Temporal Logic + Timed Automata: Expressiveness and Decidability

Ahmed Bouajjani[1] Yassine Lakhnech[2*]

[1] Verimag, Miniparc-Zirst, Rue Lavoisier, 38330 Montbonnot St-Martin, France.
email: Ahmed.Bouajjani@imag.fr

[2] Institut für Informatik und Praktische Mathematematik Christian-Albrechts-Universität zu Kiel, Preußerstr. 1-9, D-24105 Kiel, Germany.
email: yl@informatik.uni-kiel.d400.de

Abstract

We investigate the verification problem of timed automata w.r.t. linear-time dense time temporal logics. We propose a real-time logic TATL which extends the propositional temporal logic with constraints involving timed automata. These constraints allow to require that the computation segment since some designated point is accepted by a given timed automaton. We identify an expressively powerful fragment of this logic, called TATL^{+} for which the verification problem is decidable. To establish this result, we define a fragment of TATL, called $\mathrm{TATL}^{\times}$, such that the negation of any formula in TATL^{+} is equivalent to a $\mathrm{TATL}^{\times}$ formula, and then, we prove that every $\mathrm{TATL}^{\times}$ formula can be characterized by a nondeterministic timed Büchi automaton.

1 Introduction

Real-time temporal logics are widely accepted as suitable specification formalisms for systems with hard timing constraints [5, 7, 12, 11, 2, 8, 4]. These logics are interpreted on either discrete or dense time models.

Several works investigate the verification problem (model-checking) of real-time temporal logics, i.e., whether all the computations (trajectories in the dense-time case) of a timed system satisfy some property expressed by a formula in these logics. The first decidability results have been obtained for discrete time logics like TPTL [5]. This logic is obtained by extending the linear-time propositional temporal logic PTL [13] with the so-called *reset* quantification allowing to introduce clocks that are used in timing constraints (comparisons with natural constants). For instance, the formula

$$\Box(P \Rightarrow x.\Diamond(Q \wedge x \leq 5)) \tag{1}$$

expresses the bounded response property saying that *whenever P holds, a state satisfying Q will be eventually reached within* 5 *time units.*

Unfortunately, the logic TPTL is undecidable for dense-time models. Actually, there are mainly two dense-time temporal logics for which the verification problem has been shown to be decidable. The first one is the branching-time logic

* This work has been performed while this author was visiting Verimag.

TCTL [2, 1, 10] obtained by extending the logic CTL [6] using the reset quantification and timing constraints in a similar way as for TPTL. This decidability result is based on the fact that TCTL induces a time-abstract bisimulation on timed automata that allows to solve the verification problem by reasoning on a finite-state region graph. The second logic is the linear-time logic MITL [4] which can be seen as the fragment of dense-time TPTL obtained by imposing two restrictions: One of these restrictions is that, between the occurrence of each clock x in some constraint and its binding ("$x.$"), there is exactly one temporal operator, i.e., one until ($\mathcal{U}$) operator. The other restriction is that punctual constraints, i.e., equalities as $x = 5$, are forbidden. Recently, the logic MITL has been extended in [15] by using automata operators instead of the until operator; these automata operators are (untimed) finite-state automata on finite sequences. The obtained logic, called EMITL, is shown to be also decidable.

In this paper, we adopt a different approach to get a temporal logic with a decidable verification problem. We introduce a linear-time temporal logic which is an extension of dense-time TPTL by using timed automata instead of timing constraints. Then, we identify a fragment of this logic for which the verification problem is decidable and where general reset quantification as well as punctual constraints are allowed.

The logic we define, called Timed Automata Temporal Logic (TATL), is an extension of PTL by a *position quantification* allowing to relate *position variables* with points in trajectories. Then, *automata constraints* can be used to express the fact that the trajectory segment since the point associated with some position variable is accepted by some given timed automaton. For instance, the TPTL formula (1) can be written in TATL as $\Box(P \Rightarrow u.\Diamond(Q \wedge A^u_{\leq 5}))$ where $A_{\leq 5}$ recognizes the set of bounded trajectories of length less or equal than 5.

While constraints in TPTL are simple timing constraints on trajectory segments, automata constraints in TATL allow to express finer conditions combining constraints on the order of occurrence of state properties with timing constraints between these occurrences.

Clearly, the verification problem of TATL is undecidable since it is already undecidable for dense-time TPTL [5, 9]. Then, we define a syntactical fragment of TATL, called TATL$^+$, and show that its verification problem is decidable. This fragment covers a wide class of interesting real-time properties like bounded invariance and bounded response properties. We prove that it is strictly more expressive than deterministic timed Muller automata. Moreover, concerning untimed properties, we show that TATL$^+$ can express any stuttering-closed ω-regular property.

To prove the decidability of the verification problem of TATL$^+$, we introduce another fragment of TATL, called TATL$^\times$, such that the negation of each TATL$^+$ formula can be transformed into an equivalent formula in TATL$^\times$. Then, we prove that for every TATL$^\times$ formula φ, we can construct a nondeterministic timed Büchi automaton which recognizes the set of trajectories satisfying φ.

The remainder of this paper is organized as follows. In Section 2, we introduce some basic notations and definitions. In Section 3, we define timed automata on bounded or unbounded trajectories. In Section 4, we introduce the logic TATL and its fragments TATL$^+$ and TATL$^\times$. In Section 5, we study the expressiveness of TATL$^+$ and TATL$^\times$ and compare them with timed automata. Section 6 presents decidability results, and concluding remarks are given in Section 7.

2 Preliminaries

2.1 Timing Constraints

Let $\mathcal{V}$ be a set of variables, called *clocks*, ranging over positive reals ($\mathbb{R}_{\geq 0}$). A *timing constraint* over $\mathcal{V}$ is a boolean combination of constraints of the form $x \prec c$ where $x \in \mathcal{V}$, $\prec \in \{<, \leq\}$, and c is a natural constant ($c \in \mathbb{N}$). The symbols $<$ and $\leq$ represent the usual (strict and nonstrict) ordering relations over $\mathbb{R}_{\geq 0}$. Let $\mathcal{C}_{\mathcal{V}}$ be the set of timing constraints over $\mathcal{V}$.

A *valuation* over $\mathcal{V}$ is a function in $[\mathcal{V} \rightarrow \mathbb{R}_{\geq 0}]$. A satisfaction relation is defined as usual between valuations and constraints. Given a valuation ν and $f \in \mathcal{C}_{\mathcal{V}}$, we denote by $\nu \models f$ the fact that ν satisfies f.

We denote by $\mathbf{0}$ the valuation that associates with each clock the value 0. Given a valuation ν and a set of variables $X \subseteq \mathcal{V}$, we denote by $\nu[X \mapsto 0]$ the valuation which associates with each variable in X the value 0, and coincides with ν on all the other variables. For every $t \in \mathbb{R}_{\geq 0}$, we denote by $\nu + t$ the valuation ν' such that $\forall x \in \mathcal{V}$, $\nu'(x) = \nu(x) + t$.

2.2 States, Interval Sequences, and Trajectories

Let $\mathcal{P}$ be a finite set of atomic propositions. A *state* is a subset of $\mathcal{P}$. Let Σ be the set of all possible states, i.e., $\Sigma = 2^{\mathcal{P}}$.

Let I be a nonempty subinterval of $\mathbb{R}_{\geq 0}$. Then, we denote respectively by $lb(I)$ and $ub(I)$ the lower and upper bounds of I. We denote also by $\ell(I)$ the length of the interval I, i.e., $\ell(I) = ub(I) - lb(I)$. Two nonempty intervals I_1 and I_2 are adjacent if $I_1 \cap I_2 = \emptyset$ and $lb(I_2) = ub(I_1)$.

Let I be a left-closed nonempty subinterval of $\mathbb{R}_{\geq 0}$ such that $lb(I) = 0$. An *interval sequence* over I is a finite or infinite sequence of nonempty intervals $\vec{I} = I_0 I_1 \cdots I_i \cdots$ such that

- $\forall i < |\vec{I}|$, the two intervals I_i and I_{i+1} are adjacent,
- $\forall t \in I.\ \exists i < |\vec{I}|.\ t \in I_i$,
- If I is bounded (resp. unbounded), the sequence $\vec{I}$ is finite (resp. infinite).

Now, consider a mapping $\tau : I \rightarrow \Sigma$. We say that τ is *finite-variable* if in every finite subinterval of I, τ changes its value a finite number of times (τ has a finite number of discontinuity points). Then, we call *trajectory* over I any mapping $\tau : I \rightarrow \Sigma$ which is finite-variable. We say that a trajectory $\tau : I \rightarrow \Sigma$ is *bounded* (resp. *unbounded*) if the interval I is bounded (resp. unbounded).

It can be seen that for every trajectory τ, there exists an interval sequence $\vec{I} = I_0 I_1 \cdots I_i \cdots$ such that $\forall i < |\vec{I}|.\ \forall t, t' \in I_i.\ \tau(t) = \tau(t')$, which means that τ is piecewise constant. Given a trajectory τ over an interval I, we denote by $\mathcal{D}(\tau)$ the set of infinite sequences $\sigma = s_0 s_1 \cdots \in \Sigma^\omega$ such that there exists an interval sequence $\vec{I}$ over I satisfying $\forall i \in \omega.\ \forall t \in I_i.\ \tau(t) = s_i$. This notation is generalized in the obvious way to sets of trajectories.

Let $\tau : I \longrightarrow \Sigma$ be a trajectory, and I' a bounded closed nonempty subinterval of I. Then, we denote by $\tau(I')$ the bounded trajectory $\tau' : [0, \ell(I')] \longrightarrow \Sigma$ such that $\forall t \in [0, \ell(I')].\ \tau'(t) = \tau(t + lb(I'))$.

3 Timed Automata

We introduce timed automata on either bounded or unbounded trajectories. These automata are similar to those used in [4].

3.1 Timed Transition Tables

Definition Recall that $\Sigma = 2^{\mathcal{P}}$ where $\mathcal{P}$ is a finite set of atomic propositions. A *timed transition table* $\mathcal{T}$ over Σ consists of the following components:

- $\mathcal{X}$, a finite set of clocks,
- $\mathcal{Q}$, a finite set of control locations,
- $\mathcal{I}nit$, a set of initial control locations ($\mathcal{I}nit \subseteq \mathcal{Q}$),
- $\mathcal{E}$, a set of edges. Each edge is a triplet $E = (q, X, q')$ where $q, q' \in \mathcal{Q}$ are the source and target locations, and $X \subseteq \mathcal{X}$ is the set of reset variables,
- Π, a function in $[\mathcal{Q} \longrightarrow \Sigma]$, associating a state with each control location,
- Γ, a function in $[\mathcal{Q} \longrightarrow \mathcal{C}_{\mathcal{X}}]$, associating with each control location q a timing constraint under which the computation is allowed to stay in q.

Runs, Paths, and Generated Trajectories Let I be a left-closed subinterval of $\mathbb{R}_{\geq 0}$ such that $lb(I) = 0$. Then, a *run* of $\mathcal{T}$ over the interval I is a finite or infinite sequence $\rho = \langle q_0, \nu_0, I_0\rangle\langle q_1, \nu_1, I_1\rangle \cdots \langle q_i, \nu_i, I_i\rangle \cdots$ such that

- $q_0 \in \mathcal{I}nit$, $\nu_0 = \mathbf{0}$, and
- $I_0 I_1 \cdots I_i \cdots$ is an interval sequence over I, and
- $\forall i < |\rho|.\ \forall t \in I_i.\ \nu_i + (t - lb(I_i)) \models \Gamma(q_i)$, and
- $\forall i < |\rho|$.
  - either $q_i = q_{i+1}$ and $\nu_{i+1} = \nu_i + \ell(I_i)$,
  - or $\exists (q_i, X, q_{i+1}) \in \mathcal{E}.\ \nu_{i+1} = (\nu_i + \ell(I_i))[X \mapsto 0]$.

With every run $\rho = \langle q_0, \nu_0, I_0\rangle\langle q_1, \nu_1, I_1\rangle \cdots \langle q_i, \nu_i, I_i\rangle \cdots$ defined as above, we associate a *path* of $\mathcal{T}$ over I which is a mapping $\theta : I \longrightarrow \mathcal{Q}$ such that $\forall i < |\rho|.\ \forall t \in I_i.\ \theta(t) = q_i$. Then, the *trajectory generated* by ρ is defined as the mapping $\tau = \Pi \circ \theta$.

Standard form We say that a timed transition table is in *standard form* if for every control location q, the condition $\Gamma(q)$ is of the form

$$\bigwedge_{x \in \mathcal{X}} (a_x \prec_1^x x \prec_2^x b_x)$$

where the $\prec_j^x$'s are in $\{<, \leq\}$, the a_x's in $\mathbb{N}$, and the b_x's in $\mathbb{N} \cup \{+\infty\}$.

It can be verified that every timed transition table can be transformed into another one which is in standard form and generates the same set of trajectories. From now on, we only consider timed transition tables in standard form.

Deterministic Timed Transition Tables Let $\mathcal{T}$ be a timed transition table. Given q a control location of $\mathcal{T}$ with $\Gamma(q) = \bigwedge_{x\in\mathcal{X}}(a_x \prec_1^x x \prec_2^x b_x)$, and $X \subseteq \mathcal{X}$, we denote by $\Gamma_X(q)$ the constraint $\bigwedge_{x\in X}(a_x \prec_1^x x \prec_2^x b_x)$. Then, we say that $\mathcal{T}$ is *deterministic* if

- $\forall q, q' \in \mathcal{I}nit.\ \Pi(q) = \Pi(q')$ implies that $\Gamma(q) \wedge \Gamma(q')$ is equivalent to $false$, and moreover, $\forall s \in \Sigma$, the union of the conditions $\Gamma(q)$, for all the control locations $q \in \mathcal{I}nit$ such that $\Pi(q) = s$, is equivalent to $true$,
- $\forall \langle q, X, q'\rangle \in \mathcal{E}$, if $\Pi(q) = \Pi(q')$ then either $\Gamma_{\mathcal{X}\setminus X}(q) \wedge \Gamma_{\mathcal{X}\setminus X}(q')$ is equivalent to $false$, or $\exists x \in X$ such that $\Gamma_{\{x\}}(q)$ is equivalent to $x = c$, for some constant $c \in \mathbb{N}$,
- $\forall \langle q, X_1, q_1\rangle, \langle q, X_2, q_2\rangle \in \mathcal{E}$, if $\Pi(q_1) = \Pi(q_2)$ then $\Gamma_{\mathcal{X}\setminus X_1}(q_1) \wedge \Gamma_{\mathcal{X}\setminus X_2}(q_2)$ is equivalent to $false$,
- $\forall q \in \mathcal{Q}$, $\forall X \subseteq \mathcal{X}$, $\forall s \in \Sigma$ such that $s \neq \Pi(q)$, the union of the conditions $\Gamma(q')$, for all the edges $\langle q, X, q'\rangle$ with $\Pi(q') = s$, is equivalent to $true$,
- $\forall q \in \mathcal{Q}$, $\forall X \subseteq \mathcal{X}$, the union of $\Gamma(q)$ with the conditions $\Gamma(q')$, for all the edges $\langle q, X, q'\rangle$ with $\Pi(q') = \Pi(q)$, is equivalent to $true$.

Notice that when $\mathcal{T}$ is deterministic, for every (bounded or unbounded) trajectory τ, there exists a unique path θ of $\mathcal{T}$ such that $\tau = \Pi \circ \theta$.

3.2 Timed Automata on Bounded Trajectories

Definition A bounded trajectories timed automaton over Σ is a pair $\mathcal{A} = (\mathcal{T}, \mathcal{F})$, where $\mathcal{T} = (\mathcal{X}, \mathcal{Q}, \mathcal{I}nit, \mathcal{E}, \Pi, \Gamma)$ is a timed transition table over Σ, and $\mathcal{F} \subseteq \mathcal{Q}$, is a set of *final* control locations. The automaton $\mathcal{A}$ is deterministic if its table $\mathcal{T}$ is deterministic.

Let $\rho = \langle q_0, \nu_0, I_0\rangle \cdots \langle q_n, \nu_n, I_n\rangle$ be a finite run of $\mathcal{T}$. Then, we say that ρ is an *accepting run* of $\mathcal{A}$ if $q_n \in \mathcal{F}$. We denote by $L(\mathcal{A})$ the set of bounded trajectories generated by accepting runs of $\mathcal{A}$.

Closure Properties Let (D)TA be the class of the sets of trajectories that are definable by (deterministic) bounded trajectories timed automata. It can be shown straightforwardly that TA is closed under union and intersection. The class TA is, however, not closed under complementation (this can be proved as in [3]). On the other hand, it can be easily seen that DTA is closed under all the boolean operators.

3.3 Timed Automata on Unbounded Trajectories

Definition Let $\mathcal{T} = (\mathcal{X}, \mathcal{Q}, \mathcal{I}nit, \mathcal{E}, \Pi, \Gamma)$ be a timed transition table. A Büchi (resp. Muller) acceptance condition is a set $\mathcal{F} \subseteq \mathcal{Q}$ (resp. $\Delta \subseteq 2^{\mathcal{Q}}$).

A timed Büchi (resp. Muller) automaton over Σ, is a pair $\mathcal{A} = (\mathcal{T}, \mathcal{C})$ where $\mathcal{T}$ is a timed transition table over Σ, and $\mathcal{C}$ is a Büchi (resp. Muller) acceptance condition. As in the case of automata on bounded trajectories, an automaton $\mathcal{A}$ is deterministic if its table $\mathcal{T}$ is deterministic.

Let $\rho = \langle q_0, \nu_0, I_0\rangle \cdots \langle q_i, \nu_i, I_i\rangle \cdots$ be an infinite run of $\mathcal{T}$. We denote by $Inf(\rho)$ the set of contol locations q such that $q = q_i$ for infinitely many i's. Then, let $\mathcal{C}$ be a Büchi (resp. Muller) acceptance condition. We say that ρ is an *accepting* *run* of the automaton $\mathcal{A} = (\mathcal{T}, \mathcal{C})$ if $Inf(\rho) \cap \mathcal{C} \neq \emptyset$ (resp. $Inf(\rho) \in \mathcal{C}$). We denote by $L(\mathcal{A})$ the set of unbounded trajectories generated by accepting runs of $\mathcal{A}$.

Expressiveness, Closure Properties, and Emptiness Problem We give hereafter some results on unbounded trajectories timed automata that are relevant for this paper. In [3], a theory is developed for a slightly different version of timed automata. However, the results of [3] can be transposed easily to the automata we consider here.

Let (D)TBA (resp. (D)TMA) denotes the class of sets of trajectories definable by (deterministic) timed Büchi automata (resp. (deterministic) timed Muller automata). First, we recall that these classes are related by the inclusion relations:

$$\text{DTBA} \subset \text{DTMA} \subset \text{TBA} = \text{TMA}$$

Moreover, the classes DTBA and TBA (and hence TMA) are closed under intersection and union but not under complementation, whereas the class DTMA is closed under all the boolean operations.

Finally, we recall that the emptiness problem for TBA is decidable [3, 4].

4 Timed Automata Temporal Logic

We introduce in this section the real-time logic TATL which is an extension of the propositional temporal logic PTL [13] with bounded trajectories timed automata.

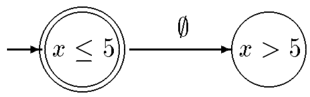
4.1 The logic TATL

Let $\mathcal{P}$ be a finite set of atomic propositions, and let $\Sigma = 2^{\mathcal{P}}$. We use letters $P, Q, \ldots$ to range over elements of $\mathcal{P}$. Let us introduce a set $\mathcal{W}$ of *position variables*. We use letters $u, v, \ldots$ to range over $\mathcal{W}$. Finally, we use letters $A, B, \ldots$ to range over bounded trajectories automata over Σ. The set of formulas of TATL is defined by:

$$\varphi ::= P \mid \neg\varphi \mid \varphi \vee \varphi \mid u.\varphi \mid A^u \mid \varphi \mathcal{U} \varphi$$

We consider as abbreviations the usual boolean connectives as conjunction ($\wedge$) and implication ($\Rightarrow$), $\Diamond\varphi$ for $true\mathcal{U}\varphi$, and $\Box\varphi$ for $\neg\Diamond\neg\varphi$.

TATL formulas are interpreted on unbounded trajectories. The operators $\mathcal{U}$, $\Diamond$, and $\Box$, are the classical *until*, *eventually*, and *always* operators of PTL. In the formula $u.\varphi$, the current time is associated with the position variable u. Then, subformulas of φ of the form A^u are used to express the fact that the bounded trajectory since the time associated with u is accepted by the automaton A. We call formulas of the form A^u *automata constraints*. For example, let $A_{\leq 5}$ be the deterministic automaton represented by the following picture:



Then, the formula $u.\Diamond(P \wedge A^u_{\leq 5})$ expresses the fact that "*P will be eventually true within* 5 *time units*".

In the formula $u.\varphi$ (e.g., $u.\Diamond(P \wedge A^u_{\leq 5})$), the construction "$u.$", called *position quantification*, binds the position variable u in the subformula φ. We suppose without loss of generality that in every formula, each position variable is bound at most once. Then, every variable appearing in some formula is either *bound* or *free*. We denote by $\mathcal{F}ree(\varphi)$ the set of position variables occurring free in φ. A formula φ is *closed* if all the variables occurring in it are bound (i.e., $\mathcal{F}ree(\varphi) = \emptyset$), otherwise φ is *open*.

The formal semantics of TATL is defined using a satisfaction relation $\models$ between unbounded trajectories, time values, and formulas. Since formulas may be open, the satisfaction relation between a trajectory τ, a time value t, and a formula φ, is defined w.r.t. a *position association* E that associates with each position variable $u \in \mathcal{F}ree(\varphi)$ the time (which is less or equal than t) where u has been introduced. Then, the position association E is updated along the trajectory τ due to the introduction of new position variables. We denote by $E[u \mapsto t]$ the position association whose domain is the extension of the domain of E by u, and which associates with u the time value t and coincides with E on all the other position variables.

Now, let τ be an unbounded trajectory. For every $t \in \mathbb{R}_{\geq 0}$, every position association E (such that for every u in the domain of E, $0 \leq E(u) \leq t$), and every TATL formula φ, we define the meaning of $\langle \tau, t\rangle \models_E \varphi$ inductively on the structure of φ.

$$
\begin{array}{lll}
\langle \tau, t\rangle \models_E P & \text{iff} & P \in \tau(t) \\
\langle \tau, t\rangle \models_E \neg\varphi & \text{iff} & \langle \tau, t\rangle \not\models_E \varphi \\
\langle \tau, t\rangle \models_E \varphi_1 \vee \varphi_2 & \text{iff} & \langle \tau, t\rangle \models_E \varphi_1 \text{ or } \langle \tau, t\rangle \models_E \varphi_2 \\
\langle \tau, t\rangle \models_E u.\varphi & \text{iff} & \langle \tau, t\rangle \models_{E[u \mapsto t]} \varphi \\
\langle \tau, t\rangle \models_E A^u & \text{iff} & \tau([E(u), t]) \in L(A) \\
\langle \tau, t\rangle \models_E \varphi_1 \mathcal{U} \varphi_2 & \text{iff} & \exists t' \geq t.\ \langle \tau, t'\rangle \models_E \varphi_2, \text{ and } \forall t'' \in [t, t').\ \langle \tau, t''\rangle \models_E \varphi_1
\end{array}
$$

When φ is a closed formula, we omit the position association E, and write simply $\langle \tau, t\rangle \models \varphi$. Given a closed formula φ, we denote by $[\![\varphi]\!]$ the set of unbounded trajectories τ such that $\langle \tau, 0\rangle \models \varphi$. The formula φ is *satisfiable* iff $[\![\varphi]\!] \neq \emptyset$; it is *valid* iff $[\![\varphi]\!]$ is the set of all possible unbounded trajectories.

4.2 Dense-Time TPTL

Dense-time TPTL [5] is a real-time temporal logic allowing timing constraints to reason about the time elapsed between two states. For instance, the formula (1) given in the introduction is a typical TPTL formula expressing a bounded response property. In that formula, the variable x represents a clock which is reset when P holds, and which is tested when Q is encountered.

Actually, the logic TPTL can be defined as a fragment of the logic TATL. Indeed, for any natural constant c, let us consider the two deterministic automata $A_{<c}$ and $A_{\leq c}$ represented respectively by the following pictures:

and let $u \prec c$ stand for $A^u_{\prec c}$ with $\prec \in \{<, \leq\}$. Then, the logic TPTL is the fragment of TATL corresponding to the set of formulas given by the following grammar:

$$\varphi ::= P \mid \neg\varphi \mid \varphi \vee \varphi \mid u.\varphi \mid u \prec c \mid \varphi \mathcal{U} \varphi$$

and augmented by all the abbreviations we have considered previously. Moreover, we consider that $\varphi_1 \mathcal{U}_{>0} \varphi_2$ abbreviates $u.(\varphi_1 \mathcal{U}\,(u > 0 \wedge \varphi_2))$ where u is a new position variable such that $u \notin \mathcal{F}ree(\varphi_1) \cup \mathcal{F}ree(\varphi_2)$.

4.3 On the semantics of the until operator

The semantics of the operator $\mathcal{U}$ says that a formula $\varphi_1 \mathcal{U} \varphi_2$ is true at some point t of a trajectory τ if and only if there exists some points t' where φ_2 is true, and φ_1 is true at every point in the right-open interval $[t, t')$. Now, let us consider a trajectory τ where, for some point t', $\varphi_1 \wedge \neg\varphi_2$ is true along $[t, t']$ and $\neg\varphi_1 \wedge \varphi_2$ is true along $(t', +\infty)$. Then, it is easy to see that $\varphi_1 \mathcal{U} \varphi_2$ is not true at the point t since, for every point $p > t'$, there exists a point $p' \in (t', p)$ where φ_1 is false. So, given two adjacent intervals where φ_1 and φ_2 are respectively true, the semantics of $\varphi_1 \mathcal{U} \varphi_2$ is in general sensitive to whether the second interval is left-closed or not. An alternative semantics of the until operator can be adopted which is not sensitive to this fact [10]. It corresponds to the definition of the operator $\widetilde{\mathcal{U}}$:

$$\varphi_1 \widetilde{\mathcal{U}} \varphi_2 = (\varphi_1 \vee \varphi_2) \mathcal{U} \varphi_2$$

A nice feature of the operator $\widetilde{\mathcal{U}}$ is that the negation of a formula $\varphi_1 \widetilde{\mathcal{U}} \varphi_2$ can be expressed in a similar way as in PTL using another until operator defined by:

$$\varphi_1 \overline{\mathcal{U}} \varphi_2 = \varphi_1 \mathcal{U} (\varphi_1 \wedge \varphi_2)$$

Indeed, the following equivalences can be proved from TATL semantics.

$$\neg(\varphi_1 \widetilde{\mathcal{U}} \varphi_2) = (\Box\neg\varphi_2) \vee (\neg\varphi_2 \overline{\mathcal{U}} \neg\varphi_1) \tag{2}$$

$$\neg(\varphi_1 \overline{\mathcal{U}} \varphi_2) = (\Box\neg\varphi_2) \vee (\neg\varphi_2 \widetilde{\mathcal{U}} \neg\varphi_1) \tag{3}$$

However, the expression of the negation of an $\mathcal{U}$ formula is more complicated:

$$\neg(\varphi_1 \mathcal{U} \varphi_2) = \neg(\varphi_1 \widetilde{\mathcal{U}} \varphi_2) \vee (\neg\varphi_2 \overline{\mathcal{U}}\, u.(u = 0 \widetilde{\mathcal{U}} (u > 0 \wedge \neg\varphi_1))) \tag{4}$$

Finally, notice that $\Diamond\varphi$ is equivalent to $true \widetilde{\mathcal{U}} \varphi$ as well as to $true \overline{\mathcal{U}} \varphi$.

4.4 The fragments TATL$^+$, TATL$^-$, and TATL$^\times$

We introduce hereafter the syntactical fragments of the logic TATL we consider for expressiveness and decidability issues in the next sections.

The fragment TATL$^+$ consists of the set of formulas φ defined by:

$$\begin{aligned} \varphi &::= \psi \mid u.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \varphi \widetilde{\mathcal{U}} \psi \\ \psi &::= P \mid A^u, A \text{ is deterministic } \mid \neg\psi \mid \psi \vee \psi \mid \psi \widetilde{\mathcal{U}} \psi \mid \psi \overline{\mathcal{U}} \psi \end{aligned}$$

whereas the fragment TATL$^-$ corresponds to the set of formulas φ defined by:

$$\varphi ::= \psi \mid u.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \overline{\mathcal{U}} \varphi$$
$$\psi ::= P \mid A^u, A \text{ is deterministic } \mid \neg\psi \mid \psi \vee \psi \mid \psi \widetilde{\mathcal{U}} \psi \mid \psi \overline{\mathcal{U}} \psi$$

and TATL$^\times$ corresponds to the set formulas φ defined by:

$$\varphi ::= \psi \mid u.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \psi \mathcal{U} \varphi \mid \psi \widetilde{\mathcal{U}} \varphi$$
$$\psi ::= P \mid \neg P \mid A^u \mid \psi \vee \psi \mid \psi \wedge \psi \mid \Box\psi \mid \psi \widetilde{\mathcal{U}} \psi \mid \psi \overline{\mathcal{U}} \psi$$

Notice that formulas of the form $\psi \overline{\mathcal{U}} \varphi$ are also in TATL$^\times$ since, by definition, they correspond to formulas of the form $\psi \mathcal{U} (\psi \wedge \varphi)$ that are clearly in TATL$^\times$.

Using (2) and (3) together with standard laws on boolean connectives, duality of $\Box$ and $\Diamond$, the fact that DTA is closed under complementation, and that $\neg u.\varphi$ is equivalent to $u.\neg\varphi$, we can prove by structural induction the following facts:

Proposition 4.1 *For every TATL$^-$ closed formula φ, there exists a TATL$^\times$ closed formula φ' such that $[\![\varphi]\!] = [\![\varphi']\!]$.*

Proposition 4.2 *For every TATL$^+$ closed formula φ, there exists a TATL$^-$ closed formula φ' such that $[\![\neg\varphi]\!] = [\![\varphi']\!]$, and conversely.*

Thus, TATL$^-$ charaterizes exactly the complements of the TATL$^+$ definable sets of trajectories, and conversely. While the negation of every TATL$^+$ formula is equivalent to a TATL$^\times$ formula, it can be seen that the converse, however, does not hold. The main reason is that the negations of $\mathcal{U}$ formulas are not TATL$^+$ definable in general (see (4)).

Finally, we introduce the fragments of TPTL, called TPTL$^+$, TPTL$^-$, and TPTL$^\times$, corresponding respectively to TATL$^+$, TATL$^-$, and TATL$^\times$ formulas where only constraints of the form $u \prec c$ are used. It is clear that Propositions 4.1 and 4.2 still hold when restricted to these fragments of TPTL.

Notice that, for instance, the bounded response formula (1) given in the introduction is in TPTL$^+$.

5 Expressiveness

In this section, we compare the expressive power of TATL$^+$, TATL$^-$, TATL$^\times$, and timed automata. First of all, we prove that TATL$^+$ as well as TATL$^-$ are strictly more expressive than deterministic timed Muller automata. Moreover, we prove that TATL$^+$ is not comparable with TATL$^-$, TATL$^\times$, and nondeterministic timed automata. On the other hand, we prove that TATL$^\times$ formulas can be characterized by nondeterministic timed automata; and this is our main result in this section.

Let us start by showing that DTMA is included in TATL$^+$ and TATL$^-$ (and hence, in TATL$^\times$). Let $\mathcal{A} = (\mathcal{T}, \Delta)$ be a deterministic timed Muller automaton over $\Sigma = 2^{\mathcal{P}}$, with $\Delta = \{\mathcal{F}_1, \cdots, \mathcal{F}_n\}$. For every control location q in $\mathcal{A}$, we denote by A_q the bounded trajectories timed automaton $(\mathcal{T}, \{q\})$, i.e., the automaton which accepts the set of bounded trajectories that reaches the location q. Then, it can easily be seen that the set of trajectories recognized by the automaton $\mathcal{A}$ is characterized by the formula

$$u.\bigvee_{i=1}^{n}((\bigwedge_{q\in\mathcal{F}_i}\Box\Diamond A_q^u)\wedge(\bigwedge_{q\notin\mathcal{F}_i}\neg\Box\Diamond A_q^u)) \tag{5}$$

which is in both fragments TATL$^+$ and TATL$^-$. So, we have the following result.

Theorem 5.1 *For every deterministic timed Muller automaton $\mathcal{A}$ over Σ, there exists a closed formula φ in TATL$^+$ and TATL$^-$ such that $L(\mathcal{A}) = [\![\varphi]\!]$.*

By observing (5), we can deduce that every DTMA can be charaterized by a boolean combination of formulas of the form $u.\Box\Diamond A^u$, where A is a DTA. Actually, the converse also holds by the facts: (1) it can be shown that formulas of the form $u.\Box\Diamond A^u$, where A is deterministic, correspond exactly to DTBA, (2) every DTBA is a DTMA, and (3) DTMA is closed under all boolean operations.

Now, let us examine the expressiveness of TATL$^+$ and TATL$^-$ in terms of (untimed) ω-sequences of states. A set of sequences $S \subseteq \Sigma^\omega$ is *stuttering-closed* if whenever it contains a sequence $\sigma a \sigma'$, where $\sigma \in \Sigma^*$, $a \in \Sigma$, and $\sigma' \in \Sigma^\omega$, it contains also the sequence $\sigma a a \sigma'$, and vice versa. Then, it can be seen that every stuttering-closed ω-regular (deterministic Muller ω-automata definable) subset of Σ^ω can be charaterized by a DTMA, and hence, by TATL$^+$ and TATL$^-$ formulas.

Theorem 5.2 *For every stuttering-closed ω-regular set $S \subseteq \Sigma^\omega$, there exists a closed formula φ in TATL$^+$ and TATL$^-$ such that $S = \mathcal{D}([\![\varphi]\!])$.*

Actually, we can show that TATL$^+$ is strictly more expressive than DTMA. Inspired by [3], we exhibit a formula in TATL$^+$, actually in TPTL$^+$, that cannot be characterized by nondeterministic timed Muller automata. Let us consider the formula

$$\Phi = \Box(P \Rightarrow u.\Diamond(Q \wedge u = 1)) \tag{6}$$

Then, an automaton recognizing the set of trajectories satisfying Φ must check that whenever P holds, Q will also hold after exactly 1 time unit. Since in any time interval the number of times the truth of P and Q can change is unbounded, it can be seen that such an automaton should dispose of infinitly many clocks, and hence, we have the following result.

Proposition 5.1 *There exists a TATL$^+$ closed formula Φ such that $[\![\Phi]\!]$ is not definable by (finite) nondeterministic timed Muller automata.*

Now, we show that also TATL$^-$ is strictly more expressive than DTMA. For that, let us consider the negation of the formula Φ of (6). By Proposition 4.2, the formula $\neg\Phi$ is equivalent to a TATL$^-$ formula. Suppose that there exists a deterministic timed Muller automaton which recognizes $[\![\neg\Phi]\!]$. Then, since the class DTMA is closed under complementation, $[\![\Phi]\!]$ should also be definable by some deterministic timed Muller automaton, which contradicts Proposition 5.1. Hence, we obtain the following result.

Proposition 5.2 *There exists a TATL$^-$ closed formula φ (equivalent to $\neg\Phi$) such that $[\![\varphi]\!]$ is not definable by deterministic timed Muller automata.*

We show hereafter our main result stating that TATL$^\times$ closed formulas can be characterized by nondeterministic timed Büchi automata, i.e., for every TATL$^\times$ closed formula φ, we can construct a nondeterministic Büchi automaton $\mathcal{A}_\varphi$ such that $L(\mathcal{A}_\varphi) = [\![\varphi]\!]$. We present here the intuition of this construction and leave the details to the full paper.

The essential reason why such a construction is possible is that TATL$^\times$ formulas allow to consider on trajectories only a finite number of points (positions) where automata constraints can be initialized. Indeed, we can show that each TATL$^\times$ formula can be transformed into an equivalent formula in *canonical form*, i.e., a disjunction of formulas of the form

$$\psi_0 \wedge u_1.\,(\psi_1\,\mathcal{W}_{>0}\,(\psi_1' \wedge \cdots u_n.\,(\psi_n\,\mathcal{W}_{>0}\,(\psi_n' \wedge u_{n+1}.\,\psi_{n+1}))\cdots)) \tag{7}$$

where $\mathcal{W}$ stands for either $\mathcal{U}$ or $\widetilde{\mathcal{U}}$, and the ψ_i's are "$u.$"-free TATL formulas. It is convenient to use in formulas in canonical form the constrained operator $\mathcal{W}_{>0}$ instead of $\mathcal{W}$. Notice that this is not a restriction by the fact that $\phi_1 \mathcal{W} \phi_2$ is equivalent to $\phi_2 \vee (\phi_1\,\mathcal{W}_{>0}\,\phi_2)$, where $\mathcal{W}$ is either $\mathcal{U}$ or $\widetilde{\mathcal{U}}$.

Then, given a closed formula φ, the basic idea behind our construction is to consider separately the following problems:

1. checking that a trajectory τ is *propositionally consistent*, which means that, if we abstract from automata constraints and consider φ as a formula of the propositional temporal logic, then the sequences in $\mathcal{D}(\tau)$ satisfy φ,
2. checking that a trajectory is *dense-time semantics consistent*, which means essentially that it is consistent with the dense-time semantics of $\mathcal{U}$,
3. checking that a trajectory is *automata-constraints consistent*, which means that it respects the automata constraints involved in φ.

Let us illustrate on a small example the principle of this construction. Consider the formula in canonical form

$$\phi = u.\,(P_0\,\mathcal{U}_{>0}\,v.\,(P_1\,\widetilde{\mathcal{U}}_{>0}\,B^v))$$

First of all, we associate with each formula of the form $\phi_1 \mathcal{U}_{>0} \phi_2$ a new atomic proposition which allows to distinguish the (right-open) interval where ϕ_1 holds and the point where ϕ_2 holds. Let Q be the atomic proposition associated with the $\mathcal{U}_{>0}$ operator of ϕ. Then, we consider the extended formula

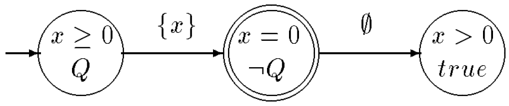
$$\phi_{ext} = u.\,((P_0 \wedge Q)\,\mathcal{U}_{>0}\,(\neg Q \wedge v.\,(P_1\,\widetilde{\mathcal{U}}_{>0}\,B^v)))$$

It is easy to see that ϕ and ϕ_{ext} are equivalent modulo projection on the set of the (old) atomic propositions $\mathcal{P}$ (i.e., abstraction from Q).

Next, we reduce the problem (2) above to the problems (1) and (3). For that, we consider an equivalent formula to ϕ_{ext} defined by:

$$\phi_{ext}' = u.\,((P_0 \wedge Q)\,\widetilde{\mathcal{U}}_{>0}\,(A^u \wedge v.\,(P_1\,\widetilde{\mathcal{U}}_{>0}\,B^v)))$$

where A is the deterministic automaton represented by:



Thus, in the formula ϕ'_{ext} we replace each $\mathcal{U}_{>0}$ operator by the less restrictive one $\widetilde{\mathcal{U}}_{>0}$. However, the introduction of the constraint using the automaton A ensures that all the trajectories satisfying ϕ'_{ext} satisfy also ϕ_{ext}.

To check that a trajectory τ is *propositionally consistent*, we abstract from the actual meaning of the automata constraints and manipulate them as atomic propositions. So, let us associate with the formulas A^u and B^v the new atomic propositions $[A^u]$ and $[B^v]$. We also introduce two other propositions at-ℓ_0 and at-ℓ_1 and consider the formulas $\pi_0 =$ at-$\ell_0 \wedge \neg$at-ℓ_1, and $\pi_1 =$ at-$\ell_1 \wedge \neg$at-ℓ_0. These two formulas allow to mark the point p_v corresponding to the position variable v, that is, the point p_v where the automaton B should be initialized is marked by the end of the truth of π_0 and the beginning of the truth of π_1. Then, let us consider the new set of atomic propositions $\mathcal{P}' = \{[A^u], [B^v], Q, \text{at-}\ell_0, \text{at-}\ell_1\} \cup \mathcal{P}$, and let $\delta_0 = P_0 \wedge Q \wedge \pi_0$ and $\delta_1 = P_1 \wedge \pi_1$. Then, a trajectory is propositionally consistent if it satisfies the formula

$$\phi^* = u.\,(\delta_0\,\widetilde{\mathcal{U}}_{>0}\,([A^u] \wedge v.\,(\delta_1\,\widetilde{\mathcal{U}}_{>0}\,[B^v])))$$

Now, to guaranty that a trajectory τ satisfies indeed ϕ'_{ext} (and hence, ϕ_{ext}), we must check that it is *automata-constraints consistent*, that is, for every $t \in \mathbb{R}_{\geq 0}$, if $[A^u] \in \tau(t)$ (resp. $[B^v] \in \tau(t)$), then $\tau[0,t] \in L(A)$ (resp. $\tau[p_v,t] \in L(B)$).

Then, the set of trajectories that satisfy ϕ_{ext} can be seen as the intersection of two sets of trajectories: the set of propositionally consistent trajectories and the set of automata-constraints consistent trajectories. Therefore, we construct two timed Büchi automata $\mathcal{B}_{\phi^*}$ and $\mathcal{B}_{\phi}^{const}$ over $\Sigma' = 2^{\mathcal{P}'}$ which recognize respectively these two sets of trajectories.

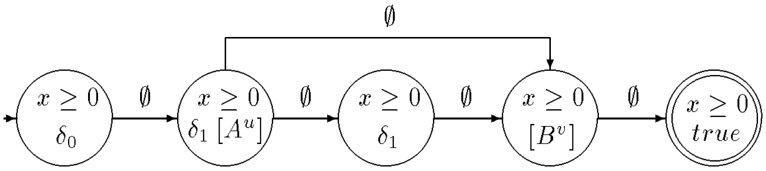
We present the construction of $\mathcal{B}_{\phi^*}$ as an adaptation of the construction of a Büchi automaton recognizing the set of infinite sequences in $(\Sigma')^\omega$ satisfying some appropriate PTL formula. Indeed, let us consider the PTL formula

$$\phi^*_{prop} = \delta_0\,\mathcal{U}\,(\delta_0 \wedge \bigcirc([A^u] \wedge (\delta_1\,\mathcal{U}\,(\delta_1 \wedge \bigcirc[B^v]))))$$

where $\bigcirc$ is the PTL next operator. This formula is obtained from ϕ^* by a syntactic transformation which consists of replacing recursively each subformula of the form $\phi_1\,\widetilde{\mathcal{U}}_{>0}\,\phi_2$ by the formula $\phi_1\,\mathcal{U}\,(\phi_1 \wedge \bigcirc\phi_2)$. It can be verified that the infinite sequences satisfying ϕ^*_{prop} and the trajectories satisfying ϕ^* are related as follows:

$$\forall\sigma.\,\forall\tau.\;\sigma \in \mathcal{D}(\tau) \text{ implies that } \sigma \in [\![\phi^*_{prop}]\!] \text{ iff } \tau \in [\![\phi^*]\!]$$

Then, following [14], we construct a nondeterministic Büchi automaton $\mathcal{B}_{\phi^*_{prop}}$ which recognizes the set of infinite sequences in $(\Sigma')^\omega$ satisfying (according to PTL semantics) the formula ϕ^*_{prop}. The obtained automaton can be transformed straightforwardly (by adding trivial timing constraints) to a timed automaton $\mathcal{B}_{\phi^*}$ which recognizes the set of trajectories satisfying ϕ^*. The automaton $\mathcal{B}_{\phi^*}$ can be represented by the following picture:



It remains to construct the automaton $\mathcal{B}_\phi^{const}$. This is done according to the following principle: The automaton has two layers; in the first one, it simulates the automaton A, and when it moves to the second one, it starts simulating the automaton B while continuing the simulation of A. The label of every control location of the first layer (resp. second layer) contains at-ℓ_0 but not at-ℓ_1 (resp. at-ℓ_1 but not at-ℓ_0). Moreover, the label of a control location contains the proposition $[A^u]$ (resp. $[B^v]$), if and only if this location corresponds to a final location of the automaton A (resp. B).

Finally, we consider an automaton $\mathcal{A}$ over Σ' contructed as a product of the automata $\mathcal{B}_{\phi^*}$ and $\mathcal{B}_\phi^{const}$ so that it recognizes the set of trajectories $L(\mathcal{B}_{\phi^*}) \cap L(\mathcal{B}_\phi^{const})$. The automaton $\mathcal{A}_\phi$ is obtained as the projection of $\mathcal{A}$ on Σ. The construction of $\mathcal{A}_\phi$ for all the TATL$^\times$ formulas is given in the full paper. Then, we have the following result.

Theorem 5.3 *For every TATL$^\times$ closed formula φ, we can construct a nondeterministic timed Büchi automaton $\mathcal{A}_\varphi$ such that $L(\mathcal{A}_\varphi) = [\![\varphi]\!]$.*

Then, by Proposition 5.1 and Theorem 5.3, we obtain the following fact.

Corollary 5.1 *There exists a TATL$^+$ closed formula Φ such that $[\![\Phi]\!]$ is not definable in TATL$^\times$.*

Now, we show the converse, i.e., TATL$^+$ cannot characterize all TATL$^\times$ formulas (actually, it cannot characterize even all TATL$^-$ formulas). Consider again the TATL$^+$ formula Φ (6). By Proposition 4.2, the formula $\neg\Phi$ is equivalent to a TATL$^-$ formula. Now, suppose that $\neg\Phi$ is definable in TATL$^+$. Then, again by Proposition 4.2, Φ is equivalent to a TATL$^-$ formula, and hence, by Theorem 5.3, that $[\![\Phi]\!]$ is in TMA; this contradicts Proposition 5.1. Thus, the formula $\neg\Phi$ is not definable in TATL$^+$, and we obtain the following result.

Proposition 5.3 *There exists a TATL$^-$ closed formula φ (equivalent to $\neg\Phi$) such that $[\![\varphi]\!]$ is not definable in TATL$^+$.*

Then, by Corollary 5.1 and Proposition 5.3, we obtain that the two fragments TATL$^\times$ and TATL$^-$ are not comparable with TATL$^+$. Moreover, as a consequence of Proposition 5.3, Theorem 5.3, and Proposition 5.1, we can deduce also that TMA and TATL$^+$ are not comparable.

6 Decidability

In this section, we present decidability results that are consequences of Theorem 5.3. Our main result is that the verification problem of timed automata w.r.t. TATL$^+$ formulas is decidable.

The first decidability result we have concerns the satisfiability problem of TATL$^\times$. Consider a TATL$^\times$ formula φ. By Theorem 5.3, we can construct a nondeterministic timed Büchi automaton $\mathcal{A}_\varphi$ such that $L(\mathcal{A}_\varphi) = [\![\varphi]\!]$. Then, the formula φ is satisfiable iff $L(\mathcal{A}_\varphi) \neq \emptyset$. Since, the emptiness problem of timed automata is decidable, we obtain the following result.

Theorem 6.1 *The satisfiability problem of TATL$^\times$ is decidable.*

Now, if we consider a TATL$^+$ formula φ, then, by Proposition 4.2, its negation $\neg\varphi$ can be transformed into an equivalent formula $\overline{\varphi}$ in TATL$^-$, and hence in TATL$^\times$ (by Proposition 4.1). Clearly, the formula φ is valid iff the formula $\overline{\varphi}$ is not satisfiable. Then, as a consequence of Theorem 6.1, we deduce that:

Corollary 6.1 *The validity problem of TATL$^+$ is decidable.*

Finally, let us consider the verification problem of timed automata w.r.t. TATL$^+$ formulas. The verification problem consists of, given a timed system modelled as a timed automaton $\mathcal{A}_{syst}$, and a property expressed as formula φ, checking whether all the behaviours (trajectories) of the system satisfy the property, i.e., that $L(\mathcal{A}_{syst}) \subseteq [\![\varphi]\!]$.

Given a TATL$^+$ formula φ, let $\mathcal{A}_{\overline{\varphi}}$ be the timed automaton which recognizes the set of trajectories satisfying a TATL$^\times$ formula $\overline{\varphi}$ which is equivalent to $\neg\varphi$. Then, verifying that $\mathcal{A}_{syst}$ satisfies φ reduces to checking whether $L(\mathcal{A}_{syst}) \cap L(\mathcal{A}_{\overline{\varphi}}) = \emptyset$. Since timed automata are closed under intersection, and their emptiness problem is decidable, we obtain the following decidability result:

Theorem 6.2 *The verification problem of timed automata w.r.t. TATL$^+$ formulas is decidable.*

7 Conclusion

We have investigated the verification problem of timed automata w.r.t. a linear-time dense time temporal logic. The logic we consider, TATL, is an extension of TPTL with timed automata constraints. In this logic, we can define positions on trajectories using the position quantification "$u.$", and use constraints saying that the trajectory segment since some position is accepted by a given timed automaton.

We have identified an expressively powerful fragment of the logic TATL, namely the fragment TATL$^+$, for which the verification problem is decidable. Consequently, we obtain a fragment of TPTL, i.e., the fragment TPTL$^+$, for which the verification problem is decidable.

The existing results in the literature concerning the verification problem of linear-time dense time temporal logics concern the logic MITL [4] and its extension EMITL [15]. Our decidability result concerns a class of timed properties which is not comparable with the properties definable in these logics. Indeed, in both MITL and EMITL, decidability is achieved by forbidding punctual timing constraints, whereas TATL$^+$ allows these kind of constraints (see for instance the formula (6) in Section 5). Moreover, each timing constraint in these logics is related to one temporal operator (which is $\mathcal{U}$ in MITL, or defined by some finite-state automaton on finite sequences in EMITL), whereas the constraints in TATL$^+$ are related to position variables, refering to particular points on a trajectory, following the more general style of TPTL. For instance, general position quantification is essential to the characterization of deterministic timed Muller automata in TATL$^+$ (see (5) in Section 5). Finally, notice that we use in TATL timed automata to express constraints on finite trajectories, whereas in EMITL untimed automata on finite sequences are used as logical operators.

Acknowledgment We thank S. Yovine for his valuable comments.

References

1. R. Alur. *Techniques of Automatic Verification of Real-Time Systems.* PhD thesis, Stanford Univ., 1991.
2. R. Alur, C. Courcoubetis, and D. Dill. Model-Checking for Real-Time Systems. In *LICS'90.* IEEE, 1990.
3. R. Alur and D. Dill. A Theory of Timed Automata. *TCS*, 126, 1994.
4. R. Alur, T. Feder, and T. Henzinger. The Benefits of Relaxing Punctuality. In *PODC'91*, 1991.
5. R. Alur and T. Henzinger. A Really Temporal Logic. In *FOCS'89.* IEEE, 1989.
6. E.M. Clarke, E.A. Emerson, and P. Sistla. Automatic Verification of Finite State Concurrent Systems using Temporal Logic Specifications: A Practical Approach. In *POPL'83.* ACM, 1983.
7. E.A. Emerson, A. Mok, A.P. Sistla, and J. Srinivasan. Quantitative Temporal Reasoning. In *1st Workshop on Computer Aided Verification.* 1989.
8. E. Harel, O. Lichtenstein, and A. Pnueli. Explicit-Clock Temporal Logic. In *LICS'90.* IEEE, 1990.
9. T. Henzinger. *Temporal Specification and Verification of Real-Time Systems.* PhD thesis, Stanford University, 1991.
10. T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic Model-Checking for Real-Time Systems. In *LICS'92.* IEEE, 1992.
11. R. Koymans. Specifying Real-Time Properties with Metric Temporal Logic. *Journal of Real-Time Systems*, 2, 1990.
12. J.S. Ostroff. *Temporal Logic for Real-Time Systems.* Research Stud. Press, 1989.
13. A. Pnueli. The Temporal Logic of Programs. In *FOCS'77.* IEEE, 1977.
14. M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86.* IEEE, 1986.
15. Th. Wilke. Specifying Timed State Sequences in Powerful Decidable Logics and Timed Automata. In *FTRTFT'94.* LNCS 863, 1994.