *Proc. 16th International Conference on Concurrency Theory (CONCUR05)*, vol. 3653 of *LNCS*, p. 110–124. Springer, 2005.
10. P. Bouyer, T. Brihaye, and N. Markey. Improved Undecidability Results on Priced Timed Automata. *Information Processing Letters*, 98(5):188–194, 2006.
11. P. Bouyer, E. Brinksma, and K. G. Larsen. Staying alive as cheaply as possible. In *Proc. 7th Int. Workshop on Hybrid Systems: Computation and Control (HSCC 2004), Philadelphia, PA, USA, Mar. 2004*, vol. 2993 of *LNCS*, p. 203–218. Springer, 2004.
12. Th. Brihaye, V. Bruyère, and J.-F. Raskin. Model-checking for weighted timed automata. In *Proc. Joint Conf. Formal Modelling and Analysis of Timed Systems (FORMATS 2004) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT 2004), Grenoble, France, Sep. 2004*, vol. 3253 of *LNCS*, p. 277–292. Springer, 2004.
13. V. Bruyère, E. Dall'Olio, and J.-F. Raskin. Durations, parametric model-checking in timed automata with presburger arithmetic. In *Proc. 20th Ann. Symp. Theoretical Aspects of Computer Science (STACS 2003), Berlin, Germany, Feb. 2003*, vol. 2607 of *LNCS*, p. 687–698. Springer, 2003.
14. Z. Chaochen, C. Hoare, and A. Ravn. A calculus of duration. *Information Processing Letters*, 40(5):269–276, 1991.
15. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model-checker for hybrid systems. *Journal of Software Tools for Technology Transfer*, 1(1–2):110–122, 1997.
16. Th. A. Henzinger. The theory of hybrid automata. In *Proc. 11th IEEE Symp. Logic in Computer Science (LICS '96), New Brunswick, NJ, USA, July 1996*, p. 278–292. IEEE Comp. Soc. Press, 1996.
17. Th. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model-checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
18. Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Decidable integration graphs. *Information and Computation*, 150(2):209–243, 1999.
19. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
20. K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *Journal of Software Tools for Technology Transfer*, 1(1–2):134–152, 1997.
21. J. Ouaknine and J. Worrell. On the decidability of Metric Temporal Logic. In *Proc. 20th IEEE Symp. Logic in Computer Science (LICS 2005), Chicago, IL, USA, June 2005*, p. 188–197. IEEE Comp. Soc. Press, 2005.
22. S. Yovine. Kronos: A verification tool for real-time systems. *Journal of Software Tools for Technology Transfer*, 1(1–2):123–133, 1997.