A New Modality for Almost Everywhere Properties in Timed Automata*

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

[1] LSV, CNRS & ENS de Cachan
61 av. du Président Wilson, 94235 Cachan Cedex, France
Emails: {mokadem,bouyer,fl}@lsv.ens-cachan.fr
[2] LAMSADE, CNRS & Université Paris-Dauphine
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
Email: berard@lamsade.dauphine.fr

Abstract. The context of this study is timed temporal logics for timed automata. In this paper, we propose an extension of the classical logic TCTL with a new Until modality, called "Until almost everywhere". In the extended logic, it is possible, for instance, to express that a property is true at all positions of all runs, except on a negligible set of positions. Such properties are very convenient, for example in the framework of boolean program verification, where transitions result from changing variable values. We investigate the expressive power of this modality and in particular, we prove that it cannot be expressed with classical TCTL modalities. However, we show that model-checking the extended logic remains PSPACE-complete as for TCTL.

1 Introduction

Verification of timed temporal logic properties. Temporal logic provides a fundamental framework for formally specifying systems and reasoning about them. Furthermore, model-cheking techniques lead to the automatic verification that a finite-state model of a system satisfies some temporal logic specification. Since the introduction of timed automata [AD90,AD94] and timed logics like MITL, L_ν or TCTL [AH92,LLW95,AFH96], model-checking has been extended to real-time models [HNSY94] and analysis tools have been developped [DOTY96, HHWT95,LPY97] and successfully applied to numerous case studies.
Among these case studies, some examples concern the verification of programs which handle boolean or integer variables. The usual way to build a (possibly timed) model of the program consists in defining the discrete control states as tuples of variable values. The transitions are thus equipped with updates for the variables (and possibly time constraints). In such a model, a variable may change its value exactly upon leaving a control state and reaching another one, which gives an ambiguous semantics: a variable can have several different

* Work partially supported by the project VSMT of ENS de Cachan

values at a given time. This may lead to detect errors in the system, which are only due to the modeling phase. Such problems occur in the area of industrial automation, for the verification of Programmable Logic Controllers. In this case, programs are written from a set of languages described by the IEC-61131-3 specification [IEC93].

Example. Consider the SFC (Sequential Function Chart, one of the languages of the IEC standard) in Figure 1 below. It describes the control program of a device, designed to start some machine when two buttons (`L` and `R` for left and right button respectively) are pushed within 0.5 seconds. If only one button is pushed (then $\mathtt{L}+\mathtt{R}$ is true) and the 0.5 seconds delay is reached (time-out `Et` has occurred), then the whole process must be started again. After the machine has started, it stops as soon as one button is released, and it can start again only after both buttons have been released ($\overline{\mathtt{L}}.\overline{\mathtt{R}}$ is true).

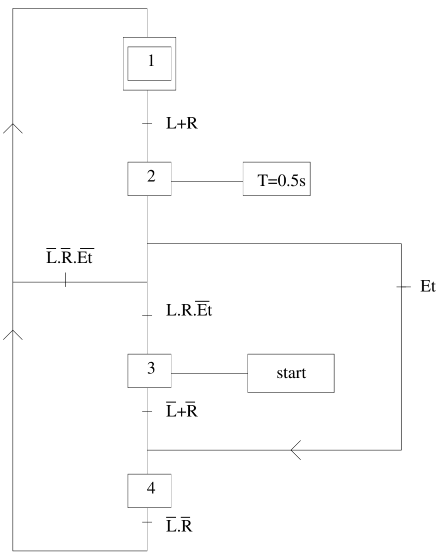


Fig. 1. SFC program for the two button machine

This device can be modeled with three timed automata (Figure 2), which communicate through the boolean variables `L` and `R`. The two automata for the buttons simply give arbitrary values in $\{0, 1\}$ to `L` and `R`, while the automaton for the control program is a straightforward translation of the SFC, with the only addition of an initialization step. The latter automaton handles a clock to measure the time interval of length 0.5. Note that some transitions must be urgent: for instance, the transition into state `running`, which sets the output variable `s` to 1, must be taken as soon as both buttons are pushed (if $t < 0.5$).

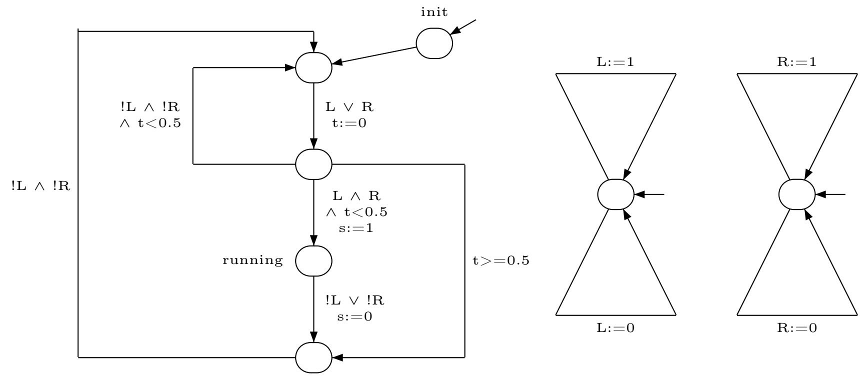


Fig. 2. Timed automata for the control program and the buttons

Consider now the following property: *it is always true that the machine has started only if both buttons have been pushed*, *i.e.* if `s=1` then `L=1` and `R=1`. This property does not hold because the automaton is still in state `running` when one of the buttons has been released, even if the transition into the next state will occur instantaneously afterward. What we should require instead is that this property be true *almost everywhere*, meaning that it could be false only on intervals with null duration.

A similar problem can occur when a sequence of transitions must be executed in an atomic way. To this purpose, a convenient feature was introduced in Uppaal: when a location of a timed automaton is labeled as *committed*, no time delay is permitted in this location and a new action transition has to be performed to leave this location. This mechanism is used in particular to obtain n-ary synchronization when only binary synchronization is possible. For example, the sequence $s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3$ executes atomically if location s_2 is committed. Like above, a given property may be true before s_1 and after s_3 but false in the intermediate location s_2 where the control stays for a null duration. Again in this case, a property true "almost everywhere" would be sufficient.

Some solutions. A basic method to solve the particular example of the "two buttons machine" described above would be to synchronize the update transitions of the L and R variables with the control transitions. This would amount to remove the variables in the model, introducing synchronizing channels instead. However, the resulting models do not faithfully represent the control program of the device, which receives the values of L and R by intermediate variables updated through sensors. Since the control program may later be translated into some other language of the standard (like Ladder Diagram), the model should remain as close as possible to the original specification.

A simple way of dealing with the general case consists in defining restricted semantics for timed automata, requiring that at most one configuration be associated with a given time. This holds for instance when only strictly increasing time sequences are permitted. However, when practical issues are considered, it is often useful to assume that several actions are executed in an atomic way (as described above for synchronization). Moreover, this hypothesis yields simpler models in the specification step and reduce the time needed for reachability analysis. Restricting the expressive power of a model is generally not a good idea. When such atomicity hypotheses are made, it is then possible to modify the property to be checked, requiring it to be true only in specified states where no ambiguity can occur. Such methods were used for instance in the verification with HyTech of the ABR protocol [BFKM03]. But this is an *ad-hoc* construction, where all the details of the system must be carefully investigated.

Finally, one could think of introducing an observer automaton. For example, to test if some atomic proposition a is true almost everywhere, such an automaton would move to an error state if it has stayed in $\neg a$ for a non null duration. However, it is well known that this method does not apply to full TCTL, but is restricted to a fragment expressing safety properties [ABBL03].

Contribution. In this paper, we propose a solution that does not depend on the model, which can thus remain as it was originally designed (often in a long and difficult process) for a given system. This solution consists in extending the syntax of the TCTL logic with an *almost everywhere* until modality U^{a}. We obtain for instance formulae like $\mathsf{AG}^{\mathsf{a}}\varphi$, meaning that property φ is true almost everywhere.

Section 2 recalls the main features of the timed automata model and gives definitions for the syntax and semantics of our extended logic. In Section 3, we investigate the expressive power of this extension, comparing it with TCTL. In particular, we prove that the modality U^{a} cannot be expressed with TCTL operators and conversely that U^{a} cannot express TCTL modalities. Finally, in the last section, we show that model-checking the extended logic $\mathsf{TCTL}^{\mathsf{ext}}$ is decidable by some labeling procedure, with the same complexity as TCTL.

2 Timed Automata and $\mathsf{TCTL}^{\mathsf{ext}}$

Let $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$ denote respectively the sets of natural and non-negative real numbers. Let X be a set of real valued clocks. The set of *valuations* is the set $\mathbb{R}_{\geq 0}^X$ of mappings from X to $\mathbb{R}_{\geq 0}$. 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 clock valuations. For every $v \in \mathbb{R}_{\geq 0}^X$ and $d \in \mathbb{R}_{\geq 0}$, we use $v + d$ to denote the time assignment which maps each clock $x \in X$ to the value $v(x) + d$. For a subset r of 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* $q \xrightarrow{g,r}_A q'$*).* $\mathsf{Inv}_A : Q_A \rightarrow \mathcal{C}(X)$ *assigns an* invariant *to each control state. Finally* $l_A : Q_A \rightarrow 2^{\mathsf{AP}}$ *labels every location with a subset of* AP.

A *configuration* of a TA A is a pair (q, v), where $q \in Q_A$ is the current location and $v \in \mathbb{R}^X_{\geq 0}$ 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 $(q, v + t)$ for some $t \in \mathbb{R}$ 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:

- $S = \{(q, v) \mid q \in Q_A \text{ and } v \in \mathbb{R}^X_{\geq 0} \text{ s.t. } v \models \mathsf{Inv}_A(q)\}$ 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 $q' = q$, $v' = v + t$ and $v + t' \models \mathsf{Inv}_A(q)$ for any $t' \leq t$. This is a delay transition, written $(q, v) \xrightarrow{t} (q, v + t)$,
  - or $\exists q \xrightarrow{g,r}_A q'$ and $v \models g$, $v' = v[r \leftarrow 0]$ and $v' \models \mathsf{Inv}_A(q')$. This is an action transition, written $(q, v) \rightarrow_a (q', v')$.
- $l : S \rightarrow 2^{\mathsf{AP}}$ labels every state (q, v) with the subset $l_A(q)$ of AP .

A 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 a run can always be described as an alternating infinite sequence $s_0 \xrightarrow{t_0}\rightarrow_a s_1 \xrightarrow{t_1}\rightarrow_a \cdots$ for some $t_i \in \mathbb{R}$. Such a run ρ goes through any configuration s' reachable from some s_i by a delay transition of duration $t \in [0, t_i]$. We write Exec(s) for the set of all runs starting from s. A configuration can occur several times along some run ρ. A particular occurrence p of a configuration is called a *position*, we write $p \in \rho$. For such a p, the corresponding configuration is denoted by s_p.
The standard notions of prefix, suffix and subrun apply for 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 $p \overset{\sigma}{\mapsto} p'$.
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 $\xrightarrow{0}$). Note that the set of positions along ρ is totally ordered by $<_\rho$, independently of the representation of the run.
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.

2.2 Definition of $\mathsf{TCTL}^{\mathsf{ext}}$.

We extend the syntax of TCTL to express that a formula holds almost everywhere: $\mathsf{TCTL}^{\mathsf{ext}}$ is obtained by adding the two modalities $\mathsf{E}\_\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\_$ and $\mathsf{A}\_\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\_$ to TCTL.

Definition 2 (Syntax of $\mathsf{TCTL}^{\mathsf{ext}}$). $\mathsf{TCTL}^{\mathsf{ext}}$ *formulae are given by the following grammar:*

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

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

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

$$\mathsf{EF}^{a}_{\sim c}\,\varphi \overset{def}{=} \mathsf{E}(\top\,\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\,\varphi) \qquad \mathsf{AF}^{\mathsf{a}}_{\sim\mathsf{c}}\,\varphi \overset{def}{=} \mathsf{A}(\top\,\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\,\varphi)$$
$$\mathsf{EG}^{\mathsf{a}}_{\sim\mathsf{c}}\,\varphi \overset{def}{=} \neg\mathsf{AF}^{\mathsf{a}}_{\sim\mathsf{c}}\neg\varphi \qquad \mathsf{AG}^{\mathsf{a}}_{\sim\mathsf{c}}\,\varphi \overset{def}{=} \neg\mathsf{EF}^{\mathsf{a}}_{\sim\mathsf{c}}\neg\varphi$$

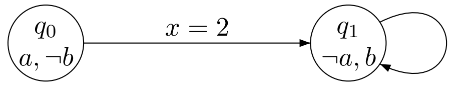
Definition 3 (Semantics of $\mathsf{TCTL}^{\mathsf{ext}}$). *The following clauses define when a state s of some TTS $\mathcal{T} = \langle S, s_{init}, \rightarrow, l\rangle$ satisfies a $\mathsf{TCTL}^{\mathsf{ext}}$ formula φ, written $s \models \varphi$, by induction over the structure of φ (the semantics of boolean operators is omitted).*

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

$$\begin{array}{ll}
\rho \models \varphi\mathsf{U}_{\sim c}\psi & \text{iff } \exists p \in \rho \text{ s.t. } \mathsf{Time}(\rho^{\leq p}) \sim c \;\wedge\; s_p \models \psi \;\wedge\; \forall p' <_\rho p,\; s_{p'} \models \varphi \\
\rho \models \varphi\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\psi & \text{iff there exists a subrun } \sigma \text{ s.t. } \hat{\mu}(\sigma) > 0, \exists p \in \sigma,\; \mathsf{Time}(\rho^{\leq p}) \sim c, \\
& \forall p' \in \sigma, s_{p'} \models \psi,\; \hat{\mu}(\{p' \mid p' <_\rho p \;\wedge\; s_{p'} \not\models \varphi\}) = 0
\end{array}$$

Note that in the case of the *almost* modality U^{a}, we ask that φ holds almost everywhere before ψ occurs. Moreover, we require that ψ holds not only at a single position (which has a measure equal to 0), like in the usual framework, but on a whole interval around the position satisfying the time constraint.

For example, $\mathsf{AG}^{a}_{\geq 0}\varphi$ specifies that along every run, the set of positions at which φ does not hold has a measure equal to 0, *i.e.* φ holds *almost everywhere* along all paths. It was precisely this kind of property we wanted to be able to express. Note that the positions where some formula φ does not hold are not restricted to discrete transitions, contrary to some intuition. Indeed, consider the automaton below, with two atomic propositions a and b, and the formula $\varphi = \mathsf{E}a\mathsf{U}_{=1}b$. Let ρ be the run starting in $(q_0, 0)$ -there is only one-. Clearly φ is not satisfied in each position of ρ except in $(q_0, 1)$, then $(q_0, 0) \not\models \mathsf{AG}(\neg\varphi)$ but $(q_0, 0) \models \mathsf{AG}^{a}(\neg\varphi)$.



The standard TCTL logic is the fragment of $\mathsf{TCTL}^{\mathsf{ext}}$ without $\mathsf{E\_U^a_{\sim c}\_}$ and $\mathsf{A\_U^a_{\sim c}\_}$, while the logic $\mathsf{TCTL^a}$ is the restriction of $\mathsf{TCTL}^{\mathsf{ext}}$ where classical $\mathsf{E\_U}_{\sim c}\mathsf{\_}$ and $\mathsf{A\_U}_{\sim c}\mathsf{\_}$ are forbidden.
The size $|\varphi|$ of a formula φ is defined in the standard way, with constants written in binary notation.

3 Expressiveness of $\mathsf{U^a}$ Modality

In this section we show that the modality $\mathsf{U^a}$ cannot be expressed with TCTL operators and conversely that $\mathsf{U^a}$ cannot express TCTL modalities.
Formally we say that two formulae φ and ψ are *equivalent* for a class of models C whenever their truth value is the same for any element of C, this is denoted $\varphi \overset{C}{\equiv} \psi$ or just $\varphi \equiv \psi$ when C is clear from the context. Let $\mathcal{L}$ and $\mathcal{L}'$ be two logical languages interpreted over the same models. $\mathcal{L}'$ is said to be *as expressive as* $\mathcal{L}$ (denoted $\mathcal{L} \preceq \mathcal{L}'$) iff for any formula $\varphi \in \mathcal{L}$ there exist $\varphi' \in \mathcal{L}'$ s.t. $\varphi \equiv \varphi'$. Moreover $\mathcal{L}'$ is *strictly more expressive* than $\mathcal{L}$ (written $\mathcal{L} \prec \mathcal{L}'$) iff $\mathcal{L} \preceq \mathcal{L}'$ and $\mathcal{L}' \not\preceq \mathcal{L}$.

3.1 $\mathsf{TCTL} \prec \mathsf{TCTL}^{\mathsf{ext}}$

First we show that $\mathsf{U^a}$ cannot be expressed with standard U modality. The proof is based on classical techniques used in untimed temporal logics (see for ex. [Eme91,EH86]). However, adapting them to the timed framework results in more involved constructions.
Let Ψ be the $\mathsf{TCTL^a}$ formula $\mathsf{E}(a\mathsf{U^a_{>0}}\mathsf{b})$. We will prove that there is no TCTL formula equivalent to Ψ. Consider the timed automata M_i and N_i with $i \geq 1$ in Figure 3. Clearly we have $M_i, (q_i, 0) \models \Psi$ while $N_i, (q'_i, 0) \not\models \Psi$. The next lemma states that M_i and N_i satisfy the same TCTL formula whose size is less than i.
We first introduce some notations. Given two configurations s and s', we write $s \equiv^k_{\mathsf{TCTL}} s'$ iff for any $\varphi \in \mathsf{TCTL}$ with $|\varphi| \leq k$, we have $s \models \varphi \Leftrightarrow s' \models \varphi$. We write $s \equiv_{\mathsf{TCTL}} s'$ iff $s \equiv^k_{\mathsf{TCTL}} s'$ for any $k \geq 1$.
Automata M_i and N_i contain only one clock, any configuration is then defined as a pair (ℓ, t) where ℓ is a location and $t \in \mathbb{R}_{\geq 0}$ is a value for x. Moreover the automata have only one cycle on r_0: for any configuration of the form (q_j, t), (q'_j, t), (r_j, t), or (r'_j, t) with $j \geq 1$, there is at most one such position along ρ.
Proof of expressiveness will be a consequence of the following Lemma:

Lemma 4. *Given the automata described in Figure 3, $\forall k \geq 1$, $\forall i \geq k$ and $\forall t \in \mathbb{R}$, we have:*

$$(q_i, t) \equiv^k_{\mathsf{TCTL}} (q'_i, t) \qquad\qquad (r_i, t) \equiv^k_{\mathsf{TCTL}} (r'_i, t)$$

Let ρ be a run starting in (q'_i, t) in N_i with $i > 0$. The run ρ is characterized by the time elapsed δ_0 in q'_i, the time elapsed δ_1 in r'_i and a suffix ρ_1 in N_{i-1} or M_{i-1}. Then ρ has the following structure:

$$(q'_i, t) \xrightarrow{\delta_0} (q'_i, \delta_0 + t) \rightarrow_a (r'_i, 0) \xrightarrow{\delta_1} (r'_i, \delta_1) \rightarrow_a \overset{\rho_1}{\mapsto}$$

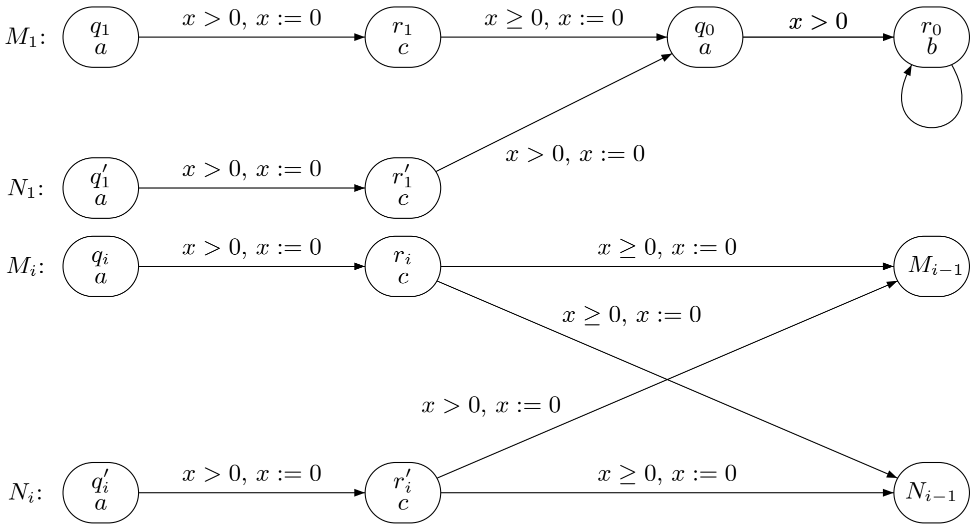


Fig. 3. Automata M_i and N_i, $i = 1, 2, \ldots$

Note that the suffix ρ_1 is in M_{i-1} only if $\delta_1 > 0$. Let $f_{M_i}(\rho)$ be the run of M_i defined by: $(q_i, t) \xrightarrow{\delta_0} (q_i, \delta_0 + t) \rightarrow_a (r_i, 0) \xrightarrow{\delta_1} (r_i, \delta_1) \rightarrow_a \overset{\rho_1}{\mapsto}$. The same can be done for a run issued from (r'_i, t), but in this case there is only the delay transition labeled by δ_1. Note that ρ and $f_{M_i}(\rho)$ share the same suffix ρ_1.
Given a run ρ in M_i from (q_i, t) or (r_i, t), one can also define a corresponding run $f_{N_i}(\rho)$ in N_i whenever the delay δ_1 spent in r_i is strictly positive.

Proof (of Lemma 4). The proof is done by induction over k, the size of formulae. First note that, given the guards and the resets on transitions of M_i and N_i, we clearly have for every $j \geq 0$ and locations $\ell \in \{q_j, r_j, q'_j, r'_j\}$

$$(r_j, 0) \equiv_{\mathsf{TCTL}} (r_j, t) \qquad \forall t > 0 \tag{1}$$

$$(\ell, t) \equiv_{\mathsf{TCTL}} (\ell, t') \qquad \forall t, t' > 0 \tag{2}$$

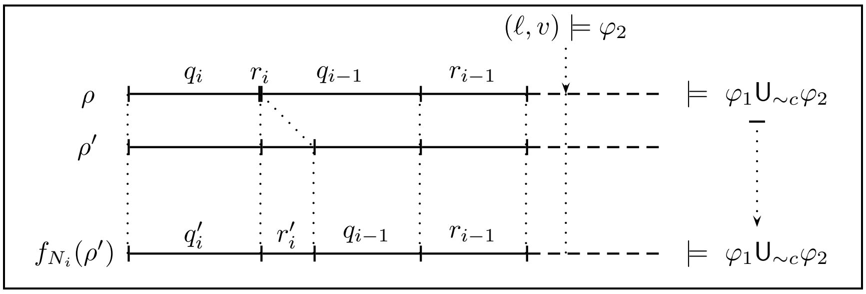
For formulae of size $k = 1$, the equivalences of the lemma hold because q_i and q'_i (resp. r_i and r'_i) are labeled by the same atomic propositions.
We assume now that $k > 1$ and that equivalences of the lemma hold for formulae with size $< k$. The case of boolean combinations is obvious, so we now concentrate on formulae $\mathsf{A}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$ and $\mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$.
From equivalences (1) and (2) and from induction hypothesis, if ρ is a run in N_i, then $f_{M_i}(\rho)$ exists and $\rho \models (\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \iff f_{M_i}(\rho) \models (\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$. Similarly, if ρ is a run in M_i and if $f_{N_i}(\rho)$ exists, then $\rho \models (\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \iff f_{N_i}(\rho) \models (\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$. Note that there exist some runs ρ in M_i for which there is no corresponding $f_{N_i}(\rho)$ (when there is no delay in location r_i).

We thus deduce immediately that

$$\begin{cases} (q_i,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \implies (q'_i,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \\ (r_i,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \implies (r'_i,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \\ (q'_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \implies (q_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \\ (r'_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \implies (r_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2) \end{cases}$$

To get all equivalences of Lemma 4, we need some extra work for several implications.

– Assume that $(q_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$ and take a run ρ from state (q_i,t) satisfying $\varphi_1 \mathsf{U}_{\sim c} \varphi_2$ with no corresponding run $f_{N_i}(\rho)$ (the delay in location r_i is thus 0). We note (ℓ,v) the position along ρ which satisfies φ_2 while all previous positions satisfy φ_1. If that position is before $(q_{i-1},0)$, then taking a run which starts with the prime version of the prefix of ρ ending in (ℓ,v), by induction hypothesis, we get a run which satisfies $\varphi_1 \mathsf{U}_{\sim c} \varphi_2$. Otherwise we need to change delays in ρ (to get a run ρ') as follows: on ρ, there is no delay in location r_i, we add one small delay in this state, small enough such that the run is unchanged after state r_{i-1} (the accumulated delays in states r_i and q_{i-1} in ρ' corresponds to the delay in q_{i-1} on run ρ, see the figure below) and such that if $\ell = q_{i-1}$ (in which case $v > 0$ by assumption), then the corresponding position on ρ' is some (q_{i-1},v') with $v' > 0$.



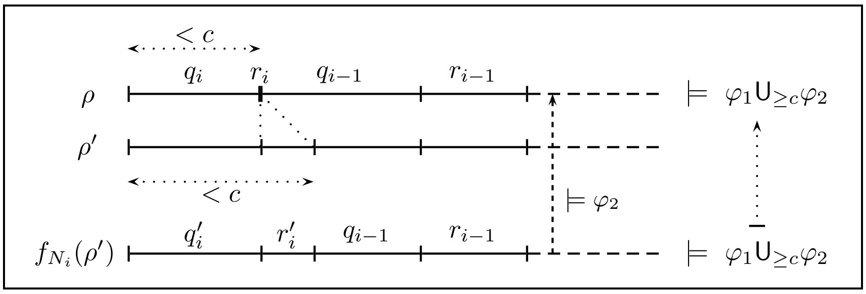
The run ρ' then satisfies $\varphi_1 \mathsf{U}_{\sim c} \varphi_2$: the position which corresponds to (ℓ,v) on ρ' also satisfies φ_2, and all previous positions satisfy φ_1 (using equivalences (1) and (2)). We thus get that $f_{N_i}(\rho')$ also satisfies $\varphi_1 \mathsf{U}_{\sim c} \varphi_2$. Thus, $(q'_i,t) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$.
A similar construction can be done to prove that $(r_i,0) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$ implies $(r'_i,0) \models \mathsf{E}(\varphi_1 \mathsf{U}_{\sim c} \varphi_2)$.

– For the formula $\mathsf{A}(\varphi_1 \mathsf{U}_{\prec c} \varphi_2)$ where $\prec$ is either $<$ or $\leq$ and $c > 0$, we consider a location $\ell \in \{q_i, r_i, q'_i, r'_i\}$. The following then holds:
  - if $t > 0$, $(\ell,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\prec c} \varphi_2)$ iff $(\ell,t) \models \varphi_2$ as we can take a run waiting at least c time units in location ℓ, and for some delay $d \prec c$, $(\ell, t+d)$ will have to satisfy φ_2 (which entails by (2) that (ℓ,t) must satisfy φ_2)
  - similarly $(\ell,0) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\prec c} \varphi_2)$ iff $(\ell,0) \models \varphi_2$ or $((\ell,0) \models \varphi_1$ and $(\ell,t) \models \varphi_2$ for every $t > 0)$

  Using induction hypothesis (on formulae φ_1 and φ_2), we get that $(\ell',t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\prec c} \varphi_2)$ implies $(\ell,t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\prec c} \varphi_2)$ if $\ell \in \{q_i, r_i\}$.

– We consider formula $\mathsf{A}(\varphi_1 \mathsf{U}_{=c} \varphi_2)$ with $c > 0$. Any reachable state from some (ℓ, t) can be reached in exactly c units of time and in strictly less than c units of time (because there is no real constraints on delays in states). This formula is then equivalent to $\varphi_1 \wedge \varphi_2$ over states (ℓ, t) with $\ell \in \{q_i, r_i, q'_i, r'_i\}$ and $t > 0$, and $(\ell, 0) \models \mathsf{A}(\varphi_1 \mathsf{U}_{=c} \varphi_2)$ iff $(\ell, 0) \models \varphi_1$ and all reachable states from $(\ell, 0)$ satisfy $\varphi_1 \wedge \varphi_2$ (ℓ is in $\{q_i, r_i, q'_i, r'_i\}$). Using induction hypothesis, we get that $(\ell', t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{=c} \varphi_2)$ implies $(\ell, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{=c} \varphi_2)$ for $\ell \in \{q_i, r_i\}$.

– We assume that $(q'_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$ and we want to prove that $(q_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$. We consider a run ρ in M_i starting in (q_i, t) such that $f_{N_i}(\rho)$ is not defined (the delay in state r_i is 0). We will construct a run in N_i from state (q'_i, t) "equivalent" to ρ, and distinguish two cases, depending on the delay δ in location q_i. We first consider the case where $\delta < c$.



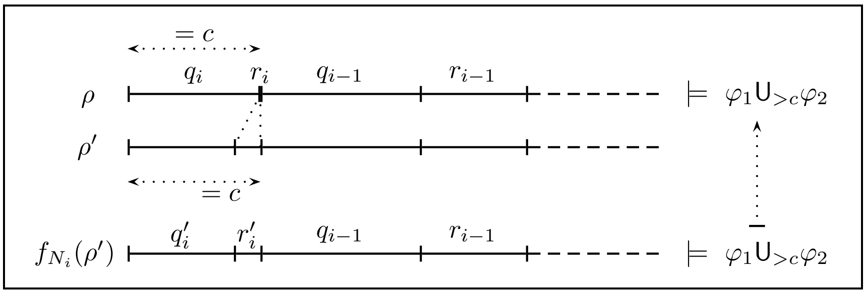
In ρ, the delay in q_i is $< c$ whereas the delay in r_i is null. We first construct a run ρ' with a positive delay in r_i (however smaller than the initial delay of ρ in state q_{i-1}) such that the accumulated delay in q_i and r_i is still $< c$ (see the figure above). From ρ' we construct run $f_{N_i}(\rho')$ in N_i. Using induction hypothesis, at all positions, the two runs ρ' and $f_{N_i}(\rho')$ agree on properties φ_1 and φ_2. As $(q'_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$, this implies that $f_{N_i}(\rho') \models \varphi_1 \mathsf{U}_{\geq c} \varphi_2$, and thus that $\rho' \models \varphi_1 \mathsf{U}_{\geq c} \varphi_2$. In particular, φ_1 has to hold in states (r_i, t) for every $t \geq 0$. Moreover, property φ_2 holds at some position along ρ', and φ_2 will also hold at the same position on ρ. We thus get that ρ also satisfies property $\varphi_1 \mathsf{U}_{\geq c} \varphi_2$.

We now assume that $\delta \geq c$. From ρ which does not delay in state r_i, we construct a run ρ' which waits a small amout of time (as in the previous case), and then consider the corresponding run $f_{N_i}(\rho')$ in N_i. By assumption, this runs satisfies $\varphi_1 \mathsf{U}_{\geq c} \varphi_2$. Then several cases can happen: (i) the property φ_2 holds in some $(q'_i, t+d)$ with $d \geq c$, in which case φ_2 also holds in $(q_i, t+d)$ by induction hypothesis, and φ_1 holds in all $(q_i, t+d')$ for $d' < d$ (also by induction hypothesis) which implies that $\rho \models \varphi_1 \mathsf{U}_{\geq c} \varphi_2$; (ii) the property holds in some (r'_i, d) for some $d \geq 0$, which implies that φ_2 also holds in (r_i, d) by i.h. and thus that $(r_i, 0) \models \varphi_2$ using (1), thus $\rho \models \varphi_1 \mathsf{U}_{\geq c} \varphi_2$; (iii) the property φ_2 holds for some other state (ℓ, d), which will be also true on run ρ, thus in that case also $\rho \models \varphi_1 \mathsf{U}_{\geq c} \varphi_2$.

In both cases we can conclude that $(q_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$.

Similar constructions can be done to prove that $(r'_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$ implies $(r_i, t) \models \mathsf{A}(\varphi_1 \mathsf{U}_{\geq c} \varphi_2)$.

– Formula $\mathsf{A}(\varphi_1 \mathsf{U}_{>c} \varphi_2)$ is almost handled in a similar way as $\mathsf{A}(\varphi \mathsf{U}_{\geq c} \varphi_2)$. Like before, we consider a run ρ in M_i which has no corresponding run $f_{N_i}(\rho)$. If δ is the delay in location q_i, we have also to distinguish three cases (instead of two): cases where $\delta < c$ or $\delta > c$ can be done exactly as previously. The only different case is when $\delta = c$. As previously we first construct a run ρ' which waits some positive delay in location r_i, and then consider run $f_{N_i}(\rho')$ which has to satisfy $\varphi_1 \mathsf{U}_{>c} \varphi_2$, and then using induction hypothesis we get that $\rho' \models \varphi_1 \mathsf{U}_{>c} \varphi_2$, from which we get that $\rho \models \varphi_1 \mathsf{U}_{>c} \varphi_2$ (using equivalences (1) and (2)). In that case, the delay in location q_i is shortened, and the accumulated delay in q_i and r_i (in run ρ') is precisely c, as seen in the figure below.



– It is easy to see that formula $\mathsf{A}(\varphi_1 \mathsf{U}_{=0} \varphi_2)$ is equivalent to φ_2 over states of M_i and N_i.

This concludes the proof of Lemma 4. □

Now we have the following result:

Theorem 5. $\mathsf{TCTL}^{\mathsf{ext}}$ *is strictly more expressive than* TCTL.

Proof. This is a consequence of Lemma 4: assume that there exists a TCTL formula Φ equivalent to formula $\mathsf{E}(a \mathsf{U}^{\mathsf{a}}_{>0} \mathsf{b})$. Then $(q_i, 0) \models \Phi$ and $(q'_i, 0) \not\models \Phi$ for any $i \geq 0$, but this contradicts $(q_i, 0) \equiv^{|\Phi|}_{\mathsf{TCTL}} (q'_i, 0)$ for any $i \geq |\Phi|$ provided by Lemma 4. □

3.2 $\mathsf{TCTL}^{\mathsf{a}} \prec \mathsf{TCTL}^{\mathsf{ext}}$

However, modality U^{a} is no help to express the classical U modality:

Theorem 6. $\mathsf{TCTL}^{\mathsf{ext}}$ *is strictly more expressive than* $\mathsf{TCTL}^{\mathsf{a}}$.

Proof. Let A be the automaton described in Figure 4.
It can be easily proven that (q_0, t) and (q'_0, t) agree on the same $\mathsf{TCTL}^{\mathsf{a}}$ formulae. Indeed the only difference is that the state $(r'_0, 0)$ belongs to any run from q'_0. But this state has to be left immediately and then this position has a measure null along any run and cannot have an effect on the truth value of $\mathsf{TCTL}^{\mathsf{a}}$ formulae. □

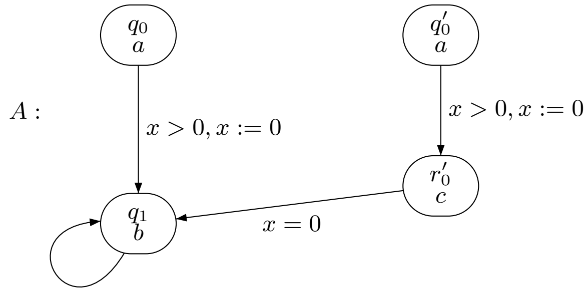


Fig. 4. $(q_0, 0) \models \mathsf{E}(a\mathsf{U}b)$, $(q'_0, 0) \not\models \mathsf{E}(a\mathsf{U}b)$, but $(q_0, 0) \equiv_{\mathsf{TCTL}^a} (q'_0, 0)$.

4 Model-Checking $\mathsf{TCTL}^{\mathsf{ext}}$

We now address the model-checking problem for $\mathsf{TCTL}^{\mathsf{ext}}$: given a TA A and a formula $\Phi \in \mathsf{TCTL}^{\mathsf{ext}}$, we want to decide whether Φ holds for A or not. The number of states of the TTS $\mathcal{T}_A$ is infinite, we then use the standard *region graph* technique introduced by Alur, Courcoubetis and Dill [ACD93] for TCTL model-checking. This method consists in defining an equivalence $\cong$ over clocks valuations such that (1) (q, v) and (q, v') satisfy the same formulae when $v \cong v'$, and (2) the quotient $\mathbb{R}_{\geq 0}^X/\cong$ is finite. Then model-checking TCTL reduces to model-checking a CTL-like logic over a (finite) abstracted graph. This technique can be extended to $\mathsf{TCTL}^{\mathsf{ext}}$ by using the same equivalence over valuations as the one used for TCTL.
Given A and some clock $x \in X$, we use $c_x \in \mathbb{N}$ to denote the maximal constant that x is compared with in the guards and invariants of A. Let $\cong$ be the following equivalence [AD90] over clocks valuations of $v, v' \in \mathbb{R}_{\geq 0}^X$: $v \cong v'$ iff (1) $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \vee (v(x) > c_x \wedge v'(x) > c_x)$ for any $x \in X$, and (2) for any $x, y \in X$ s.t. $v(x) \leq c_x$ and $v(y) \leq c_y$, we have: $\mathsf{frac}(v(x)) \leq \mathsf{frac}(v(y)) \Leftrightarrow \mathsf{frac}(v'(x)) \leq \mathsf{frac}(v'(y))$ and $\mathsf{frac}(v(x)) = 0 \Leftrightarrow \mathsf{frac}(v'(x)) = 0$. This equivalence is of finite index. An equivalence class of $\cong$ is called a *region* and $[v]$ denotes the class of v. Now we can show that this equivalence is consistent with the truth values of $\mathsf{TCTL}^{\mathsf{ext}}$ formulae:

Lemma 7. *Given a TA* $A = \langle X, Q_A, q_{init}, \rightarrow_A, \mathsf{Inv}_A, l_A \rangle$*,* $q \in Q_A$*, a formula* $\Phi \in \mathsf{TCTL}^{\mathsf{ext}}$ *and* $v, v' \in \mathbb{R}_{\geq 0}^X$ *s.t.* $v \cong v'$*, we have:* $(q, v) \models \Phi \;\Leftrightarrow\; (q, v') \models \Phi$*.*

Proof (sketch). The proof follows the same steps as the corresponding one for TCTL. First, given a run $\rho \in \mathrm{Exec}(q, v)$, we can build a run $\rho' \in \mathrm{Exec}(q, v')$ where the same action transitions are taken at “almost” the same times and where the regions visited for a duration strictly positive are the same. Let $\rho \in \mathrm{Exec}(q, v)$ be the run $(q_0, v_0) \xrightarrow{t_0}\rightarrow_a (q_1, v_1) \xrightarrow{t_1}\rightarrow_a \ldots$ with $q_0 = q$ and $v_0 = v$. Let $\delta_i = \sum_{j<i} t_j$ be the time at which the i-th action transition takes place, and $\delta_0 = 0$. Let v_i^* be the extended valuation over $X \cup \{\delta\}$ – where δ is a new symbol – defined by $v_i^*(x) = v_i(x)$ and $v_i^*(\delta) = \delta_i$. Now we consider the equivalence $\cong$ extended to

valuations over $X \cup \{\delta\}$ by assuming $c_\delta = \infty$. Like in [ACD93], we can build a run $\rho' \in \mathrm{Exec}(q, v')$ of the form $(q_0, v'_0) \xrightarrow{t'_0}\!\!\rightarrow_a (q_1, v'_1) \xrightarrow{t'_1}\!\!\rightarrow_a \ldots$ with $v'_0 = v'$ such that for any i we have: $v_i^* \cong v_i'^*$. This clearly entails that there is no strictly positive delay between the i-th and $(i+1)$-th action transitions in ρ iff there is no strictly positive delay between the i-th and $(i+1)$-th action transitions in ρ'.
We now prove the lemma by structural induction over the $\mathsf{TCTL}^{\mathsf{ext}}$ formulae. Since the property holds for TCTL formulae, we only have to consider the U^{a} modalities.
Assume $(q, v) \models \mathsf{E}\varphi\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\psi$ and assume that the truth value of φ and ψ are homogeneous over regions $(q, [u])$ (*i.e.* for any region γ, they hold for any valuation of γ, or for no valuation of γ). There exists some run $\rho \in \mathrm{Exec}(q, v)$ with a subrun σ s.t. : $\hat{\mu}(\sigma) > 0$, $\exists p \in \sigma$ s.t. $\mathsf{Time}(\rho^{\leq p}) \sim c$, $\forall p' \in \sigma$ we have $s_{p'} \models \psi$ and $\hat{\mu}(\{p' \mid p' <_\rho p \wedge s_{p'} \not\models \varphi\}) = 0$. Now consider a run ρ' corresponding to ρ as described above. Clearly there exists a subrun σ' in ρ' corresponding to the same regions as σ, and then these regions also satisfy ψ. Moreover, like for the TCTL case, there exists some position p' in ρ' s.t. $\mathsf{Time}(\rho'^{\leq p'}) \sim c \Leftrightarrow \mathsf{Time}(\rho^{\leq p}) \sim c$.
The set of positions $\{p' | p' <_\rho p \wedge s_{p'} \not\models \varphi\}$ corresponds to a set of regions along ρ where no time elapses. In ρ' the same regions are visited and no delay transition occur. Then this set will also have a null measure. Thus $(q, v') \models \mathsf{E}\varphi\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\psi$.
The same argument can be used for $\mathsf{A}\varphi\mathsf{U}^{\mathsf{a}}_{\sim\mathsf{c}}\psi$ because any run from (q, v) has a corresponding run from (q, v') and vice versa. $\square$

Given some region $\gamma \in \mathbb{R}_{\geq 0}^X/\cong$, the *successor* region of γ, when it exists, is the region distinct from γ s.t. for any $v \in \gamma$, there exists some $t \in \mathbb{R}_{\geq 0}$ s.t. $v + t \in \mathsf{Succ}(\gamma)$ and $v + t' \in \gamma \cup \mathsf{Succ}(\gamma)$ for any $0 \leq t' < t$. We will write $\gamma(x) \sim c$ when any valuation v in γ satisfies $v(x) \sim c$. Finally the region $\gamma[r \leftarrow 0]$ denotes the region $[v[r \leftarrow 0]]$ for any $v \in \gamma$.
Model-checking $\mathsf{TCTL}^{\mathsf{ext}}$ reduces to a model-checking problem for a CTL-like logic over a finite graph, called the *region graph*. Let X^* be the set of clocks $X \cup \{x_\Phi\}$. The new clock x_Φ is used to handle subscripts $\sim c$ in U modalities, the value c_{x_Φ} is the maximal constant occurring in a subscript. For any subscript $\sim c$ in Φ we add new atomic propositions $p_{<c}$, $p_{>c}$ and $p_{=c}$, that hold for regions γ s.t. $\gamma(x_\Phi) \sim c$. Let p_b be another proposition that holds for *boundary regions*: $\gamma \models p_b$ iff there is some clock $x \in X^*$ with $\mathsf{frac}(x) = 0$ in γ. Let $\mathsf{AP}^+ = \mathsf{AP} \cup \{p_b, p_{<c}, \ldots\}$ be the extended set of atomic propositions.
We can now recall the region graph of [ACD93]: For a TA $A = \langle X, Q_A, q_{\mathrm{init}}, \rightarrow_A, \mathsf{Inv}_A, l_A\rangle$ and a $\mathsf{TCTL}^{\mathsf{ext}}$ formula Φ, the region graph $\mathcal{R}_{A,\Phi}$ is the finite fair graph $(V, \rightarrow, l, F)$ with:

- $V = \{(q, \gamma) \mid q \in Q_A \text{ and } \gamma \in \mathbb{R}_{\geq 0}^{X^*}/\cong\}$
- The set of transitions $\rightarrow = \rightarrow_t \cup \rightarrow_a$ contains two kinds of transitions:
  - $(q, \gamma) \rightarrow_t (q, \mathsf{Succ}(\gamma))$ if $\mathsf{Succ}(\gamma) \models \mathsf{Inv}_A(q)$.
  - $(q, \gamma) \rightarrow_a (q, \gamma')$ s.t. there exists $q \xrightarrow{g,r}_A q'$ with $\gamma \models g$, $\gamma' = \gamma[r \leftarrow 0]$ and $\gamma' \models \mathsf{Inv}_A(q')$.
- $l : V \rightarrow 2^{\mathsf{AP}^+}$ labels the vertices with the atomic propositions it satisfies: $l(q, \gamma)$ contains $l_A(q)$ and the propositions for γ.

– F is a set of fairness constraints: $F = \{F_x | x \in X^*\}$ with $F_x = \{(q,\gamma) | \gamma(x) = 0 \vee \gamma(x) > c_x\}$. A fair path in $\mathcal{R}_{A,\Phi}$ has to visit infinitely often a configuration in F_x for any $x \in X^*$.

We now define $\mathcal{R}^+_{A,\Phi}$ an extension of $\mathcal{R}_{A,\Phi}$ where we consider the transitive closure of $\rightarrow_a$: $\mathcal{R}^+_{A,\Phi} = (V, \rightarrow, l, F)$ where V, l and F are defined as for $\mathcal{R}_{A,\Phi}$, and $\rightarrow = \rightarrow_t \cup \rightarrow_a^+$. Then an action transition in $\mathcal{R}^+_{A,\Phi}$ $(q,\gamma) \rightarrow_a^+ (q',\gamma')$ corresponds to a sequence of action transitions in A which can be performed with no delay in between. Note that all the intermediate configurations along such a sequence are visited but the set of their positions is of measure 0 w.r.t. $\hat{\mu}$. We call these configurations *transient* configurations, and more formally, a configuration along a run ρ is non-transient iff its region is non-boundary and the previous or the next transition on ρ is a delay transition (a strictly positive delay has to elapse in the state along ρ). We will use this extended region graph when looking for the existence of a run satisfying $\varphi \mathsf{U}^{\mathsf{a}}_{\sim \mathsf{c}} \psi$ because we do not need to consider such intermediate transient configuration.

We reduce model-checking $\mathsf{TCTL}^{\mathsf{ext}}$ to model-checking CTL over $\mathcal{R}^+_{A,\Phi}$. We will use the classical $\mathsf{E\_U\_}$ and $\mathsf{A\_U\_}$ operators where E and A deal with paths in $\mathcal{R}_{A,\Phi}$, whereas E^+ and A^+ deal with paths in $\mathcal{R}^+_{A,\Phi}$, that is when transitions corresponding to transitive closure of action transitions in $\mathcal{R}_{A,\Phi}$ are allowed. Finally we also assume that for any state (q,γ) of $\mathcal{R}_{A,\Phi}$, there is a fair path rooted at (q,γ).

It remains to describe a labeling procedure to label every state of $\mathcal{R}_{A,\Phi}$ with the Φ-subformulae it satisfies. This is done by adapting the procedure for the TCTL case [ACD93], using the graphs $\mathcal{R}_{A,\Phi}$ and $\mathcal{R}^+_{A,\Phi}$. For example, in the case of formula $\mathsf{EG}^{\mathsf{a}}_{\leq \mathsf{c}} \varphi$, a state (q,γ) is labeled by $\mathsf{EG}^{\mathsf{a}}_{\leq \mathsf{c}} \varphi$ iff $(q, \gamma[x_\Phi \leftarrow 0])$ satisfies the CTL formula:

$$\mathsf{E}^+(p_b \vee \varphi)\mathsf{U}\Big(p_{=c} \wedge \big((\varphi \wedge \mathsf{EX} p_{>c}) \vee \mathsf{EX}(p_{>c} \wedge \varphi)\big)\Big)$$

where the next operator (EX) ensures that the position for which the right-hand side of the Until has to hold, is the last position at duration $= c$ along a run.
This leads to the following result:

Theorem 8. *Given a TA A and a* $\mathsf{TCTL}^{\mathsf{ext}}$ *formula Φ, deciding whether Φ holds for A is a PSPACE-complete problem.*

5 Conclusion

In this work, we studied the extension $\mathsf{TCTL}^{\mathsf{ext}}$ of the classical logic TCTL, obtained by introducing a new modality $\mathsf{U}^{\mathsf{a}}_{\sim \mathsf{c}}$. The superscript a means "almost everywhere" and expresses the fact that a property must be true except on a negligible set of positions. We proved that this modality cannot be expressed by the classical ones, and conversely. We also proposed a model-checking procedure for $\mathsf{TCTL}^{\mathsf{ext}}$, with the same complexity result than TCTL, where the classical constructions must be adapted to take into account the set of negligible positions on a run. Further work could consist in extending this new modality for the verification of "permanent" properties, i.e. properties that hold on an sufficiently large interval, the length of which could be a parameter.

References

[ABBL03] Luca Aceto, Patricia Bouyer, Augusto Burgueño, and Kim G. Larsen. The power of reachability testing for timed automata. *Theoretical Computer Science*, 300(1–3):411–475, 2003.

[ACD93] Rajeev Alur, Costas Courcoubetis, and David Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

[AD90] Rajeev Alur and David Dill. Automata for modeling real-time systems. In *Proc. 17th International Colloquium on Automata, Languages and Programming (ICALP'90)*, volume 443 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 1990.

[AD94] Rajeev Alur and David Dill. A theory of timed automata. *Theoretical Computer Science (TCS)*, 126(2):183–235, 1994.

[AFH96] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. The benefits of relaxing punctuality. *Journal of the Association for Computing Machinery (JACM)*, 43(1):116–146, 1996.

[AH92] Rajeev Alur and Thomas A. Henzinger. Logics and models of real-time: a survey. In *Real-Time: Theory in Practice, Proc. REX Workshop 1991*, volume 600 of *Lecture Notes in Computer Science*, pages 74–106. Springer, 1992.

[BFKM03] Béatrice Bérard, Laurent Fribourg, Francis Klay, and Jean-François Monin. A compared study of two correctness proofs for the standardized algorithm of abr conformance. *Formal Methods in System Design*, 22(1):59–86, 2003.

[DOTY96] Conrado Daws, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. The tool KRONOS. In *Proc. Hybrid Systems III: Verification and Control (1995)*, volume 1066 of *Lecture Notes in Computer Science*, pages 208–219. Springer, 1996.

[EH86] E. Allen Emerson and Joseph Y. Halpern. "Sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

[Eme91] E. Allen Emerson. *Temporal and Modal Logic*, volume B (Formal Models and Semantics) of *Handbook of Theoretical Computer Science*, pages 995–1072. MIT Press Cambridge, 1991.

[HHWT95] Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HyTech: the next generation. In *Proc. 16th IEEE Real-Time Systems Symposium (RTSS'95)*, pages 56–65. IEEE Computer Society Press, 1995.

[HNSY94] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic model-checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.

[IEC93] IEC (International Electrotechnical Commission). *IEC Standard 61131-3: Programmable controllers - Part 3*, 1993.

[LLW95] François Laroussinie, Kim G. Larsen, and Carsten Weise. From timed automata to logic – and back. In *Proc. 20th International Symposium on Mathematical Foundations of Computer Science (MFCS'95)*, volume 969 of *Lecture Notes in Computer Science*, pages 529–539. Springer, 1995.

[LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *Journal of Software Tools for Technology Transfer (STTT)*, 1(1–2):134–152, 1997.