On Presburger Liveness of Discrete Timed Automata

Zhe Dang[1], Pierluigi San Pietro[2] and Richard A. Kemmerer[3]

[1] School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164, USA

[2] Dipartimento di Elettronica e Informazione
Politecnico di Milano, Italia

[3] Department of Computer Science
University of California at Santa Barbara, CA 93106, USA

Abstract. Using an automata-theoretic approach, we investigate the decidability of liveness properties (called *Presburger liveness properties*) for timed automata when Presburger formulas on configurations are allowed. While the general problem of checking a temporal logic such as TPTL augmented with Presburger clock constraints is undecidable, we show that there are various classes of Presburger liveness properties which are decidable for *discrete* timed automata. For instance, it is decidable, given a discrete timed automaton $\mathcal{A}$ and a Presburger property P, whether there exists an ω-path of $\mathcal{A}$ where P holds infinitely often. We also show that other classes of Presburger liveness properties are indeed undecidable for discrete timed automata, e.g., whether P holds infinitely often *for each* ω-path of $\mathcal{A}$. These results might give insights into the corresponding problems for timed automata over dense domains, and help in the definition of a fragment of linear temporal logic, augmented with Presburger conditions on configurations, which is decidable for model checking timed automata.

1 Introduction

Timed automata [3] are widely regarded as a standard model for real-time systems, because of their ability to express quantitative time requirements in the form of clock *regions*: a clock or the difference of two clocks is compared against an integer constant, e.g., $x - y > 5$, where x and y are clocks. A fundamental result in the theory of timed automata is that region reachability is decidable. This has been proved by using the region technique [3]. This result is very useful since in principle it allows some forms of automatic verification of timed automata. In particular, it helps in developing a number of temporal logics [2, 6, 13, 15, 4, 16], in investigating the model-checking problem and in building model-checking tools [12, 17, 14] (see [1, 18] for surveys).

In real-world applications [7], clock constraints represented as clock regions are useful but often not powerful enough. For instance, we might want to argue whether a non-region property such as $x_1 - x_2 > x_3 - x_4$ (i.e., the difference of clocks x_1 and x_2 is larger than that of x_3 and x_4) always holds when a timed automaton starts from clock values satisfying another non-region property. Hence, it would be useful to

consider Presburger formulas as clock constraints, were it not for the fact that a temporal logic like TPTL [6] is undecidable when augmented with Presburger clock constraints [6]. However, recent work [9, 10] has found decidable characterizations of the binary reachability of timed automata, giving hope that *some* important classes of non-region properties are still decidable for timed automata.

In this paper, we look at *discrete* timed automata (dta), i.e., timed automata where clocks are integer-valued. Discrete time makes it possible to apply, as underlying theoretical tools, a good number of automata-theoretic techniques and results. Besides the facts that discrete clocks are usually easier to handle than dense clocks also for practitioners, and that dtas are useful by themselves as a model of real-time systems [5], results on dtas may give insights into corresponding properties of dense timed automata [11].

The study of *safety* properties and *liveness* properties is of course of the utmost importance for real-life applications. In [10] (as well as in [9]), it has been shown that the Presburger safety analysis problem is decidable for discrete timed automata. That is, it is decidable whether, given a discrete timed automaton $\mathcal{A}$ and two sets I and P of configurations of $\mathcal{A}$ (tuples of control state and clock values) definable by Presburger formulas, $\mathcal{A}$ always reaches a configuration in P when starting from a configuration in I.

In this paper we concentrate on the Presburger liveness problem, by systematically formulating a number of Presburger liveness properties and investigating their decidability. For instance, we consider the $\exists$-Presburger-i.o. problem: whether there exists an ω-path p for $\mathcal{A}$ such that p starts from I and P is satisfied on p infinitely often. Another example is the $\forall$-Presburger-eventual problem: whether for all ω-paths p that start from I, P is eventually satisfied on p.

The main results of this paper show that (using an obvious notation, once it is clear that $\exists$ and $\forall$ are path quantifiers):

- The $\exists$-Presburger-i.o. problem and the $\exists$-Presburger-eventual problem are both decidable. So are their duals, the $\forall$-Presburger-almost-always problem and the $\forall$-Presburger-always problem.
- The $\forall$-Presburger-i.o. problem and the $\forall$-Presburger-eventual problem are both undecidable. So are their duals, the $\exists$-Presburger-almost-always problem and the $\exists$-Presburger-always problem.

These results can be helpful in formulating a weak form of a Presburger linear temporal logic and in defining a fragment thereof that is decidable for model-checking dta. The proofs are based on the definition of a version of dta, called *static* dta, which does not have enabling conditions on transitions. The decidability of the previous Presburger liveness problems is the same for dta and static dta. Hence, proofs can be easier, since static dta are much simpler to deal with than dta.

The paper is organized as follows. Section 2 introduces the main definitions, such as discrete timed automata and the Presburger liveness properties. Section 3 shows the decidability of the $\exists$-Presburger-i.o. and of the $\exists$-Presburger-eventual problems, by introducing static dta. Section 4 shows the undecidability of the $\forall$-Presburger-i.o. and of the $\forall$-Presburger-eventual problems. Section 5 discusses some aspects related to the introduction of Presburger conditions in temporal logic, and to the extension of our results to dense time domains.

The proofs of some lemmas and theorems can be found in the full version of the paper available at `www.eecs.wsu.edu/~zdang`.

2 Preliminaries

A timed automaton [3] is a finite state machine augmented with a number of real-valued clocks. All the clocks progress synchronously with rate 1, except when a clock is reset to 0 at some transition. In this paper, we consider *integer-valued* clocks. A *clock constraint* (or a *region*) is a Boolean combination of *atomic clock constraints* in the following form: $x\#c, x-y\#c$ where $\#$ denotes $\leq, \geq, <, >$, or $=$, c is an integer, x, y are integer-valued clocks. Let $\mathcal{L}_X$ be the set of all clock constraints on clocks X. Let $\mathbf{N}$ be the set of nonnegative integers.

Definition 1. *A discrete timed automaton (dta) is a tuple $\mathcal{A} = \langle S, X, E\rangle$ where S is a finite set of (control) states, X is a finite set of clocks with values in $\mathbf{N}$, and $E \subseteq S \times 2^X \times \mathcal{L}_X \times S$ is a finite set of edges or transitions.*

Each edge $\langle s, \lambda, l, s'\rangle$ denotes a transition from state s to state s' with *enabling condition* $l \in \mathcal{L}_X$ and a set of clock resets $\lambda \subseteq X$. Note that λ may be empty: in this case, the edge is called a *clock progress transition*. Since each pair of states may have more than one edge between them, in general $\mathcal{A}$ is nondeterministic.

The semantics of *dta*s is defined as follows. We use $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{X}, \boldsymbol{Y}$ to denote clock vectors (i.e., vectors of clock values) with $\boldsymbol{V}_x$ being the value of clock x in $\boldsymbol{V}$. $\boldsymbol{I}$ denotes the identity vector in $\mathbf{N}^{|X|}$; i.e., $\boldsymbol{I}_x = 1$ for each $x \in X$.

Definition 2. (configuration, one-step transition relation $\rightarrow^{\mathcal{A}}$) *A configuration $\langle s, \boldsymbol{V}\rangle \in S \times (\mathbf{N})^{|X|}$ is a tuple of a control state s and a clock vector $\boldsymbol{V}$. $\langle s, \boldsymbol{V}\rangle \rightarrow^{\mathcal{A}} \langle s', \boldsymbol{V}'\rangle$ denotes a* one-step transition *from configuration $\langle s, \boldsymbol{V}\rangle$ to configuration $\langle s', \boldsymbol{V}'\rangle$ satisfying all the following conditions:*

- *There is an edge $\langle s, \lambda, l, s'\rangle$ in $\mathcal{A}$ connecting state s to state s',*
- *The enabling condition of the edge is satisfied, that is, $l(\boldsymbol{V})$ is true,*
- *Each clock changes according to the edge. If there are no clock resets on the edge, i.e., $\lambda = \emptyset$, then clocks progress by one time unit, i.e., $\boldsymbol{V}' = \boldsymbol{V} + \boldsymbol{I}$. If $\lambda \neq \emptyset$, then for each $x \in \lambda$, $\boldsymbol{V}'_x = 0$ while for each $x \notin \lambda$, $\boldsymbol{V}'_x = \boldsymbol{V}_x$.*

A configuration $\langle s, \boldsymbol{V}\rangle$ is a *deadlock configuration* if there is no configuration $\langle s', \boldsymbol{V}'\rangle$ such that $\langle s, \boldsymbol{V}\rangle \rightarrow^{\mathcal{A}} \langle s', \boldsymbol{V}'\rangle$. $\mathcal{A}$ is *total* if every configuration is not a deadlock configuration. A *path* is a finite sequence $\langle s_0, \boldsymbol{V}^0\rangle \cdots \langle s_k, \boldsymbol{V}^k\rangle$ such that $\langle s_i, \boldsymbol{V}^i\rangle \rightarrow^{\mathcal{A}} \langle s_{i+1}, \boldsymbol{V}^{i+1}\rangle$ for each $0 \leq i \leq k-1$. A path is a *progress* path if there is at least one clock progress transition on the path. An *ω-path* is an infinite sequence $\langle s_0, \boldsymbol{V}^0\rangle \cdots \langle s_k, \boldsymbol{V}^k\rangle \cdots$ such that each prefix $\langle s_0, \boldsymbol{V}^0\rangle \cdots \langle s_k, \boldsymbol{V}^k\rangle$ is a path. An ω-path is *divergent* if there is an infinite number of clock progress transitions on the ω-path. Without loss of generality, in this paper we consider timed automata without event labels [3], since they can be built into the control states.

Let Y be a finite set of variables over integers. For all integers a_y with $y \in Y$, b and c (with $c > 0$), $\sum_{y\in Y} a_y y < b$ is an *atomic linear relation* on Y and $\sum_{y\in Y} a_y y \equiv_b c$ is a *linear congruence* on Y. A *linear relation* on Y is a Boolean combination (using $\neg$ and $\wedge$) of atomic linear relations on Y. A *Presburger formula* on Y is the Boolean

combination of atomic linear relations on Y and of linear congruences on Y. A set P is *Presburger-definable* if there exists a Presburger formula $\mathcal{F}$ on Y such that P is exactly the set of the solutions for Y that make $\mathcal{F}$ true. Since Presburger formulas are closed under quantifications, we will allow quantifiers over integer variables.

Write $\langle s, \boldsymbol{V} \rangle \rightsquigarrow^{\mathcal{A}} \langle s', \boldsymbol{V}' \rangle$ if $\langle s, \boldsymbol{V} \rangle$ reaches $\langle s', \boldsymbol{V}' \rangle$ through a path in $\mathcal{A}$. The binary relation $\rightsquigarrow^{\mathcal{A}}$ can be considered as a subset of configuration tuples and called *binary reachability*. It has been shown recently that,

Theorem 1. *The binary reachability $\rightsquigarrow^{\mathcal{A}}$ is Presburger-definable [9, 10].*

The *Presburger safety analysis problem* is to consider whether $\mathcal{A}$ can only reach configurations in P starting from any configuration in I, given two Presburger-definable sets I and P of configurations. Because of Theorem 1, the Presburger safety analysis problem is decidable [10] for dtas.

In this paper, we consider *Presburger liveness analysis problems* for dtas, obtained by combining a path-quantifier with various modalities of satisfaction on an ω-path. Let I and P be two Presburger-definable sets of configurations, and let p be an ω-path $\langle s_0, \boldsymbol{V}^0 \rangle, \langle s_1, \boldsymbol{V}^1 \rangle \ldots$. Define the following modalities of satisfactions of P and I over p:

- p is *P-i.o.* if P is satisfied infinitely often on the ω-path, i.e., there are infinitely many k such that $\langle s_k, \boldsymbol{V}^k \rangle \in P$.
- p is *P-always* if for each k, $\langle s_k, \boldsymbol{V}^k \rangle \in P$.
- p is *P-eventual* if there exists k such that $\langle s_k, \boldsymbol{V}^k \rangle \in P$.
- p is *P-almost-always* if there exists k such that for all $k' > k$, $\langle s_{k'}, \boldsymbol{V}^{k'} \rangle \in P$.
- p *starts* from I if $\langle s_0, \boldsymbol{V}^0 \rangle \in I$.

Definition 3. (Presburger liveness analysis problems) *Let $\mathcal{A}$ be a dta and let I and P be two Presburger-definable sets of configurations of $\mathcal{A}$. The $\exists$-Presburger-i.o. (resp. always, eventual and almost-always) problem is to decide whether the following statement holds: there is an ω-path p starting from I that is P-i.o. (resp. P-always, P-eventual and P-almost-always). The $\forall$-Presburger-i.o. (resp. always, eventual and almost-always) problem is to decide whether the following statement holds: for every ω-path p, if p starts from I, then p is P-i.o. (resp. always, eventual and almost-always).*

3 Decidability Results

In this section, we show that the $\exists$-Presburger-i.o. problem is decidable for dtas. Proofs of an infinitely-often property usually involve analysis of cycles in the transition system. However, for dtas, this is difficult for the following reasons. A discrete timed automaton $\mathcal{A}$ can be treated as a transition graph on control states with clock reset sets properly assigned to each edge, and augmented with tests (i.e., clock constraints) on edges. The tests are dynamic – the results of the tests depend upon the current values of each clock and obviously determine which edges may be taken. This is an obstacle to applying cyclic analysis techniques on the transition graph of $\mathcal{A}$.

A solution to these difficulties is to introduce *static* discrete timed automata, i.e., dtas with all the enabling conditions being simply *true*. The lack of enabling conditions simplifies the proof that the $\exists$-Presburger-i.o. problem is decidable for static dtas. Then, we show that each $\exists$-Presburger-i.o. problem for a dta can be translated into an $\exists$-Presburger-i.o. problem for a static dta, and hence it is decidable as well.

3.1 The ∃-Presburger-i.o. problem for static dtas

Let $\mathcal{A}$ be a static dta. We show that the ∃-Presburger-i.o. problem for static dtas is decidable. Given two sets I and P of configurations definable by Presburger formulas, an ω-path $p = \langle s_0, \boldsymbol{V}^0\rangle \cdots \langle s_k, \boldsymbol{V}^k\rangle \cdots$ is a *witness* if it is a solution of the ∃-Presburger-i.o. problem, i.e., p is P-i.o. and p starts from I ($\langle s_0, \boldsymbol{V}^0\rangle \in I$). There are two cases to a witness p: (1) p is not divergent; (2) p is divergent. For Case (1), we can establish the following lemma by expressing the existence of p into a Presburger formula obtained from the binary reachability of $\mathcal{A}$.

Lemma 1. *The existence of a non-divergent witness is decidable.*

The difficult case, however, is when the witness p is divergent. The remainder of this subsection is devoted to the proof that the existence of a divergent witness is decidable. For now, we fix a choice of a control state s and a set $X_r \subseteq X$ of clocks (there are only finitely many of them). To ensure that p is divergent, each path from $\langle s_{k_i} = s, \boldsymbol{V}^{k_i}\rangle$ to $\langle s_{k_{i+1}} = s, \boldsymbol{V}^{k_{i+1}}\rangle$ is picked so that it contain at least one clock progress transition, i.e., a *progress cycle*, as follows.

Definition 4. *For all clock vectors $\boldsymbol{V}, \boldsymbol{V}'$, we write $\langle s, \boldsymbol{V}\rangle \rightsquigarrow^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{V}'\rangle$ if*

1. *there exists $\langle s_0, \boldsymbol{V}^0\rangle \in I$ such that $\langle s_0, \boldsymbol{V}^0\rangle \rightsquigarrow^{\mathcal{A}} \langle s, \boldsymbol{V}\rangle$, i.e., $\langle s, \boldsymbol{V}\rangle$ is reachable from a configuration in I,*
2. *$\langle s, \boldsymbol{V}'\rangle \in P$,*
3. *$\langle s, \boldsymbol{V}\rangle \rightsquigarrow^{\mathcal{A}} \langle s, \boldsymbol{V}'\rangle$ through a progress path on which all the clocks in X_r are reset at least once and all the clocks not in X_r are never reset.*

The proof proceeds as follows. First, we show (Lemma 2) that the relation $\rightsquigarrow^{\mathcal{A}}_{X_r}$ is Presburger-definable. Then, since $\mathcal{A}$ is finite state, there exists a P-i.o. ω-path p iff there is a state s such that P holds infinitely often on p at state s. This is equivalent to saying (Lemma 3) that there exist clock vectors $\boldsymbol{V}^1, \boldsymbol{V}^2, \ldots$ such that $\langle s, \boldsymbol{V}^i\rangle \rightsquigarrow^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{V}^{i+1}\rangle$ for each $i > 0$. Since the actual values of the clocks in X_r may be abstracted away (Lemma 4 and Definition 5) and the clocks in $X - X_r$ progress synchronously, this is equivalent to saying that there exist $\boldsymbol{V}, d^1 > 0, d^2 > 0, \ldots$ such that $\boldsymbol{V}^i_x = \boldsymbol{V}_x + d^i$ for all $x \in X - X_r$(Lemma 5). The set $\{d^i\}$ may be defined with a Presburger formula, as shown in Lemma 7, since each d^i may always be selected to be of the form $c^i + f(c^i)$, where the set $\{c^i\}$ is a periodic set (hence, Presburger definable) and f is a Presburger-definable function. This is based on the fact that static automata have no edge conditions, allowing us to increase the length d of a progress cycle to a length nd (Lemma 6), for every $n > 0$. The decidability result on the existence of a divergent witness follows directly from Lemma 7.

Lemma 2. *$\rightsquigarrow^{\mathcal{A}}_{X_r}$ is Presburger-definable. That is, given $s \in S$, $\langle s, \boldsymbol{V}\rangle \rightsquigarrow^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{V}'\rangle$ is a Presburger formula, when the clock vectors $\boldsymbol{V}, \boldsymbol{V}'$ are regarded as integer variables.*

Based upon the above analysis, the following lemma is immediate:

Lemma 3. *There is a divergent witness p iff there are s, X_r and clock vectors $\boldsymbol{V}^1, \boldsymbol{V}^2, \ldots$ such that $\langle s, \boldsymbol{V}^i\rangle \rightsquigarrow^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{V}^{i+1}\rangle$ for each $i > 0$.*

$\langle s, \boldsymbol{V}\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{W}\rangle$ denotes the following scenario. Starting from some configuration in I, $\mathcal{A}$ can reach $\langle s, \boldsymbol{V}\rangle$ and return to s again with clock values $\boldsymbol{W}$. The cycle at s is a progress one such that each clock in X_r resets at least once and all clocks not in X_r do not reset. Since $\mathcal{A}$ is static, the cycle can be represented by a sequence $s_0 s_1 \cdots s_t$ of control states, with $s_0 = s_t = s$, and such that, for each $0 \leq i < t$, there is an edge in $\mathcal{A}$ connecting s_i and s_{i+1}. Observe that, since each $x \in X_r$ is reset in the cycle, the starting clock values $\boldsymbol{V}_x$ for $x \in X_r$ at $s_0 = s$ are insensitive to the ending clock values $\boldsymbol{W}_x$ with $x \in X_r$ at $s_t = s$ (those values of $\boldsymbol{W}_x$ only depend on the sequence of control states). We write $\boldsymbol{V} =_{X-X_r} \boldsymbol{U}$ if $\boldsymbol{V}$ and $\boldsymbol{U}$ agree on the values of the clocks not in X_r, i.e., $\boldsymbol{V}_x = \boldsymbol{U}_x$, for each $x \in X - X_r$. The insensitivity property is stated in the following lemma.

Lemma 4. *For all clock vectors $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$, if $\langle s, \boldsymbol{V}\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{W}\rangle$ and $\langle s, \boldsymbol{U}\rangle$ is reachable from some configuration in I with $\boldsymbol{V} =_{X-X_r} \boldsymbol{U}$, then $\langle s, \boldsymbol{U}\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{W}\rangle$.*

Also note that, since all clocks not in X_r do not reset on the cycle, the differences $\boldsymbol{W}_x - \boldsymbol{V}_x$ for each $x \in X - X_r$ are equal to the *duration* of the cycle (i.e., the number of progress transitions in the cycle). The following technical definition allows us to "abstract" clock values for X_r away in $\langle s, \boldsymbol{V}\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{W}\rangle$.

Definition 5. *For all clock vectors $\boldsymbol{Y}$ and positive integers d, we write $\boldsymbol{Y} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y} + d\boldsymbol{I}$ if there exist two clock vectors $\boldsymbol{V}$ and $\boldsymbol{W}$ such that $\langle s, \boldsymbol{V}\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{W}\rangle$ with $\boldsymbol{Y} =_{X-X_r} \boldsymbol{V}$ and $\boldsymbol{Y} + d\boldsymbol{I} =_{X-X_r} \boldsymbol{W}$.*

Obviously, in the previous definition, the cycle from $\langle s, \boldsymbol{V}\rangle$ to $\langle s, \boldsymbol{W}\rangle$ has duration d. Also, the relation $\leadsto^{\mathcal{A}}_{\langle s, X_r\rangle}$ is Presburger-definable (over $\boldsymbol{Y}$ and d).

Lemma 5. *There exists a divergent witness for $\mathcal{A}$ iff there are $s, X_r, \boldsymbol{Y}, d^1, d^2, \ldots$ such that $0 \leq d^1 < d^2 < \ldots$ and $\boldsymbol{Y} + d^i\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y} + d^{i+1}\boldsymbol{I}$, for each $i \geq 1$.*

The following technical lemma, based on Definition 5 and Lemma 5, will be used in the proof of Lemma 7.

Lemma 6. *For all $\boldsymbol{Y}, \boldsymbol{Y}', n > 1, d > 0$, if $\boldsymbol{Y} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y} + d\boldsymbol{I}$ and $\boldsymbol{Y} + nd\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y}'$, then $\boldsymbol{Y} + d\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y}'$.*

Lemma 7. *It is decidable whether there exists a divergent witness for a static dta $\mathcal{A}$.*

Proof. We claim that, there are s, X_r such that the Presburger formula

$(*)\quad \exists \boldsymbol{Y} \forall m > 0 \exists d_1 \geq m \exists d_2 > 0 \ \left(\ \boldsymbol{Y} + d_1\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y} + (d_1 + d_2)\boldsymbol{I} \ \right)$

holds if and only if there is a divergent witness for $\mathcal{A}$. The statement of the lemma then follows immediately.

Assume there is a divergent witness. Hence, by Lemma 3, there exist $\boldsymbol{V}^1, \boldsymbol{V}^2, \ldots$ and $d^1, d^2, \ldots$ such that, for each $i \geq 1$, $\langle s, \boldsymbol{V}^i\rangle \leadsto^{\mathcal{A}}_{X_r} \langle s, \boldsymbol{V}^{i+1}\rangle$ with a progress cycle of duration $d^i > 0$. Let $\boldsymbol{Y}$ be such that $\boldsymbol{Y} =_{X-X_r} \boldsymbol{V}^1$. By Definition 5, $\boldsymbol{Y} + \left(\sum_{j=1}^{i-1} d^j\right)\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y} + \left(\sum_{j=1}^{i} d^j\right)\boldsymbol{I}$ for each $i \geq 1$. For each $m > 0$, let $d_1 = \sum_{j=1}^{m} d^j$, $d_2 = d^{m+1}$. It is immediate that (*) holds.

Conversely, let $\boldsymbol{Y}_0$ be one of the vectors $\boldsymbol{Y}$ such that $(*)$ holds. Apply skolemization to the formula $\exists d_2 > 0 \left(\boldsymbol{Y}_0 + d_1\boldsymbol{I} \leadsto^{\mathcal{A}}_{\langle s, X_r\rangle} \boldsymbol{Y}_0 + (d_1 + d_2)\boldsymbol{I}\right)$, by introducing

a function $f(d_1)$ to replace the variable d_2. Since (*) holds, then the formula $H(d_1)$, defined as $\boldsymbol{Y}_0 + d_1\boldsymbol{I} \rightsquigarrow^{\mathcal{A}}_{\langle s,X_r\rangle} \boldsymbol{Y}_0 + (d_1 + f(d_1))\boldsymbol{I}$, holds for infinitely many values of d_1. Combining the fact that $H(d_1)$ is Presburger-definable (because $\boldsymbol{Y}_0$ is fixed), there is a periodic set included in the infinite domain of H, i.e., there exist $n > 1, k \geq 0$ such that for all $d \geq 0$ if $d \equiv_n k$ then $H(d)$ holds. Let c^0 be any value in the periodic set, and let $c^i = c^{i-1} + nf(c^{i-1})$, for every $i \geq 1$. Obviously, every c^i satisfies the periodic condition: $c^i \equiv_n k$, and therefore $H(c^i)$ holds. Hence, for every $i \geq 1$, $\boldsymbol{Y}_0 + c^i\boldsymbol{I} \rightsquigarrow^{\mathcal{A}}_{\langle s,X_r\rangle} \boldsymbol{Y}_0 + (c^i + f(c^i))\boldsymbol{I}$.

Since $\boldsymbol{Y}_0 + c^{i+1}\boldsymbol{I} = \boldsymbol{Y}_0 + c^i\boldsymbol{I} + nf(c^i)\boldsymbol{I} \rightsquigarrow^{\mathcal{A}}_{\langle s,X_r\rangle} \boldsymbol{Y}_0 + (c^{i+1} + f(c^{i+1}))\boldsymbol{I}$, we may apply Lemma 6, with: $\boldsymbol{Y} = \boldsymbol{Y}_0 + c^i\boldsymbol{I}$, $d = f(c^i)$, $\boldsymbol{Y}' = \boldsymbol{Y} + (c^{i+1} + f(c^{i+1}))\boldsymbol{I}$. Lemma 6 then gives $\boldsymbol{Y} + d\boldsymbol{I} \rightsquigarrow^{\mathcal{A}}_{\langle s,X_r\rangle} \boldsymbol{Y}'$, i.e., $\boldsymbol{Y}_0 + (c^i + f(c^i))\boldsymbol{I} \rightsquigarrow^{\mathcal{A}}_{\langle s,X_r\rangle} \boldsymbol{Y}_0 + (c^{i+1} + f(c^{i+1}))\boldsymbol{I}$, for every $i \geq 1$. By Lemma 5, with $d^i = c^i + f(c^i)$, there is a divergent witness. ∎

By Lemmas 1 and 7, we have:

Theorem 2. *The ∃-Presburger-i.o. problem is decidable for static dtas.*

3.2 The ∃-Presburger-i.o. problem for dtas

In the full paper, we use a technique modified from [10] to show that the tests in $\mathcal{A}$ can be eliminated. That is, $\mathcal{A}$ can be effectively transformed into $\mathcal{A}''$ where all the tests are simply *true* and $\mathcal{A}''$ has (almost) the same static transition graph as $\mathcal{A}$. This is based on an encoding of the tests of $\mathcal{A}$ into the finite state control of $\mathcal{A}''$. Now we look at the ∃-Presburger-i.o. problem for $\mathcal{A}$. Recall that the problem is to determine, given two Presburger-definable sets I and P of configurations of $\mathcal{A}$, whether there exists a P-i.o. ω-path p starting from I. We relate the instance of the ∃-Presburger-i.o. problem for $\mathcal{A}$ to an instance of the ∃-Presburger-i.o. problem for $\mathcal{A}''$:

Lemma 8. *Given a dta $\mathcal{A}$, and two Presburger-definable sets I and P of configurations of $\mathcal{A}$, there exist a static dta $\mathcal{A}''$ and two Presburger definable sets I'' and P'' of configurations of $\mathcal{A}''$ such that: the existence of a witness to the ∃-Presburger-i.o. for $\mathcal{A}$, given I and P, is equivalent to the existence of a witness to the ∃-Presburger-i.o. for $\mathcal{A}''$, given I'' and P''.*

Since $\mathcal{A}''$ is a static dta, the decidability of the ∃-Presburger-i.o. for $\mathcal{A}$ follows from Theorem 2 and Lemma 8.

Theorem 3. *The ∃-Presburger-i.o. problem and the ∀-Presburger-almost-always problem are decidable for dtas.*

3.3 Decidability of the ∃-Presburger-eventual Problem

Given a dta $\mathcal{A}$, and two Presburger-definable sets I and P of configurations, the ∃-Presburger-eventual problem is to decide whether there exists a P-eventual ω-path p starting from I. Define I' to be the set of all configurations in P that can be reached from a configuration in I. From Theorem 1, I' is Presburger-definable. Let P' be simply *true*. It can be shown that the existence of a witness for the ∃-Presburger-eventual problem (given I and P) is equivalent to the existence of a witness for the ∃-Presburger-i.o. problem (given I' and P'). ¿From Theorem 3,

Theorem 4. *The $\exists$-Presburger-eventual problem and the $\forall$-Presburger-always problem are decidable for dtas.*

It should be noted that there is a slight difference between the $\forall$-Presburger-always problem and the Presburger safety analysis problem mentioned before. The difference is that the Presburger safety analysis problem considers (finite) paths while the $\forall$-Presburger-always problem considers ω-paths.

4 Undecidability Results

The next three subsections show that the undecidability of the $\forall$-Presburger-eventual problem and of the $\forall$-Presburger-i.o. problem. We start by demonstrating the fact that a two-counter machine can be implemented by a generalized version of a dta. This fact is then used in the following two subsections to show the undecidability results.

4.1 Counter Machines and Generalized Discrete Timed Automata

Consider a counter machine M with counters $x_1, \cdots, x_k$ over nonnegative integers and with a finite set of locations $\{l_1, \cdots, l_n\}$. M can increment, decrement and test against 0 the values of the counters. It is well-known that a two-counter machine can simulate a Turing machine.

We now define *generalized* discrete timed automata. They are defined similarly to dtas but for each edge $\langle s, \lambda, l, s' \rangle$ the formula l is of the form $\sum_i a_i x_i \# c$, where a_i and c are integers. Generalized dtas are Turing-complete, since they can simulate any counter machine:

Lemma 9. *Given a deterministic counter machine M, there exists a deterministic generalized dta that can simulate M.*

From now on, let M be a deterministic counter machine and let $\mathcal{A}$ be a deterministic generalized dta that implements M. We may assume that $\mathcal{A}$ is total (i.e., there are no deadlock configurations), since $\mathcal{A}$ can be made total by adding a new self-looped state s_f, and directing every a deadlock configuration to this new state. Now we define the *static version* $\mathcal{A}^-$, to which $\mathcal{A}$ can be modified as follows. $\mathcal{A}^-$ is a discrete timed automaton with the enabling condition on each edge being simply *true*. Each state in $\mathcal{A}^-$ is a pair of states in $\mathcal{A}$. $\langle \langle s_1, s_1' \rangle, \lambda_1, true, \langle s_2, s_2' \rangle \rangle$ is an edge of $\mathcal{A}^-$ iff there are edges $\langle s_1, \lambda_1, l_1, s_1' \rangle$ and $\langle s_2, \lambda_2, l_2, s_2' \rangle$ in $\mathcal{A}$ with $s_1' = s_2$. We define a set P, called the *path restriction of* $\mathcal{A}$, of configurations of $\mathcal{A}^-$ as follows. For each configuration $\langle \langle s, s' \rangle, \boldsymbol{V} \rangle$ of $\mathcal{A}^-$, $\langle \langle s, s' \rangle, \boldsymbol{V} \rangle \in P$ iff there exists an edge $e = \langle s, \lambda, l, s' \rangle$ such that the clock values $\boldsymbol{V}$ satisfy the linear relation l in e. Clearly, P is Presburger-definable. Since $\mathcal{A}$ is total and deterministic, the above edge e always exists and is unique for each configuration $\langle s, \boldsymbol{V} \rangle$ of $\mathcal{A}$. Using this fact, we have,

Theorem 5. *Let $\mathcal{A}$ be a total and deterministic generalized dta with path restriction P, and let $\mathcal{A}^-$ be the static version of $\mathcal{A}$. An ω-sequence $\langle s_0, \boldsymbol{V}^0 \rangle \cdots \langle s_k, \boldsymbol{V}^k \rangle \cdots$ is an ω-path of $\mathcal{A}$ iff $\langle \langle s_0, s_1 \rangle, \boldsymbol{V}^0 \rangle \cdots \langle \langle s_k, s_{k+1} \rangle, \boldsymbol{V}^k \rangle \cdots$ is an ω-path of $\mathcal{A}^-$ with $\langle \langle s_k, s_{k+1} \rangle, \boldsymbol{V}^k \rangle \in P$ for each k.*

4.2 Undecidability of the ∀-Presburger-eventual Problem

We consider the negation of the ∀-Presburger-eventual problem, i.e., the ∃-Presburger-always problem, which can be formulated as follows: given a discrete timed automaton $\mathcal{A}$ and two Presburger-definable sets I and P of configurations, decide whether there exists a $\neg P$-always ω-path of $\mathcal{A}$ starting from I.

Consider a deterministic counter machine M with the initial values of the counters being 0 and the first instruction labeled l_0. Let $\mathcal{A}$ be the deterministic generalized dta implementing M, as defined by Lemma 9, with P being the path restriction of $\mathcal{A}$. As before, $\mathcal{A}$ is total. Let $\mathcal{A}^-$ be the static version of $\mathcal{A}$. It is well known that the halting problem for (deterministic) counter machines is undecidable. That is, it is undecidable, given M and an instruction label l, whether M executes the instruction l. Define P' to be the set of configurations $\langle\langle s, s'\rangle, \boldsymbol{V}\rangle \in P$ with $s \neq l$. Let I be the set of initial configurations of $\mathcal{A}^-$ with all the clocks being 0 and the first component of the state (note that each state in $\mathcal{A}^-$ is a state pair of $\mathcal{A}$) being l_0. I is finite, thus Presburger-definable. ¿From Theorem 5 and the fact that $\mathcal{A}$ implements M, we have: M does not halt at l iff $\mathcal{A}^-$ has a P'-always ω-path starting from a configuration in I. Thus, we reduce the negation of the halting problem to the ∃-Presburger-always problem for dtas with configuration sets P' and I. Therefore,

Theorem 6. *The ∃-Presburger-always problem and the ∀-Presburger-eventual problem are undecidable for discrete timed automata.*

4.3 Undecidability of the ∀-Presburger-i.o. Problem

In this subsection, we show that the ∃-Presburger-almost-always problem is undecidable. Therefore, the ∀-Presburger-i.o. problem is also undecidable. In the previous subsection, we have shown that the existence of a P-always ω-path of $\mathcal{A}$ is undecidable. But this result does not directly imply that the existence of a P-almost-always ω-path is also undecidable.

In fact, let $\mathcal{A}^-$ be the static version of a generalized discrete timed automaton $\mathcal{A}$ that implements a deterministic counter machine M, let P be the path restriction of $\mathcal{A}$, and let p be an ω-path of $\mathcal{A}^-$. In the previous subsection, we argued that the existence of a P'-always ω-path p is undecidable where P' is $P \cap \{\langle\langle s, s'\rangle, \boldsymbol{V}\rangle : s \neq l\}$ with l being a given instruction label in M. But when considering a P'-almost-always path p, the situation is different: p may have a prefix that does not necessarily satisfy P' (i.e., it does not obey the exact enabling conditions on the edges in $\mathcal{A}$).

Consider a deterministic two-counter machine M with an input tape, and denote with $M(i)$ the result of the computation of M when given $i \in \mathbf{N}$ in input. It is known that the finiteness problem for deterministic two-counter machines (i.e., finitely many i such that $M(i)$ halts) is undecidable. Now we reduce the finiteness problem to the ∃-almost-always problem for dtas.

We can always assume that M halts when and only when it executes an operation labeled *halt*. Let M' be a counter machine (without input tape) that enumerates all the computations of M on every $i \in \mathbf{N}$. M' works as follows. We use $M_j(i)$ to denote the j-th step of the computation of $M(i)$. If $M(i)$ halts in less than j steps, then we assume that $M_j(i)$ is a special null operation that does nothing. Thus, the entire computation of $M(i)$ is an ω-sequence $M_1(i), \cdots, M_j(i), \cdots$ (when $M(i)$ halts, the sequence is

composed of a finite prefix, the halt operation and then infinitely many occurrences of the special null operation). Each step of the computation may or may not execute the instruction labeled *halt*, but of course an halt may be executed only at most once for each input value i. M' implements the following program:

$k := 0$; $z := 0$;
while true do
  $k := k + 1$;
  for $i := 0$ *to* $k - 1$ *do*
    $z := 1$;
    simulate $M(i)$ *for the first* k *steps* $M_1(i), M_2(i), \ldots, M_k(i)$*;*
    if $M_k(i)$ *executes the instruction labeled* halt*, then* $z := 0$*;*

M' is still a deterministic counter machine (with various additional counters to be able to simulate M and keep track of k, i, z). In the enumeration, whenever $M_k(i)$ executes the instruction labeled *halt* (at most once for each i, by the definition of M' as above), M' sets the counter z to 0, bringing it back to 1 immediately afterwards – M' resets z to 0 for only finitely many times iff the domain of M (i.e., the set of i such that $M(i)$ halts) is finite. Let $\mathcal{A}^-$ be the static version of a generalized discrete timed automaton $\mathcal{A}$ that implements M'. Let P be the path restriction of $\mathcal{A}$. P' is $P \cap \{\langle\langle s, s'\rangle, \boldsymbol{V}\rangle : \boldsymbol{V}_z \neq 0\}$. It can be established, by using Lemma 9 and Theorem 5, that there are only finitely many i such that $M(i)$ halts iff $\mathcal{A}^-$ is $\exists$-Presburger-almost-always for P' and I where I contains only the initial configuration. Therefore,

Theorem 7. *The $\exists$-Presburger-almost-always problem and the $\forall$-Presburger-i.o. problem are undecidable for discrete timed automata.*

5 Discussions and Future Work

It is important to provide a uniform framework to clarify what kind of temporal Presburger properties can be automatically checked for timed automata. Given a dta $\mathcal{A}$, the set of linear temporal logic formulas $\mathcal{L}_\mathcal{A}$ with respect to $\mathcal{A}$ is defined by the following grammar: $\phi := P|\neg\phi|\phi \wedge \phi|\bigcirc\phi|\phi U \phi$, where P is a Presburger-definable set of configurations of $\mathcal{A}$, $\bigcirc$ denotes "next", and U denotes "until". Formulas in $\mathcal{L}_\mathcal{A}$ are interpreted on ω-sequences p of configurations of $\mathcal{A}$ in a usual way. We use p^i to denote the ω-sequence resulting from the deletion of the first i configurations from p. We use p_i to indicate the i-th element in p. The satisfiability relation $\models$ is recursively defined as follows, for each ω-sequence p and for each formula $\phi \in \mathcal{L}_\mathcal{A}$ (written $p \models \phi$):

$p \models P$ if $p_1 \in P$,
$p \models \neg\phi$ if not $p \models \phi$,
$p \models \phi_1 \wedge \phi_2$ if $p \models \phi_1$ and $p \models \phi_2$,
$p \models \bigcirc\phi$ if $p^1 \models \phi$,
$p \models \phi_1 U \phi_2$ if $\exists j (p^j \models \phi_2$ and $\forall k < j (p^k \models \phi_1))$.

where the variables i, j, k range over $\mathbf{N}$. We adopt the convention that $\Diamond\phi$ (eventual) abbreviates $(trueU\phi)$ and $\Box\phi$ (always) abbreviates $(\neg\Diamond\neg\phi)$.

Given $\mathcal{A}$ and a formula $\phi \in \mathcal{L}_\mathcal{A}$, the model-checking problem is to check whether each ω-path p of $\mathcal{A}$ satisfies $p \models \phi$. The satisfiability-checking problem, which is the dual of the model-checking problem, is to check whether there is an ω-path p of $\mathcal{A}$ satisfying $p \models \phi$. The results of this paper show that:

- The satisfiability-checking problem is decidable for formulas in $\mathcal{L}_{\mathcal{A}}$ in the form $I \wedge \Box\Diamond P$ and $I \wedge \Diamond P$, where I and P are Presburger.
- The model-checking problem is undecidable for formulas in $\mathcal{L}_{\mathcal{A}}$, even when the formulas are in the form $\Box\Diamond P$ and $\Diamond P$.
- Hence, both the satisfiability-checking problem and the model-checking problem are undecidable for the entire $\mathcal{L}_{\mathcal{A}}$, even when the "next" operator $\bigcirc$ is excluded from the logic $\mathcal{L}_{\mathcal{A}}$.

Future work may include investigating a fragment of $\mathcal{L}_{\mathcal{A}}$ that has a decidable satisfiability-checking/model-checking problem. For instance, we don't know whether the satisfiability-checking problem is decidable for $I \wedge \Box\Diamond P \wedge \Box\Diamond Q$ (i.e., find an ω-path that is both P-i.o. and Q-i.o). A decidable subset of $\mathcal{L}_{\mathcal{A}}$ may be worked out along the recent work of Comon and Cortier [8] on model-checking a decidable subset of a Presburger (in the discrete case) LTL for one-cycle counter machines.

In [6], an extension of TPTL, called Presburger TPTL, is proposed and it is shown to be undecidable for discrete time. The proof in [6] does not imply (at least, not in an obvious way) the undecidability of the $\forall$-Presburger-i.o. problem and the $\forall$-Presburger-eventual problem in the paper. In that proof, the semantics of Presburger TPTL (over discrete time domain) is interpreted on timed state sequences. The transition relation of a two-counter machine can be encoded into Presburger TPTL by using $\bigcirc, U$ and the freeze quantifier. This gives the undecidability of the logic [6]. On the other hand, $\Box\Diamond P$ and $\Diamond P$ in this paper are interpreted on sequences of configurations (in contrast to timed state sequences). Formulas like $\Box\Diamond P$ and $\Diamond P$ are state formulas. That is, without using $\bigcirc$ and without introducing freeze quantifiers, we have no way to remember clock values in one configuration and use them to compare those in another configuration along p. Therefore, the transition relation of a two-counter machine cannot be encoded in our logic $\mathcal{L}_{\mathcal{A}}$. But we are able to show in this paper that computations of a two-counter machine can be encoded by ω-paths, restricted under $\Box\Diamond P$ or $\Diamond P$, of a dta, leading to the undecidability results of this paper.

We are also interested in considering the same set of liveness problems for a dense time domain. We believe that the decidability results (for the $\exists$-Presburger-i.o. problem and the $\exists$-Presburger-eventual problem) also hold for dense time when the semantics of a timed automaton is carefully defined. A possible approach is to look at Comon and Jurski's flattening construction [9]. The undecidability results in this paper can be naturally extended to the dense time domain when the ω-paths in this paper are properly redefined for dense time.

Thanks to anonymous reviewers for a number of useful suggestions.

References

1. R. Alur, *"Timed automata"*, *CAV'99*, LNCS 1633, pp. 8-22
2. R. Alur, C. Courcoubetis, and D. Dill, *"Model-checking in dense real time," Information and Computation*, **104** (1993) 2-34
3. R. Alur and D. Dill, *"Automata for modeling real-time systems," Theoretical Computer Science*, **126** (1994) 183-236
4. R. Alur, T. Feder, and T. A. Henzinger, *"The benefits of relaxing punctuality," J. ACM*, **43** (1996) 116-146

5. R. Alur, T. A. Henzinger, *"Real-time logics: complexity and expressiveness," Information and Computation*, **104** (1993) 35-77
6. R. Alur, T. A. Henzinger, *"A really temporal logic," J. ACM*, **41** (1994) 181-204
7. A. Coen-Porisini, C. Ghezzi and R. Kemmerer, *"Specification of real-time systems using ASTRAL," IEEE Transactions on Software Engineering*, 23 (1997) 572-598
8. H. Comon and V. Cortier, *"Flatness is not a weakness,"* Proc. Computer Science Logic, 2000.
9. H. Comon and Y. Jurski, *"Timed automata and the theory of real numbers," CONCUR'99*, LNCS 1664, pp. 242-257
10. Z. Dang, O. H. Ibarra, T. Bultan, R. A. Kemmerer, and J. Su, *"Binary reachability analysis of discrete pushdown timed automata," CAV'00*, LNCS 1855, pp. 69-84
11. T. A. Henzinger, Z. Manna, and A. Pnueli, *"What good are digital clocks?," ICALP'92*, LNCS 623, pp. 545-558
12. T. A. Henzinger and Pei-Hsin Ho, *"HyTech: the Cornell hybrid technology tool," Hybrid Systems II*, LNCS 999, pp. 265-294
13. T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine, *"Symbolic model checking for real-time systems," Information and Computation*, **111** (1994) 193-244
14. K. G. Larsen, P. Pattersson, and W. Yi, *"UPPAAL in a nutshell," International Journal on Software Tools for Technology Transfer*, **1** (1997): 134-152
15. F. Laroussinie, K. G. Larsen, and C. Weise, *"From timed automata to logic - and back," MFCS'95*, LNCS 969, pp. 529-539
16. F. Wang, *"Parametric timing analysis for real-time systems," Information and Computation*, 130 (1996): 131-150
17. S. Yovine, *"A verification tool for real-time systems," International Journal on Software Tools for Technology Transfer*, **1** (1997): 123-133
18. S. Yovine, *"Model checking timed automata," Embedded Systems'98*, LNCS 1494, pp. 114-152