

The Timestamp of Timed Automata

Amnon Rosenmann

Institute of Discrete Mathematics
Graz University of Technology, Steyrergasse 30, A-8010 Graz, Austria
rosenmann@math.tugraz.at

Abstract

Given a member A of the class of non-deterministic timed automata with silent transitions (eNTA), we show how one can effectively compute its timestamp: the set of all pairs of time values and the corresponding actions of all observable timed traces of A , and also a deterministic timed automaton with the same timestamp as that of A . The timestamp is eventually periodic and is constructed via a finite periodic augmented region automaton. A consequence of this construction is the periodicity of the language of timed automata with respect to suffixes. Applications include the decidability of the 1-bounded language inclusion problem for the class eNTA, and a partial method, not bounded by time or number of steps, for the general language non-inclusion problem for eNTA.

Keywords: timed automaton, timestamp of a timed automaton, periodic augmented region automaton, reachability problem, eventual periodicity

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 [AD94] as abstract models of real-time systems, and were implemented in tools like UPPAAL [LPY97], Kronos [BDM⁺98] and RED [Wan04].

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 [AD94], the reachability problem becomes a decidable problem (though of complexity PSPACE-complete), since the region automaton contains only reachable regions.

Research on the reachability problem went beyond the above basic question. In [CY92] it is shown that the problem of the minimum and maximum reachability time is also PSPACE-complete. In another work, [CJ99], 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 [CY92],[AKV98], [TY01], [WZP03], [AM99], [HP06], [CHKM11], [HOW12]. In this 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 $\mathfrak{TS}(A)$, is more precisely defined to be the set of all pairs (t, a) that appear in the observable timed traces of A . In other words, we transform each timed trace of A from a sequence of pairs (t, a) to a set of such pairs, and then take the union, over all timed traces, of these sets. Note that for this definition it does not matter whether we consider infinite runs or finite ones.

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 [AD94], [Fin06], and the latter is strictly included in the set of languages defined by the class eNTA of non-deterministic timed automata with silent transitions [BPDG98]. 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 [BBBB09, BPDG98, AFH99, BDFP04, OW04, ORW09, OW10, AM04] 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.

One application of the effectiveness of computing the timestamp is the decidability of the 1-bounded language inclusion problem for the class eNTA. That is, for any two $A, B \in \text{eNTA}$, it is decidable whether the set of timed words of length 1 (the set of pairs of time values of first possible occurrences of observable actions, or events, together with the corresponding actions), of A is included in that of B . Another application is a partial method, which is not bounded by time or number of steps, for demonstrating a witness for the refutation of the general language inclusion problem for eNTA.

The computation of the timestamp is done through the construction of a

periodic augmented region automaton. It is a region automaton augmented with a global non-resetting clock t , and which contains 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 6.3). Periodic transitions were introduced in [CG00], 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, 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.

In terms of the language of a timed automaton $A \in \text{eNTA}$, we show that it 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 [Bea98].

In Section 2 basic definitions concerning timed automata are given. Then, in Section 3 we describe the trail and timestamp of a single path of a timed automaton, more from a geometric than from an algebraic point of view, after treating the absolute global clock t as part of the system. The augmented and infinite augmented region automaton, $\mathfrak{R}^t(A)$ and $\mathfrak{R}_\infty^t(A)$, are presented in Section 4, and then, in Section 5, we explore the time-periodicity in them, so that $\mathfrak{R}_\infty^t(A)$ can be folded into the finite periodic augmented region automaton $\mathfrak{R}_{\text{per}}^t(A)$ (Section 6). In last section (Section 7) we construct the entire eventually periodic timestamp. We also show that the 1-bounded universality and 1-bounded language inclusion problems are decidable for the class eNTA. As for the general language inclusion problem in eNTA, the timestamp, or better - the more informative automaton $\mathfrak{R}_{\text{per}}^t$, may serve as a tool in demonstrating the non-inclusion relation between the languages of two members of eNTAs.

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 $\mathbb{R}_{\geq 0}$, the set of non-negative real numbers. It consists of a finite set of *locations* with a finite set of *transitions* between the locations, while time, measured by the clocks, is continuous.

As long as the system is in some location q , all clocks advance at the same rate. Then at some point in time, the system may make a transition τ to location q' (which may be q itself). Such a transition can occur if, first, the definition of the timed automaton contains the transition τ from q to q' . Secondly, the *transition guard*, which consists of a set of constraints in the form of strict or

weak integer upper and/or lower bounds on the values of the clocks, should be satisfied at the time of the transition. 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 can be taken at the same time, with the same action, from the same location q , but to two different locations q' and q'' . However, we would like to note that even in DTA, the class of deterministic timed automata, at each time there may exist a choice of either to stay in the current location or to take a transition, and if a transition is taken there may be more than one possible (although with different actions).

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 2.1 (eNTA). *A non-deterministic timed automaton with silent transitions* $A \in \text{eNTA}$ 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* $v(c)$ is a function $v : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$. 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 $A \in \text{eNTA}$ 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 the valuation v satisfies the guard g and $v' = v[\mathcal{C}_{rst}]$.

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

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

The run ρ of A induces the *timed trace (timed word)*

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

with $a_i \in \Sigma_\epsilon$ and $t_i = \sum_{j=1}^i d_j$. 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.

The class of *non-deterministic* timed automata, denoted NTA, is defined as in Definition 2.1, except that the transition labels Σ refer only to observable transitions. If, in addition, each timed trace of the timed automaton is induced by a unique run then the timed automaton belongs to the class of *deterministic* timed automata, denoted DTA.

Remark 2.2. *it is common to include in the definition of a timed automaton a set of constraints called location invariants, which are upper bounds on the clocks values while staying in a location. However, for the purpose of this paper these constraints are not needed and we may consider them to be more of a ‘syntactic sugar’ because the invariants of location q may be incorporated in the guards of the transitions to q (for the clocks that are not reset at the transitions) and in those emerging from q .*

Remark 2.3. *The distinction between accepting and non-accepting locations, between observable and silent transitions, and between deterministic and non-deterministic automata is irrelevant to reachability problems. The language accepted by the automaton does depend on the observability of the transitions and the acceptance of the locations, however these restrictions do not matter for the analysis and results concerning the reachability problems that are presented in the paper.*

3 The Trail and Timestamp of a Single Path

In this section we describe the trail and timestamp of a single path of a timed automaton. Given a timed automaton $A \in \text{eNTA}$ over s clocks x_1, \dots, x_s , we add to it a non-resetting global clock t that displays absolute time. All clocks, including t , start with value 0 and then progress at the same rate.

We take the geometric point of view and look at a run as inducing a trajectory of an object moving continuously (except for the projections occurring

during clocks reset) in the non-negative part of the $tx_1 \cdots x_s$ -space. Then, we shift our perspective from single runs to the set of all runs along a path of A . The resulting geometric domain, which we call a 'trail' of the path, is more 'solid' and deterministic. It is subject to a natural hyper-triangulation, where each such hyper-triangle or simplex is called 'region'. The initial region is the point of origin $\mathbf{0}$, and then each region has a unique successor in direction $(1, 1, \dots, 1)$. When some clocks are reset on a transition then the successor of a region is again a region, which is simply the source region with the reset clocks set to be the constant 0.

Definition 3.1 (Trajectory of a run). Let $\{t, x_1, \dots, x_s\}$ be an ordered set of clocks of $A \in \text{eNTA}$. Let ϱ be a run of duration T of A . The *trajectory* of ϱ is the set of points (t, x_1, \dots, x_s) in the $tx_1 \cdots x_s$ -space visited during ϱ , where $0 \leq t \leq T$.

Definition 3.2 (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 .

A finite *path* in A has the form $\gamma = q_0\tau_1q_1\tau_2 \cdots \tau_nq_n$ of alternating locations and transitions, and we always assume that q_0 is the initial location. Such a path is an abstraction of a run since the temporal part is omitted. Given a path γ in A , there may be many possible runs along γ , and we say that γ is *feasible* when there is at least one run along it.

Definition 3.3 (Trail of a path). The *trail of a path* γ , denoted by Θ_γ , is the union of the trajectories of all runs along γ .

Definition 3.4 (Timestamp of a path). The *timestamp of a path* γ of A is the union of the timestamps of all runs ϱ along γ .

We call each instance of a transition along γ an *event*. That is, a transition is a static object which joins two locations of the timed automaton, whereas an event refers to a specific occurrence of a transition within the path γ . Hence, several events along a path may refer to the same transition of the timed automaton.

Definition 3.5 (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.

Assume that A has $s \geq 1$ clocks x_1, \dots, x_s , which we call the *regular* clocks in what follows. Let γ be a feasible path in A . The trajectory of a run ϱ along γ , consisting of all points $(t = x_0, x_1, \dots, x_s)$ during ϱ , forms a (discontinuous, in general) piecewise linear curve in the $x_0 \cdots x_s$ -space.

The trail of γ , Θ_γ , is the union of the trajectories of all runs along γ . The *trail legs*, the parts of the trail between clocks reset, are in the form of *zones* [DT98], a conjunction of diagonal constraints $x_i - x_j < n_{ij}$ or $x_i - x_j \leq n_{ij}$, $n_{ij} \in \mathbb{Z}$, and transition constraints (as described in Section 2) $x_i \sim n_i$, where $\sim \in \{<, \leq, =, \geq, >\}$, $n_i \in \mathbb{N}_0$. The diagonal constraints are invariants of the trail

legs since all clocks proceed at the same rate, and they determine the ordering among the fractional parts of the clocks values, as described below. However, the diagonal constraints describe unbounded sets, and the transition constraints bound these domains. Each trail leg can be partitioned into *simplicial trails*, which are (possibly unbounded) parallelotopes consisting of a sequence of *regions* [AD94] arranged along the directional vector $\mathbf{1} = (1, 1, \dots, 1)$.

Each region $\mathbf{n} + \Delta$ is in the form of an open (unless it is a point) simplex Δ of dimension 0 (a point), 1 (a line), 2 (a triangle), 3 (a tetrahedron) or higher, up to dimension $s + 1$, that resides inside an $(s + 1)$ -dimensional unit hyper-cube with the vertices of the simplex in the lattice \mathbb{N}_0^{r+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

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

where $\preceq_i \in \{=, <\}$. The integral point $\mathbf{n} \in \mathbb{N}_0^{r+1}$ consists of the integral parts of the values of the clocks x_0, x_1, \dots, 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. When the simplicial trail S is k -dimensional then the immediate time-successor of an open k -simplex (a simplex of dimension k , $1 \leq k \leq r + 1$) is a $(k - 1)$ -simplex and vice-versa, where each $(k - 1)$ -simplex is a face of its neighboring k -simplices. A region which is in the form of a k -simplex refers to the case where the fractional parts of the clocks are all non-zero, and then its immediate time-successor is a $(k - 1)$ -simplex, in which the integral part of the clock with maximal fractional part is increased by 1 while its fractional part is set to zero. The order between the other clocks remains as before. The switch from a $k - 1$ -simplex into a k -simplex occurs when a clock of fractional part 0 turns into a positive fractional part and the order of the fractional parts of the clocks as well as their integral values remains as before. Thus, at each switch there is a cyclic shift in the fractional parts of the clocks, which results in a periodic sequence of simplices along a simplicial trail.

Let $d_i \geq 0$ be the feasible *duration* of the i -th event along a path γ . That is, $d_i = M_i - m_i$, where M_i is the supremum, over the runs along γ , of the time at which the i -th event of the run occurs, and m_i is defined as the infimum of the same set. In case of an automaton with a single clock x , if x resets on this transition then the size of the temporal part of the timestamp of the i -th event increases by d_i , resulting in an increase in the width of the parallelogram that represents the trail of γ after the i -th event, and possibly increasing the dimension of the trail from 1 to 2. Otherwise, the width remains as before. In case of multiple clocks, the dimension of the trail can increase, decrease or stay the same after an event with reset of clocks: clocks with the same fractional part can be separated, resulting in an increase of the dimension, while clocks whose fractional parts become identical (namely, 0) contribute to a decrease of the dimension.

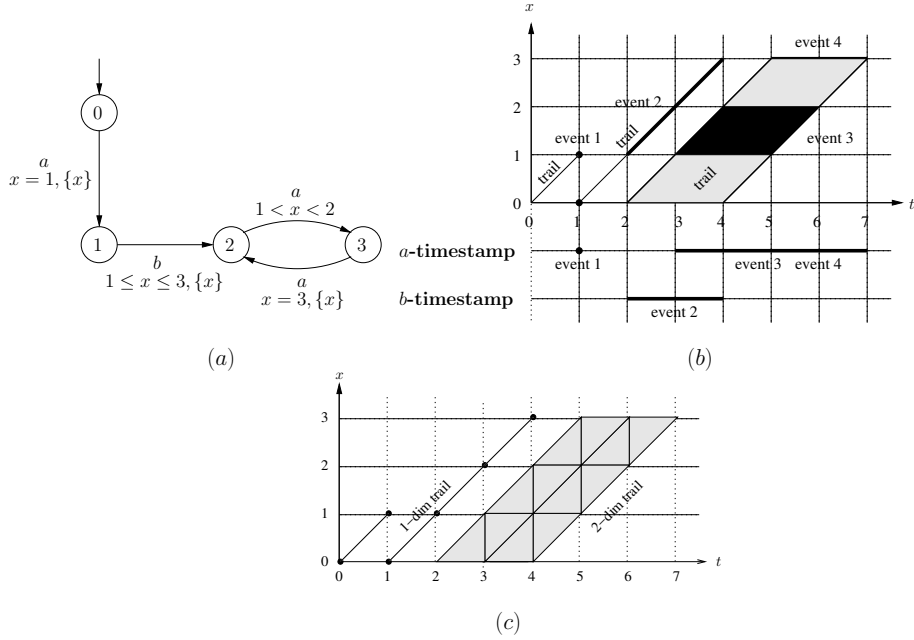


Figure 1: Trail, timestamp and regions of a path (single clock)

Let us look at a simple example of the trail and timestamp of a path in an automaton with a single clock.

Example 3.6. In Fig. 1(a) a timed automaton is drawn, and in Fig. 1(b) we see the trail and timestamp of the finite path $\gamma : (0) \xrightarrow{a} (1) \xrightarrow{b} (2) \xrightarrow{a} (3) \xrightarrow{a} (2)$, where 'a-timestamp' refers to the projection on the t -axis of the elements (t, a) of the timestamp, and similarly for 'b-timestamp'. The first event occurs when $x = 1$ and the timestamp is $\{1\} \times \{a\}$. Then x resets and the trail (a straight line of slope 1) continues from the t -axis. Event 2 occurs when $1 \leq x \leq 3$ with timestamp $[2, 4] \times \{b\}$ and a reset of x . After that event the trail is two-dimensional (a parallelogram). Event 3 occurs when $1 < x < 2$ without clock reset, and the orthogonal projection to the t -axis gives the timestamp $(3, 6) \times \{a\}$ (here $(3, 6)$ is the open interval $3 < t < 6$). The fourth event happens when $x = 3$ and its timestamp is $[5, 7] \times \{a\}$. The timestamp of γ is the union of the above sets, that is, $S_1 \times \{a\} \cup S_2 \times \{b\}$, with $S_1 = \{1\} \cup (3, 7]$ and $S_2 = [2, 4]$.

Proposition 3.7. 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. The trail of each path is composed of simplices as in (1) residing on the integral grid. The intersection of such a simplex Δ with a domain satisfying a transition constraint of the form $x_i \sim n_i$, where $\sim \in \{<, \leq, =, \geq, >\}$, $n_i \in \mathbb{N}_0$

is either the whole of Δ or the empty set. A possible reset of clocks x_i during an event results in mapping Δ to another simplex Δ' , which may be of smaller dimension. Thus, it suffices to show that the timestamp of a single simplex Δ is of the required form. But the temporal part of the timestamp of Δ is the set $n + S$, where $n \in \mathbb{N}_0$ and S is the set of values of the clock $t = x_0$ in Δ . Since S is either $\{0\}$ or the open interval $(0, 1)$, we get that the timestamp of Δ is either an action-labeled integral point $\{n\}$ or an action-labeled open unit interval $(n, n + 1)$. \square

4 Augmented and Infinite Augmented Region Automaton

4.1 Infinite Augmented Region Automaton

Given a (finite) timed automaton A , the region automaton $\mathfrak{R}(A)$ [AD94] is a finite *discretized* version of A , such that time is abstracted and both automata define the same untimed language. Instead of looking at the clocks-space as a continuous space it is partitioned into regions. 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 Section 3 or an unbounded region, in which the value of at least one of the clocks passed the maximal integer value M that appears in the guards of the automaton (for convenience, we apply for all the regular clocks x_1, \dots, x_s the same value $M = \max_i(M_i)$, where M_i is the maximal integer value that appears in a constraint regarding clock x_i).

The region automaton is constructed by first partitioning the space of clock valuations into equivalent classes, where two valuation belong to the same equivalent class if and only if the integral parts of the clocks valuations are the same, as well as the order among the fractional parts, unless a clock passes the bound on the maximal integer appearing in transition guards, in which case the actual integral and fractional values of that clock are irrelevant. 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 [AD94]), the region automaton can be effectively constructed. As shown in [AD94], one can determine through it the 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 .

Suppose now that we add to A a clock t which measures the absolute time, that is, t progresses like the regular clocks, starting with value 0 at the initial state, but does not appear in any transition guard of A and is never reset to 0 on a transition. Clearly, the set of runs and the set of timed traces of A are not affected by the addition of t .

Next, we construct the region automaton that corresponds to A extended with the global clock t , but the partitioning of the clocks-space into regions is according to all integral values of t and not only the values $0, 1, \dots, M$ as is the case with the regular clocks. That is, the time-regions associated with the

continuous time t are the alternating point and open unit interval: $\{0\}$, $(0, 1)$, $\{1\}$, $(1, 2)$ and so on. As for the regular clocks, when the value of a clock x_i passes M then we represent its value by the symbol \top . We call this model infinite augmented region automaton and denote it by $\mathfrak{R}_\infty^t(A)$.

Definition 4.1 (Infinite augmented region automaton). Given a non-deterministic timed automaton with silent transitions $A \in \text{eNTA}$ extended with the clock t that measures absolute time, a corresponding *infinite augmented region automaton* $\mathfrak{R}_\infty^t(A)$ is a tuple $(V, v_0, E, \Sigma_\epsilon)$, where:

1. V is an infinite (in general) set of vertices. Each vertex is a triple

$$(q, \mathbf{n}, \Delta), \quad (2)$$

where q is a location of A and the pair (\mathbf{n}, Δ) is a region, with

$$\mathbf{n} = (n_0, n_1, \dots, n_s) \in \mathbb{N}_0 \times \{0, 1, \dots, M, \top\}^s \quad (3)$$

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

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

$$(q, r) \xrightarrow{a} (q', r') \quad (4)$$

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

$$(q, v) \xrightarrow{d} (q, v + d) \xrightarrow{a} (q', v'), \quad (5)$$

such that 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 that are the labels of the edges.

We note that there may be infinitely-many edges going-out of the same region in $\mathfrak{R}_\infty^t(A)$ (see Fig. 2(b)).

Proposition 4.2. *For each positive integer n , one can effectively construct the part of $\mathfrak{R}_\infty^t(A)$ which contains all regions with $t \leq n$ and all in-coming edges of these regions.*

Proof. There are finitely-many regions obeying the constraint $t \leq n$. These regions and their in-coming edges can be constructed the same way as a standard region automaton is constructed, starting with the initial location and proceeding step by step according to the immediate time-successor regions (which

include the clock t) and according to the transitions of A . Indeed, the additional clock t is only responsible for a finer partition of regions, but its introduction does not affect the transition guards of A . Note also that since the clock t never resets, there are no edges connecting regions with $t > n$ to regions with $t \leq n$. Hence, the number of edges of $\mathfrak{R}_\infty^t(A)$ restricted to $t \leq n$ is finite. \square

The benefit of introducing the clock t into the region automaton is that we can know approximately at what absolute time an action occurs. For example, suppose that A has a single clock x and that x is reset on a transition from location q to location q' . Then, in the corresponding region automaton the information about the time spent at location q before moving to q' is lost. In $\mathfrak{R}_\infty^t(A)$, however, if we take the absolute time at which an action occurs to be $n + 0.5$ when entering a region whose time-region (the value of t) is the open interval $(n, n + 1)$, and the absolute time n when entering a region whose time-region is exactly $t = n$, then it is possible to construct from it an (infinite) *approximate timed automaton* with a single clock and which differs from A by at most 0.5 time units at each action.

The timestamp of the timed automaton A , denoted $\mathfrak{T}\mathfrak{S}(A)$, 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 . Let us also define the timestamp of $\mathfrak{R}_\infty^t(A)$. For each region r of $\mathfrak{R}_\infty^t(A)$, the part that refers to clock t , i.e. the set of values of t in this region, is either an integral point $\{n\}$ or an open unit interval $(n, n + 1)$.

Definition 4.3 (Timestamp of $\mathfrak{R}_\infty^t(A)$). The timestamp of $\mathfrak{R}_\infty^t(A)$, $\mathfrak{T}\mathfrak{S}(\mathfrak{R}_\infty^t(A))$, is the union of sets $s \times a$, where s is a time-region of t that is part of a region of a vertex of $\mathfrak{R}_\infty^t(A)$ and $a \in \Sigma$ is a label of an edge of $\mathfrak{R}_\infty^t(A)$ that is directed towards this vertex.

Proposition 4.4. $\mathfrak{T}\mathfrak{S}(A) = \mathfrak{T}\mathfrak{S}(\mathfrak{R}_\infty^t(A))$.

Proof. By definition of the infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$, its regions are exactly the clock-regions which are visited by runs of the timed automaton A extended with the clock t . In particular, the time-regions of $\mathfrak{R}_\infty^t(A)$ are the time-regions that are visited by the runs on the extended timed automaton. Thus, $\mathfrak{T}\mathfrak{S}(A) \subseteq \mathfrak{T}\mathfrak{S}(\mathfrak{R}_\infty^t(A))$. By Proposition 3.7 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 . \square

4.2 Augmented Region Automaton

A second construction is a finite automaton called augmented region automaton, denoted $\mathfrak{R}^t(A)$. Here we consider only the fractional part of t and ignore its integral part, that is, we take the quotient of the continuous time t modulo the integers, resulting in the circular unit interval $[0, 1]$, where 1 is identified with 0 (thus, isomorphic to the half-open interval $[0, 1)$). $\mathfrak{R}^t(A)$ is obtained by

folding $\mathfrak{R}_\infty^t(A)$, that is, $\mathfrak{R}_\infty^t(A)$ is mapped onto $\mathfrak{R}^t(A)$ by identify vertices that contain the same data, except for the integral part of t , as well as identifying the corresponding edges. This kind of folding with respect to the equivalence relation between any two vertices which are identical except for the integral value of t is justified since the clock t does not appear in any transition guard of $\mathfrak{R}_\infty^t(A)$. Thus, if we ignore t , these equivalent vertices would have all been represented by the same vertex in the standard region automaton.

Alternatively, $\mathfrak{R}^t(A)$ may be constructed from A similarly to the construction of $\mathfrak{R}_\infty^t(A)$ by first adding to A the clock t and then forming the region automaton with respect to the regular clocks and the new clock t while ignoring the integral part of t . Note that since there are only 2 different time-regions of t , namely $\{0\}$ and $(0, 1)$, the number of regions of $\mathfrak{R}^t(A)$ is finite. This construction is, however, less informative than $\mathfrak{R}_\infty^t(A)$. In order to make the two region automata equally informative we assign each edge of $\mathfrak{R}^t(A)$ a non-negative integral *weight* m , which may also be marked as m^* as explained below.

Definition 4.5 (Augmented region automaton). Given a non-deterministic timed automaton with silent transitions $A \in \text{eNTA}$, extended with the clock t that measures the passing time, a corresponding (finite) *augmented region automaton* $\mathfrak{R}^t(A)$ is a tuple $(V, v_0, E, \Sigma_\epsilon, [0..M], *)$, where:

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

$$\mathbf{n} = (n_1, \dots, n_s) \in \{0, 1, \dots, M, \top\}^s \quad (6)$$

containing the integral parts of the clocks x_1, \dots, x_s , and Δ is the simplex defined by the fractional parts of the clocks t, x_1, \dots, 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' . The edge also contains a weight $m \in [0..M]$, which equals the difference $\lfloor t_1 \rfloor - \lfloor t_0 \rfloor$ between the integral part of the value of t in the target location and that in the source location in the corresponding run of A . Thus, there may be more than one edge between two vertices of $\mathfrak{R}^t(A)$, each one with a distinguished weight. When the weight of an edge is marked as m^* then it represents infinitely-many consecutive values $m, m + 1, m + 2, \dots$ as weights between the two vertices, with m being the minimal value of such a sequence, (corresponding to a transition to or from a region r in which all regular clocks have passed the maximal integer M that appears in a constraint).
4. $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ is the finite set of actions.

5. $[0..M]$ is the set of weights on the edges.
6. $'*'$ is the symbol representing a marked weight.

Proposition 4.6. $\mathfrak{R}^t(A)$ and $\mathfrak{R}_\infty^t(A)$ are equally informative.

Proof. Clearly, since $\mathfrak{R}^t(A)$ may be obtained from $\mathfrak{R}_\infty^t(A)$ then it cannot be more informative.

In the other direction, it can be shown that for each positive integer n , one can effectively construct $\mathfrak{R}_\infty^t(A)$ up to time $t = n$, as in Proposition 4.2. Indeed, since the transitions in A do not rely on t , by taking the quotient of $\mathfrak{R}_\infty^t(A)$ 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 appears as a weight on the corresponding edge of $\mathfrak{R}^t(A)$. It is also clear that whenever the time difference is greater than M then since M is the maximal integer that appears as a constraint in A then all values that are greater than M are also possible time differences between the same (when ignoring t) regions, and in this case the edge weight m is marked with $*$.

It follows that one can construct $\mathfrak{R}_\infty^t(A)$ up to level $t = n$ by unfolding $\mathfrak{R}^t(A)$ and recovering the absolute time t by summing up the weights of the edges along the taken paths. \square

As with $\mathfrak{R}_\infty^t(A)$, we can construct from $\mathfrak{R}^t(A)$ an approximate automaton, this time a finite and deterministic one, which approximates A with a maximal error of $1/2$ time units at each observed transition. This automaton has only one clock and this clock resets at every transition. The maximal error $\epsilon = 1/2$ could be further reduced to $1/n$ by allowing transitions in the approximate automaton to occur at times p/n , $p \in \mathbb{N}_0$ and only on such times.

Example 4.7. In Fig. 2(a) we see a timed automaton A containing a transition to an unbounded region. The corresponding infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$ is shown in Fig. 2(b). Each vertex of $\mathfrak{R}_\infty^t(A)$ is represented by a rounded rectangle containing the original location of A (circled, on the left), the integral values of t and of x (in the top of the rectangle) and the simplex (in the bottom). Notice that when the value of x is greater than $M = 0$ it is marked by \top and its fractional part is ignored. To the left of $\mathfrak{R}_\infty^t(A)$ we see the discretization of time t into time-regions, and each vertex of $\mathfrak{R}_\infty^t(A)$ is drawn in the level of its time-region. In Fig. 2(c) the augmented region automaton $\mathfrak{R}^t(A)$ is shown. Here the integral part of the value of t is ignored. The edge labeled by 0^* represents the infinitely-many differences in the integral parts of the values of t : $0, 1, 2, \dots$. Similarly, the edge labeled with 1^* refers to the differences $1, 2, 3, \dots$.

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.

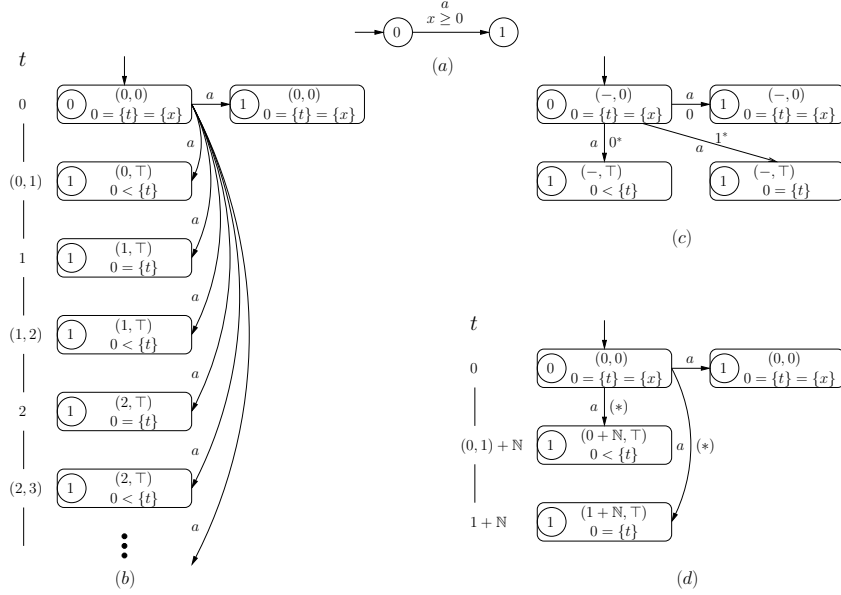


Figure 2: (a) $A \in \text{TA}$; (b) The infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$; (c) The augmented region automaton $\mathfrak{R}^t(A)$; (d) A periodic augmented region automaton $\mathfrak{R}_{\text{per}}^t(A)$

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 transition in A that can be taken. 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 5.1 (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 5.2 ((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 5.3. *There exists a minimal positive integer $t_{\text{nz}} \leq D + 1$, 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.*

Proof. Indeed, if $\mathfrak{R}^t(A)$ does not contain non-Zeno cycles then $t_{\text{nz}} = D + 1$ and the claim holds vacuously. So, suppose that $\mathfrak{R}^t(A)$ contains non-Zeno Cycles.

Then each path of duration $D + 1$ must contain non-Zeno cycles because otherwise the Zeno cycles could have been removed, without changing the duration of the path, resulting in a simple path of duration $D + 1$ - a contradiction. \square

In case $\mathfrak{R}^t(A)$ contains non-Zeno cycles the actual value of t_{nz} can be less than $D + 1$ since it is not required that the path contains a non-Zeno cycle but 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 $t_{\text{nz}} = t_0 + 1$.

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 5.4 (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 5.5 (Period of $\mathfrak{R}^t(A)$). A time *period*, or just period, of the augmented region automaton $\mathfrak{R}^t(A)$ is an integer L which is a common multiple of the members of the set $\{d(\pi) \mid \pi \in C\}$, for some (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.

We remark that if we want to compute a minimal period $L > M$ we need to conduct a thorough exploration of the duration of cycles in $\mathfrak{R}^t(A)$, taking into account their common factors, but this computation is not needed for the results presented here.

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

As before, t_{nz} denotes the non-Zeno threshold time, C denotes a fixed minimal covering set of non-Zeno cycles and L is a period of $\mathfrak{R}^t(A)$ with respect to the set C .

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

Definition 5.6 (L -shift in time). Given a subgraph G of $\mathfrak{R}_\infty^t(A)$, 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 , and by $v + L$ when $V = \{v\}$.

Lemma 5.7. *If $\mathfrak{R}_\infty^t(A)$ is not bounded in time then*

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

Proof. First we show that the inclusion holds for the set of vertices of the above subgraphs. Let γ be a path of $\mathfrak{R}_\infty^t(A)$ which terminates in a vertex $v_1 \in \mathfrak{R}_\infty^t(A)|_{t \geq t_{\text{nz}}}$. 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 $d(\gamma) \geq t_{\text{nz}}$ then by Lemma 5.3 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 $\mathfrak{R}_\infty^t(A)$ of this extended path.

As for the out-going edges, if $v_1 \xrightarrow{a} w_1$ is an edge of $\mathfrak{R}_\infty^t(A)|_{t \geq t_{\text{nz}}}$ then there exists an edge $v_2 \xrightarrow{a} w_2$ of $\mathfrak{R}_\infty^t(A)|_{t \geq t_{\text{nz}} + L}$, with $v_2 = v_1 + L$ and $w_2 = w_1 + L$, since the set of out-going edges from each vertex does not rely on the value of t but on the values of the other clocks. \square

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

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

Theorem 5.8. *If the infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$ is not bounded in time then it is eventually periodic: there exists an integral time $t_{\text{per}} > 0$ such that*

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

Proof. By Lemma 5.7,

$$V_k + L \subseteq V_{k+1}, \quad \text{for } k \geq 0.$$

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 5.7 holds also here. \square

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

Proposition 5.9. *if $|V_k| = |V_{k+1}| = |V_{k+2}|$ for some k then we can set $t_{\text{per}} = t_{\text{nz}} + kL$.*

Proof. The equalities $|V_k| = |V_{k+1}| = |V_{k+2}|$ are equivalent to $V_{k+1} = V_k + L$ and $V_{k+2} = V_{k+1} + L$. By induction, it suffices to show that these equalities imply that $|V_{k+2}| = |V_{k+3}|$, that is, $V_{k+3} = V_{k+2} + L$. Let $v \in V_{k+3}$. We need to show that there exists $v' \in V_{k+2}$ such that $v = v' + L$.

Suppose that v is reached by an edge from a vertex $u \in V_{k+1} \cup V_{k+2}$. Since $V_{k+2} = V_{k+1} + L = V_k + 2L$, there exists a vertex $u' = u - L \in V_k \cup V_{k+1}$ and this vertex is connected to a vertex $v' = v - L \in V_{k+2}$.

Otherwise, v is reached by an edge e_1 from a vertex u in V_k or earlier, and the time difference d between u and v is greater than $2L$. This implies that the projection $p(e_1) \in \mathfrak{R}^t(A)$ is of unbounded time delay m^* . Since $L > M$ then $d - L > M$. Hence, there is another edge e_2 in $\mathfrak{R}_\infty^t(A)$, that is also a pre-image of $p(e_1)$, and which joins u to a vertex $v' \in V_{k+2}$, where $v = v' + L$. \square

Example 5.10. *This example refers to the timed automaton of Fig. 4 (a). In order to make the analysis of its time-periodic structure simpler, we changed the guard on the transition from location 1 to location 2 to be simpler (Fig. 3 (a)), so that in the resulting infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$ (Fig. 3 (b)) we can clearly see two different cycles of period 6 (circled in dotted lines) (the edges with label c are only partly shown). We then added the original guard between locations 1 and 2 (Fig. 4 (a)). In the additional part in $\mathfrak{R}_\infty^t(A)$ (Fig. 4 (b)) we see two more cycles, one of period 11 and one of period 5. We can still use a period of length 6 for this more complex automaton, but the existence of cycles of other lengths results in a longer time until reaching the repeated periodic part of the entire automaton.*

As is known, a timed automaton may be totally non-periodic in the sense that no single timed trace of it is eventually periodic (see Example 6.3). However, a special kind of periodicity, which we call *suffix-periodicity*, holds between different timed traces, as shown in the following theorem.

Theorem 5.11. *If $A \in \text{eNTA}$ is not bounded in time then its language $\mathfrak{L}(A)$ is suffix-periodic: if $t_r > t_{\text{per}}$ and*

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

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

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

Proof. Suppose that λ is the observable timed trace of some run ρ of A . This run corresponds to a path in $\mathfrak{R}_\infty^t(A)$ whose r -th transition reaches a vertex v with some time-region α with $t_r \in \alpha$. By Theorem 5.8 there exists a path γ in $\mathfrak{R}_\infty^t(A)$ which reaches a vertex $u = v + k$. That is, if $v = (q, (n_0, n_1, \dots, n_s), \Delta)$ then u is identical to v except for the integral part of t , which is increased by k : $u = (q, (n_0 + k, n_1, \dots, n_s), \Delta)$, or, in other words, the time-region α' of u is a translate by k of the time-region α of v . Hence, since $t_r \in \alpha$ then $t_r + k \in \alpha'$. As we saw in Section 3, the trail of the path γ (the union of the trajectories along

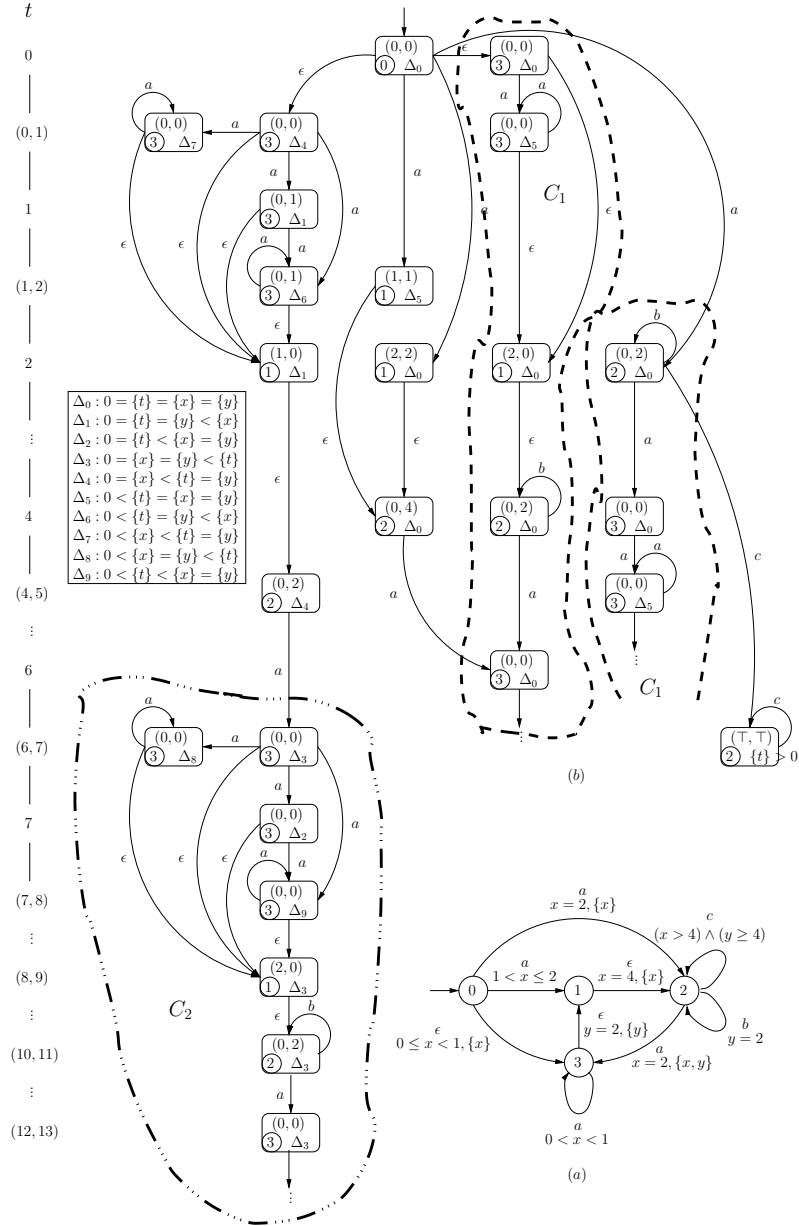


Figure 3: (a) The simplified $A \in \text{eNTA}$; (b) $\mathfrak{R}_\infty^t(A)$ with period 6

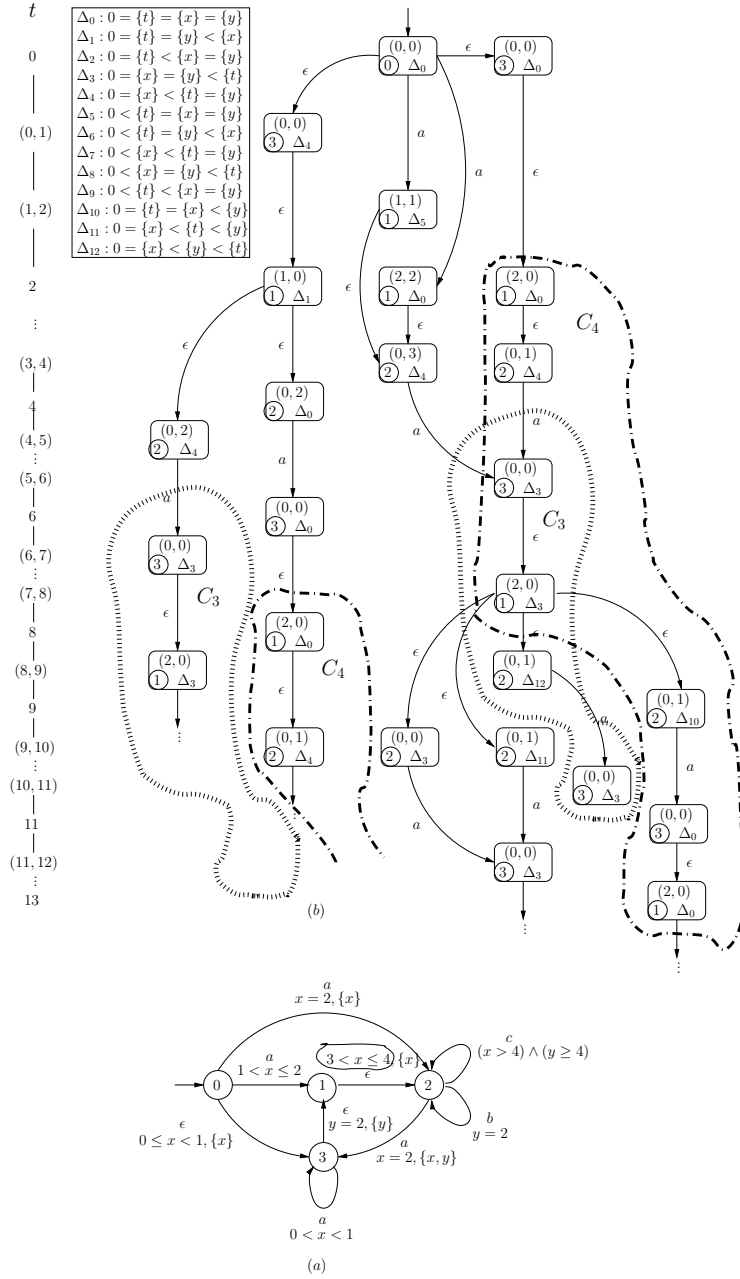


Figure 4: (a) The original $A \in \text{eNTA}$; (b) The additional part of $\mathfrak{R}_\infty^t(A)$ with cycles of lengths 11 and 5

γ) is composed of regions in the form of simplices. Thus, for every value of the time-region α' , in particular for $t_r + k$, there exists a run ϱ' of A which reaches location q at the exact time $t_r + k \in \alpha'$ on an observable action a_r . From that time on, the run ϱ' can imitate the behavior of ϱ by keeping a time difference k in the taken transitions. The result then follows. \square

6 Periodic Augmented Region Automaton

After revealing the periodic structure of $\mathfrak{R}_\infty^t(A)$, it is natural to fold it into a finite graph, which we call periodic augmented region automaton and denote by $\mathfrak{R}_{\text{per}}^t(A)$. The construction of $\mathfrak{R}_{\text{per}}^t(A)$ is done by first taking the subgraph of $\mathfrak{R}_\infty^t(A)$ of time $t < t_{\text{per}} + L$ and then folding the infinite subgraph of $\mathfrak{R}_\infty^t(A)$ of time $t \geq t_{\text{per}} + L$ onto the subgraph of time $t_{\text{per}} \leq t < t_{\text{per}} + L$, which becomes the periodic subgraph. Thus, each vertex of the periodic subgraph represents infinitely-many vertices of $\mathfrak{R}_\infty^t(A)$. Similarly, the out-going edges of the periodic subgraph are periodic edges. In addition, some of the edges of $\mathfrak{R}_{\text{per}}^t(A)$ are marked with $(*)$ or $(*+)$, as explained below.

Definition 6.1 (Periodic augmented region automaton). Given an infinite augmented region automaton $\mathfrak{R}_\infty^t(A)$ with period L and periodicity starting time t_{per} , a (finite) fold of it called *periodic augmented region automaton* $\mathfrak{R}_{\text{per}}^t(A)$ 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. There are two types of vertices: regular and periodic. The regular vertices are those vertices of $\mathfrak{R}_\infty^t(A)$ of time $t < t_{\text{per}}$. The *periodic* vertices of $\mathfrak{R}_{\text{per}}^t(A)$ are the vertices of $\mathfrak{R}_\infty^t(A)$ of time $t_{\text{per}} \leq t < t_{\text{per}} + L$ turned into periodic vertices by setting for each $t_{\text{per}} \leq n < t_{\text{per}} + L$ the integral value $\lfloor t \rfloor = n$ of t , to be $\lfloor t \rfloor = n + L\mathbb{N}_0$. Each periodic vertex of $\mathfrak{R}_{\text{per}}^t(A)$ represents the infinitely-many vertices of $\mathfrak{R}_\infty^t(A)$ which have different integral time $\lfloor t \rfloor \in \{n + kL \mid k = 0, 1, 2, \dots\}$ and otherwise are identical.
2. E is the set of edges, which are the projected edges of $\mathfrak{R}_\infty^t(A)$: Each edge joining two vertices of $\mathfrak{R}_\infty^t(A)$ is mapped to an edge with the same action label that joins the projected vertices. Some of the edges are marked with $(*)$ or $(*+)$ and some are not. An unmarked edge of $\mathfrak{R}_{\text{per}}^t(A)$ that starts in a regular vertex of time t satisfying $\lfloor t \rfloor = n_1$ and terminates in a regular vertex with $\lfloor t \rfloor = n_2$ or in a periodic vertex of time $\lfloor t \rfloor = n_2 + L\mathbb{N}_0$ lifts to an edge of $\mathfrak{R}_\infty^t(A)$ from a vertex of time t , where $\lfloor t \rfloor = n_1$, to a vertex of time t with $\lfloor t \rfloor = n_2$. If the edge e of $\mathfrak{R}_{\text{per}}^t(A)$ joins a periodic vertex of time t , $\lfloor t \rfloor = n_1 + L\mathbb{N}_0$, to a periodic vertex satisfying $\lfloor t \rfloor = n_2 + L\mathbb{N}_0$, $t_{\text{per}} \leq n_1, n_2 < t_{\text{per}} + L$, then the following holds. If $n_1 \leq n_2$ then e represents the infinitely-many edges of $\mathfrak{R}_\infty^t(A)$ connecting the corresponding vertices of the following integral time $\lfloor t \rfloor$: $n_1 \rightarrow n_2$, $n_1 + L \rightarrow n_2 + L$, $n_1 + 2L \rightarrow n_2 + 2L$ and so on. If $n_1 > n_2$ then the corresponding edges of $\mathfrak{R}_\infty^t(A)$ are of integral time $\lfloor t \rfloor$: $n_1 \rightarrow n_2 + L$, $n_1 + L \rightarrow n_2 + 2L$, $n_1 + 2L \rightarrow n_2 + 3L$ and so on.

When an edge from a regular vertex v is marked with $(*)$ then it means that this edge represents infinitely-many edges stemming from the same vertex of $\mathfrak{R}_\infty^t(A)$ and terminating in infinitely many vertices of $\mathfrak{R}_\infty^t(A)$, all of the same time modulo L : if the edge terminates in a vertex w of time n_2 or $n_2 + L\mathbb{N}_0$ then it represents the edges of $\mathfrak{R}_\infty^t(A)$, all starting in the pre-image (under the projection to $\mathfrak{R}_{\text{per}}^t(A)$) of v and ending in the vertices which are pre-images of w and of integral time $n_2, n_2 + L, n_2 + 2L$, and so on. If the marked edge of $\mathfrak{R}_{\text{per}}^t(A)$ starts in a periodic vertex v of integral time $n_1 + L\mathbb{N}_0$ then it represents infinitely-many edges that start in the pre-image of v in $\mathfrak{R}_\infty^t(A)$ of integral time n_1 , with the same rules as above for a marked edge from a regular vertex, and also the infinitely-many edges from the pre-image of v of integral time $n_1 + L$, and the infinitely-many edges of $\mathfrak{R}_\infty^t(A)$ that start in the pre-image of v of integral time $n_1 + 2L$ and so on.

An edge can also be marked with $(*+)$. The same rules that apply to an edge marked with $(*)$ apply here, except that the target vertices of an edge marked with $(*)$ 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.
4. $B = \{(*), (*+)\}$ is the set of symbols marking the edges.

Proposition 6.2. $\mathfrak{R}_{\text{per}}^t(A)$ is well-defined and as informative as $\mathfrak{R}_\infty^t(A)$.

Proof. Clearly, since $\mathfrak{R}_{\text{per}}^t(A)$ may be obtained from $\mathfrak{R}_\infty^t(A)$ then it cannot be more informative. It suffices then to show that for each positive integer n , $\mathfrak{R}_\infty^t(A)$ can be effectively constructed from $\mathfrak{R}_{\text{per}}^t(A)$ up to time $t = n$. Well, for $t < t_{\text{per}}$, $\mathfrak{R}_{\text{per}}^t(A)$ is identical to $\mathfrak{R}_\infty^t(A)$. Then, by Theorem 5.8, the graph of $\mathfrak{R}_\infty^t(A)$ becomes periodic in the sense that the subgraph of time $t_{\text{per}} \leq t < t_{\text{per}} + L$ repeats itself, except for the integral part of t , which progresses indefinitely in $\mathfrak{R}_\infty^t(A)$ but can be expressed modulo the period L , as is done in $\mathfrak{R}_{\text{per}}^t(A)$. Indeed, since the transitions in A do not rely on t , by taking the quotient of $\mathfrak{R}_\infty^t(A)$ modulo L from time $t \geq t_{\text{per}}$, the only loss of information is the exact time difference in t between the target and source regions. But due to the periodicity in $\mathfrak{R}_\infty^t(A)$, this information can be finitely presented. Hence, since the edges of $\mathfrak{R}_\infty^t(A)$ whose initial vertices are of time $t \geq t_{\text{per}} + L$ are translates of similar edges that start at time $t_{\text{per}} \leq t < t_{\text{per}} + L$, it suffices to examine the latter.

So, let e be an edge of $\mathfrak{R}_\infty^t(A)$ which joins a vertex u of integral time $[t] = n_1$, $t_{\text{per}} \leq n_1 < t_{\text{per}} + L$, with a vertex v of integral time $[t] = n_2$, and suppose that $n_1 \leq n_2 \pmod L$. Suppose also that u is not joined to a vertex $v' = v - L$. Then, since $L > M$ there are only two cases: either $n_2 - n_1 < L$ or $L \leq n_2 - n_1 < 2L$. In order to distinguish between these cases, the latter case is marked by a plus sign that is added to the corresponding edge of $\mathfrak{R}_{\text{per}}^t(A)$ from a vertex of integral time $[t] = n_1 + L\mathbb{N}_0$ to a vertex of integral time $[t] = t_{\text{per}} + ((n_2 - t_{\text{per}}) \pmod L) + L\mathbb{N}_0$. When u is also connected to a vertex $v' = v - L$ then we

let v'' be of minimal integral time modulo L to which u is connected, that is $v'' = v - iL$, for some $i > 0$, and there is no edge from u to $v'' - L$ (here v, v' and v'' are identical except for the integral time of t). If v'' is of integral time n then necessarily u is connected to infinitely-many vertices of integral time $n + kL$, $k \geq 0$, and all these edges are captured in $\mathfrak{R}_{\text{per}}^t(A)$ by marking with $(*)$ the edge from the corresponding vertex of integral time $\lfloor t \rfloor = n_1 + LN_0$ to a vertex of integral time $\lfloor t \rfloor = t_{\text{per}} + ((n - t_{\text{per}}) \bmod L) + LN_0$.

The case where $n_1 > n_2 \bmod L$ is handled similarly. It is now clear that in order to construct $\mathfrak{R}_{\infty}^t(A)$ up to time $t = n$ we only need to unfold $\mathfrak{R}_{\text{per}}^t(A)$ up to this time by obeying the above rules. \square

We remark that the periodic subgraph of $\mathfrak{R}_{\text{per}}^t(A)$ refers to the time $t_{\text{per}} \leq t < t_{\text{per}} + L$, a time interval of the form $[a, b)$. For our convenience, we may also define the periodic time interval to be of the form $(a, b]$. In Fig. 2 we see the four different representations of a very simple automaton A , with Fig. 2(d) being the periodic augmented region automaton with the periodic part referring to the time interval $(0, 1]$.

Example 6.3. *The timed automaton shown in Fig. 5(a) is taken from [AD94], where it demonstrates non-periodicity: the time difference between an a -transition and the following b -transition is strictly decreasing along a run. However, the periodicity among the collection of timed traces is seen in the periodic augmented region automaton, where the period here is of size 1, and the vertices in times $(2, 3) + \mathbb{N}_0$ and $3 + \mathbb{N}_0$ are periodic. Notice also that there are edges marked with $(*)$ which represent infinitely-many edges with the same source.*

6.1 Complexity

Let $N = N(\mathfrak{R}^t(A))$ denote the number of vertices in the augmented region automaton $\mathfrak{R}^t(A)$. If κ denotes the number of clocks, including the absolute clock t , λ the number of locations in A and $\mu = M + 2$, where M is the maximal integer appearing in a guard of A , then

$$N \leq \lambda(2\mu)^\kappa \kappa!. \quad (7)$$

Indeed, the number of combinations of the integral values of the clocks is bounded by μ^κ (in fact, t is assigned a single value), there are $\kappa!$ different orderings of the fractional parts of the clocks $\{x_i\}$, and the term 2^κ refers to all possibilities of inequality or equality between each pair of adjacent $\{x_i\}, \{x_j\}$ in an ordering.

Let us look now at the number of vertices in $\mathfrak{R}_{\text{per}}^t(A)$. At each time-level the number of vertices is bounded by N . Since $t_{\text{nz}} \leq MN$ then there are at most MN^2 vertices of time $t \leq t_{\text{nz}}$. After passing t_{nz} we have the subgraphs \tilde{G}_k of time length L , where L is the period. Each such subgraph has at most NL vertices. Since the number of vertices in the subgraphs forms an almost increasing sequence (until an equality occurs two consecutive times), the number

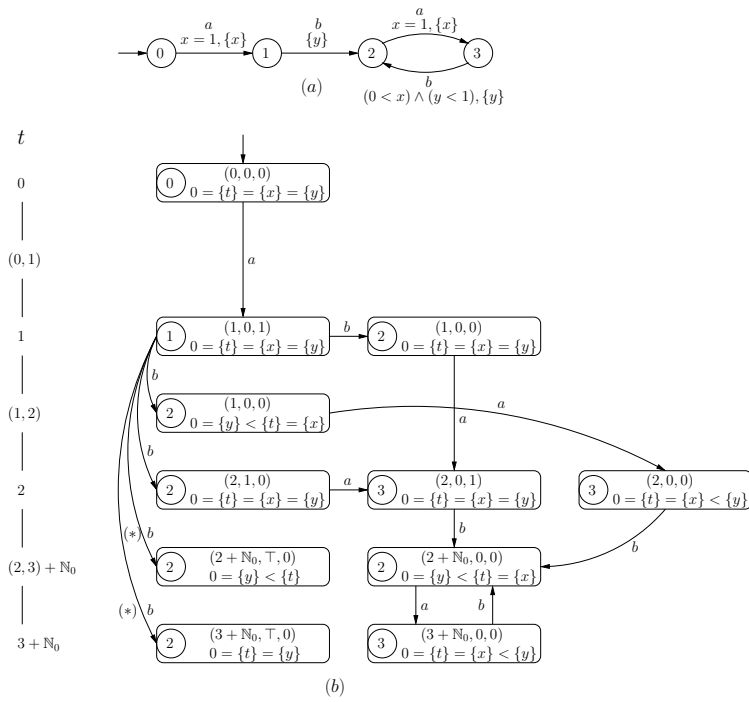


Figure 5: a) $A \in \text{TA}$; b) $\mathfrak{R}_{\text{per}}^t(A)$, a periodic augmented region automaton of A

of vertices from time t_{nz} to time t_{per} is bounded by $(NL)^2$. Thus, the number $N(\mathfrak{R}_{\text{per}}^t(A))$ of vertices in $\mathfrak{R}_{\text{per}}^t(A)$ satisfies

$$N(\mathfrak{R}_{\text{per}}^t(A)) \leq (L^2 + M)N^2(1 + o(1)) \quad (8)$$

as $N \rightarrow \infty$.

The largest factor in (8) may come from the period L , so let us compute an upper bound of L . L is the least common multiple of the durations $d(\pi)$ of cycles that form a covering set of non-Zeno cycles. For each such cycle π , $d(\pi) \leq MN$ since the length of a simple cycle is bounded by the number N of vertices in $\mathfrak{R}^t(A)$ and the time difference between two vertices along a path is at most M . Thus, a bound on L is given by the least common multiple of $1, 2, \dots, MN$, which is by the prime number theorem

$$L \leq \text{lcm}(1, 2, \dots, MN) = e^{MN(1+o(1))} \quad (9)$$

as $MN \rightarrow \infty$.

Example 6.4. *When computing the period L , in the worst case the numbers $d(\pi)$ are pairwise prime and the vertices of the cycles π form a disjoint union of sets which (almost) covers the set of vertices of $\mathfrak{R}^t(A)$. So, suppose that $\mathfrak{R}^t(A)$ is in the form of n simple cycles, where each cycle is connected to the initial vertex by an additional edge. Suppose also that the length of cycle i is p_i , the i -th prime number, $i = 1, \dots, n$. Let us assume that $M = 1$ and each edge is of weight 1. The number of vertices in $\mathfrak{R}^t(A)$ is $N = 1 + \sum_{i=1}^n p_i \sim (1/2)n^2 \log n$. Then $L = \text{lcm}\{p_1, \dots, p_n\} = \prod_{i=1}^n p_i = e^{n \log n(1+o(1))}$, the primorial $p_n\#$. This upper bound is closer to $e^{M\sqrt{N}}$ than to the bound e^{MN} of (9).*

7 The Timestamp

Recall that the timestamp $\mathfrak{T}\mathfrak{S}(A)$ 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 .

Theorem 7.1. *The timestamp of a timed automaton 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 5.11, 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 $t_{\text{per}} + [t]$, where $0 \leq [t] < L$ is the representative of t in the quotient group $\mathbb{R}/L\mathbb{Z}$.

By Proposition 3.7, 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 4.2 it is effectively computable. \square

The timestamp of a timed automaton is an abstraction of its language. It does not preserve the timestamps of single timed traces. However, the timestamp is eventually periodic and computable, hence the timestamp inclusion problem is decidable. Thus, due to the general undecidability of the language inclusion problem in non-deterministic timed automata, one may use the timestamp for refutation purpose.

Corollary 7.2. *Given two timed automata $A, B \in \text{eNTA}$ 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: $\mathfrak{L}(A) \not\subseteq \mathfrak{L}(B)$.*

The timestamp is easily extracted from $\mathfrak{R}_{\text{per}}^t$ (in fact, it is enough to take the subgraph of \mathfrak{R}_{∞}^t 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 7.3. For each $a \in \Sigma^A$, let A_a be the restriction of A to a -actions. A_a is constructed by turning each transition in A , except for the transitions labeled by a , into a 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 letters (pairs) (b, t) , $b \neq a$.

Example 7.4. *The timestamp of the a -transitions of the automaton of Fig. 5 is $\mathfrak{TS}(A_a) = \mathbb{N}$, and that of the b -transitions is $\mathfrak{TS}(A_b) = [1, \infty)$.*

7.1 Timestamp Automaton

Given a timed automaton A , we will show how to construct a *timestamp automaton* \tilde{A} associated with it, which shares with A the same timestamp. Such an automaton is decomposable into the timestamp automata of the automata A_a .

Definition 7.5 (Timestamp automaton). Given a timed automaton $A \in \text{eNTA}$, a timestamp automaton \tilde{A} is a deterministic timed automaton with a single clock and with timestamp identical to that of A . It consists of the timestamp automata \tilde{A}_a , $a \in \Sigma$, having a joint initial vertex. Each timestamp automaton \tilde{A}_a has the form of a single path $\tilde{\gamma}_a$ of positive length, which may end in a loop $\tilde{\pi}_a$, thus giving \tilde{A} the form of a bouquet.

Theorem 7.6. *Given a timed automaton $A \in \text{eNTA}$, one can effectively construct a timestamp automaton \tilde{A} .*

Proof. We construct \tilde{A}_a by following the ordered connected components (intervals) of the timestamp $\mathfrak{TS}(A_a)$ (here 'interval' includes also singletons $\{n\}$). To each such time interval corresponds the next transition guard in $\tilde{\gamma}_a$, where the lower and upper constraint on the clock x in the transition guard are exactly the left and right end-points of the interval. In case $\mathfrak{TS}(A_a)$ contains a finite

number of intervals (possibly the last interval of infinite length) then we are done.

Otherwise, $\mathfrak{IS}(A_a)$ contains infinitely-many intervals, which form an eventually periodic sequence with respect to the sizes of the intervals and the distances among them. Then we need to attach a loop at the end of $\tilde{\gamma}_a$. We distinguish between two cases.

Case (i): The periodic part of $\mathfrak{IS}(A_a)$ contains an integral point n (not necessarily as an isolated point). Then we first split the interval, say $[a, b)$, to which n belongs into disjoint intervals $[a, n), \{n\}, (n, b)$, such that the point n belongs to a singleton. Then we extend $\tilde{\gamma}_a$ until reaching $\{n\}$, so that the last transition of $\tilde{\gamma}_a$ is constrained to $x = n$ while resetting x . From that point begins the loop $\tilde{\pi}_a$, which obeys the same rules as applied to $\tilde{\gamma}_a$, with x being reset only when finishing the loop (see Fig. 6 (a)).

Case (ii): The periodic part of $\mathfrak{IS}(A_a)$ does not contain an integral point, that is, it is a union of open unit intervals $(n, n + 1)$. Then, if necessary, we split the last interval before starting the loop into two with the second component a unit open interval (we know that this last interval is not a singleton). This unit interval refers to the last transition of $\tilde{\gamma}_a$ and we reset x on that transition. Then, all transitions within the loop $\tilde{\pi}_a$ are forced to occur at integral times, with x being reset when completing the whole loop (see Fig. 6 (b)) (hence, in both cases the clock x is reset in each \tilde{A}_a only on a transition to the vertex $v_a \in \tilde{\gamma}_a \cap \tilde{\pi}_a$). The idea is that if we enter the loop at a fractional time, say $c = 0.3$, then all the next transitions will take place at times $n + 0.3$, but since c can be arbitrarily chosen within the open interval $(0, 1)$ then the set of all runs will cover the entire timestamp. \square

Example 7.7. Let A be a timed automaton with timestamp

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

Then a possible timestamp automaton of A is given in Fig. 6.

Example 7.8. The language of the timed automaton $A \in \text{eNTA}$ of Fig. 7 (a) is $\mathcal{L}(A) = \{(t_0, a), (t_1, a), \dots, (t_n, a) \mid i < t_i < i + 1, i \in \mathbb{N}_0\}$ (supposing all locations are ‘accepting’). The timestamp of A is $\mathfrak{IS}(A) = \{(n, n + 1) \mid n \in \mathbb{N}_0\} \times \{a\}$. A is not determinizable. The reason is that 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 the guards can use is the clock reset at time 0, hence the guards need to refer to all natural numbers, which is impossible by the finiteness of the automaton. Nevertheless, the timestamp automaton associated with A , seen in Fig. 7 (b), is deterministic.

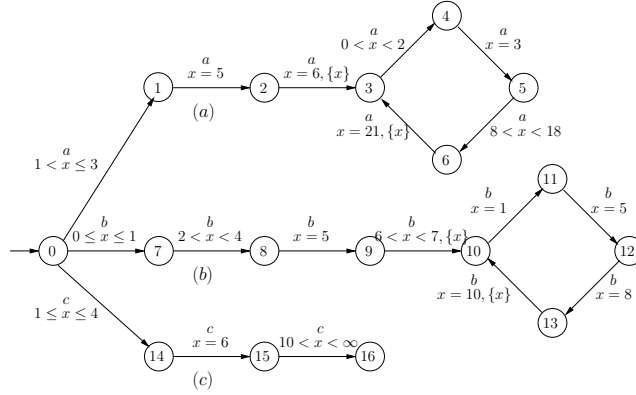


Figure 6: Timestamp automata of a) $\mathfrak{TS}(A_a)$; b) $\mathfrak{TS}(A_b)$; c) $\mathfrak{TS}(A_c)$

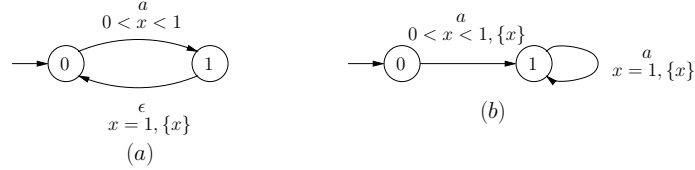


Figure 7: a) A non-determinizable $A \in \text{eNTA}$; b) A timestamp automaton \tilde{A}

7.2 Timestamp of First Observable Event

We give here a simple application of the timestamp construction. The language universality problem asks whether the language of a timed automaton consists of all possible timed traces. It is known [AD94] to be undecidable in general. Here we address a simpler problem, which is decidable. Given a fixed positive integer k , we say that the language $\mathfrak{L}(A)$ has the *k-bounded universality* property if $\mathfrak{L}_k(A) := \{w \in \mathfrak{L}(A) \mid |w| \leq k\}$ contains all possible observable timed traces of length at most k .

Proposition 7.9. *The 1-bounded language universality problem for eNTA is decidable.*

Proof. We add to A a location which is a 'sink' and redirect each observable transition to it. Then we compute the timestamp of the connected component of the initial location and check whether it equals $\mathbb{R}_{\geq 0}$. \square

In fact, the same construction for computing the first timestamp may be used for comparing $A_1, A_2 \in \text{eNTA}$. Let $L = \text{lcm}(L_1, L_2)$, where L_1, L_2 are periods of A_1, A_2 respectively, and let $t_{\text{per}} = \max(t_{\text{per}_1}, t_{\text{per}_2})$. Then we compute the first timestamp of A_1, A_2 with period parameters t_{per} and L . Due to the eventual periodicity of the timestamps it suffices to conduct the comparison up

to time $t_{\text{per}} + L$. The next proposition about the 1-bounded language inclusion, $\mathfrak{L}_1(A_1) \subseteq \mathfrak{L}_1(A_2)$, is a generalization of Proposition 7.9.

Theorem 7.10. *The 1-bounded language inclusion problem for eNTA is decidable.*

Notice that the set of timed traces of observable length 1 may be unbounded by length or by time since each such timed trace may contain a prefix of silent transitions, and these prefixes may be unbounded in length or time due to cycles of silent transitions, also with clock reset. Hence, even the seemingly simple 1-bounded language problems do not fit into the setting of bounded time, proved to be decidable in [OW10]), or to the setting of bounded number of steps, which was proved to be decidable in [LRNA15].

We would also like to remark that by constructing an automaton which contains a sequence of several copies A_i of A , directing the observable transitions from A_i to A_{i+1} and transforming the necessary transitions from observable to silent transitions, one can compute the k -th timestamp (the timestamp of all k -th observable actions, when considering all runs of A), or the timestamp of the k -th until the l -th observable action, or the timestamp that excludes the ones between the k -th and the l -th, or the timestamp of events that obey specific paths of A or specific sequence of actions, etc.

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. However, the timestamp may be seen as overly abstract since it does not take into account the order in which events occur. One can construct discretized and deterministic automata, in which time is not continuous but jumps in steps of $1/n$ units. These discretized versions of timed automata can approximate (continuous) timed automata with an error that is as small as we wish. The price for such constructions is higher complexity: the better the approximation the higher the computation complexity. It is thus important to find more efficient deterministic models which can give good

approximations of real-valued models and at the same time reduce the probability of 'false positive' language inclusion results. More efficient representations of time-periodic structures similar to zones or other symbolic representations [MPS11] should also be sought for.

When $k > 1$, the decidability of the k -bounded language inclusion problem for eNTA is in general still open, as far as we know. It cannot be handled by just considering the timestamp, which merges the times at which events of all runs occur whereas the information about the separate timed traces is being lost. This is another interesting question that may be worth studying.

Acknowledgement. This work was supported by the Austrian Science Fund (FWF): Project P29355-N35.

References

- [AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
- [AFH99] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211(1-2):253–273, 1999.
- [AKV98] Rajeev Alur, Robert P. Kurshan, and Mahesh Viswanathan. Membership questions for timed and hybrid automata. In *Real-Time Systems Symposium*, pages 254–263, 1998.
- [AM99] Eugene Asarin and Oded Maler. As soon as possible: Time optimal control for timed automata. In *HSCC*, pages 19–30, 1999.
- [AM04] Rajeev Alur and P. Madhusudan. Decision problems for timed automata: A survey. In *Formal Methods for the Design of Real-Time Systems, International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 1–24, 2004.
- [BBBB09] Christel Baier, Nathalie Bertrand, Patricia Bouyer, and Thomas Brihaye. When are timed automata determinizable? In *ICALP (2)*, pages 43–54, 2009.
- [BDFP04] Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit. Updatable timed automata. *Theor. Comput. Sci.*, 321(2-3):291–345, 2004.
- [BDM⁺98] Marius Bozga, Conrado Daws, Oded Maler, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. Kronos: A model-checking tool for real-time systems. In *CAV*, pages 546–550, 1998.

- [Bea98] Danièle Beauquier. Pumping lemmas for timed automata. In *FoS-SaCS*, pages 81–94, 1998.
- [BPDG98] Béatrice Bérard, Antoine Petit, Volker Diekert, and Paul Gastin. Characterization of the expressive power of silent transitions in timed automata. *Fundam. Inform.*, 36(2-3):145–182, 1998.
- [CG00] Christian Choffrut and Massimiliano Goldwurm. Timed automata with periodic clock constraints. *Journal of Automata, Languages and Combinatorics*, 5(4):371–404, 2000.
- [CHKM11] Taolue Chen, Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre. Reachability probabilities in markovian timed automata. In *CDC-ECC*, pages 7075–7080, 2011.
- [CJ99] Hubert Comon and Yan Jurski. Timed automata and the theory of real numbers. In *CONCUR*, pages 242–257, 1999.
- [CY92] Costas Courcoubetis and Mihalis Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4):385–415, 1992.
- [DT98] Conrado Daws and Stavros Tripakis. Model checking of real-time reachability properties using abstractions. In *TACAS*, pages 313–329, 1998.
- [Fin06] Olivier Finkel. Undecidable problems about timed automata. In *FORMATS*, pages 187–199, 2006.
- [HOW12] Christoph Haase, Joël Ouaknine, and James Worrell. On the relationship between reachability problems in timed and counter automata. In *Reachability Problems*, pages 54–65, 2012.
- [HP06] Thomas A. Henzinger and Vinayak S. Prabhu. Timed alternating-time temporal logic. In *FORMATS*, pages 1–17, 2006.
- [LPY97] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *STTT*, 1(1-2):134–152, 1997.
- [LRNA15] Florian Lorber, Amnon Rosenmann, Dejan Nickovic, and Bernhard K. Aichernig. Bounded determinization of timed automata with silent transitions. In *FORMATS*, pages 1–17, 2015.
- [MPS11] Georges Morbé, Florian Pigorsch, and Christoph Scholl. Fully symbolic model checking for timed automata. In *CAV*, pages 616–632, 2011.
- [ORW09] Joël Ouaknine, Alexander Rabinovich, and James Worrell. Time-bounded verification. In *CONCUR*, pages 496–510, 2009.

- [OW04] Joël Ouaknine and James Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *LICS*, pages 54–63, 2004.
- [OW10] Joël Ouaknine and James Worrell. Towards a theory of time-bounded verification. In *ICALP (2)*, pages 22–37, 2010.
- [TY01] Stavros Tripakis and Sergio Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.
- [Wan04] Farn Wang. Efficient verification of timed automata with bdd-like data structures. *STTT*, 6(1):77–97, 2004.
- [WZP03] Bożena Wozna, Andrzej Zbrzezny, and Wojciech Penczek. Checking reachability properties for timed automata via SAT. *Fundam. Inform.*, 55(2):223–241, 2003.