The Timestamp of Timed Automata

Amnon Rosenmann(✉)

Graz University of Technology, Steyrergasse 30, 8010 Graz, Austria
rosenmann@math.tugraz.at

Abstract. Let eNTA be the class of non-deterministic timed automata with silent transitions. Given $A \in$ eNTA, we effectively compute its timestamp: the set of all pairs (time value, action) of all observable timed traces of A. We show that the timestamp is eventually periodic and that one can compute a simple deterministic timed automaton with the same timestamp as that of A. As a consequence, we have a partial method, not bounded by time or number of steps, for the general language non-inclusion problem for eNTA. We also show that the language of A is periodic with respect to suffixes.

Keywords: Timed automata · Timestamp of timed automata · Reachability problem · Language inclusion for timed automata

1 Introduction

Timed automata (TA) are finite automata extended with clocks that measure the time that elapsed since past events in order to control the triggering of future events. They were defined by Alur and Dill in their seminal paper [1] as abstract models of real-time systems and were implemented in tools like UPPAAL [20], Kronos [10], RED [27] and PRISM [19].

A fundamental problem in this area is the reachability problem, which in its basic form asks whether a given location of a timed automaton is reachable from the initial location. The set of states of the system (i.e., locations and valuation to the clocks) is, in general, an infinite uncountable set. However, through the construction of a region automaton, which contains finitely-many equivalence classes of regions [1], the reachability problem becomes a decidable problem (though of complexity PSPACE-complete).

Research on the reachability problem went beyond the above basic question. In [14] it is shown that the problem of the minimum and maximum reachability time is also PSPACE-complete. In another work, [13], which is more of a theoretical nature, the authors show that some problems on the relations between states may be defined in the decidable theory of the domain of real numbers equipped with the addition operation. In particular, the reachability problem between any two states is decidable. For other aspects of the reachability problem, also in the context of variants and extensions of timed automata (e.g. with game and probability characteristics) we refer to [3,5,11,14,17,18,26,28]. In this

É. André and M. Stoelinga (Eds.): FORMATS 2019, LNCS 11750, pp. 181–198, 2019.
https://doi.org/10.1007/978-3-030-29662-9_11

paper we generalize the reachability problem in another direction. We show that the problem of computing the set of all time values on which any observable transition occurs (and thus, a location is reached by an observable transition) is solvable. This set, called the *timestamp* of the automaton A and denoted $\mathbf{TS}(A)$, is more precisely defined to be the set of all pairs (t, a) that appear in the observable timed traces of A. Note that for this definition it does not matter whether we consider infinite runs or finite ones.

We show that the timestamp is in the form of a union of action-labeled open intervals with integral end-points, and action-labeled points of integral values. When the timestamp is unbounded in time then it is eventually periodic.

The set of languages defined by the class DTA of deterministic timed automata is strictly included in the set of languages defined by the class NTA of non-deterministic timed automata [1,16], and the latter is strictly included in the set of languages defined by the class eNTA of non-deterministic timed automata with silent transitions [8]. The fundamental problem of inclusion of the language accepted by a timed automaton A (e.g. the implementation) in the language accepted by the timed automaton B (e.g. the specification) is undecidable for the class NTA but decidable for the class DTA. On the other hand, for special sub-classes or modifications it was shown that decidability exists (see [2,4,6,8,9,21–24] for a partial list). However, the abstraction (or over-approximation) represented in the form of a timestamp is a discrete object, in which questions like inclusion of timestamps or universality are decidable. In fact, we show that for any given non-deterministic timed automaton with silent transitions, one can construct a simple deterministic timed automaton having the same timestamp.

The computation of the timestamp is done through the construction of a periodic augmented region automaton $\mathfrak{R}^t_{\mathrm{per}}(A)$. It is a region automaton augmented with a global non-resetting clock t and containing periodic regions and periodic transitions: they are defined modulo a time period $L \in \mathbb{N}$. This kind of abstraction demonstrates a periodic nature which is absent, in general, from timed traces: there are timed automata with no timed traces that are eventually periodic (see Example 1). Periodic transitions were introduced in [12], where it was shown that they increase the expressiveness of DTA, though they are less expressive than silent transitions.

The construction of the periodic automaton is preceded by defining the infinite augmented region automaton $\mathfrak{R}^t_{\infty}(A)$, in which the values of the clock t are unbounded. Then, after exhibiting the existence of a pattern that repeats itself every L time units, we fold the infinite automaton into a finite one according to this periodic structure.

Our construction shows that the language of a timed automaton $A \in$ eNTA is periodic with respect to suffixes: for every run ϱ with suffix ς that occurs after passing a fixed computable time there are infinitely-many runs of A with the same suffix ς, but with the suffix shifted in time by multiples of L. Note that this result does not follow from the pumping lemma, which does not hold in general in timed automata [7].

Due to lack of space, some proofs are either sketched or completely missing (for a longer version see [25]).

2 Timed Automata with Silent Transitions

A timed automaton is an abstract model aiming at capturing the temporal behavior of real-time systems. It is a finite automaton extended with a finite set of clocks defined over $\mathbb{R}_{\geq 0}$. It consists of a finite set of *locations* q with a finite set of *transitions* τ between the locations, while time, measured by the clocks, is continuous. A transition at time t can occur only if the condition expressed as a *transition guard* is satisfied at t. The transition is immediate - no clock is advancing in time. However, some of the clocks may be reset to zero.

There are two sorts of transitions: *observable* transitions, which can be traced by an outside observer, and *silent* transitions, which are inner transitions and thus cannot be observed from the outside. There are finitely-many types of observable transitions, each type labeled by a unique *action* $a \in \Sigma$, whereas all the silent transitions have the same label ϵ. In NTA, the class of non-deterministic timed automata, there exist states in which two transitions from the same location q can be taken at the same time and with the same action but to two different locations q' and q''. When this situation cannot happen, the TA is deterministic.

Let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and let $\mathcal{P}(S)$ be the power set of a set S. A transition guard is a conjunction of constraints of the form $c \sim n$, where c is a clock, $\sim \in \{<, \leq, =, \geq, >\}$ and $n \in \mathbb{N}_0$. A formal definition of eNTA is as follows.

Definition 1 (eNTA). *A* non-deterministic timed automaton with silent transitions $A \in$ eNTA *is a tuple* $(\mathcal{Q}, q_0, \Sigma_\epsilon, \mathcal{C}, \mathcal{T})$*, where:*

1. $\mathcal{Q}$ *is a finite set of locations and* q_0 *is the initial location;*
2. $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ *is a finite set of transition labels, called actions, where* Σ *refers to the observable actions and* ϵ *represents a silent transition;*
3. $\mathcal{C}$ *is a finite set of clock variables;*
4. $\mathcal{T} \subseteq \mathcal{Q} \times \Sigma_\epsilon \times \mathcal{G} \times \mathcal{P}(\mathcal{C}) \times \mathcal{Q}$ *is a finite set of transitions of the form* $(q, a, g, \mathcal{C}_{rst}, q')$*, where:*
   (a) $q, q' \in \mathcal{Q}$ *are the source and the target locations respectively;*
   (b) $a \in \Sigma_\epsilon$ *is the transition action;*
   (c) $g \in \mathcal{G}$ *is the* transition guard*;*
   (d) $\mathcal{C}_{rst} \subseteq \mathcal{C}$ *is the subset of clocks to be reset.*

A clock *valuation* v is a function $v : \mathcal{C} \to \mathbb{R}_{\geq 0}$. We denote by $\mathcal{V}$ the set of all clock valuations and by $\mathbf{d}$ the valuation which assigns the value d to every clock. Given a valuation v and $d \in \mathbb{R}_{\geq 0}$, we define $v + d$ to be the valuation $(v + d)(c) := v(c) + d$ for every $c \in \mathcal{C}$. The valuation $v[\mathcal{C}_{rst}]$, $\mathcal{C}_{rst} \subseteq \mathcal{C}$, is defined to be $v[\mathcal{C}_{rst}](c) = 0$ for $c \in \mathcal{C}_{rst}$ and $v[\mathcal{C}_{rst}](c) = v(c)$ for $c \notin \mathcal{C}_{rst}$.

The *semantics* of $A \in$ eNTA is given by the *timed transition system* $[\![A]\!] = (S, s_0, \mathbb{R}_{\geq 0}, \Sigma_\epsilon, T)$, where:

1. $S = \{(q, v) \in \mathcal{Q} \times \mathcal{V}\}$ is the set of states, with $s_0 = (q_0, \mathbf{0})$ the initial state;
2. $T \subseteq S \times (\Sigma_\epsilon \cup \mathbb{R}_{\geq 0}) \times S$ is the transition relation. The set T consists of
   (a) *Timed transitions (delays):* $(q, v) \xrightarrow{d} (q, v + d)$, where $d \in \mathbb{R}_{\geq 0}$;
   (b) *Discrete transitions (jumps):* $(q, v) \xrightarrow{a} (q', v')$, where $a \in \Sigma_\epsilon$ and there exists a transition $(q, a, g, \mathcal{C}_{rst}, q')$ in $\mathcal{T}$, such that for each clock c, $v(c)$ satisfies the constraints of g regarding c, and $v' = v[\mathcal{C}_{rst}]$.

A (finite) *run* ϱ of $A \in$ eNTA is a sequence of alternating timed and discrete transitions of the form

$$(q_0, \mathbf{0}) \xrightarrow{d_1} (q_0, \mathbf{d}_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{d_2} \cdots \xrightarrow{d_k} (q_{k-1}, v_{k-1} + d_k) \xrightarrow{a_k} (q_k, v_k)$$

and *duration* $T = \sum_{j=1}^{k} d_j$. The run ϱ of A induces the *timed trace* (*timed word*)

$$\lambda = (t_1, a_1), (t_2, a_2), \ldots, (t_k, a_k),$$

with $a_i \in \Sigma_\epsilon$ and $t_i = \Sigma_{j=1}^{i} d_j$. From the latter we can extract the *observable timed trace* (*observable timed word*), which is obtained by deleting from λ all the pairs containing silent transitions. Note that when the TA is deterministic then each timed trace refers to a unique run. We remark that we did not include the location invariants in the definition of timed automata since these invariants can be incorporated in the transition guards. We also do not distinguish between accepting and non-accepting locations as they do not change the analysis and results concerning the reachability problems that are dealt with here. Thus, the *language* $\mathfrak{L}(A)$ of A refers here to the set of observable timed traces of A without restricting it to those observable timed traces of runs that end in acceptable locations.

3 The Trail and Timestamp of a Single Path

Given a timed automaton $A \in$ eNTA over s clocks $x_1, \ldots, x_s$, we add to it a non-resetting global clock t that displays absolute time. A finite *path* in A has the form $\gamma = q_0 \tau_1 q_1 \tau_2 \cdots \tau_n q_n$ of alternating locations and transitions, with q_0 the initial location and τ_i a transition between q_{i-1} and q_i, $i = 1, \ldots, n$, that is, a path here refers to the standard definition in a directed graph. A run of the TA induces a *trajectory* in the non-negative part of the $tx_1 \cdots x_s$-space that is a piecewise-linear curve (the discontinuity is the clocks reset).

Definition 2 (Trajectory of a run). *Let* $\{t, x_1, \ldots, x_s\}$ *be an ordered set of clocks of* $A \in$ eNTA*. Let* ϱ *be a run of duration* T *of* A*. The* trajectory *of* ϱ *is the set of points* $(t, x_1, \ldots, x_s)$ *in the* $tx_1 \cdots x_s$*-space visited during* ϱ*, where* $0 \leq t \leq T$*.*

Next, we define the *trail* of a path.

Definition 3 (Trail of a path). *The* trail of a path γ *is the union of the trajectories of all feasible runs along* γ*, that is, runs that follow the locations and discrete transitions of* γ*.*

The *trail legs*, the parts of the trail between clocks reset, are in the form of *zones* [15], a conjunction of diagonal constraints $x_i - x_j < n_{ij}$ or $x_i - x_j \leq n_{ij}$, $n_{ij} \in \mathbb{Z}$, bounded by transition constraints $x_i \sim n_i$, where $\sim \in \{<, \leq, =, \geq, >\}$, $n_i \in \mathbb{N}_0$. Each trail leg can be further partitioned into *simplicial trails*, which are (possibly unbounded) parallelotopes consisting of a sequence of *regions* [1] arranged along the directional vector $\mathbf{1} = (1, 1, \ldots, 1)$. Each region $\mathbf{n} + \Delta$ is in the form of an open (unless it is a point) simplex Δ that is a hyper-triangle of dimension $0 \leq d \leq s+1$. The simplex Δ is characterized by the fractional values $\{x_i\}$ of the clock variables, and each point in the simplex satisfies the same fixed ordering of the form

$$0 \preceq_1 \{x_{i_1}\} \preceq_2 \{x_{i_2}\} \preceq_3 \cdots \preceq_s \{x_{i_s}\} < 1, \tag{1}$$

where $\preceq_i \in \{=, <\}$. The integral point $\mathbf{n} \in \mathbb{N}_0^{s+1}$ consists of the integral parts of the values of the clocks $x_0, x_1, \ldots, x_s$, and it indicates the lowest point in the $x_0 \cdots x_s$-space of the boundary of the region. Each region has a unique *immediate time-successor*, which is the next region along the directional vector $\mathbf{1}$, as long as no clock is reset on an event.

Definition 4 (Timestamp of a run). *The* timestamp of a run ϱ *is the set of pairs* $(t_i, a_i) \in \mathbb{R}_{\geq 0} \times \Sigma$ *of the observable timed trace induced by* ϱ*.*

Definition 5 (Timestamp of a path). *The* timestamp of a path γ *of* A *is the union of the timestamps of all runs* ϱ *along* γ*.*

Each instance of a transition along γ is an *event*.

Definition 6 (Timestamp of an event in a path). *The* timestamp of an event *in a path* γ *is the union of the timestamps of that event of all runs along* γ*. It is the part of the timestamp of the path that refers to that event.*

Proposition 1. *The timestamp of each event is either a labeled integral point or a labeled (open, closed or half-open) interval between points* m *and* n*,* $m < n$*,* $m \in \mathbb{N}_0$ *and* $n \in \mathbb{N} \cup \infty$*.*

Proof (sketch). It follows from the fact that the trail of each path is composed of simplices as in (1) residing on the integral grid, and such are the intersections with domains defined by transition constraints and projections due to clocks resets. Thus, it suffices to show that the timestamp of a single simplex Δ is of the required form and this follows from the fact that the simplex vertices are of integral value.

An alternative proof is via a linear programming problem over the variables t_i, where t_i represents the time of event i along a path γ, by showing that the minimum and maximum (if not infinite) of the solution set is integral. □

Definition 7 (Timestamp of a timed automaton). *The timestamp* $\mathbf{TS}(A)$ *of a timed automaton* A *is the set of all pairs* (t, a)*, such that an observable transition with action* a *occurs at time* t *in some run of* A*.*

4 Augmented and Infinite Augmented Region Automaton

4.1 Infinite Augmented Region Automaton

Given a (finite) timed automaton A, the region automaton $\mathfrak{R}(A)$ [1] is a finite *discretized* version of A, such that time is abstracted and both automata define the same untimed language. Each vertex in $\mathfrak{R}(A)$ records a location q in A and a region r, which is either in the form of a simplex (as described in Sect. 3) or an unbounded region, in which the value of at least one of the clocks is $\top$, meaning that it passed the maximal integer value M that appears in the transition guards. The regions partition the space of clock valuations into equivalence classes, where two valuations belong to the same equivalence class if and only if they agree on the clocks with $\top$ value and on the integral parts and the order among the fractional parts of the other clocks. The edges of $\mathfrak{R}(A)$ are labeled by the transition actions, and they correspond to the actual transitions that occur in the runs of A. Using the time-successor relation over the clock regions (see [1]), the region automaton can be effectively constructed. As shown in [1], through the region automaton the questions of reachable locations and states of A and the actions along the (possibly infinitely-many) paths that lead to these locations, i.e. the untimed language of A, become decidable.

Now we define the *infinite augmented region automaton* $\mathfrak{R}^t_\infty(A)$. First, we add to A a clock t that measures absolute time, does not appear in the transition guards, is never reset to 0 and does not affect the runs and timed traced of A. Next, we construct the region automaton augmented with t. The construction is similar to the construction of the standard region automaton with respect to the regular clocks (all clocks except for t) and the maximal bound M, that is, the time regions of each regular clock x_i are $\{0\}$, $(0,1)$, $\{1\}, (1,2), \dots, M, > M$, the latter being unbounded and refers to all values of x greater than M. The integration of the clock t is as follows. The construction of regions is as usual by considering the integral parts and the order of the fractional parts of all clocks, including t. The only difference is that the integral part of t is in $\mathbb{N}_0$ and not bounded by M. Thus, the infinitely-many time-regions associated with t are the alternating point and open unit interval: $\{0\}$, $(0,1)$, $\{1\}$, $(1,2), \dots$ (see Fig. 1(b)). Hence, $\mathfrak{R}^t_\infty(A)$ contains information about absolute time that is lacking from the standard region automaton.

Definition 8 (Infinite augmented region automaton). *Given* $A \in$ eNTA *extended with the clock* t *that measures absolute time, a corresponding* infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$ *is a tuple* $(V, v_0, E, \Sigma_\epsilon)$*, where:*

1. *V is an infinite (in general) set of vertices of the form $(q, \mathbf{n}, \Delta)$, where q is a location of A and the pair $(\mathbf{n}, \Delta)$ is a region, with*

$$\mathbf{n} = (n_0, n_1, \dots, n_s) \in \mathbb{N}_0 \times \{0, 1, \dots, M, \top\}^s \qquad (2)$$

containing the integral parts of the clocks $t, x_1, \dots, x_s$, and Δ is the simplex defined by the order of the fractional parts of the clocks.

2. *$v_0 = (q_0, \mathbf{0}, \mathbf{0})$ is the initial vertex with q_0 the initial location of A and with all clocks having integral part and fractional part equal to 0.*
3. *E is the set of edges. There is an edge*

$$(q, r) \xrightarrow{a} (q', r') \tag{3}$$

labeled with a in $\mathfrak{R}^t_\infty(A)$ if and only if there is a run of A which contains a timed transition followed by a discrete transition of the form

$$(q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v'), \tag{4}$$

such that the clock valuation v over $t, x_1, \ldots, x_s$ represents a point in the region r and the clock valuation v' represents a point in the region r'.
4. *$\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ is the finite set of actions that are edge labels.*

We note that there may be infinitely-many edges going-out of the same region in $\mathfrak{R}^t_\infty(A)$ (see Fig. 1(b)).

Proposition 2. *For each positive integer n, one can effectively construct the part of $\mathfrak{R}^t_\infty(A)$ which contains all regions with $t \leq n$ and all in-coming edges of these regions.*

The timestamp of the TA A, denoted $\mathbf{TS}(A)$, is the union of the timestamps of all observable transitions of A, that is, the set of all pairs (t, a), such that an observable transition with action a occurs at time t in some run of A. We define also the timestamp of $\mathfrak{R}^t_\infty(A)$.

Definition 9 (Timestamp of $\mathfrak{R}^t_\infty(A)$). *The timestamp of $\mathfrak{R}^t_\infty(A)$, $\mathbf{TS}(\mathfrak{R}^t_\infty(A))$, is the union of sets $s \times a$, where s is a time-region of t (an integral point $\{n\}$ or an open unit interval $(n, n+1)$) that is part of a region of a vertex of $\mathfrak{R}^t_\infty(A)$ and $a \in \Sigma$ is a label of an edge of $\mathfrak{R}^t_\infty(A)$ that is directed towards this vertex.*

Proposition 3. $\mathbf{TS}(A) = \mathbf{TS}(\mathfrak{R}^t_\infty(A))$.

Proof. By definition of the infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$, its regions are exactly the clock-regions which are visited by runs of the TA A extended with the clock t. In particular, the time-regions of $\mathfrak{R}^t_\infty(A)$ are the time-regions that are visited by the runs on the extended TA. Thus, $\mathbf{TS}(A) \subseteq \mathbf{TS}(\mathfrak{R}^t_\infty(A))$. By Proposition 1, this is an equality since for each open interval $(n, n+1)$ representing absolute time that is visited in some run of A on an action a, the set of all runs of A cover all the points of this interval with the same action a. □

4.2 Augmented Region Automaton

A second construction is the augmented region automaton, denoted $\mathfrak{R}^t(A)$, in which we consider only the fractional part of t and ignore its integral part.

$\mathfrak{R}^t(A)$ is a finite folding of $\mathfrak{R}^t_\infty(A)$, obtained by identifying vertices that contain the same data except for the integral part of t, and the corresponding edges. Thus, t has only two time-regions: $\{0\}$ and $(0, 1)$. As a compensation, we assign weights to the edges of $\mathfrak{R}^t(A)$, as explained below.

Definition 10 (Augmented region automaton). *Given a non-deterministic timed automaton with silent transitions $A \in$ eNTA, extended with the absolute-time clock t, a corresponding (finite)* augmented region automaton $\mathfrak{R}^t(A)$ *is a tuple $(V, v_0, E, \Sigma_\epsilon, W^*)$, where:*

1. *V is the set of vertices. Each vertex is a triple $(q, \mathbf{n}, \Delta)$, where q is a location of A and the pair $(\mathbf{n}, \Delta)$ is a region, with*

$$\mathbf{n} = (n_1, \ldots, n_s) \in \{0, 1, \ldots, M, \top\}^s \tag{5}$$

containing the integral parts of the clocks $x_1, \ldots, x_s$, and Δ is the simplex defined by the fractional parts of the clocks $t, x_1, \ldots, x_s$.
2. *$v_0 = (q_0, \mathbf{0}, \mathbf{0})$ is the initial vertex.*
3. *E is the set of edges. There is an edge $(q, r) \xrightarrow{a} (q', r')$ labeled with action a if and only if there is a run of A which contains a timed transition followed by a discrete transition of the form $(q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v')$, such that, when ignoring the integral part of the time measured by t, the clock valuation v represents a point in the region r and the clock valuation v' represents a point in the region r'.*
4. *$\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ is the finite set of actions.*
5. *W^* is the set of weights on the edges. Each weight m, possibly marked with '$*$', is $m = \lfloor t_1 \rfloor - \lfloor t_0 \rfloor \in [0..M]$, where $\lfloor t_1 \rfloor$ is the integral part of the value of t in the target location and $\lfloor t_0 \rfloor$ - in the source location in the corresponding run of A.*

There may be more than one edge between two vertices of $\mathfrak{R}^t(A)$, each one with a distinguished weight. A marked weight m^* represents infinitely-many consecutive values $m, m+1, m+2, \ldots$ as weights between the same two vertices, with m being the minimal value of such a sequence. It refers to a transition to or from a region r in which all regular clocks have passed the maximal integer M appearing in a transition guard as is illustrated in Fig. 1.

The languages $\mathfrak{L}(\mathfrak{R}^t(A))$ of $\mathfrak{R}^t(A)$ and $\mathfrak{L}(\mathfrak{R}^t_\infty(A))$ of $\mathfrak{R}^t_\infty(A)$ consist of all observable timed traces but, in contrast to the language $\mathfrak{L}(A)$ of A, in each pair (t_i, a_i) the time t_i is not exact: it is either an exact integer n or an arbitrary value of an interval $(n, n+1)$ that satisfies $t_i \geq t_{i-1}$. Thus, $\mathfrak{L}(\mathfrak{R}^t(A))$ and $\mathfrak{L}(\mathfrak{R}^t_\infty(A))$ are less abstract than the untimed language $\mathfrak{L}(\mathfrak{R}(A))$ of the region automaton $\mathfrak{R}(A)$ but are more abstract than $\mathfrak{L}(A)$: one cannot, in general, distinguish between a transition that occurs without any time delay, e.g. when $x_i \geq 0$, and a transition that demands a time delay, e.g. when $x_i > 0$. When comparing $\mathfrak{L}(\mathfrak{R}^t(A))$ and $\mathfrak{L}(\mathfrak{R}^t_\infty(A))$ then, since $\mathfrak{R}^t(A)$ may be obtained from $\mathfrak{R}^t_\infty(A)$, it is clear that $\mathfrak{L}(\mathfrak{R}^t(A))$ cannot be less abstract than $\mathfrak{L}(\mathfrak{R}^t_\infty(A))$. But, in fact, these region automata are equally informative: for each positive integer n, one can

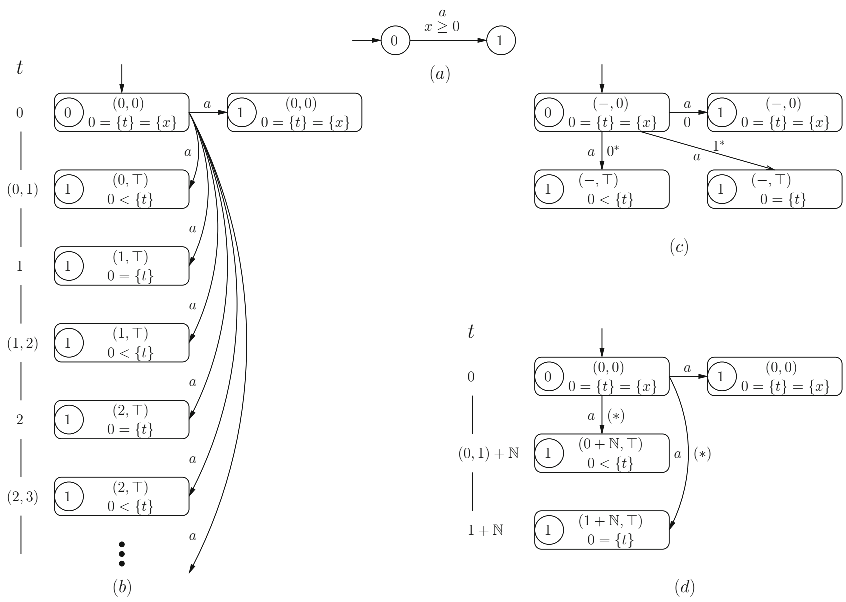

Fig. 1. (a) $A \in$ TA; (b) The infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$; (c) The augmented region automaton $\mathfrak{R}^t(A)$; (d) A periodic augmented region automaton $\mathfrak{R}^t_{\text{per}}(A)$. Each rectangle represents a vertex containing the location of A (circled, left), the integral values of t and x (top) and the simplex (bottom).

effectively construct $\mathfrak{R}^t_\infty(A)$ up to time $t = n$, as in Proposition 2, by unfolding $\mathfrak{R}^t(A)$ and recovering absolute time t by summing up the weights of the edges along the taken paths. Indeed, since the transitions in A do not rely on t, by taking the quotient of $\mathfrak{R}^t_\infty(A)$ by 'forgetting' the integral part of t, the only loss of information is the time difference in t between the target and source regions, but then this information is regained in the form of weight on the corresponding edge of $\mathfrak{R}^t(A)$. Thus, we have the following.

Proposition 4. $\mathfrak{L}(\mathfrak{R}^t(A)) = \mathfrak{L}(\mathfrak{R}^t_\infty(A))$.

5 Eventual Periodicity

In this section we address the main topic of this paper: exploring the time-periodic property of TA. In addition to demonstrating its existence, we show how one can actually compute the parameters of a period.

5.1 Non-Zeno Cycles in $\mathfrak{R}^t(A)$

$\mathfrak{R}^t(A)$ is in the form of a finite connected directed graph with an initial vertex. Every edge of $\mathfrak{R}^t(A)$ corresponds to a feasible transition in A (contained in a run of A). In what follows, a 'path' in $\mathfrak{R}^t(A)$ is a directed path that starts at the initial vertex g_0, unless otherwise stated.

Definition 11 (Duration of a path). *Given a path γ in $\mathfrak{R}^t(A)$, its minimal integral* duration*, or simply duration, $d(\gamma) \in \mathbb{N}_0$ is the sum of the weights on its edges, where a weight m^* is counted as m.*

Definition 12 ((Non)-Zeno cycle). *A cycle of $\mathfrak{R}^t(A)$ of duration* 0 *is called a* Zeno cycle*. Otherwise, it is a* non-Zeno cycle*.*

A path is called *simple* if no vertex of it repeats itself, and we let D be the maximal duration of a simple path in $\mathfrak{R}^t(A)$.

Lemma 1. *There exists a minimal positive integer $t_{\mathrm{nz}} \leq D+1$, the non-Zeno threshold time, such that every path γ of $\mathfrak{R}^t(A)$ that is of (minimal) duration t_{nz} or more contains a vertex belonging to some non-Zeno cycle.*

In order to compute t_{nz} we can explore the simple paths of $\mathfrak{R}^t(A)$, say in a breadth-first manner, up to the time t_0 in which each such path either cannot be extended to a path of a larger duration or any extension of it hits a vertex belonging to some non-Zeno cycle. Then $t_{\mathrm{nz}} = t_0 + 1$, which may be much smaller than $D + 1$.

5.2 A Period of $\mathfrak{R}^t(A)$

A set S is *minimal* with respect to some property if for every element $e \in S$ the set $S \setminus \{e\}$ does not satisfy the property.

Definition 13 (Covering set of non-Zeno cycles). *A set C of non-Zeno cycles of $\mathfrak{R}^t(A)$ is called a* covering set of non-Zeno cycles *if every path γ of $\mathfrak{R}^t(A)$ whose duration $d(\gamma)$ is at least t_{nz} intersects a cycle in C in a common vertex.*

Without loss of generality, we may assume that a covering set of non-Zeno cycles is minimal.

Definition 14 (Period of $\mathfrak{R}^t(A)$). *A time* period *(or just period) L of $\mathfrak{R}^t(A)$ is a common multiple of the set of durations $d(\pi)$, $\pi \in C$, for some fixed (minimal) covering set of non-Zeno cycles C. For convenience, we also set L to be greater than M, unless $\mathfrak{R}^t(A)$ does not contain non-Zeno cycles, in which case we define L to be 0.*

5.3 Eventual Periodicity of $\mathfrak{R}^t_\infty(A)$

Let t_{nz}, C, L be as above, with C fixed. We denote by $\mathfrak{R}^t_\infty(A)|_{t \geq n}$ the subgraph of $\mathfrak{R}^t_\infty(A)$ that starts at time-level n, that is, the set of vertices of $\mathfrak{R}^t_\infty(A)$ with absolute time $t \geq n$ and their out-going edges.

Definition 15 (L-shift in time). *Given a subgraph G of $\mathfrak{R}^t_\infty(A)$, an* L-shift in time *of G, denoted $G+L$, is the graph obtained by adding the value L to each value of the integral part of the clock t in G and leaving the rest of the data unaltered. We also denote by $V(G)+L$ the L-shift in time for the set of vertices of G, with $v+L$ in case $V=\{v\}$.*

Lemma 2. *If $\mathfrak{R}^t_\infty(A)$ is not bounded in time then*

$$\mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{nz}}} + L \subseteq \mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{nz}}+L}.$$

Proof. First we show that the inclusion holds for the set of vertices of the above subgraphs. Let γ be a path of $\mathfrak{R}^t_\infty(A)$ which terminates in a vertex $v_1 \in \mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{nz}}}$. Let $\gamma' = p(\gamma)$ be the image of γ under the projection to $\mathfrak{R}^t(A)$. If γ contains an edge e_1 whose image $e'_1 = p(e_1)$ is labeled by a marked weight m^* then we can replace e_1 by another edge $e_2 \in p^{-1}(e'_1)$ whose delay is greater by L than the delay of e_1. So, suppose that e_1 starts in the vertex u_1 and terminates in w_1. Then e_2 starts in u_1 and terminates in the vertex $w_2 = w_1 + L$ and then the path continues as in γ but with an L-shift in time, terminating in the vertex $v_2 = v_1 + L$. Otherwise, no edge of γ' has a marked weight. Since $d(\gamma) \geq t_{\mathrm{nz}}$ then by Lemma 1 and the definition of L, γ' contains a vertex v' that belongs to a non-Zeno cycle π and whose duration is a factor of L. Hence, by a 'pumping' argument, we can extend γ' with $L/d(\pi)$ cycles of π that start and end in v' and then reach the vertex $v_2 = v_1 + L$ in the pre-image in $\mathfrak{R}^t_\infty(A)$ of this extended path.

The inclusion of the out-going edges follows from the fact that the out-going edges do not depend on the value of t. □

Let us denote by V_k, $k = 0, 1, 2, \ldots$, the set of vertices

$$V_k = V(\mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{nz}}+kL}) \setminus V(\mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{nz}}+(k+1)L}).$$

Theorem 1. *If the infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$ is not bounded in time then it is eventually periodic: there exists an integral time $t_{\mathrm{per}} > 0$ such that*

$$\mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{per}}} + L = \mathfrak{R}^t_\infty(A)|_{t\geq t_{\mathrm{per}}+L}.$$

Proof. By Lemma 2, $V_k + L \subseteq V_{k+1}$, for $k \geq 0$. But there is a bound on the number of possible vertices of V_k since t is bounded, hence the sequence V_k eventually stabilizes. The result then follows since for the out-going edges the same argument given in the proof of Lemma 2 holds also here. □

When $\mathfrak{R}^t_\infty(A)$ is finite then we can set t_{per} to be $t_{\max} + 1$, where $t_{\max}$ is the maximal integral time of $\mathfrak{R}^t_\infty(A)$. By the following proposition, a possible value for t_{per} can be effectively computed when $\mathfrak{R}^t_\infty(A)$ is infinite.

Proposition 5. *if $|V_k| = |V_{k+1}| = |V_{k+2}|$ for some k then we can set $t_{\mathrm{per}} = t_{\mathrm{nz}} + kL$.*

As is known, a TA may be totally non-periodic in the sense that no single timed trace of it is eventually periodic (see Example 1). However, a special kind of periodicity, which we call *suffix-periodicity*, holds between different timed traces, as shown in the following theorem.

Theorem 2. *If $A \in$ eNTA is not bounded in time then its language $\mathfrak{L}(A)$ is suffix-periodic: if $t_r > t_{\mathrm{per}}$ and*

$$\lambda = (t_1, a_1), \ldots, (t_{r-1}, a_{r-1}), (t_r, a_r), (t_{r+1}, a_{r+1}), \ldots, (t_{r+m}, a_{r+m})$$

is an observable timed trace of $\mathfrak{L}(A)$ then, for each $k \in L\mathbb{Z}$, if $t_r + k > t_{\mathrm{per}}$ then there exists an observable timed trace $\lambda' \in \mathfrak{L}(A)$ such that

$$\lambda' = (t'_1, a'_1), \ldots, (t'_s, a'_s), (t_r + k, a_r), (t_{r+1} + k, a_{r+1}), \ldots, (t_{r+m} + k, a_{r+m}).$$

6 Periodic Augmented Region Automaton

After revealing the periodic structure of $\mathfrak{R}^t_\infty(A)$, it is natural to fold it into a finite graph according to this period, which we call periodic augmented region automaton and denote by $\mathfrak{R}^t_{\mathrm{per}}(A)$. The construction of $\mathfrak{R}^t_{\mathrm{per}}(A)$ is done by first taking the subgraph of $\mathfrak{R}^t_\infty(A)$ of time $t < t_{\mathrm{per}} + L$ and then folding the infinite subgraph of $\mathfrak{R}^t_\infty(A)$ of time $t \geq t_{\mathrm{per}} + L$ onto the subgraph of time $t_{\mathrm{per}} \leq t < t_{\mathrm{per}} + L$, which becomes the periodic subgraph, as explained below. For an edge e, we denote by $\iota(e)$ and $\tau(e)$ the initial, resp. terminal, vertex of e.

Definition 16 (Periodic augmented region automaton). *Given an infinite augmented region automaton $\mathfrak{R}^t_\infty(A)$ with period L and periodicity starting time t_{per}, a finite projection $p(\mathfrak{R}^t_\infty(A))$ of it, called* periodic augmented region automaton *and denoted $\mathfrak{R}^t_{\mathrm{per}}(A)$, is a tuple $(V, v_0, E, \Sigma_\epsilon, B)$, where:*

1. *V is the set of vertices, with $v_0 = (q_0, \mathbf{0}, \mathbf{0})$ the initial vertex. For each $v \in \mathfrak{R}^t_{\mathrm{per}}(A)$, if $u \in p^{-1}(v) \subseteq \mathfrak{R}^t_\infty(A)$ then u equals v in all fields, except possibly for the integral part of t. If $v.\lfloor t \rfloor < t_{\mathrm{per}}$ then $u = v$ and v is a* regular *vertex. Otherwise, v is a* periodic *vertex, $v.\lfloor t \rfloor$ is written as $n + L\mathbb{N}_0$, for some $t_{\mathrm{per}} \leq n < t_{\mathrm{per}} + L$, $p^{-1}(v)$ is infinite and $\{u.\lfloor t \rfloor \mid p(u) = v\} = \{n + kL \mid k = 0, 1, 2, \ldots\}$.*
2. *E is the set of edges, which are the projected edges of $\mathfrak{R}^t_\infty(A)$ under the map p. Each edge joining two vertices of $\mathfrak{R}^t_\infty(A)$ is mapped to an edge with the same action label that joins the projected vertices. Some of the edges are marked with a symbol of $B = \{(*), (*+)\}$. The description below is technical and refers to the different types of edges that occur when folding $\mathfrak{R}^t_\infty(A)$: whether the source of the edge is a regular* (**R**) or a periodic (**P**) vertex (in the latter case the preimage in $\mathfrak{R}^t_\infty(A)$ contains infinitely-many edges, one from each of the preimage vertices), whether it is unmarked (**U**) or marked (**M**) (in the latter case there are infinitely-many edges starting from each of the vertices in the preimage source vertices), and finally the plus sign (+) represents the case where in the preimage the target vertices are not of value n but $n + L$.

- **UR:** (unmarked, regular) If $e \in \mathfrak{R}^t_{\mathrm{per}}(A)$ is unmarked and $\iota(e)$ is regular then $\iota(e).\lfloor t \rfloor = n_1 < t_{\mathrm{per}}$, $\tau(e).\lfloor t \rfloor = n_2$ or $\tau(e).\lfloor t \rfloor = n_2 + L\mathbb{N}_0$ and $p^{-1}(e) = \{e'\}$, with $\iota(e').\lfloor t \rfloor = n_1$ and $\tau(e').\lfloor t \rfloor = n_2$.
- **UP:** (unmarked, periodic) If $e \in \mathfrak{R}^t_{\mathrm{per}}(A)$ is unmarked and $\iota(e)$ is periodic then $\iota(e).\lfloor t \rfloor = n_1 + L\mathbb{N}_0$, $\tau(e).\lfloor t \rfloor = n_2 + L\mathbb{N}_0$, $t_{\mathrm{per}} \le n_1, n_2 < t_{\mathrm{per}} + L$ and the preimage of e in $\mathfrak{R}^t_{\infty}(A)$ are the infinitely-many edges satisfying the following. If $n_1 \le n_2$ then $p^{-1}(e) = \{e' \mid \iota(e').\lfloor t \rfloor = n_1 + kL, \tau(e').\lfloor t \rfloor = n_2 + kL, k = 0, 1, 2, \ldots\}$, and if $n_1 > n_2$ then $p^{-1}(e) = \{e' \mid \iota(e').\lfloor t \rfloor = n_1 + kL, \tau(e').\lfloor t \rfloor = n_2 + (k+1)L, k = 0, 1, 2, \ldots\}$.
- **MR:** (marked, regular) If $e \in \mathfrak{R}^t_{\mathrm{per}}(A)$ is marked with '$(*)$' and $\iota(e)$ is regular, with $\iota(e).\lfloor t \rfloor = n_1$ and $\tau(e).\lfloor t \rfloor = n_2$ or $n_2 + L\mathbb{N}_0$, then $p^{-1}(e) = \{e' \mid \iota(e').\lfloor t \rfloor = n_1, \tau(e').\lfloor t \rfloor = n_2 + kL, k = 0, 1, 2, \ldots\}$, that is, infinitely-many edges starting from the same vertex.
- **MP:** (marked, periodic) If $e \in \mathfrak{R}^t_{\mathrm{per}}(A)$ is marked with '$(*)$' and $\iota(e)$ is periodic, with $\iota(e).\lfloor t \rfloor = n_1 + L\mathbb{N}_0$ and $\tau(e).\lfloor t \rfloor = n_2 + L\mathbb{N}_0$, then its preimage in $\mathfrak{R}^t_{\infty}(A)$ contains all the edges according to both rules **UP** and **MR**.
- **MP+:** (marked, periodic, shifted) If $e \in \mathfrak{R}^t_{\mathrm{per}}(A)$ is marked with '$(*+)$' then the same rules that apply to an edge marked with '$(*)$' hold, except that the target vertices are of L-shift in time compared to those of an edge marked with '$(*)$'.

3. $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ is the finite set of actions.

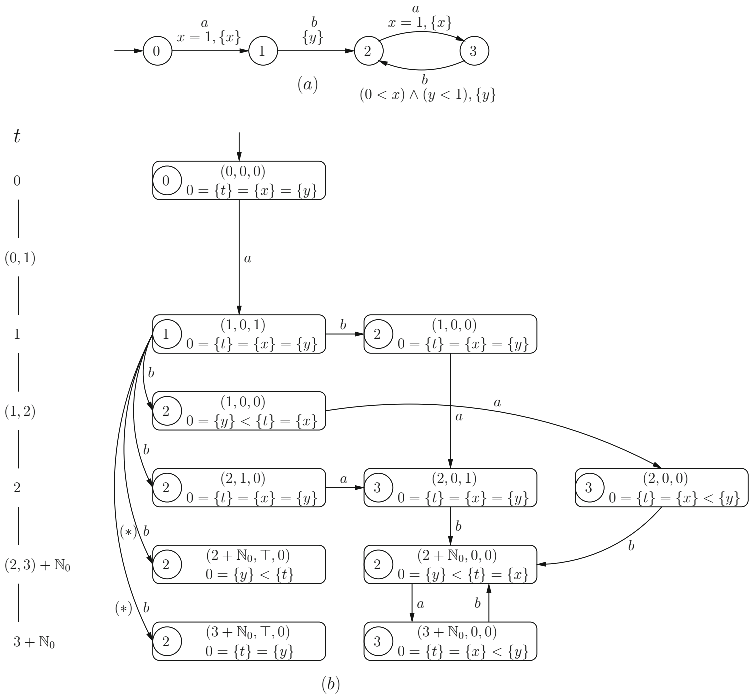
0
1
2
3
a
$x = 1, \{x\}$
b
$\{y\}$
a
$x = 1, \{x\}$
b
$(0 < x) \wedge (y < 1), \{y\}$
(a)
t
0
$(0,1)$
1
$(1,2)$
2
$(2,3) + \mathbb{N}_0$
$3 + \mathbb{N}_0$
$(0,0,0)$
$0 = \{t\} = \{x\} = \{y\}$
a
$(1,0,1)$
$0 = \{t\} = \{x\} = \{y\}$
b
$(1,0,0)$
$0 = \{t\} = \{x\} = \{y\}$
b
$(1,0,0)$
$0 = \{y\} < \{t\} = \{x\}$
a
a
b
$(2,1,0)$
$0 = \{t\} = \{x\} = \{y\}$
a
$(2,0,1)$
$0 = \{t\} = \{x\} = \{y\}$
$(2,0,0)$
$0 = \{t\} = \{x\} < \{y\}$
$(*)$
b
b
b
$(2 + \mathbb{N}_0, \top, 0)$
$0 = \{y\} < \{t\}$
$(2 + \mathbb{N}_0, 0, 0)$
$0 = \{y\} < \{t\} = \{x\}$
$(*)$
b
a
b
$(3 + \mathbb{N}_0, \top, 0)$
$0 = \{t\} = \{y\}$
$(3 + \mathbb{N}_0, 0, 0)$
$0 = \{t\} = \{x\} < \{y\}$
(b)

Fig. 2. (a) $A \in \mathrm{TA}$; (b) $\mathfrak{R}^t_{\mathrm{per}}(A)$, a periodic augmented region automaton of A

We remark that instead of periodic time interval of type $[a, b)$ we can define it analogously to be of type $(a, b]$ as in Fig. 1(d), where the periodic time is $(0, 1]$.

Example 1. The TA shown in Fig. 2(a) is taken from [1], where it demonstrates non-periodicity: the time difference between an a-transition and the following b-transition is strictly decreasing along a run. $\mathfrak{R}^t_{\text{per}}(A)$, however, becomes periodic (Fig. 2(b)).

Proposition 6. *$\mathfrak{R}^t_{\text{per}}(A)$ is well-defined and as informative as $\mathfrak{R}^t_{\infty}(A)$.*

7 The Timestamp

Theorem 3. *The timestamp of a TA A is a union of action-labeled integral points and open unit intervals with integral end-points. It is either finite or forms an eventually periodic (with respect to time t) subset of $\mathbb{R}_{\geq 0} \times \Sigma$ and is effectively computable.*

Proof. By Theorem 2, if the timestamp is not finite then it becomes periodic, with period L, after time $t = t_{\text{per}}$. Thus, if it can effectively be computed up to time $t_{\text{per}} + L$, then in order to find whether there is an observable transition with action a at time $t_{\text{per}} + L + t$ one only needs to check the timestamp at time $t_{\text{per}} + (t \mod L)$.

By Proposition 1, the timestamp up to time $t_{\text{per}} + L$ is a finite number of labeled integral points and open intervals between integral points and by Proposition 2, it is effectively computable. □

The timestamp of a TA is an abstraction of its language. However, the timestamp is eventually periodic and computable, hence the timestamp inclusion problem is decidable.

Corollary 1. *Given two timed automata $A, B \in$ eNTA over the same alphabet (action labels), the question of non-inclusion of their timestamps is decidable, thus providing a decidable sufficient condition for the (in general, undecidable) question of non-inclusion of their languages: $\mathfrak{L}(A) \nsubseteq \mathfrak{L}(B)$.*

The timestamp is easily extracted from $\mathfrak{R}^t_{\text{per}}$ (in fact, it is enough to take the subgraph of $\mathfrak{R}^t_{\infty}$ up to level $t_{\text{per}} + L$). We just form the union of the time-regions up to level $t_{\text{per}} + L$, where each time-region is either a point $\{n\}$ or an open interval $(n, n+1)$, along with the labels of the actions of the in-going edges. The timestamp in the interval $t_{\text{per}} \leq t < t_{\text{per}} + L$ then repeats itself indefinitely.

Definition 17. *For each $a \in \Sigma$, let A_a be the restriction of A to a-actions, obtained by substituting each $b \in \Sigma \setminus \{a\}$ with ϵ, representing the silent transition.*

Thus, the language of A_a is the 'censored' language of A, which is the outcome of deleting from each word (timed trace) all pairs (b, t), $b \neq a$.

Example 2. The timestamp of the a-transitions of the automaton of Fig. 2 is $\mathbf{TS}(A_a) = \mathbb{N}$, and that of the b-transitions is $\mathbf{TS}(A_b) = [1, \infty)$.

7.1 Timestamp Automaton

Given a TA A, one can effectively construct a deterministic TA $\tilde{A}$, called a *timestamp automaton* of A with the same timestamp as that of A.

Definition 18 (Timestamp automaton). *Given a timed automaton $A \in$ eNTA, a timestamp automaton $\tilde{A}$ is a deterministic (finite) timed automaton with a single clock and with timestamp identical to that of A. It is the union of the timestamp automata $\tilde{A}_a$, $a \in \Sigma$, having a common initial vertex. Each $\tilde{A}_a$ is in the shape of a linear graph and possibly ending in a simple loop.*

Theorem 4. *Given a timed automaton $A \in$ eNTA, one can effectively construct a timestamp automaton $\tilde{A}$.*

Example 3. Let A be a TA with timestamp

$$\begin{aligned}
\mathbf{TS}(A_a) &= (1,3] \cup \{5\} \cup (6 + ([0,2) \cup \{3\} \cup (8,18)) + 21\mathbb{N}_0) \times \{a\}, \\
\mathbf{TS}(A_b) &= [0,1] \cup (2,4) \cup \{5\} \cup (6 + ((0,1) \cup (1,2) \cup (5,6) \cup (8,9)) + 10\mathbb{N}_0) \\
&\quad \times \{b\}, \\
\mathbf{TS}(A_c) &= [1,4] \cup \{6\} \cup (10,\infty) \times \{c\}.
\end{aligned}$$

Then a possible timestamp automaton of A is given in Fig. 3.

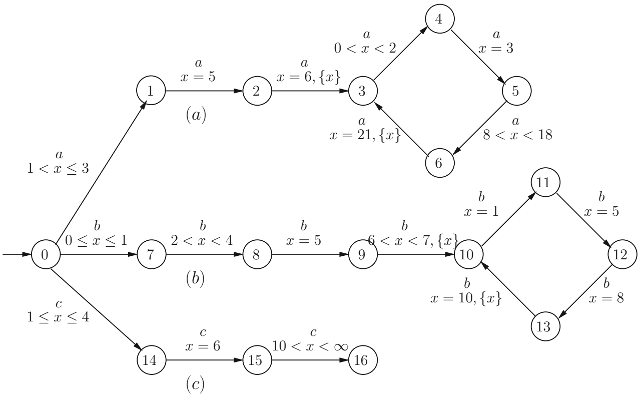

Fig. 3. Timestamp automata of (a) $\mathbf{TS}(A_a)$; (b) $\mathbf{TS}(A_b)$; (c) $\mathbf{TS}(A_c)$

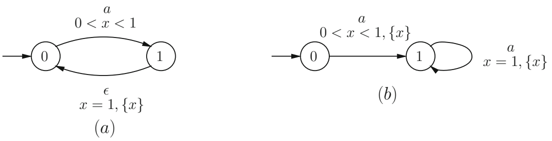

Fig. 4. (a) A non-determinizable $A \in$ eNTA; (b) A timestamp automaton $\tilde{A}$

Example 4. The language of the TA $A \in$ eNTA of Fig. 4(a) is

$$\mathfrak{L}(A) = \{(t_0, a), (t_1, a), \ldots, (t_n, a) \mid i < t_i < i + 1, i = 0, \ldots, n - 1, n \in \mathbb{N}_0\}$$

(supposing all locations are 'accepting'). The timestamp of A is the set of all positive non-integral reals: $\mathbf{TS}(A) = \mathbb{R}_{\geq 0} \setminus \mathbb{N}_0$. A is not determinizable. Each transition occurs between the next pair of successive natural numbers. The guard of each such transition must refer to a clock which was reset on some previous integral time. But since all transitions occur on non-integral time, the only clock that can be referred to is a clock x that is reset at time 0 and hence the transition guards need to be of the form $n < x < n + 1$ for each $n \in \mathbb{N}_0$, which makes the automaton infinite. Nevertheless, the timestamp automaton associated with A, seen in Fig. 4(b), is deterministic.

8 Conclusion and Future Research

The timestamp of a non-deterministic timed automaton with silent transitions (eNTA) consists of the set of all action-labeled times at which locations can be reached by observable transitions. The problem of computing the timestamp is a generalization of the basic reachability problem, a fundamental problem in model checking, thus being of interest from the theoretical as well as from the practical point of view. In this paper we showed that the timestamp can be effectively computed, also when the timed automata are non-deterministic and include silent transitions.

One of the major problems in testing and verification of abstract models of real-time systems is the inclusion of the language of one timed automaton in the language of another timed automaton. This problem is, in general, undecidable. Thus, since (non)-inclusion of timestamps of timed automata is a decidable problem, we have a tool which provides a sufficient condition for language non-inclusion in timed automata.

Acknowledgements. This research was partly supported by the Austrian Science Fund (FWF) Project P29355-N35.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994). https://doi.org/10.1016/0304-3975(94)90010-8
2. Alur, R., Fix, L., Henzinger, T.A.: Event-clock automata: a determinizable class of timed automata. Theor. Comput. Sci. **211**(1–2), 253–273 (1999). https://doi.org/10.1016/S0304-3975(97)00173-4
3. Alur, R., Kurshan, R.P., Viswanathan, M.: Membership questions for timed and hybrid automata. In: Real-Time Systems Symposium, pp. 254–263 (1998). https://doi.org/10.1109/REAL.1998.739751
4. Alur, R., Madhusudan, P.: Decision problems for timed automata: a survey. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 1–24. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30080-9_1

5. Asarin, E., Maler, O.: As soon as possible: time optimal control for timed automata. In: Vaandrager, F.W., van Schuppen, J.H. (eds.) HSCC 1999. LNCS, vol. 1569, pp. 19–30. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48983-5_6
6. Baier, C., Bertrand, N., Bouyer, P., Brihaye, T.: When are timed automata determinizable? In: ICALP (2), pp. 43–54 (2009). https://doi.org/10.1007/978-3-642-02930-1_4
7. Beauquier, D.: Pumping lemmas for timed automata. In: Nivat, M. (ed.) FoSSaCS 1998. LNCS, vol. 1378, pp. 81–94. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0053543
8. Bérard, B., Petit, A., Diekert, V., Gastin, P.: Characterization of the expressive power of silent transitions in timed automata. Fundam. Inform. **36**(2–3), 145–182 (1998). https://doi.org/10.3233/FI-1998-36233
9. Bouyer, P., Dufourd, C., Fleury, E., Petit, A.: Updatable timed automata. Theor. Comput. Sci. **321**(2–3), 291–345 (2004). https://doi.org/10.1016/j.tcs.2004.04.003
10. Bozga, M., Daws, C., Maler, O., Olivero, A., Tripakis, S., Yovine, S.: Kronos: a model-checking tool for real-time systems. In: Hu, A.J., Vardi, M.Y. (eds.) CAV 1998. LNCS, vol. 1427, pp. 546–550. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0028779
11. Chen, T., Han, T., Katoen, J., Mereacre, A.: Reachability probabilities in Markovian timed automata. In: CDC-ECC, pp. 7075–7080 (2011). https://doi.org/10.1109/CDC.2011.6160992
12. Choffrut, C., Goldwurm, M.: Timed automata with periodic clock constraints. J. Autom. Lang. Comb. **5**(4), 371–403 (2000). https://doi.org/10.25596/jalc-2000-371
13. Comon, H., Jurski, Y.: Timed automata and the theory of real numbers. In: Baeten, J.C.M., Mauw, S. (eds.) CONCUR 1999. LNCS, vol. 1664, pp. 242–257. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48320-9_18
14. Courcoubetis, C., Yannakakis, M.: Minimum and maximum delay problems in real-time systems. Form. Methods Syst. Des. **1**(4), 385–415 (1992). https://doi.org/10.1007/BF00709157
15. Daws, C., Tripakis, S.: Model checking of real-time reachability properties using abstractions. In: Steffen, B. (ed.) TACAS 1998. LNCS, vol. 1384, pp. 313–329. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054180
16. Finkel, O.: Undecidable problems about timed automata. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 187–199. Springer, Heidelberg (2006). https://doi.org/10.1007/11867340_14
17. Haase, C., Ouaknine, J., Worrell, J.: On the relationship between reachability problems in timed and counter automata. In: Finkel, A., Leroux, J., Potapov, I. (eds.) RP 2012. LNCS, vol. 7550, pp. 54–65. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33512-9_6
18. Henzinger, T.A., Prabhu, V.S.: Timed alternating-time temporal logic. In: Asarin, E., Bouyer, P. (eds.) FORMATS 2006. LNCS, vol. 4202, pp. 1–17. Springer, Heidelberg (2006). https://doi.org/10.1007/11867340_1
19. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47
20. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. STTT **1**(1–2), 134–152 (1997). https://doi.org/10.1007/s100090050010
21. Lorber, F., Rosenmann, A., Nickovic, D., Aichernig, B.K.: Bounded determinization of timed automata with silent transitions. Real Time Syst. **53**(3), 291326 (2017). https://doi.org/10.1007/s11241-017-9271-x

22. Ouaknine, J., Rabinovich, A., Worrell, J.: Time-bounded verification. In: Bravetti, M., Zavattaro, G. (eds.) CONCUR 2009. LNCS, vol. 5710, pp. 496–510. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04081-8_33
23. Ouaknine, J., Worrell, J.: On the language inclusion problem for timed automata: closing a decidability gap. In: LICS, pp. 54–63 (2004). https://doi.org/10.1109/LICS.2004.1319600
24. Ouaknine, J., Worrell, J.: Towards a theory of time-bounded verification. In: ICALP (2), pp. 22–37 (2010). https://doi.org/10.1007/978-3-642-14162-1_3
25. Rosenmann, A.: The timestamp of timed automata. arXiv abs/1412.5669v4 (2019). http://arxiv.org/abs/1412.5669
26. Tripakis, S., Yovine, S.: Analysis of timed systems using time-abstracting bisimulations. Form. Methods Syst. Des. **18**(1), 25–68 (2001). https://doi.org/10.1023/A:1008734703554
27. Wang, F.: Efficient verification of timed automata with BDD-like data structures. STTT **6**(1), 77–97 (2004). https://doi.org/10.1007/s10009-003-0135-4
28. Wozna, B., Zbrzezny, A., Penczek, W.: Checking reachability properties for timed automata via SAT. Fundam. Inform. **55**(2), 223–241 (2003)