Christel Baier, Nathalie Bertrand, Patricia Bouyer and Thomas Brihaye

When are Timed Automata Determinizable?

Research Report LSV-09-08

14 April 2009

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

When are Timed Automata Determinizable?

Christel Baier[1], Nathalie Bertrand[2], Patricia Bouyer[3], and Thomas Brihaye[4]

[1] Technische Universität Dresden, Germany
[2] INRIA Rennes Bretagne Atlantique, France
[3] LSV, CNRS & ENS Cachan, France
[4] Université de Mons, Belgium

Abstract. In this paper, we propose an abstract procedure which, given a timed automaton, produces a language-equivalent deterministic infinite timed tree. We prove that under a certain boundedness condition, the infinite timed tree can be reduced into a classical *deterministic* timed automaton. The boundedness condition is satisfied by several subclasses of timed automata, some of them were known to be determinizable (event-clock timed automata, automata with integer resets), but some others were not. We prove for instance that strongly non-Zeno timed automata can be determinized. As a corollary of those constructions, we get for those classes the decidability of the universality and of the inclusion problems, and compute their complexities (the inclusion problem is for instance EXPSPACE-complete for strongly non-Zeno timed automata).

1 Introduction

Timed automata have been proposed by Alur and Dill in the early 90s as a model for real-time systems [2]. A timed automaton is a finite automaton which can manipulate real-valued variables called clocks, that evolve synchronously with the time, can be tested and reset to zero. One of the fundamental properties of this model is that, although the set of configurations is in general infinite, checking reachability properties is decidable. From a language-theoretic point of view, this means that checking emptiness of the timed language accepted by a timed automaton can be decided (and is a PSPACE-complete problem). The proof relies on the construction of the so-called region automaton, which finitely abstracts behaviours of a timed automaton. Since then, its appropriateness as a model for the verification of real-time systems has been confirmed, with the development of verification algorithms and dedicated tools.

There are however two weaknesses to that model: a timed automaton cannot be determinized, and inclusion (and universality) checking is undecidable [2], except for deterministic timed automata. This basically forbids the use of timed automata as a specification language. Understanding and coping with these weaknesses have attracted lots of research, and, for instance, testing whether a timed automaton is determinizable has been proved undecidable [6]. Also, the undecidability of universality has been further investigated, and rather restricted classes of timed automata suffer from that undecidability result [1]. On the other hand,

classes of timed automata have been exhibited, that either can be effectively determinized (for instance event-clock timed automata [3], or timed automata with integer resets [9]), or for which universality can be decided (for instance single-clock timed automata [7]).

In this paper, we describe a generic construction that is applicable to every timed automaton, and which, under certain conditions, yields a deterministic timed automaton, which is language-equivalent to the original timed automaton. The idea of the procedure is to unfold the timed automaton into a finitely-branching infinite tree that records the timing constraints that have to be satisfied using one clock per level of the tree (hence infinitely many clocks). When reading a finite timed word in that infinite tree, we may reach several nodes of the tree, but the timing information stored in the clocks is independent of the run in the tree. Thanks to this kind of *input-determinacy* property, we can determinize this infinite object, yielding another finitely-branching infinite tree. And, under a boundedness condition on the amount of timing information we need to store, we will be able to fold back the tree into a deterministic timed automaton. This boundedness condition is not a syntactical condition on the original timed automaton, but will be satisfied by large classes of timed automata: event-clock timed automata [3], timed automata with integer resets [9], and strongly non-Zeno timed automata. Furthermore, our construction yields automata of exponential-size in the first case, and doubly-exponential-size automata otherwise. In particular, our approach provides an EXPSPACE algorithm to check universality (and inclusion) for a large class of timed automata, and we prove that this complexity is tight. Our algorithm can easily be adapted into a PSPACE one, in the special case of event-clock timed automata, allowing to recover the known result of [3].

2 Timed automata

Preliminaries. Given X a finite or infinite set of clocks and M a non-negative integer, a clock *valuation* over X bounded by M is a mapping $v : X \to \mathbb{T}_M$ where $\mathbb{T}_M = [0, M] \cup \{\bot\}$. We assume furthermore that $\bot > M$. The notation $\bot$ is for abstracting values of clocks that are above some fixed value M. This is rather non-standard (though used for instance in [8]) but it will be convenient in this paper. We note $\bar{0}$ the valuation that assigns 0 to all clocks. If v is a valuation over X and bounded by M, and $t \in \mathbb{R}_+$, then $v + t$ denotes the valuation which assigns to every clock $x \in X$ the value $v(x) + t$ if $v(x) + t \leq M$, and $\bot$ otherwise (in particular, if $v(x) = \bot$, then $(v + t)(x) = \bot$). For $Y \subseteq X$ we write $[Y \leftarrow 0]v$ for the valuation equal to v on $X \setminus Y$ and to $\bar{0}$ on Y, and $v_{|Y}$ for the valuation v restricted to clocks in Y. A(n M-bounded) *guard* (or *constraint*) over X is a finite conjunction of constraints of the form $x \sim c$ where $x \in X$, $c \in \mathbb{N} \cap [0, M]$ and $\sim \in \{<, \leq, =, \geq, >\}$. We denote by $\mathcal{G}_M(X)$ the set of M-bounded guards over X. Given a valuation v and a guard g we write $v \models g$ whenever v satisfies g.

A timed word over Σ is a finite sequence of pairs $(a_1, t_1)(a_2, t_2) \dots (a_k, t_k)$ such that for every i, $a_i \in \Sigma$ and $(t_i)_{1 \leq i \leq k}$ is a nondecreasing sequence in $\mathbb{R}_+$.

Timed automata. A *timed automaton* is a tuple $\mathcal{A} = (L, \ell_0, L_{\mathsf{acc}}, X, M, E)$ such that: (i) L is a finite set of locations, (ii) $\ell_0 \in L$ is the initial location, (iii) $L_{\mathsf{acc}} \subseteq L$ is the set of final locations, (iv) X is a finite set of clocks, (v) $M \in \mathbb{N}$, and (vi) $E \subseteq L \times \mathcal{G}_M(X) \times \Sigma \times 2^X \times L$ is a finite set of edges. Constant M is called the maximal constant of $\mathcal{A}$.

The semantics of a timed automaton $\mathcal{A}$ is given as a timed transition system $\mathcal{T}_{\mathcal{A}} = (S, s_0, S_{\mathsf{acc}}, (\mathbb{R}_+ \times \Sigma), \rightarrow)$ with set of states $S = L \times \mathbb{T}_M^X$, initial state $s_0 = (\ell_0, \bar{0})$, set of accepting states $S_{\mathsf{acc}} = L_{\mathsf{acc}} \times \mathbb{T}_M^X$, and transition relation $\rightarrow \subseteq S \times (\mathbb{R}_+ \times \Sigma) \times S$ composed of moves of the form $(\ell, v) \xrightarrow{\tau, a} (\ell', v')$ whenever there exists an edge $(\ell, g, a, Y, \ell') \in E$ such that $v+\tau \models g$ and $v' = [Y \leftarrow 0](v+\tau)$.

A run ϱ of $\mathcal{A}$ is a finite sequence of moves, *i.e.*, $\varrho = s_0 \xrightarrow{\tau_1, a_1} s_1 \ldots \xrightarrow{\tau_k, a_k} s_k$. It is said initial whenever $s_0 = (\ell_0, \bar{0})$. An initial run is accepting if it ends in an accepting location. The timed word $u = (a_1, t_1)(a_2, t_2)\ldots(a_k, t_k)$ is said to be read on ϱ whenever $t_i = \sum_{j=1}^{i} \tau_j$ for every $1 \leq i \leq k$. We write $\mathcal{L}(\mathcal{A})$ for the set of timed words (or timed language) accepted by $\mathcal{A}$, that is the set of timed words u such that there exists an initial and accepting run ϱ which reads u.

A timed automaton $\mathcal{A}$ is *deterministic* whenever for every timed word u, there is at most one initial run which reads u. It is *strongly non-Zeno* whenever there exists $K \in \mathbb{N}$ such that for every run $\varrho = s_0 \xrightarrow{\tau_1, a_1} s_1 \ldots \xrightarrow{\tau_k, a_k} s_k$ in $\mathcal{A}$, $k \geq K$ implies $\sum_{i=1}^{k} \tau_i \geq 1$. This condition is rather standard in timed automata [4].

Example 1. An example of timed automaton is depicted in Fig. 1. This automaton will be used as a running example throughout the paper in order to illustrate the different steps of our construction. This automaton is not deterministic and accepts the timed language $\{(a, t_1)(a, t_2)\cdots(a, t_{2n}) \mid n \geq 1,\ 0 < t_1 < t_2 < \cdots < t_{2n-1}$ and $t_{2n} - t_{2n-2} = 1\}$, with the convention that $t_0 = 0$. The timed word $(a, 0.5)(a, 1.6)(a, 2.9)$ can be read on the initial run $(\ell_0, 0) \xrightarrow{0.5, a} (\ell_3, 0) \xrightarrow{1.1, a} (\ell_0, 0) \xrightarrow{1.3, a} (\ell_1, \bot)$ but is not accepted. The last configuration of the above run is $(\ell_1, \bot)$ because the value of clock x should be 1.3, but as it is larger than the maximal constant 1, we abstract the precise value into $\bot$.

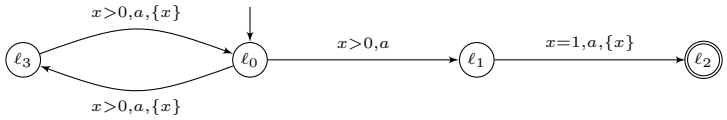


Fig. 1. A timed automaton $\mathcal{A}$

On timed bisimulations. A *strong timed (resp. time-abstract) simulation relation* between two timed transition systems $\mathcal{T}_i = (S_i, s_{i,0}, S_{i,\mathsf{acc}}, (\Sigma \cup \mathbb{R}_+), \rightarrow_i)$ for $i \in \{1, 2\}$ is a relation $\mathfrak{R} \subseteq S_1 \times S_2$ such that if $s_1\ \mathfrak{R}\ s_2$ and $s_1 \xrightarrow{t_1, a} s_1'$ for some $t_1 \in \mathbb{R}_+$ and $a \in \Sigma$, then there exists $s_2' \in S_2$ (resp. $t_2 \in \mathbb{R}_+$ and $s_2' \in S_2$) such that $s_2 \xrightarrow{t_1, a} s_2'$ (resp. $s_2 \xrightarrow{t_2, a} s_2'$) and $s_1'\ \mathfrak{R}\ s_2'$. A *strong timed (resp. time-abstract) bisimulation relation* between two timed transition $\mathcal{T}_i$ for $i \in \{1, 2\}$ is a relation $\mathfrak{R} \subseteq S_1 \times S_2$ such that both $\mathfrak{R}$ and $\mathfrak{R}^{-1}$ are strong timed (resp. time-abstract) simulation relations. The above relations *preserve* initial (resp.

accepting) states whenever $s_{1,0}\ \mathfrak{R}\ s_{2,0}$ (resp. $s_1\ \mathfrak{R}\ s_2$ and $s_i \in S_{i,\mathsf{acc}}$ implies $s_{3-i} \in S_{3-i,\mathsf{acc}}$). Note that the notion of strong timed bisimulation which preserves initial and accepting states is stronger than that of language equivalence.

The classical region construction. We let X be a finite set of clocks, and $M \in \mathbb{N}$. We define the equivalence relation $\equiv_{X,M}$ between valuations in $\mathbb{T}_M$ as follows: $v \equiv_{X,M} v'$ iff (i) for every clock $x \in X$, $v(x) \leq M$ iff $v'(x) \leq M$; (ii) for every clock $x \in X$, if $v(x) \leq M$, then $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$, and (ii) for every pair of clocks $(x, y) \in X^2$ such that $v(x) \leq M$ and $v'(x) \leq M$, $\{v(x)\} \leq \{v(y)\}$ iff $\{v'(x)\} \leq \{v'(y)\}$. [5] The equivalence relation is called the *region equivalence* for the set of clocks X w.r.t. M, and an equivalence class is called a *region*. We note Reg_M^X for the set of such regions. A region r' is a time-successor of a region r if there is $v \in r$ and $t \in \mathbb{R}_+$ such that $v + t \in r'$. If v is a valuation, we will write $[v]$ for the region to which v belongs (when X and M are clear in the context).

It is a classical result [2] that given a timed automaton $\mathcal{A}$ with maximal constant M and set of clocks X, the relation $\mathfrak{R}_{X,M}$ between configurations of $\mathcal{A}$ defined by $(\ell, v)\ \mathfrak{R}_{X,M}\ (\ell, v')$ iff $v \equiv_{X,M} v'$ is a time-abstract bisimulation.

3 Some transformations

In this section, we describe a general construction that aims at determinizing a timed automaton. We know however that not all timed automata can be determinized [2], and even that we cannot decide whether a timed automaton can be determinized [6]. We will thus give conditions that will ensure (i) that our procedure can be properly applied, and (ii) that the resulting timed automaton is deterministic and accepts the same timed language as the original automaton. We will then analyze the complexity of the procedure, and apply it to several subclasses of timed automata, some of which were known to be determinizable, some other were not known to be determinizable.

This construction consists in four steps: (i) an unfolding of the original automaton into an infinite timed tree, (ii) a region abstraction, (iii) a symbolic determinization, and (iv) a reduction of the number of clocks, allowing to fold the tree back into a timed automaton. These steps are described in the following subsections. Due to page limitation, we will give no formal definitions of the objects we build in our construction, and better illustrate the construction on the running example. All details of the construction can be found in the appendix.

3.1 Construction of an equivalent infinite timed tree

In this first step, we unfold the timed automaton $\mathcal{A}$ into a finitely-branching *infinite timed tree* $\mathcal{A}^\infty$ that has infinitely many clocks (one clock per level of the tree), we call $Z = \{z_0, z_1, \ldots\}$ this infinite set of clocks. The idea of this unfolding is to use a fresh clock reset at each level of the tree in order to record

[5] Where $\lfloor \alpha \rfloor$ (resp. $\{\alpha\}$) denotes the integral (resp. fractional) part of α.

the timing constraints that have to be satisfied in $\mathcal{A}$. Each node n of $\mathcal{A}^\infty$ is labelled by a pair $(\ell, \sigma) \in L \times Z^X$ where ℓ records the location of $\mathcal{A}$ that node n simulates and σ describes how the clocks of $\mathcal{A}$ are encoded using the clocks of $\mathcal{A}^\infty$ (if $\sigma(x) = z_i$, the value clock x would have in $\mathcal{A}$ is the current value of clock z_i). The advantage of this infinite timed tree is that it enjoys some *input-determinacy* property: when reading a finite timed word u in $\mathcal{A}^\infty$, there may be several runs in the tree that read u, but the timing information stored in the clocks is independent of the run in the tree (see Remark 4).

Example 2. Part of the infinite timed tree $\mathcal{A}^\infty$ associated with the timed automaton $\mathcal{A}$ of Fig. 1 is depicted in Fig. 2. Notice that a fresh clock is reset at each level; for instance z_2 is reset on all edges from level-1 to level-2 nodes (*i.e.* $n_1 \to n_3$ and $n_2 \to n_4$). The timed tree $\mathcal{A}^\infty$ corresponds to the unfolding of $\mathcal{A}$: the two branches starting from the node n_0 represent the possible choice in state ℓ_0 of $\mathcal{A}$; the same phenomenon also happens in n_4. The label of n_4 is (ℓ_0, z_2); it means that node n_4 represents the location ℓ_0 of $\mathcal{A}$ and that the value of clock x can be recovered from the current value of clock z_2. It is important to observe how the second component of the label evolves. First consider the edge $n_4 \to n_5$; it represents the transition from ℓ_0 to ℓ_1 in $\mathcal{A}$, which does not reset clock x; the reference for clock x is the same in n_5 as it is in n_4, that is why the label of n_5 is (ℓ_1, z_2). Now consider the edge $n_4 \to n_6$; it represents the transition from ℓ_0 to ℓ_3 in $\mathcal{A}$, which resets clock x; the reference for clock x thus becomes z_3, the clock which has just been reset, that is why the label of n_6 is (ℓ_3, z_3).

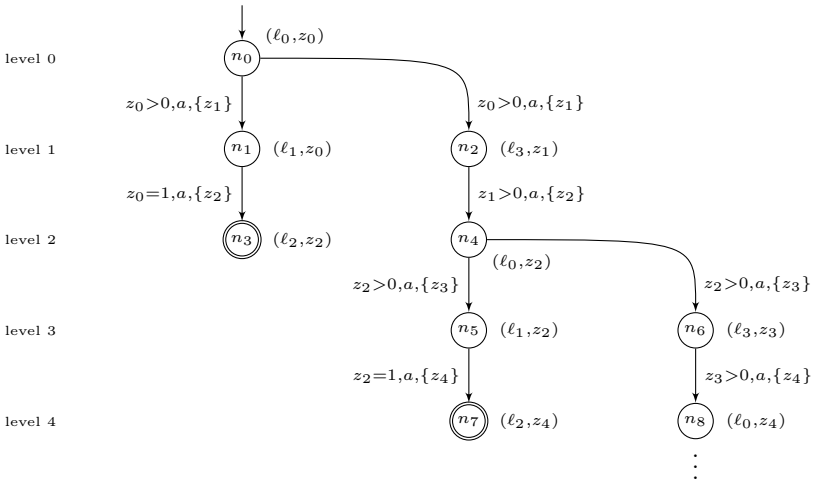


Fig. 2. The infinite timed tree $\mathcal{A}^\infty$ associated with the timed automaton $\mathcal{A}$ of Fig. 1.

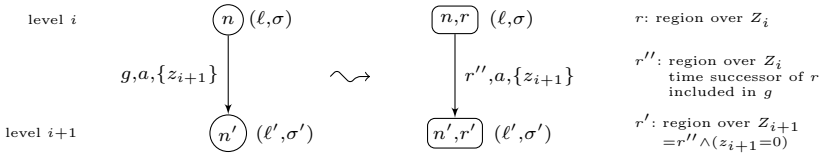
The correctness of this unfolding is stated in the follow lemma.

Lemma 3. *The relation $\mathfrak{R}_1$ between states of $\mathcal{A}$ and states of $\mathcal{A}^\infty$ defined by $(\ell, v \circ \sigma)\ \mathfrak{R}_1\ (n, v)$ if label$(n) = (\ell, \sigma)$ is a strong timed bisimulation.*

Remark 4. In $\mathcal{A}^\infty$, for every finite timed word u, there is a unique valuation $v_u \in \mathbb{T}^Z$ such that for every initial run ϱ in $\mathcal{A}^\infty$ that reads u, ϱ ends in some configuration (n, v_u) with $level(n) = |u|$. Indeed, if the timed word u is of the form $(a_1, t_1)...(a_{|u|}, t_{|u|})$, any initial run ϱ reading u necessarily ends in a configuration (n, v_u) where $level(n) = |u|$ and $v_u(z_j) = t_{|u|} - t_j$ for any $j \leq |u|$.

3.2 A region abstraction

In this second step, we extend in a natural way the classical region equivalence to the above infinite timed tree: at level i of the tree, only clocks in $Z_i = \{z_0, \cdots, z_i\}$ are relevant (all other clocks have not been used yet), we thus consider regions over that set of clocks. We use $R(\mathcal{A}^\infty)$ to denote this region abstraction, and we interpret it in a timed manner. We do not illustrate this transformation step on our running example, since $R(\mathcal{A}^\infty)$ is easily obtained from $\mathcal{A}^\infty$, but only depict the transformation on an edge, see below:



It is worth noting that, in $R(\mathcal{A}^\infty)$, any state reached after a transition is of the form $((n, r), v)$, where n is a node of $\mathcal{A}^\infty$ (of some level, say i), r is a region over Z_i, and v is a valuation over Z_i which belongs to r. It is not difficult to see that, as in the standard region construction in timed automata, two states $((n, r), v_1)$ and $((n, r), v_2)$ with $v_1, v_2 \in r$ are time-abstract bisimilar. Furthermore, $R(\mathcal{A}^\infty)$ will satisfy the same input-determinacy property as $\mathcal{A}^\infty$ (see Remark 4). The correctness of $R(\mathcal{A}^\infty)$ can then be stated as follows.

Lemma 5. *The relation $\mathfrak{R}_2$ between states of $\mathcal{A}^\infty$ and states of $R(\mathcal{A}^\infty)$ defined by $(n, v)\ \mathfrak{R}_2\ ((n, r), v)$ if $v \in r$ is a strong timed bisimulation.*

3.3 Symbolic determinization

This third step is the crucial step of our construction. We will symbolically determinize the infinite timed tree $R(\mathcal{A}^\infty)$ using a rather standard subset construction, and we denote by $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ the resulting infinite tree. However there will be a subtlety in the subset construction: useless clocks will be forgotten 'on-the-fly'. More precisely, at each node, we only consider active clocks, *i.e.* clocks that appear in the label of the node (other clocks record values that do not impact on further behaviours of the system). The determinization is then performed on the 'symbolic' alphabet composed of regions over active clocks and actions, and thanks to the input-determinacy property of $R(\mathcal{A}^\infty)$, this symbolic determinization coincides with the determinization of the underlying timed transition system. Let us explain this crucial step on our running example.

Example 6. The construction of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is illustrated on Fig. 3. The determinization is performed using a classical subset construction. For example starting from node n_0, both n_1 and n_2 can be reached *via* a transition with guard $0 < z_0 < 1$. This is reflected in the leftmost $\{n_1, n_2\}$-node at the first level. It is also important to understand the meaning of active clocks. In $\mathcal{A}^\infty$, the only active clock in node n_4 is z_2. Therefore, guards on transitions leaving the node $(\{n_4\}, z_2 = 0)$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ are regions over this unique clock z_2. If we consider a node combining n_5 and n_6, active clocks will consist in the union of active clocks in both nodes, hence z_2 and z_3. For sake of readability, we have mostly omitted labels of nodes on Fig. 3, but they can be naturally inferred from those in $R(\mathcal{A}^\infty)$; for instance, the label of the top-rightmost node is $\{(\ell_1, z_0), (\ell_3, z_1)\}$, the union of the labels of n_1 and n_2 in $R(\mathcal{A}^\infty)$.

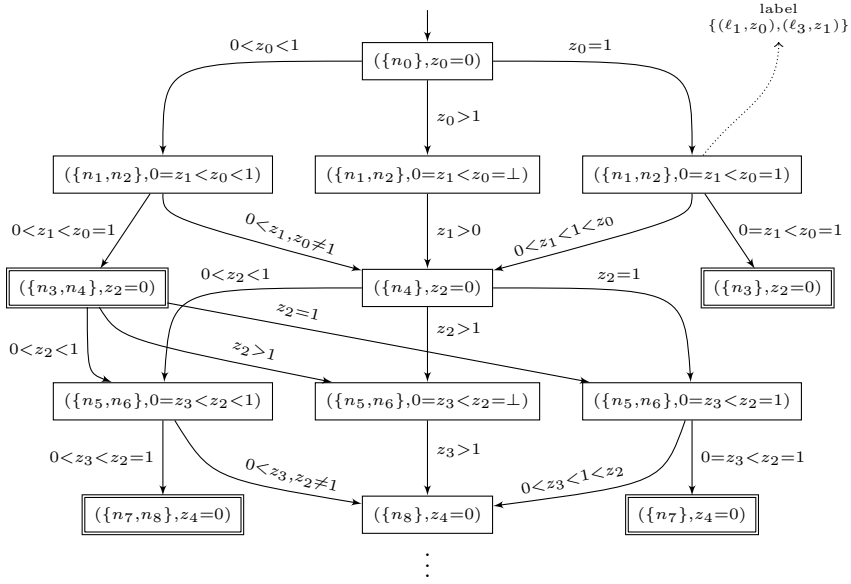


Fig. 3. The DAG induced by the infinite timed tree $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$

The subset construction induces a DAG (as seen on Fig. 3). However the rest of the construction will require a tree instead of a DAG; we thus add markers to nodes, so that we can have several copies of a node, depending on the ancestors. A node in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is thus a tuple $(\star, K, r)$ where $\star$ is a marker, K is a subset of node names in $R(\mathcal{A}^\infty)$ (they all have same level), and r is a region over the set $Act(K) = \bigcup_{n \in K, label(n) = (\ell, \sigma)} \sigma(X)$, the set of active clocks in K.

The correctness of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is stated in the following proposition.

Proposition 7. *$\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is a deterministic timed tree, and for every node $N = (\star, K, r)$ and for every valuation $v \in \mathbb{T}^{Act(K)}$ with $v \in r$,*

$$\mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N, v)) = \bigcup_{n \in K} \mathcal{L}(R(\mathcal{A}^\infty), ((n, r), v))$$

Remark 8. In case $\mathcal{A}$ has a single clock x, a level-i node of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ carries the following information: a finite set of pairs of the form $(\ell, x \mapsto z_j)$ for some $j \leq i$ and a region for clocks in Z_i. We skip details, but with this information, we can easily recover the well-quasi-order that gives the decidability of the universality problem in single-clock timed automata [7].

3.4 Reduction of the number of clocks

$\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is an infinite object (it is an infinite timed tree and it has infinitely many clocks). Our aim is to fold this tree back into a deterministic timed automaton. Obviously we cannot do so for all timed automata, and so far we have not made any assumption on $\mathcal{A}$. Given $\gamma \in \mathbb{N}$, we say that $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is *γ-clock-bounded* if in every node, the number of active clocks is bounded by γ. Under this hypothesis, we will be able to quotient $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ by an equivalence of finite index, and get a deterministic timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$ which accepts the same language as the original timed automaton $\mathcal{A}$.

The idea will be to fix a finite set of clocks $X_\gamma = \{x_1, \cdots, x_\gamma\}$, and starting from the level-0 node of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ to rename the active clocks into clocks in X_γ following a deterministic policy. Under the γ-clock-boundedness assumption, each time we will require a new clock (because a new one has become active), there will be (at least) one free clock in X_γ. Of course, we rename clocks in guards and regions as well, and change the labels of the nodes accordingly (an element of the label of a node is now a pair (ℓ, σ) where ℓ is a location of $\mathcal{A}$ and $\sigma \colon X \mapsto X_\gamma$ assigns to each clock of $\mathcal{A}$ its representative in the tree). The new object is still infinite, but it has finitely many clocks. A node is now a tuple $(\star, K, r)$ where $\star$ is a marker, K is a subset of nodes in $R(\mathcal{A}^\infty)$ and r is a region over (a subset of) X_γ. Now it is just a matter of noticing that two nodes with the same region and the same labels are isomorphic and strongly timed bisimilar (in particular they are language-equivalent). Timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$ is obtained by merging such nodes.

Example 9. In Fig. 3, it is easy to see that $\mathsf{SymbDet}(\mathcal{A}^\infty)$ is 2-clock-bounded. So one can rename the clocks to $X_2 = \{x_1, x_2\}$, for instance we can map clocks with even indices to x_1 and clocks with odd indices to x_2. After this renaming, nodes sharing the same label (that is: set of locations of $\mathcal{A}$, mappings from X to $\{x_1, x_2\}$ and regions over $\{x_1, x_2\}$) can be merged. Indeed, one can show that subtrees rooted at nodes with the same label are strongly timed bisimilar. For instance, in our running example, nodes $(\{n_0\}, z_0 = 0)$ and $(\{n_4\}, z_2 = 0)$, labelled respectively by $\{(\ell_0, z_0)\}$ and $\{(\ell_0, z_2)\}$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, are merged into a single location with region $x_1 = 0$. The resulting timed automaton is depicted on Fig. 4. In general, a location of this automaton is of the form $(\{(\ell_j, \sigma_j) \mid j \in J\}, r)$ where J is a finite set, ℓ_j is a location of $\mathcal{A}$, $\sigma_j \colon X \to X_2$, and r is a region over a subset of X_2. In our running example, there is a single clock x, hence we assimilate σ_j with the value $\sigma_j(x)$.

The correctness of the construction is stated in the following proposition.

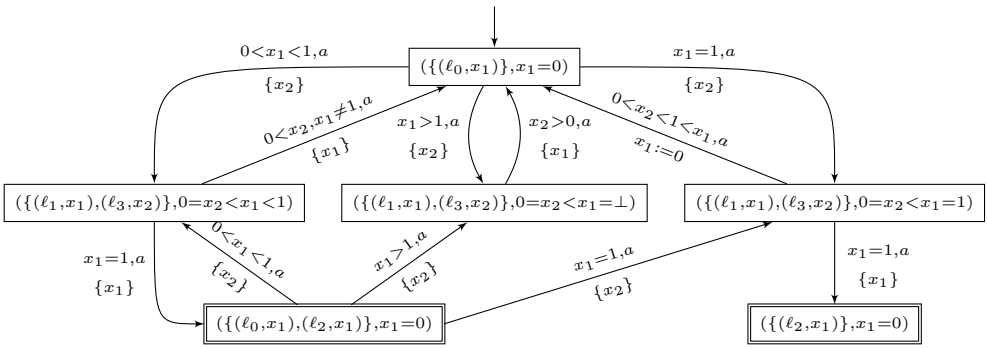


Fig. 4. The deterministic version of $\mathcal{A}$: the timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$

Proposition 10. *Assume that* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is* γ*-clock-bounded. Then,* $\mathcal{B}_{\mathcal{A},\gamma}$ *is a deterministic timed automaton, and* $\mathcal{L}(\mathcal{B}_{\mathcal{A},\gamma}) = \mathcal{L}(\mathcal{A})$.

3.5 Algorithmic issues and complexity

In this subsection, we shortly discuss the size of the effectiveness of its construction. If $\mathcal{A} = (L, \ell_0, L_{\mathsf{acc}}, X, M, E)$ is a timed automaton such that $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is γ-clock-bounded (for some $\gamma \in \mathbb{N}$), then the timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$ has roughly $\alpha(\mathcal{A},\gamma) = 2^{|L|} \cdot \gamma^{|X|} \cdot \Big((2M+2)^{(\gamma+1)^2} \cdot \gamma!\Big)$ locations because a location is characterized by a finite set of pairs (ℓ, σ) with ℓ a location of $\mathcal{A}$, $\sigma\colon X \to X_\gamma$, and a region over X_γ.

The procedure we have described goes through the construction of infinite objects. However, if we abstract away the complete construction, we know precisely how locations and transitions are derived. Hence, $\mathcal{B}_{\mathcal{A},\gamma}$ can be computed on-the-fly by guessing new transitions, and so can its complement (since $\mathcal{B}_{\mathcal{A},\gamma}$ is deterministic). A location of the automaton $\mathcal{B}_{\mathcal{A},\gamma}$ can be stored in space logarithmic in $\alpha(\mathcal{A},\gamma)$, and we will thus be able to check for universality (e.g.) in nondeterministic space $\log(\alpha(\mathcal{A},\gamma))$.

4 Our results

We will now investigate several classes of timed automata for which the procedure described in Section 3 applies.

4.1 Some classes of timed automata are determinizable

Automata satisfying the p-assumption (TA_p). Given $p \in \mathbb{N}$, we say that a timed automaton $\mathcal{A}$ satisfies the *p-assumption* if for every $n \geq p$, for every run $\varrho = (\ell_0, v_0) \xrightarrow{\tau_1,a_1} (\ell_1, v_1) \dots \xrightarrow{\tau_n,a_n} (\ell_n, v_n)$ in $\mathcal{A}$, for every clock $x \in X$, either x is reset along ϱ or $v_n(x) = \bot$. This assumption will ensure that we can apply the previous procedure, because if $\mathcal{A}$ satisfies the p-assumption, $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is p-clock-bounded. Then we observe that any strongly non-Zeno timed automaton

(we write SnZTA for this class) satisfies the p-assumption for some $p \in \mathbb{N}$ which is exponential in the size of the automaton. We thus get the following result:

Theorem 11. *For every timed automaton $\mathcal{A}$ in* SnZTA *or in* TA_p*, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is doubly-exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

Event-clock timed automata (ECTA) [3]. An *event-clock timed automaton* is a timed automaton that contains only event-recording clocks: for every letter $a \in \Sigma$, there is a clock x_a, which is reset at every occurrence of a. It is easy to see that the deterministic timed tree associated with such an automaton is $|\Sigma|$-clock-bounded. Thus, applying our procedure, we recover the result of [3], with the same complexity bound.

Theorem 12. *For every timed automaton $\mathcal{A}$ in* ECTA*, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

Timed automata with integer resets (IRTA) [9]. A *timed automaton with integer resets* is a timed automaton in which every edge $e = (\ell, g, a, Y, \ell')$ is such that Y is non empty if and only if g contains at least one atomic constraint of the form $x = c$, for some clock x. In that case, we observe that the deterministic timed tree associated with such an automaton is $(M + 1)$-clock-bounded. We thus recover the result of [9], with the same complexity bound.

Theorem 13. *For every timed automaton $\mathcal{A}$ in* IRTA*, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is doubly-exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

4.2 Deciding universality and inclusion

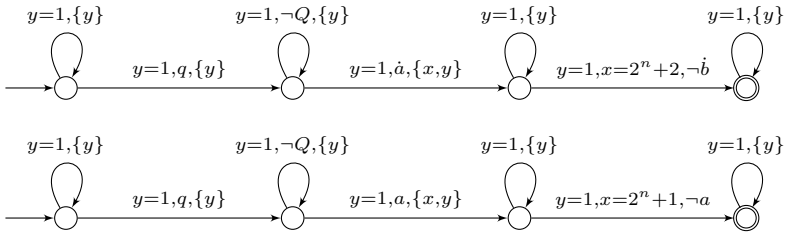
The universality and the inclusion problems are undecidable for the general class of timed automata [2]. Given $\mathcal{A}$ and $\mathcal{B}$ two timed automata, the *universality problem* asks whether $\mathcal{L}(\mathcal{A})$ is the set of all finite timed words, and the inclusion problem asks whether $\mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A})$. When $\mathcal{A}$ belongs to one of the above determinizable classes, we will be able to decide the universality and the inclusion problems (there is no need to restrict automaton $\mathcal{B}$). We establish now the precise complexity of those problems, and start by providing a lower bound for the universality problem.

Proposition 14. *Checking universality in timed automata either satisfying the p-assumption for some p or with integer resets is EXPSPACE-hard.*

Proof (sketch). The idea of the proof is as follows. Given an exponential-space Turing machine $\mathcal{M}$ with input word w_0, we define a timed automaton $\mathcal{A}_{\mathcal{M},w_0}$ such that $\mathcal{A}_{\mathcal{M},w_0}$ is universal if and only if $\mathcal{M}$ does not halt on input w_0. An

execution of $\mathcal{M}$ over w_0 is encoded by a timed word, and $\mathcal{A}_{\mathcal{M},w_0}$ will accept all finite timed words that are not encodings of halting executions of $\mathcal{M}$ on w_0. Assuming $|w_0| = n$, the maximal length of the tape is 2^n, and a configuration of $\mathcal{M}$ can be seen as a pair $\langle q, w\rangle$, where q is a control state of $\mathcal{M}$ and w is a word of length 2^n that represents the content of the tape (the position of the tape head is marked by a dotted letter). We furthermore require that actions are separated by precisely one time unit, which entails for instance that control states should be separated by precisely $2^n + 1$ time units.

A finite timed word might not be the encoding of an halting computation in $\mathcal{M}$ for several reasons: it is not the encoding of a proper execution in $\mathcal{M}$, or it does not end in the halting state, or actions do not occur at integer time points, or control states are not separated by 2^n+1 time units, *etc.* All these properties can be described using either timed automata satisfying the p-assumption, or timed automata with integer resets. For instance, a rule of the form $(q, a, b, \mathsf{right}, q')$ can be unfaithfully mimicked for two reasons: either the dotted letter (representing the position of the head) is not transferred properly (first automaton below), or the rest of the configuration is not copied properly (second automaton below).



All other cases can be handled in a similar way, which concludes the proof. □

This lower bound applies as well for the inclusion problem in the very same classes of timed automata. Note that strongly non-Zeno timed automata are never universal, but we can modify the above proof to show that the inclusion problem is EXPSPACE-hard as well for strongly non-Zeno timed automata.

Summary of the results. We can summarize our results in the following table. The column on the left indicates the subclass we consider. New results are in black and italic, and in particular we can notice that there was no lower bound known for the class IRTA.

	size of the det. TA	universality problem	inclusion problem
TA_p	*doubly exp.*	*EXPSPACE-complete*	*EXPSPACE-complete*
SnZTA	*doubly exp.*	trivial	*EXPSPACE-complete*
ECTA [3]	exp.	PSPACE-complete	PSPACE-complete
IRTA [9]	doubly exp.	EXPSPACE-*complete*	EXPSPACE-*complete*

5 Conclusion

In this paper, we proposed a general framework for the determinization of timed automata by means of an infinite timed tree. We showed that for a wide range of timed automata this infinite tree is language-equivalent to a deterministic timed automaton. The construction of this deterministic timed automaton yields the basis for algorithms to check universality or language inclusion. Concerning the complexity, these algorithms applied to event-clock timed automata [3] and timed automata with integer resets [9] provide tight bounds. In addition, our general framework yields the decidability of the universality problem for strongly non-Zeno timed automata, which was not known before.

We have focused on finite timed words, but we believe the procedure can be extended to timed automata over infinite timed words (with an ω-regular acceptance condition), by incorporating a Safra-like construction in our procedure. In that framework the strong non-Zenoness assumption will even make more sense, and we thus claim that strongly non-Zeno timed automata are determinizable!

References

1. S. Adams, J. Ouaknine, and J. Worrell. Undecidability of universality for timed automata with minimal resources. In *Proc. 5th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'07)*, volume 4763 of *Lecture Notes in Computer Science*, pages 25–37. Springer, 2007.
2. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
3. R. Alur, L. Fix, and Th. A. Henzinger. A determinizable class of timed automata. In *Proc. 6th International Conference on Computer Aided Verification (CAV'94)*, volume 818 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1994.
4. E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.
5. D. D'Souza and N. Tabareau. On timed automata with input-determined guards. In *Proc. Joint Conference on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant System (FORMATS+FTRTFT'04)*, volume 3253 of *Lecture Notes in Computer Science*, pages 68–83. Springer, 2004.
6. O. Finkel. Undecidable problems about timed automata. In *Proc. 4th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2006.
7. J. Ouaknine and J. Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *Proc. 19th Annual Symposium on Logic in Computer Science (LICS'04)*, pages 54–63. IEEE Computer Society Press, 2004.
8. J. Ouaknine and J. Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1:8), 2007.
9. P. V. Suman, P. K. Pandya, S. N. Krishna, and L. Manasa. Timed automata with integer resets: Language inclusion and expressiveness. In *Proc. 6th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'08)*, volume 5215 of *Lecture Notes in Computer Science*, pages 78–92. Springer, 2008.

Technical Appendix

Environments numbered with letters do not appear in the core of the paper.

Complements for Section 2

Infinite timed trees. An *infinite timed tree* is a tuple $\mathcal{T} = (N, N_{\mathsf{acc}}, Z, M, \mathsf{Lab}, \mathit{level}, \mathit{label}, E)$ where (i) N is a countable set of nodes, (ii) $N_{\mathsf{acc}} \subseteq N$ is a set of accepting nodes, (iii) Lab is a set of labels, (iv) $\mathit{level} : N \to \mathbb{N}$ is a function that gives a level to each node, (v) $\mathit{label} : N \to \mathsf{Lab}$ labels each node with some element of Lab, (vi) Z is a countable set of clocks, and (vii) $E \subseteq N \times \mathcal{G}_M(Z) \times \Sigma \times 2^{(Z)} \times N$ is an infinite set of edges (where $2^{(Z)}$ denotes the finite subsets of Z) such that $(n, g, a, Y, n') \in E$ implies $\mathit{level}(n') = \mathit{level}(n) + 1$, for every node n' with $\mathit{level}(n') > 0$ there is a single node n for which there can exist edges with pattern $e = (n, \_, \_, \_, n')$, and for every node n, there are finitely many edges with pattern $(n, \_, \_, \_, \_)$ (the tree is finitely branching).

The semantics of a labelled infinite timed tree is a timed transition system, defined as an easy extension of that for timed automata.

Proofs and complements for Subsection 3.1

Let $\mathcal{A} = (L, \ell_0, L_{\mathsf{acc}}, X, M, E)$ be a timed automaton. We define the infinite timed tree $\mathcal{A}^\infty = (N^\infty, N^\infty_{\mathsf{acc}}, Z, M, L \times Z^X, \mathit{level}, \mathit{label}, E^\infty)$ as follows:

- its set of clocks is $Z = \{z_i \mid i \in \mathbb{N}\}$;
- there is one node $n_{\ell_0} \in N^\infty$ with $\mathit{level}(n_{\ell_0}) = 0$ and $\mathit{label}(n_{\ell_0}) = (\ell_0, \sigma_0)$ where $\sigma_0(x) = z_0$ for each $x \in X$;
- assume that we have constructed all nodes of level i. For each such node n (with $\mathit{label}(n) = (\ell, \sigma)$) and for each transition $\ell \xrightarrow{g,a,Y} \ell'$ in $\mathcal{A}$, we add a node n' such that $\mathit{label}(n') = (\ell', \sigma')$ with $\sigma'(x) = \sigma(x)$ if $x \in X \setminus Y$ and $\sigma'(x) = z_{i+1}$ if $x \in Y$, $\mathit{level}(n') = i+1$ and we add a transition $n \xrightarrow{g_{[x \leftarrow \sigma(x)]}, a, \{z_{i+1}\}} n'$ to E^∞;
- a node $n \in N^\infty$ is accepting (in N^∞_{acc}) if $\mathit{label}(n) = (\ell, \sigma)$ with $\ell \in L_{\mathsf{acc}}$.

Note that for each node n of the tree $\mathcal{A}^\infty$, if $\mathit{level}(n) = i$ and $\mathit{label}(n) = (\ell, \sigma)$, then $\sigma(X) \subseteq Z_i = \{z_0, \ldots, z_i\}$. This can easily be proved by induction on the level i. Based on this observation, we assume in the sequel that such a σ denotes a function in Z_i^X (instead of Z^X). Moreover, this observation leads to another one, on valuations. A valuation of this "infinite" timed automaton is *a priori* a function from the countable set Z to $\mathbb{T}$, but at level i, only values of clocks in Z_i really make sense. Hence in the sequel, when writing a configuration (n, v), we will assume that v is a valuation on Z_i if $\mathit{level}(n) = i$, rather than on the countable set of clocks Z.

Lemma 3. *The relation $\mathfrak{R}_1$ between states of $\mathcal{A}$ and states of $\mathcal{A}^\infty$ defined by $(\ell, v \circ \sigma)\ \mathfrak{R}_1\ (n, v)$ if $label(n) = (\ell, \sigma)$ is a strong timed bisimulation.*

Proof. We prove that ${\mathfrak{R}_1}^{-1}$ is a strong timed simulation of $\mathcal{A}^\infty$ by $\mathcal{A}$. In the rest of the proof, we assume that $(\ell, v \circ \sigma)\ \mathfrak{R}_1\ (n, v)$ where $label(n) = (\ell, \sigma)$ and $level(n) = i$. We first focus on continuous transitions, and we assume that: $(n, v) \xrightarrow{t} (n, v+t)$ for some $t \in \mathbb{R}_+$. The continuous transition $(\ell, v \circ \sigma) \xrightarrow{t} (\ell, (v \circ \sigma)+t)$ also makes sense in $\mathcal{A}$ and we clearly have that $(\ell, (v \circ \sigma)+t)\ \mathfrak{R}_1\ (n, v+t)$. We now focus on discrete transitions and we assume that $(n, v) \xrightarrow{a} (n', v')$ in $\mathcal{A}^\infty$ with $label(n') = (\ell', \sigma')$. There exists an edge $(n, \widetilde{g}, a, \{z_{i+1}\}, n')$ in $\mathcal{A}^\infty$, with $v \models \widetilde{g}$. This edge corresponds to some edge (ℓ, g, a, Y, ℓ') in $\mathcal{A}$, with $g_{[x \leftarrow \sigma(x)]} = \widetilde{g}$. We clearly have that v satisfies the guard $\widetilde{g}$ if and only if $v \circ \sigma$ satifies g. In particular, there is a transition $(\ell, v \circ \sigma) \xrightarrow{a} (\ell', \widetilde{v})$ which makes sense in $\mathcal{A}$. It remains to prove that $\widetilde{v} = v' \circ \sigma'$. If $x \notin Y$, $\widetilde{v}(x) = v(\sigma(x))$ and $\sigma(x) = \sigma'(x)$, thus $\widetilde{v}(x) = v'(\sigma'(x))$ (because $\sigma(x) \neq z_{i+1}$). If $x \in Y$, then $\sigma'(x) = z_{i+1}$ and thus $\widetilde{v}(x) = 0 = v'(\sigma'(x)) = v'(z_{i+1})$. We thus obtain the desired result in both cases (time elapsing as well as discrete transitions).

The proof in the other direction follows exactly the same lines. □

In particular, as the relation $\mathfrak{R}_1$ preserves initial and accepting states, we immediately obtain the following corollary.

Corollary A *For every level-i node n of $\mathcal{A}^\infty$, for every valuation $v \in \mathbb{T}^{Z_i}$ $\mathcal{L}(\mathcal{A}^\infty, (n, v)) = \mathcal{L}(\mathcal{A}, (\ell, v \circ \sigma))$.*

Proofs and complements for Subsection 3.2

The classical region abstraction extends to the labelled infinite timed tree $\mathcal{A}^\infty$ using the following lemma.

Lemma B *The relation $\equiv_M$ defined on states of $\mathcal{A}^\infty$ by*

$$(n, v) \equiv_M (n, v') \text{ if } level(n) = i \text{ and } v \equiv_{Z_i, M} v'$$

is a time-abstract bisimulation.

Proof. Assume that there exists $t \in \mathbb{R}^+$ such that: $(n, v) \xrightarrow{t} (n, v + t)$ and $(n, v) \equiv_M (n, v')$. Since $v \equiv_{Z_i, M} v'$, and $\equiv_{Z_i, M}$ is a time-abstract bisimulation, there exists $t' \in \mathbb{R}^+$ such that $(v + t) \equiv_{Z_i, M} (v' + t')$. This implies that $(n, v + t) \equiv_M (n, v' + t')$ with $(n, v') \xrightarrow{t'} (n, v' + t')$.

Assume now that there exists $a \in \Sigma$ such that $(n, v) \xrightarrow{a} (\widetilde{n}, \widetilde{v})$ and $(n, v) \equiv_M (n, v')$ with $level(n) = i$ and $label(n) = (\ell, \sigma)$. In particular there exists an edge e in $\mathcal{A}^\infty$ of the form $(n, g, a, \{z_{i+1}\}, \widetilde{n})$ such that the valuation v satisfies the guard g. Since $v \equiv_{Z_i, M} v'$, and g only constrains clocks in Z_i, valuation v' also satisfies the guard g. Hence the edge e can also be taken from (n, v'),

and there exists a valuation $\widetilde{v}'$ with $(n, v') \xrightarrow{a} (\widetilde{n}, \widetilde{v}')$. It remains to prove that $(\widetilde{n}, \widetilde{v}) \equiv_M (\widetilde{n}, \widetilde{v}')$. By definition of $\mathcal{A}^\infty$, $level(\widetilde{n}) = i+1$, so we need to show that $\widetilde{v} \equiv_{Z_{i+1},M} \widetilde{v}'$. To see this, observe that v and $\widetilde{v}$ (respectively v' and $\widetilde{v}'$) agree on the values for the clocks in Z_i (since only the clock z_{i+1} has been reset on e). Moreover, $\widetilde{v}(z_{i+1}) = \widetilde{v}'(z_{i+1}) = 0$, since clock z_{i+1} was reset along edge e. Hence $\widetilde{v} \equiv_{Z_{i+1},M} \widetilde{v}'$ and thus $(\widetilde{n}, \widetilde{v}) \equiv_M (\widetilde{n}, \widetilde{v}')$. □

We construct a region abstraction for $\mathcal{A}^\infty$ that we call $R(\mathcal{A}^\infty)$. At level i the regions are those associated with the set of clocks Z_i. The region abstraction is intuitively the quotient of $\mathcal{A}^\infty$ by the region equivalence $\equiv_M$ exhibited in Lemma B. Precisely, $R(\mathcal{A}^\infty)$ is an infinite labelled timed tree defined as follows:

- $(n_{\ell_0}, z_0 = 0)$ is the initial node. As n_{ℓ_0} it has level 0 and label (ℓ_0, σ_0);
- assume all nodes of level i have been constructed. For each node (n, r) of level i and for each transition $n \xrightarrow{g,a,Y} n'$ in $\mathcal{A}^\infty$, for every basic constraint $r'' \in \mathsf{Reg}_M^{Z_i}$ such that r'' is a time-successor of r and $r'' \models g$, we add a level-$(i+1)$ node (n', r') and an edge $(n, r) \xrightarrow{r'',a,Y} (n', r')$, where $r' \in \mathsf{Reg}_M^{Z_{i+1}}$ is the region obtained from r'' by adding the contraint $z_{i+1} = 0$.
  If $label(n') = (\ell', \sigma')$, the label of the new node (n', r') is set to $(\ell', \widetilde{\sigma}')$ where $\widetilde{\sigma}'(x) = \bot$ if $r' \cap (\sigma'(x) \leq M) = \emptyset$, and $\widetilde{\sigma}'(x) = \sigma'(x)$ otherwise.[6]
- a node (n, r) is accepting if n is accepting in $\mathcal{A}^\infty$.

Get rid of expired clocks

Remark C *In the timed transition system associated with $R(\mathcal{A}^\infty)$, all configurations that are visited are of the form $((n, r), v)$ with $v \in r$. Furthermore, if $label(n, r) = (\ell, \sigma)$, then $\sigma(x) \neq \bot$ implies $r \subseteq (\sigma(x) \leq M)$.*

Lemma 5. *The relation $\mathfrak{R}_2$ defined between states of $\mathcal{A}^\infty$ and states of $R(\mathcal{A}^\infty)$ by $(n, v)\ \mathfrak{R}_2\ ((n, r), v)$ if $v \in r$ is a strong timed bisimulation.*

Proof. We first prove that $\mathfrak{R}_2$ is a timed strong similuation of $\mathcal{A}^\infty$ by $R(\mathcal{A}^\infty)$. Let n be a level-i node in $\mathcal{A}^\infty$, and v be a valuation over Z_i, and assume that $v \in r$ (hence $(n, v)\ \mathfrak{R}_2\ ((n, r), v)$). Assume that $(n, v) \xrightarrow{t,a} (n', v')$ in $\mathcal{A}^\infty$. This transition comes from some edge $n \xrightarrow{g,a,Y} n'$ in $\mathcal{A}^\infty$, with $v + t \models g$ and v' is a valuation over Z_{i+1} such that $v'(z) = v(z) + t$ for all $z \in Z_i$ and $v'(z_{i+1}) = 0$. By construction of $R(\mathcal{A}^\infty)$, there exists an edge $(n, r) \xrightarrow{r'',a,Y} (n', r')$ with $r = [v]$ $r'' = [v + t]$, and $r' = [v']$. Hence, there is a transition $((n, r), v) \xrightarrow{t,a} ((n', r'), v')$ in $R(\mathcal{A}^\infty)$. Since trivially $(n', v')\ \mathfrak{R}_2\ ((n', v'), r')$, we get the desired result.

We now prove that $\mathfrak{R}_2$ is a strong timed simulation of $R(\mathcal{A}^\infty)$ by $\mathcal{A}^\infty$. Let (n, r) be a level-i node in $R(\mathcal{A}^\infty)$ and v a valuation over Z_i such that $v \in r$. Assume $((n, r), v) \xrightarrow{t,a} ((n', r'), v')$. There must exist an edge $(n, r) \xrightarrow{r'',a,Y} (n', r')$ in $R(\mathcal{A}^\infty)$ with $r = [v]$, $r'' = [v + t]$ and $r' = [v']$. Then, by definition of

[6] Informally, we record with the label how clocks of the original automaton should be mapped onto the clocks of the tree, but in case its value is $\bot$ (meaning larger than the maximal constant), then we set it to $\bot$ because the precise reference to a clock of the tree is not relevant anymore.

$R(\mathcal{A}^\infty)$, there is an edge $n \xrightarrow{g,a,Y} n'$ with $r'' \models g$. Since $v+t \models r''$, this edge can be taken from $(n, v+t)$: then $(n,v) \xrightarrow{t,a} (n', \widetilde{v}')$ is a possible move in $\mathcal{A}^\infty$ for some valuation $\widetilde{v}'$. Obviously $\widetilde{v}' = v'$ and $(n', v')\ \mathfrak{R}_2\ ((n', r'), v')$, hence the result. □

As the construction of $R(\mathcal{A}^\infty)$ preserves initial and accepting states, we get the following corollary.

Corollary D *For every level-i node n of $\mathcal{A}^\infty$ and any valuation $v \in \mathbb{T}^{Z_i}$, if $r = [v]$, then $\mathcal{L}(R(\mathcal{A}^\infty), ((n,r),v)) = \mathcal{L}(\mathcal{A}^\infty, (n,v))$.*

This property of $R(\mathcal{A}^\infty)$ will ensure that a symbolic determinization will coincide with a real determinization of the underlying timed system.

Proofs and complements for Subsection 3.3

The infinite timed tree $R(\mathcal{A}^\infty)$, as $\mathcal{A}^\infty$, has some *input-determinacy* property [5]:

Lemma E *Let u be a finite timed word that can be read from $(n_0, \bar{0})$ in $\mathcal{A}^\infty$. There is a unique sequence $\tau_u \in (\mathsf{Reg}_M \times \Sigma \times 2^Z)^{|u|}$ (where $\mathsf{Reg}_M = \bigcup_{i\geq 0} \mathsf{Reg}_M^{Z_i}$), a unique region $r_u \in \mathsf{Reg}_M^{Z_{|u|}}$, and a unique valuation $v_u \in r_u$ such that for every initial run ϱ in $R(\mathcal{A}^\infty)$ that reads u, the underlying sequence of edges for ϱ is τ_u, and ϱ ends in some state $((n', r_u), v_u)$ with level$(n') = |u|$.*

Proof. The proof is by induction on the length of the timed word u. The base case $|u| = 0$ is trivial: there is a unique initial run on the empty word in $R(\mathcal{A}^\infty)$.

Assume that $u = w \cdot (a,t)$, that t' is the largest time-stamp appearing in w, and that the lemma holds for w. There exists a unique sequence $\tau_w \in (\mathcal{G}(Z) \times \Sigma \times 2^Z)^{|w|}$, a unique region r_w over $Z_{|w|}$ and a unique valuation v_w in r_w satisfying the conditions of the lemma. Let g be the region over $Z_{|w|}$ corresponding to valuation $v_w + t - t'$. By definition of $R(A^\infty)$, for all initial runs, the last transition fired while reading u is labelled $(g, a, \{z_{|u|}\})$. The underlying sequence of edges for all initial runs on u is thus τ_w concatenated with $(g, a, \{z_{|u|}\})$. The arrival region $r_u \in \mathsf{Reg}_M^{Z_{|u|}}$ is also uniform, and corresponds to the region g together with the constraint $z_{|u|} = 0$. Moreover the nodes have level $|u|$, since they are children of level $|w|$ $(= |u| - 1)$ nodes. This completes the induction step, and concludes the lemma. □

Lemma F *Assume n (resp. n') is a level-i (resp. level-$(i+1)$) node in $\mathcal{A}^\infty$ with label$(n) = (\ell, \sigma)$ (resp. label$(n') = (\ell', \sigma')$), and that $(n,r) \xrightarrow{g,a,\{z_{i+1}\}} (n', r')$ is a transition of $R(\mathcal{A}^\infty)$. Take any set $S \supseteq \sigma(X)$. For every region $\widetilde{r}$ over Z_i such that $\widetilde{r}_{|S} = r_{|S}$ and $(n, \widetilde{r})$ is a node of $R(\mathcal{A}^\infty)$, for every $\widetilde{g}$ that is a time-successor of $\widetilde{r}$ and such that $\widetilde{g}_{|S} = g_{|S}$, there is a transition $(n, \widetilde{r}) \xrightarrow{\widetilde{g},a,\{z_{i+1}\}} (n', \widetilde{r}')$ for some $\widetilde{r}'$ so that $\widetilde{r}'_{|S'} = r'_{|S'}$ where $S' = S \cup \{z_{i+1}\}$ (note that in particular, $S' \supseteq ((S \setminus \sigma(X)) \cup \sigma'(X))$).*

Proof. First notice that the edge $(n,r) \xrightarrow{g,a,\{z_{i+1}\}} (n',r')$ in $R(\mathcal{A}^\infty)$ comes from an edge $n \xrightarrow{g',a,\{z_{i+1}\}} n'$ in $\mathcal{A}^\infty$ (where g is a basic region-based constraint entailing g'), which itself comes from an edge $\ell \xrightarrow{g'',a,Y} \ell'$ in $\mathcal{A}$ with $g''_{[x\leftarrow\sigma(x)]} = g'$.

Let $\widetilde{v} \in \widetilde{r}$ and $\widetilde{t} \in \mathbb{R}_+$ such that $\widetilde{v} + \widetilde{t} \in \widetilde{g}$. As $\widetilde{r}_{|S} = r_{|S}$ and $\widetilde{g}_{|S} = g_{|S}$, there is $v \in r$ and $t \in \mathbb{R}_+$ such that $v + t \in g$ and $(v+t)_{|S} = (\widetilde{v} + \widetilde{t})_{|S}$. This yields a move $((n,r),v) \xrightarrow{t,a} ((n',r'),v')$ with $label(n',r') = (\ell',\sigma')$. We have that $((n,r),v)$ and $(\ell, v \circ \sigma)$ are strongly timed bisimilar (by composing the two strong timed bisimulations $\mathfrak{R}_2$ and $\mathfrak{R}_1$, see Lemma 5 and Lemma 3). Thus, in $\mathcal{A}$, there is a move $(\ell, v\circ\sigma) \xrightarrow{t,a} (\ell', v' \circ \sigma')$, and as $\sigma(X) \subseteq S$, we have that $(\widetilde{v}\circ\sigma) + \widetilde{t} = (v\circ\sigma) + t$ and thus that there is a move $(\ell, \widetilde{v}\circ\sigma) \xrightarrow{\widetilde{t},a} (\ell', v'\circ\sigma')$. Hence due to Lemma 5 and Lemma 3, there is a move $((n,\widetilde{r}),\widetilde{v}) \xrightarrow{\widetilde{t},a} ((n',\widetilde{r}'),\widetilde{v}')$ for some $\widetilde{r}'$ and some $\widetilde{v}' \in \widetilde{r}'$. Hence there is a transition $(n,\widetilde{r}) \xrightarrow{\widetilde{g},a,\{z_{i+1}\}} (n',\widetilde{r}')$ in $R(\mathcal{A}^\infty)$. We have that $\widetilde{v}'(z) = \widetilde{v}(z) + \widetilde{t} = v(z) + t = v'(z)$ if $z \in S$, and that $\widetilde{v}'(z_{i+1}) = 0 = v'(z_{i+1})$. Thus, we get that $\widetilde{r}'_{S'} = r'_{S'}$, which concludes the proof. □

Starting with $R(\mathcal{A}^\infty)$ we build an infinite timed tree $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, which is deterministic and accepts the same timed language as $R(\mathcal{A}^\infty)$. It is constructed as follows.

- its initial node is $N_0 = (\star_0, \{n_0\}, r_0)$ where $\star_0$ is a marker, r_0 is the initial region defined by $(z_0 = 0)$, and $label(N_0) = \{(\ell_0, \sigma_0)\}$ with $\sigma_0(x) = z_0$ for every $x \in X$;
- assume all level-i nodes have been built and are of the form $N = (\star, K, r)$ where:
  - $\star$ is a marker that identifies that node[7]
  - K is a finite subset of level-i nodes of $R(\mathcal{A}^\infty)$,
  - $label(N) = \{(\ell,\sigma) \mid \exists n \in K \text{ s.t. } label(n) = (\ell,\sigma)\}$,
  - setting $Act(K) = \bigcup_{(\ell,\sigma)\in label(N)} \sigma(X)$, r is a region over set of clocks $Act(K)$. $Act(K)$ is a subset of Z_i called the set of *active clocks* in K.[8] We write $Act(N)$ for $Act(K)$ whenever $N = (\star, K, r)$.

  We pick a level-i node $N = (\star, K, r)$. For every (g,a) where g is a region over $Act(K)$, we create a new node $N' = (\star', K', r')$ (where $\star'$ is a new marker not used elsewhere) whenever

$$K' = \{n' \mid \exists (n,\widetilde{r}) \xrightarrow{\widetilde{g},a,\{z_{i+1}\}} (n',\widetilde{r}') \text{ transition of } R(\mathcal{A}^\infty) \text{ s.t. } \widetilde{r}_{|Act(K)} = r, \widetilde{g}_{|Act(K)} = g\}$$

  and $\widetilde{r}'_{|Act(K')} = r'$. In the above definition, we have the following:

[7] Markers ensure that we construct a tree and not a DAG.
[8] Note that if $z_i \in Act(K)$, then $r \subseteq (z_i \leq M)$ (*i.e.* z_i is bounded in r).

$\widetilde{r}$: region over set of clocks Z_i
$\widetilde{r}'$: region over set of clocks Z_{i+1}
$\widetilde{r}_{|Act(K)}$: region over set of clocks $Act(K) \subseteq Z_i$
$\widetilde{r}'_{Act(K')}$: region over set of clocks $Act(K') \subseteq Z_{i+1}$

We add a transition $N \xrightarrow{g,a,\{z_{i+1}\}} N'$ to the tree. Node N' has level $i+1$ and label $\{label(n') \mid n' \in K'\}$;
– a level-i node N is accepting if it contains an accepting node.

Remark G *It is easy to see that a configuration (N, v) with $N = (\star, K, r)$ can be reached in the timed transition system associated with* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *iff $v \in r$. We call such a configuration a* safe *configuration.*

Also, if $N = (\star, K, r)$ is a node of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$*, and if $(\ell, \sigma) \in label(N)$, then for every clock $x \in X$, $\sigma(x) \neq \perp$ implies $r \subseteq (\sigma(x) \leq M)$.*

We will see in Proposition 7 that this construction ensures that $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is deterministic and it accepts the same finite timed words as $R(\mathcal{A}^\infty)$.

Proposition 7. $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is a deterministic timed tree, and for every node $N = (\star, K, r)$ and for every valuation $v \in \mathbb{T}^{Act(K)}$ with $v \in r$,*

$$\mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N, v)) = \bigcup_{n \in K} \mathcal{L}(R(\mathcal{A}^\infty), ((n, r), v))$$

Proof. By construction, because of the markers $\star$, this is an infinite timed tree. Assume that it is not deterministic. It means that there is a node $N = (\star, K, r)$ and two transitions $N \xrightarrow{g_1,a,\{z_i\}} N_1$ and $N \xrightarrow{g_2,a,\{z_i\}} N_2$ such that $g_1 \cap g_2 \neq \emptyset$ and $N_1 \neq N_2$. Both g_1 and g_2 are regions over $Act(K)$, hence $g_1 = g_2$. This is not possible, because we have constructed a single node for every pair (g, a) where g is a region over $Act(K)$. Hence $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is deterministic.

We turn to the second property, and start proving it from the initial configuration (N_0, v_0) of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, where $N_0 = (\star_0, K_0, r_0)$ with $r_0 = (z_0 = 0)$ and $v_0(z_0) = 0$. Let u be a finite timed word that can be read from (N_0, v_0) in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$.

– As $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is deterministic, when reading u from (N_0, v_0) we reach a single configuration (N_u, v_u) with $N_u = (\star_u, K_u, r_u)$ and $v_u \in r_u$. Note that v_u is a valuation over set of clocks $Act(K_u) \subseteq Z_{|u|}$.
– The timed word u may also be read in $R(\mathcal{A}^\infty)$ from initial configuration $((n_0, r_0), v_0)$. Applying Lemma E, we have that there is a single valuation $\widetilde{v}_u$ and a single region $\widetilde{r}_u$ (with $\widetilde{v}_u \in \widetilde{r}_u$) such that any path in $R(\mathcal{A}^\infty)$ starting from $((n_0, r_0), v_0)$ and reading u leads to some configuration $((\widetilde{n}_u, \widetilde{r}_u), \widetilde{v}_u)$. We write $\widetilde{K}_u$ for the set of such nodes $\widetilde{n}_u$. Note that $\widetilde{v}_u$ is a valuation over set of clocks $Z_{|u|}$.

We will prove the following property, by induction on the length of u:

$$K_u = \widetilde{K}_u \qquad \text{and} \qquad v_u = \widetilde{v}_{u|Act(K_u)} \tag{1}$$

This property obviously holds for the empty word. We now assume that this property holds for a timed word w, and we assume that $u = w(a,t)$. We furthermore note t' the last time stamp appearing in w. In $R(\mathcal{A}^\infty)$, there is at least one path of the form:

$$((n_0, r_0), v_0) \overset{w}{\rightsquigarrow} ((n, \widetilde{r}_w), \widetilde{v}_w) \xrightarrow{t-t',a} ((n', \widetilde{r}_u), \widetilde{v}_u).$$

Recall from Lemma E that the valuation $\widetilde{v}_w$ (resp. $\widetilde{v}_u$) is independant of the path of $R(\mathcal{A}^\infty)$ reading w (resp. u). Valuation $\widetilde{v}_u$ can be obtained from $\widetilde{v}_w$ by: for every $z \in Z_{|u|}$,

$$\widetilde{v}_u(z) = \begin{cases} 0 & \text{if } z = z_{|u|} \\ \widetilde{v}_w(z) + t - t' & \text{if } z \neq z_{|u|} \end{cases}$$

In $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, we have that

$$(N_0, v_0) \overset{w}{\rightsquigarrow} (N_w, v_w) \xrightarrow{t-t',a} (N_u, v_u)$$

and v_u is a valuation over $Act(K_u)$. The last move above comes from a transition $N_w \xrightarrow{r'',a,\{z_{|u|}\}} N_u$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. Furthermore, we have that $v_w + t - t' \models r''$ and $v_u = \big([z_{|u|} \leftarrow 0](v_w + t - t')\big)_{|Act(K_u)}$.

Take $n_u \in K_u$. By construction of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, there exist $n_w \in K_w$ and a transition $(n_w, r) \xrightarrow{\overline{r}'',a,\{z_{|u|}\}} (n_u, r')$ in $R(\mathcal{A}^\infty)$ such that $r_{|Act(K_w)} = r_w$, $\overline{r}''_{|Act(K_w)} = r''$, and $r'_{|Act(K_u)} = r_u$. Let $z \in Act(K_u)$:

$$v_u(z) = \begin{cases} 0 & \text{if } z = z_{|u|} \\ v_w(z) + t - t' & \text{if } z \in Act(K_u) \setminus \{z_{|u|}\} \end{cases}$$

As $Act(K_u) \setminus \{z_{|u|}\} \subseteq Act(K_w)$ and as $v_w(z) = \widetilde{v}_w(z)$ for every $z \in Act(K_w)$ (by induction hypothesis), we get that

$$\begin{aligned} v_u(z) &= \begin{cases} 0 & \text{if } z = z_{|u|} \\ \widetilde{v}_w(z) + t - t' & \text{if } z \in Act(K_u) \setminus \{z_{|u|}\} \end{cases} \\ &= \widetilde{v}_u(z) \end{aligned}$$

Hence, $\widetilde{v}_{u|Act(K_u)} = v_u$ and also $\widetilde{r}_{u|Act(K_u)} = r_u$.

Now, by induction hypothesis, $n_w \in K_w = \widetilde{K}_w$, hence in $R(\mathcal{A}^\infty)$, there is an execution reading w of the form

$$((n_0, r_0), v_0) \overset{w}{\rightsquigarrow} ((n_w, \widetilde{r}_w), \widetilde{v}_w)$$

and $\widetilde{v}_{w|Act(K_w)} = v_w$, and $\widetilde{r}_{w|Act(K_w)} = r_w$. In particular, $r_{|Act(K_w)} = \widetilde{r}_{w|Act(K_w)}$. Applying Lemma F, if $\underline{r}'' = [\widetilde{v}_w + t - t']$, we get that there is a transition

$(n_w, \widetilde{r}_w) \xrightarrow{\underline{r}'', a, \{z_{|u|}\}} (n_u, \underline{r}_u)$ such that $\underline{r}''_{|Act(K_w)} = \overline{r}''_{|Act(K_w)} = r''$ and $\underline{r}_{u|Act(K_u)} = r'_{|Act(K_u)} = r_u$. The transition $((n_w, \widetilde{r}_w), \widetilde{v}_w) \xrightarrow{t-t', a} ((n_u, \underline{r}_u), \widetilde{v}_u)$ is thus possible, and $n_u \in \widetilde{K}_u$.

Conversely assume that $n_u \in \widetilde{K}_u$. There is an execution

$$((n_0, r_0), v_0) \overset{w}{\rightsquigarrow} ((n_w, \widetilde{r}_w), \widetilde{v}_w) \xrightarrow{t-t', a} ((n_u, \widetilde{r}_u), \widetilde{v}_u)$$

for some $n_w \in \widetilde{K}_w = K_w$ (by i.h.), and the last move comes from a transition of the form $(n_w, \widetilde{r}_w) \xrightarrow{r'', a, \{z_{|u|}\}} (n_u, \widetilde{r}_u)$ in $R(\mathcal{A}^\infty)$, where $\widetilde{v}_w + t - t' \models r''$ and $\widetilde{v}_u = [z_{|u|} \leftarrow 0](\widetilde{v}_w + t - t')$. This transition is a witness for the construction of some N from N_w with a transition $N_w \xrightarrow{g, a, \{z_{|u|}\}} N$ with $g = r''_{|Act(K_w)}$ because $\widetilde{r}_{w|Act(K_w)} = r_w$ and $\widetilde{r}_{|Act(K_u)} = r_u$. It remains to see that this transition can be concretized into $(N_w, v_w) \xrightarrow{t-t', a} (N, v)$ for some v, because $v_w + t - t' = (\widetilde{v}_w + t - t')_{|Act(K_w)} \models g$. Hence there is a run

$$(N_0, v_0) \overset{w}{\rightsquigarrow} (N_w, v_w) \xrightarrow{t-t', a} (N, v)$$

that can be rewritten as

$$(N_0, v_0) \overset{u}{\rightsquigarrow} (N, v)$$

which yields $(N_u, v_u) = (N, v)$. Hence, $n_u \in K_u$, which concludes the proof.

In particular, we conclude that

$$\mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N_0, v_0)) = \mathcal{L}(R(\mathcal{A}^\infty), ((n_0, r_0), v_0)).$$

Now fix a configuration (N, v) of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, and say it is reachable reading the timed words w, whose duration is d. Note in particular that $v = v_w$ and that $N = (\star, K_w, r_w)$. Then:

$$\begin{aligned}
\mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N, v)) &= \{u \mid w \cdot (u + d) \in \mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N_0, v_0))\} \\
&= \{u \mid w \cdot (u + d) \in \mathcal{L}(R(\mathcal{A}^\infty), ((n_0, r_0), v_0))\} \\
&= \bigcup_{n \in K_w} \mathcal{L}(R(\mathcal{A}^\infty), ((n, r_w), \widetilde{v}_w)) \\
&= \bigcup_{n \in K_w} \mathcal{L}(\mathcal{A}^\infty, (n, \widetilde{v}_w)) \\
&= \bigcup_{n \in K_w \mid label(n) = (\ell, \sigma)} \mathcal{L}(\mathcal{A}, (\ell, \widetilde{v}_w \circ \sigma)) \\
&= \bigcup_{(\ell, \sigma) \in label(N)} \mathcal{L}(\mathcal{A}, (\ell, \widetilde{v}_w \circ \sigma)) \\
&= \bigcup_{(\ell, \sigma) \in label(N)} \mathcal{L}(\mathcal{A}, (\ell, v \circ \sigma)) \\
&\qquad \text{(because } \sigma(X) \subseteq Act(K_w)\text{)}
\end{aligned}$$

This concludes the proof of the proposition. □

Corollary H *The relation $\mathfrak{R}_3$ between safe configurations of* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *defined by*

$$(N_1, v_1)\ \mathfrak{R}_3\ (N_2, v_2) \text{ if } v_1 = v_2 \text{ and } label(N_1) = label(N_2)$$

is a strong timed bisimulation.

Proof. The proof of Proposition 7 ensures that (N_1, v_1) and (N_2, v_2) are language-equivalent in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. Since $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is deterministic, they are thus also strongly timed bisimilar. □

The previous corollary can be generalized. Let $N_1 = (\star_1, K_1, r_1)$ and $N_2 = (\star_2, K_2, r_2)$ be two nodes of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. We say that they are *equivalent w.r.t. a bijection* $\iota : Act(N_1) \to Act(N_2)$ whenever $label(N_1) = \{(\ell_j, \sigma_j) \mid j \in J\}$ implies $label(N_2) = \{(\ell_j, \iota \circ \sigma_j) \mid j \in J\}$, and $r_{1\,[z \leftarrow \iota(z)]} = r_2$.

Lemma I *The relation $\mathfrak{R}_4$ between safe configurations of* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *defined by*

$$(N_1, v_1)\ \mathfrak{R}_4\ (N_2, v_2) \text{ if there is a bijection } \iota \text{ s.t. } N_1 \text{ and } N_2 \text{ are equivalent w.r.t. } \iota \text{ and } v_1 = v_2 \circ \iota$$

is a strong timed bisimulation.

Proof. Assume that $(N_1, v_1) \xrightarrow{t,a} (N_1', v_1')$ and $(N_1, v_1)\ \mathfrak{R}_4\ (N_2, v_2)$. Thus there exists an edge $e_1 = (N_1, g_1, a, \{y_1\}, N_1')$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. Since $(N_1, v_1)\ \mathfrak{R}_4$ (N_2, v_2), valuations v_1 and v_2 satisfy $v_{1\,[z \leftarrow \iota(z)]} = v_2$, and hence $(v_1 + t)_{[z \leftarrow \iota(z)]} = (v_2 + t)$. Thus, the regions of $(v_1 + t)_{[z \leftarrow \iota(z)]}$ and $(v_2 + t)$ coincide. Moreover $[v_1 + t]$ corresponds to the guard g_1 in the edge e_1. Since N_1 and N_2 share the same set of locations, by construction of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, there is a transition $e_2 = (N_2, g_2, a, \{y_2\}, N_2')$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ such that $g_2 = g_{1\,[z \leftarrow \iota(z)]} = [v_2 + t]$. Clearly $v_1 \models g_1$ if and only if $v_2 \models g_2$. Thus the transition e_2 is firable from $(N_2, v_2 + t)$. Let (N_2', v_2') be the configuration reached from $(N_2, v_2 + t)$ after e_2 has been fired, *i.e.* we have that $(N_2, v_2) \xrightarrow{t,a} (N_2', v_2')$. It remains to prove that $(N_1', v_1')\ \mathfrak{R}_4\ (N_2', v_2')$. Bijection ι can be extended in a unique way into ι' by letting $\iota'(y_1) = y_2$ and $\iota'(y) = \iota(y)$ if $y \neq y_1$. In particular, we have that $v_2'(\iota'(y_1)) = v_2'(y_2) = v_1'(y_1) = 0$. Let us now consider $y \neq y_2$, we have that

$$\begin{aligned} v_2'(y) &= v_2(y) && \text{since } y \neq y_2 \\ &= v_1(\iota^{-1}(y)) && \text{since } v_1 = v_2 \circ \iota \\ &= v_1'(\iota^{-1}(y)) && \text{since } \iota^{-1}(y) \neq y_1 \end{aligned}$$

Thus, we have that $v_1' = v_2' \circ \iota$. A similar reasoning can be made for the labels of N_1' and N_2', and we get that N_1' and N_2' are equivalent w.r.t. ι'. This concludes the proof. □

Remark J *In case $\mathcal{A}$ has a single clock x, a level-i node of* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *carries the following information: a finite set of pairs of the form $(\ell, x \mapsto z_j)$ for some $j \leq i$ and a region for clocks in Z_i. Hence we can see such a level-i node as a pair (P, r) where $P \subseteq L \times Z_i$ and $r \in \mathsf{Reg}_M^{Z_i}$, and we will say that $(i, P, r) \preceq (i', P', r')$ whenever $i \leq i'$, and there is an injection $\kappa\colon Z_i \mapsto Z_{i'}$ such that $\left(r'_{|\kappa(Z_i)}\right) = r_{[z \leftarrow \kappa(z)]}$ and $\{(\ell, \kappa(z)) \mid (\ell, z) \in P\} \subseteq P'$. This view of a node actually coincides with the word encoding and the subword relation proposed in [7], and we can prove that $\preceq$ is indeed a well-quasi-order. We thus recover the decidability of the universality problem in single-clock timed automata (over finite timed words) [7].*

Proofs and complements for Subsection 3.4

Let $\gamma \in \mathbb{N}$. We say that $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is *γ-clock-bounded* if for every node N of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, $|Act(N)| \leq \gamma$. Under this assumption we will be able to bound the number of clocks in the infinite tree. We let $X_\gamma = \{x_1, \ldots, x_\gamma\}$, and we define inductively for every node N of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ an injective function $\Gamma_\gamma[N] : Act(N) \to X_\gamma$:

- For the initial node N_0, we let $\Gamma_\gamma[N_0](z_0) = x_1$.
- Let $N = (\star, K, r)$ be a level-i node, and assume that we have defined properly $\Gamma_\gamma[N]$. Consider a transition $N \xrightarrow{g,a,Y} N'$ in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. We define $\Gamma_\gamma[N'] : Act(N') \to X_\gamma$ as follows:

$$\Gamma_\gamma[N'](z_j) = \begin{cases} \Gamma_\gamma[N](z_j) & \text{if } z_j \in Act(N) \cap Act(N') \\ x_k & \text{if } z_j \in Act(N') \setminus Act(N), \\ & \text{where } k = \min\{h \mid x_h \notin \Gamma_\gamma[N](Act(N) \cap Act(N'))\} \end{cases}$$

  There is at most one additional clock in $Act(N')$ compared with $Act(N)$ (this is z_{i+1}), and if all clocks were used in N (*i.e.*, $\Gamma_\gamma[N](Act(N)) = X_\gamma$) and z_{i+1} becomes active, it means that a clock in $Act(N)$ has been desactivated.

Let $N = (\star, K, r)$ be a node of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. We replace node N by a node $\Gamma_\gamma(N) = (\star, K, r_{[z \leftarrow \Gamma_\gamma[N](z)]})$ and set its label to $\{(\ell, \Gamma_\gamma[N] \circ \sigma) \mid (\ell, \sigma) \in label(N)\}$. Just remark that in the above set, $\Gamma_\gamma[N] \circ \sigma$ is an application from X to X_γ. We define in a natural way the infinite timed tree $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ from $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ by replacing any transition $N \xrightarrow{g,a,Y} N'$ by a transition $\Gamma_\gamma(N) \xrightarrow{g_{[z \leftarrow \Gamma_\gamma[N](z)]}, a, \Gamma_\gamma[N'](Y)} \Gamma_\gamma(N')$.

Lemma K *If* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is γ-clock-bounded, the relation $\mathfrak{R}_5$ between states of* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *and states of* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ *defined by*

$$(N, v \circ \Gamma_\gamma[N]) \; \mathfrak{R}_5 \; (\Gamma_\gamma(N), v)$$

is a strong timed bisimulation.

Proof. Let us first notice that, by construction of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$, we have that the edge $N \xrightarrow{g,a,Y} N'$ exists in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ if and only if the edge $\Gamma_\gamma(N) \xrightarrow{g_{[z\leftarrow\Gamma_\gamma[N](z)]},a,\Gamma_\gamma[N'](Y)} \Gamma_\gamma(N')$ exists in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. Moreover, we have that the valuation v satisfies the guard $g_{[z\leftarrow\Gamma_\gamma[N](z)]}$ if and only if the valuation $v \circ \Gamma_\gamma[N]$ satisfies the guard g. From the previous discussion, one can conclude that the transition $(N, v \circ \Gamma_\gamma[N]) \xrightarrow{t,a} (N', \widetilde{v}')$ makes sense in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ if and only if the transition $(\Gamma_\gamma(N), v) \xrightarrow{t,a} (\Gamma_\gamma(N'), v')$ makes sense in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. Let us now evaluate the reached valuations in both cases in order to conclude the proof by showing that $\widetilde{v}' = v' \circ \Gamma_\gamma[N']$. For all $x \in X_\gamma$, we have that:

$$v'(x) = \begin{cases} v(x) & \text{if } \ x \notin \Gamma_\gamma[N'](Y) \\ 0 & \text{if } \ x \in \Gamma_\gamma[N'](Y). \end{cases}$$

For all $z \in Act(N')$, we have that:

$$\widetilde{v}'(z) = \begin{cases} v \circ \Gamma_\gamma[N](z) & \text{if } \ z \notin Y \\ 0 & \text{if } \ z \in Y. \end{cases}$$

If $z \in Act(N) \cap Act(N')$, we clearly have that $z \notin Y$ and $\Gamma_\gamma[N'](z) \notin \Gamma_\gamma[N'](Y)$; we can thus conclude that:

$$\begin{aligned} \widetilde{v}'(z) &= v \circ \Gamma_\gamma[N](z) && \text{since } z \notin Y \\ &= v \circ \Gamma_\gamma[N'](z) && \text{since } z \in Act(N) \cap Act(N') \\ &= v' \circ \Gamma_\gamma[N'](z) && \text{since } \Gamma_\gamma[N'](z) \notin \Gamma_\gamma[N'](Y). \end{aligned}$$

If $z \in Act(N') \setminus Act(N)$, then $z \in Y$ and $\Gamma_\gamma[N'](z) = x_k$, where x_k is the unique clock in $\Gamma_\gamma[N'](Y)$, thus

$$\widetilde{v}'(z) = 0 = v'(x_k) = v' \circ \Gamma_\gamma[N'](z).$$

Hence we have proved that $\widetilde{v}' = v' \circ \Gamma_\gamma[N']$, which concludes the proof. □

Lemma L *$\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ is a deterministic timed labelled tree.*

Proof. Since the functions $\Gamma_\gamma(N)$ are injective, for all $z, z' \in Act(N)$ we have that if $z \neq z'$ then $\Gamma_\gamma[N](z) \neq \Gamma_\gamma[N](z')$. In particular, if two guards g and g' of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ are disjoint, the corresponding guards $g_{[z\leftarrow\Gamma_\gamma[N](z)]}$ and $g'_{[z\leftarrow\Gamma_\gamma[N](z)]}$ are also disjoint in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. Since $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is a deterministic timed labelled tree (see Proposition 7), we can thus conclude that it is also the case of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. □

Corollary M *For every configuration (N, v) of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$,*

$$\mathcal{L}(\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty))), (\Gamma_\gamma(N), v)) = \mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)), (N, v \circ \Gamma_\gamma[N]))$$

Lemma N *Assume that* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is* γ*-clock-bounded. The relation* $\mathfrak{R}_6$ *between safe* [9] *configurations of* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ *defined by*

$$(\Gamma_\gamma(N_1), v_1)\ \mathfrak{R}_6\ (\Gamma_\gamma(N_2), v_2)\ \textit{if}\ v_1 = v_2\ \textit{and}\ \textit{label}(\Gamma_\gamma(N_1)) = \textit{label}(\Gamma_\gamma(N_2))$$

is a strong timed bisimulation.

Proof. First, let us notice that Lemma K ensures that $(\Gamma_\gamma(N_i), v_i)$ and $(N_i, v_i \circ \Gamma_\gamma[N_i])$ are strongly timed-bisimilar (for $i = 1, 2$).

We have that

$$\begin{aligned}\Gamma_\gamma[N_i](Act(N_i)) &= \Gamma_\gamma[N_i]\Big(\bigcup_{(\ell,\sigma)\in label(N_i)} \sigma(X)\Big)\\ &= \bigcup_{(\ell,\sigma)\in label(N_i)} \Gamma_\gamma[N_i](\sigma(X))\\ &= \bigcup_{(\ell,\sigma')\in label(\Gamma_\gamma(N_i))} \sigma'(X)\end{aligned}$$

by definition of $label(\Gamma_\gamma(N_i))$. Since $label(\Gamma_\gamma(N_1)) = label(\Gamma_\gamma(N_2))$, we get that $\Gamma_\gamma[N_1](Act(N_1)) = \Gamma_\gamma[N_2](Act(N_2))$. This allows to define a bijection $\iota : Act(N_1) \to Act(N_2)$ defined by $\iota = \Gamma_\gamma[N_2]^{-1} \circ \Gamma_\gamma[N_1]$ (this is possible as $\Gamma_\gamma[N_i]$ is injective). In particular, we have that if $label(N_1) = \{(\ell_j, \sigma_j) \mid j \in J\}$ then $label(N_2) = \{(\ell_j, \iota \circ \sigma_j) \mid j \in J\}$, and $(r_1)_{[z \leftarrow \iota(z)]} = r_2$. This means that the two nodes N_1 and N_2 are equivalent w.r.t. ι. Hence, since $v_1 \circ \Gamma_\gamma[N_1] = v_2 \circ \Gamma_\gamma[N_2] \circ \iota$, we can conclude that $(N_1, v_1 \circ \Gamma_\gamma[N_1])$ and $(N_2, v_2 \circ \Gamma_\gamma[N_2])$ are strongly bisimilar by using Lemma I.

To conclude the proof, we combine the bisimulations, using the previously defined notations, we have proved that:

$$(\Gamma_\gamma(N_1), v_1)\ \mathfrak{R}_5\ (N_1, v_1 \circ \Gamma_\gamma[N_1])\ \mathfrak{R}_4\ (N_2, v_2 \circ \Gamma_\gamma[N_2])\ \mathfrak{R}_5\ (\Gamma_\gamma(N_2), v_2)$$

Thus the relation $\mathfrak{R}_6$ is a strong timed bisimulation. □

Remark O *Furthermore, since* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ *is deterministic, the subtrees rooted at two* $\mathfrak{R}_6$*-equivalent nodes are isomorphic.*

Lemma P *Assume that* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is* γ*-clock-bounded. The relation* $\mathfrak{R}_7$ *between safe configurations of* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ *defined by*

$$(\Gamma_\gamma(N_1), v_1)\ \mathfrak{R}_7\ (\Gamma_\gamma(N_2), v_2)\ \textit{if}\ v_1 \equiv_{M,X_\gamma} v_2\ \textit{and}\ \textit{label}(\Gamma_\gamma(N_1)) = \textit{label}(\Gamma_\gamma(N_2))$$

is a time-abstract bisimulation.

[9] A safe configuration of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ is a configuration $(\Gamma_\gamma(N), v)$ with $v \in r$, when $\Gamma_\gamma(N) = (N, r)$.

Proof. Assume that we have $(\Gamma_\gamma(N_1), v_1)$, $(\Gamma_\gamma(N_2), v_2)$ and $(\Gamma_\gamma(N_1'), v_1')$ such that $(\Gamma_\gamma(N_1), v_1)\ \mathfrak{R}_7\ (\Gamma_\gamma(N_2), v_2)$ and $(\Gamma_\gamma(N_1), v_1) \xrightarrow{t_1,a} (\Gamma_\gamma(N_1'), v_1')$ for some $t_1 \in \mathbb{R}_+$. Let us notice that we have that $(\Gamma_\gamma(N_1), v_1)\ \mathfrak{R}_6\ (\Gamma_\gamma(N_2), v_1)$, thus by Lemma N, the following transition makes senses $(\Gamma_\gamma(N_2), v_1) \xrightarrow{t_1,a} (\Gamma_\gamma(N_2'), v_1')$. Since $v_1 \equiv_{M,X_\gamma} v_2$ and $\equiv_{M,X_\gamma}$ is a time-abstract bisimulation, we can find t_2 such that $[v_1 + t_1] = [v_2 + t_2]$; in particular $v_1 + t_1$ and $v_2 + t_2$ satisfy the same guards. This allows us to consider the transition $(\Gamma_\gamma(N_2), v_2) \xrightarrow{t_2,a} (\Gamma_\gamma(N_2'), v_2')$, where $v_1' \equiv_{M,X_\gamma} v_2'$; this concludes the proof. □

Given a safe configuration $(\Gamma_\gamma(N), v)$, we write $[(\Gamma_\gamma(N), v)]_{\mathfrak{R}_7}$ for its equivalence class w.r.t. the equivalence relation $\mathfrak{R}_7$. Furthermore, if $\Gamma_\gamma(N) = (\star, K, r)$ then r is a region over X_γ, and $v \in r$. Hence, we can speak of the equivalence class of $\Gamma_\gamma(N)$ instead, and write $[\Gamma_\gamma(N)]_{\mathfrak{R}_7}$. Thus, an equivalence class is characterized by its label (a finite set of pairs (ℓ, σ) with ℓ a location of $\mathcal{A}$ and $\sigma\colon X \to X_\gamma$), and a region over clocks in X_γ. There are thus finitely many such classes.

If $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is γ-clock-bounded, we define the timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$ as follows. Its set of clocks is X_γ. Its set of locations is the (finite set) $\{[\Gamma_\gamma(N)]_{\mathfrak{R}_7} \mid N \text{ node of } \mathsf{SymbDet}(R(\mathcal{A}^\infty))\}$, its initial location is $[\Gamma_\gamma(N_0)]_{\mathfrak{R}_7}$ where N_0 is the level-0 node of $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$, and a location is accepting whenever N is accepting in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ (note that this is independent of the choice of the representative as the acceptance condition in $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ only depends on the labels). There is a transition $C_1 \xrightarrow{g,a,Y} C_2$ whenever there exists some transition $\Gamma_\gamma(N_1) \xrightarrow{g,a,Y} \Gamma_\gamma(N_2)$ in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ where $\Gamma_\gamma(N_1) \in C_1$ and $\Gamma_\gamma(N_2) \in C_2$. By Lemma P this is independent of the choice of the representative.

Lemma Q *Assume that* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is* γ*-clock-bounded. The relation* $\mathfrak{R}_8$ *defined by*

$$(\Gamma_\gamma(N), v)\ \mathfrak{R}_8\ ([\Gamma_\gamma(N)]_{\mathfrak{R}_7}, v),$$

where $(\Gamma_\gamma(N), v)$ *is a safe configuration of* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$*, is a strong timed bisimulation between* $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ *and* $\mathcal{B}_{\mathcal{A},\gamma}$*.*

Proof. The proof is in the same vein as that of Lemma 5. We first prove that $\mathfrak{R}_8$ is a strong timed simulation of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ by $\mathcal{B}_{\mathcal{A},\gamma}$. Let $\Gamma_\gamma(N) = (\star, K, r)$ be a node in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$, and v be a valuation over X_γ, and assume that $v \in r$ (hence $(\Gamma_\gamma(N), v)\ \mathfrak{R}_8\ ([\Gamma_\gamma(N)]_{\mathfrak{R}_7}, v)$). Assume that $(\Gamma_\gamma(N), v) \xrightarrow{t,a} (\Gamma_\gamma(N'), v')$ in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. This transition comes from some edge $\Gamma_\gamma(N) \xrightarrow{g,a,Y} \Gamma_\gamma(N')$ in $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$, with $v + t \models g$ and v' is a valuation over X_γ such that $v'(x) = v(x)+t$ for all $x \notin Y$ and $v'(x) = 0$ otherwise. By construction, there exists an edge $[\Gamma_\gamma(N)]_{\mathfrak{R}_7} \xrightarrow{g,a,Y} [\Gamma_\gamma(N')]_{\mathfrak{R}_7}$ in $\mathcal{B}_{\mathcal{A},\gamma}$. Hence, there is a transition $([\Gamma_\gamma(N)]_{\mathfrak{R}_7}, v) \xrightarrow{t,a} ([\Gamma_\gamma(N')]_{\mathfrak{R}_7}, v')$ in $\mathcal{B}_{\mathcal{A},\gamma}$. Since trivially $(\Gamma_\gamma(N'), v')\ \mathfrak{R}_8\ ([\Gamma_\gamma(N')]_{\mathfrak{R}_7}, v')$, we get the desired result.

We now prove that $\mathfrak{R}_8$ is a strong timed simulation of $\mathcal{B}_{\mathcal{A},\gamma}$ by $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. Let $[\Gamma_\gamma(N)]_{\mathfrak{R}_7}$ be a node in $\mathcal{B}_{\mathcal{A},\gamma}$, where $\Gamma_\gamma(N) = (\star, K, r)$ and v is a valuation over X_γ such that $v \in r$. Assume $([\Gamma_\gamma(N)]_{\mathfrak{R}_7}, v) \xrightarrow{t,a} ([\Gamma_\gamma(N')]_{\mathfrak{R}_7}, v')$. There must exist an edge $[\Gamma_\gamma(N)]_{\mathfrak{R}_7} \xrightarrow{g,a,Y} [\Gamma_\gamma(N')]_{\mathfrak{R}_7}$ in $\mathcal{B}_{\mathcal{A},\gamma}$. Then, by definition of $\mathcal{B}_{\mathcal{A},\gamma}$, $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ contains an edge $\Gamma_\gamma(N) \xrightarrow{g,a,Y} \Gamma_\gamma(N'')$ such that $\mathit{label}(\Gamma_\gamma(N')) = \mathit{label}(\Gamma_\gamma(N''))$.Since $v + t \models g$, this edge can be taken from $(\Gamma_\gamma(N), v + t)$: then $(\Gamma_\gamma(N), v) \xrightarrow{t,a} (\Gamma_\gamma(N''), v')$ is a transition of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. Moreover, we clearly have that $(\Gamma_\gamma(N'), v')\ \mathfrak{R}_7\ (\Gamma_\gamma(N''), v')$. This allows us to conclude that $(\Gamma_\gamma(N'), v')\ \mathfrak{R}_8\ ([\Gamma_\gamma(N')]_{\mathfrak{R}_7}, v')$, hence the result. □

Proposition 10. *Assume that* $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ *is* γ*-clock-bounded. Then,* $\mathcal{B}_{\mathcal{A},\gamma}$ *is a deterministic timed automaton, and* $\mathcal{L}(\mathcal{B}_{\mathcal{A},\gamma}) = \mathcal{L}(\mathcal{A})$.

Proof. The determinism of $\mathcal{B}_{\mathcal{A},\gamma}$ is guaranteed by Lemma L and Lemma N. Combining Corollary A, Corollary D and Proposition 7, one can easily conclude that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty))$. Under the γ-clock-bounded hypothesis, Corollary M ensures that $\mathcal{L}(\mathsf{SymbDet}(R(\mathcal{A}^\infty)) = \mathcal{L}(\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$, we thus have that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$. The final equality: $\mathcal{L}(\mathcal{B}_{\mathcal{A},\gamma}) = \mathcal{L}(\mathcal{A})$ is obtained by means of Lemma Q. □

Remark R *Timed automaton* $\mathcal{B}_{\mathcal{A},\gamma}$ *has diagonal guards, due to the use of regions as guards, but there is actually no need of these diagonal constraints (because the order of fractional parts of clocks is given by the region we are in before firing the transition), and we could replace any region constraining a transition by the smallest (non-diagonal) guard including the region. However this would have required extra explanation, and we thought the current construction was technical enough.*

Complements for Subsection 3.5

If $\mathcal{A} = (L, \ell_0, L_{\mathsf{acc}}, X, M, E)$ is a timed automaton such that $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is γ-clock-bounded (for some $\gamma \in \mathbb{N}$), then the timed automaton $\mathcal{B}_{\mathcal{A},\gamma}$ has at most

$$\alpha(\mathcal{A}, \gamma) = 2^{|L|} \cdot \gamma^{|X|} \cdot \Big((2M+2)^{(\gamma+1)^2} \cdot \gamma!\Big)$$

locations (because, a location is characterized by a finite set of pairs (ℓ, σ) with ℓ a location of $\mathcal{A}$ and $\sigma\colon X \to X_\gamma$, and a region over clocks in X_γ), and at most

$$\alpha(\mathcal{A}, \gamma)^2 \cdot \Big((2M+2)^{(\gamma+1)^2} \cdot \gamma!\Big) \cdot |\Sigma|$$

transitions. In some cases, we can improve the way clocks in X are mapped into clocks in X_γ (see construction of $\Gamma_\gamma(\mathsf{SymbDet}(R(\mathcal{A}^\infty)))$ page x), and get better upper bounds for the size of $\mathcal{A}$.

The procedure we have described to construct (under some conditions) an equivalent deterministic timed automaton is not really effective because it goes through the construction of infinite objects. However, if we abstract away the complete construction, we know precisely how locations and transitions are derived. Indeed, given $\mathcal{A} = (L, \ell_0, L_{\mathsf{acc}}, X, M, E)$, a location of $\mathcal{B}_{\mathcal{A},\gamma}$ is a pair $(\{(\ell_j, \sigma_j) \mid j \in J\}, r)$ where J is finite, $\ell_j \in L$, $\sigma_j \colon X \mapsto X_\gamma$ and $r \in \mathsf{Reg}_M^{X_\gamma}$.

– There is a transition

$$(\{(\ell_j, \sigma_j) \mid j \in J\}, r) \xrightarrow{g,a,\{y\}} (\{(\ell'_j, \sigma'_j) \mid j \in J'\}, r')$$

whenever:

- for all $j \in J$, there is $j' \in J'$ and a transition $(\ell_j, r_j) \xrightarrow{\widetilde{g},a,\widetilde{Y}} (\ell'_j, r'_j)$ in $R(\mathcal{A})$ such that

$$\begin{cases} (r_j)_{[x \leftarrow \sigma_j(x)]} \supseteq r \\ (r'_{j'})_{[x \leftarrow \sigma'_{j'}(x)]} \supseteq r' \\ \widetilde{g}_{[x \leftarrow \sigma_j(x)]} \supseteq g \\ \sigma'_{j'}(x) = \begin{cases} \sigma_j(x) & \text{if } x \notin \widetilde{Y} \\ y & \text{if } x \in \widetilde{Y} \end{cases} \end{cases}$$

- *vice-versa* (for all $j' \in J'$, there is $j \in J \ldots$).

– There is a transition

$$(\{(\ell_j, \sigma_j) \mid j \in J\}, r) \xrightarrow{g,a,\emptyset} (\{(\ell'_j, \sigma'_j) \mid j \in J'\}, r')$$

whenever:

- for all $j \in J$, there is $j' \in J'$ and a transition $(\ell_j, r_j) \xrightarrow{\widetilde{g},a,\emptyset} (\ell'_j, r'_j)$ in $R(\mathcal{A})$ such that

$$\begin{cases} (r_j)_{[x \leftarrow \sigma_j(x)]} \supseteq r \\ (r'_{j'})_{[x \leftarrow \sigma'_{j'}(x)]} \supseteq r' \\ \widetilde{g}_{[x \leftarrow \sigma_j(x)]} \supseteq g \\ \sigma'_{j'} = \sigma_j \end{cases}$$

- *vice-versa* (for all $j' \in J'$, there is $j \in J \ldots$).

Hence, $\mathcal{B}_{\mathcal{A},\gamma}$ can be computed on-the-fly by guessing a new transition, and so can its complement (since $\mathcal{B}_{\mathcal{A},\gamma}$ is deterministic). A location of the automaton $\mathcal{B}_{\mathcal{A},\gamma}$ can be stored in space logarithmic in $\alpha(\mathcal{A}, \gamma)$, and we can thus check for universality in nondeterministic space $\log(\alpha(\mathcal{A}, \gamma))$.

Similarly, we can check whether the language of a timed automaton $\mathcal{C}$ is included in the language of $\mathcal{A}$ by using this deterministic version $\mathcal{B}(\mathcal{A}, \gamma)$ in nondeterministic space $\log(\alpha(\mathcal{A}, \gamma))$.

Proofs and complements for Subsection 4.1

Lemma S *Let $p \in \mathbb{N}$ and $\mathcal{A}$ be a timed automaton in TA_p. Then the infinite timed tree $\mathsf{SymbDet}(R(\mathcal{A}^\infty))$ is p-clock-bounded.*

Proof. By induction on the level i of a node n in $R(\mathcal{A}^\infty)$, we can easily prove that $\mathit{label}(n) = (\ell, \sigma)$ implies $\sigma(X) \subseteq \{\bot\} \cup \{z_{i-p+1}, \ldots, z_i\}$. □

Lemma T *Every strongly non-Zeno timed automaton $\mathcal{A}$ satisfies the p-assumption for some p exponential in the size of $\mathcal{A}$.*

Proof. Let $\mathcal{A}$ be a strongly non-Zeno timed automaton, and consider its region automaton $R(\mathcal{A})$. Take a run ϱ that visits a region r twice. The part of ϱ, say ϱ', between those two visits follows a cycle of $R(\mathcal{A})$. Assume $x \in X$ is a clock which is not reset along ϱ'. If r bounds clock x, it means that the global duration of ϱ' is no more than 1 time unit, and that we can iteratively construct runs of arbitrary length that will follow the cycle of $R(\mathcal{A})$, whose duration will be bounded, hence violating the strongly non-Zeno assumption. Hence region r does not bound clock x, and initially in ϱ', the value of x is $\bot$.

Set now $p-1$ as the longest acyclic sequence of transitions in $R(\mathcal{A})$. Take a run $\varrho = (\ell_0, v_0) \xrightarrow{t_1,a_1} (\ell_1, v_1) \ldots \xrightarrow{t_n,a_n} (\ell_n, v_n)$ in $\mathcal{A}$ with $n \geq p$. If we project ϱ on the regions, we get a path $(\ell_0, r_0) \xrightarrow{a_1} (\ell_1, r_1) \xrightarrow{a_2} \ldots \xrightarrow{a_n} (\ell_n, r_n)$ such that there exist $0 \leq i < j \leq n$ with $(\ell_i, r_i) = (\ell_j, r_j)$, and whenever $x \in X$ is a clock which is not reset along ϱ, it means that $v_i(x) = \bot$ and thus that $v_n(x) = \bot$. We deduce that $\mathcal{A}$ satisfies the p-assumption. The value of p can be bounded by $1 + |L| \cdot (2M+2)^{|X|+1}$. □

Theorem 11. *For every timed automaton $\mathcal{A}$ in SnZTA or in TA_p, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is doubly-exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

Proof. Thanks to Lemmas S and T, any strongly non-Zeno timed automaton is determinizable. Morover, applying previous results, the size of $\mathcal{B}_{\mathcal{A},\gamma}$ is

$$2^{|L| \cdot \gamma^{|X|}} \cdot \Big((2M+2)^{(\gamma+1)^2} \cdot \gamma! \Big)$$

with $\gamma = 1 + |L| \cdot (2M+2)^{|X|+1}$, hence doubly exponential in the size of $\mathcal{A}$. □

Theorem 12. *For every timed automaton $\mathcal{A}$ in ECTA, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

Proof (Sketch). The construction we have presented yields a doubly-exponential-size automaton. However, by slightly twisting the way we rename clocks into clocks in X_γ, we can get a singly-exponential-size automaton. If we write $X_\gamma = \{z_a \mid a \in \Sigma\}$, when taking a transition labelled with a, the clock z_a is always free and can be chosen. Thus, if $X = \{x_a \mid a \in \Sigma\}$ where clock x_a is associated

with event a, we can take all σ's so that $\sigma(x_a) = z_a$ or $\sigma(x_a) = \perp$. Hence, the size of the constructed automaton becomes:

$$2^{|L|} \cdot \left((2M+2)^{(|\Sigma|+1)^2} \cdot |\Sigma|! \right)$$

which is a single exponential in the size of $\mathcal{A}$. □

Theorem 13. *For every timed automaton $\mathcal{A}$ in* IRTA*, we can construct a deterministic timed automaton $\mathcal{B}$, whose size is doubly-exponential in the size of $\mathcal{A}$, and which recognizes the same language as $\mathcal{A}$.*

Proof (Sketch). Along a run in $R(\mathcal{A}^\infty)$, a clock can be put as active at integer time-points. Also, an active clock vanishes after more than M time units (and is replaced in the construction by $\perp$). Hence, at most $M+1$ clocks are required to be active at any time. The result follows. □

Proofs for Subsection 4.2

Proposition 14. *Checking universality in timed automata either satisfying the p-assumption for some p or with integer resets is EXPSPACE-hard.*

Proof. We do the proof by reduction from the halting problem for (deterministic) Turing machines running in exponential space.

Let $\mathcal{M}$ be a deterministic Turing machine which runs in exponential space, and fix an input w_0 of length n. The length of the tape which will be used when running $\mathcal{M}$ on w_0 is bounded by 2^n. We show how to construct a timed automaton $\mathcal{A}_{\mathcal{M},w_0}$ such that $\mathcal{A}_{\mathcal{M},w_0}$ is universal if and only if $\mathcal{M}$ doesn't stop on w_0.

We describe how to encode an execution of the Turing machine by a timed word and $\mathcal{A}_{\mathcal{M},w_0}$ will accept all finite timed words that are not encodings of a halting execution of $\mathcal{M}$ on w_0. We assume that the alphabet of $\mathcal{M}$ is $\{a,b\}$, that the set of states of $\mathcal{M}$ is Q, its initial state is q_0, its halting state is q_{Halt}, and that $\#$ is the blank character. We mark a letter with a dot if the head of the machine points to that letter. A configuration of the Turing machine can thus be viewed as a pair (q,w) where $q \in Q$, and $w \in \{a,b,\#,\dot{a},\dot{b},\dot{\#}\}^{2^n}$ (with a single dotted letter, which indicates the position of the head) is the content of the tape.

An execution of $\mathcal{M}$ over w_0 is thus a sequence of consecutive configurations $(q_0,w_0) \to (q_1,w_1) \to \cdots \to (q_p,w_p)$ (we assume each w_i has length exactly 2^n by possibly adding some $\#$'s). Such an execution is encoded by the timed word

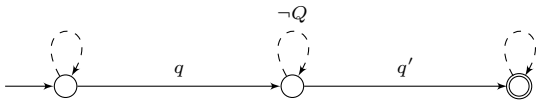
$$(q_0,0)(w_0[1],1)\dots(w_0[2^n],2^n)(q_1,2^n+1)(w_1[1],2^n+2)\dots(w_p[2^n],(2^n+1)p).$$

We will construct a timed automaton which accepts all (finite) timed words which are not encodings of an accepting execution of $\mathcal{M}$ on w_0. We will explain at the end of the proof how this timed automaton can be transformed into a

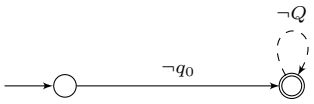
timed automaton that satisfies the p-assumption for some p or into a timed automaton with integer resets. In the next pictures, we will write Q to denote any action in q, or $\neg Q$ for any action outside Q, or any similar convention. We write no action name on a transition if it can be any. Finally, $x \neq 1$ is a shorthand for two transitions, one constrained by $x < 1$ and the other constrained by $x > 1$. For the moment, the reader should forget about dashed edges.

A (finite) timed word u might not be the encoding of an halting computation in $\mathcal{M}$ for several reasons:

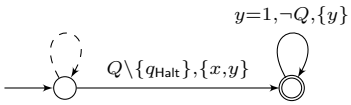
1. either there is a factor of the form $q(\neg Q)^* q'$ in the untiming of u whereas there is no rule saying that from q we can reach q':



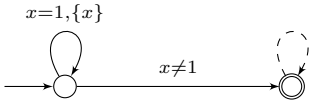
2. or the timed word does not start with the initial state of $\mathcal{M}$:



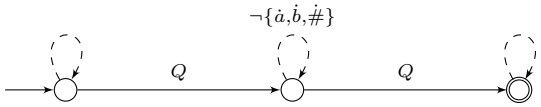
3. or the last state in u is not the halting state of $\mathcal{M}$:



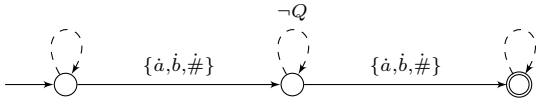
4. or there is a letter that happens at a non-integral time-point, or there is no letter at some integral time-point (except when the word stops):



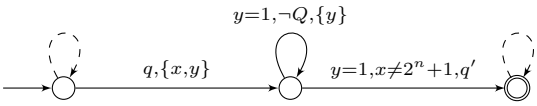
5. or there is no dotted letter between two consecutive control states in u:



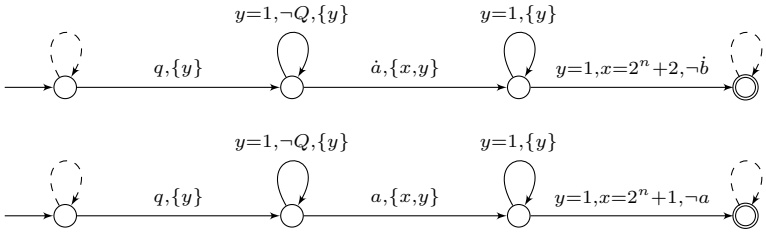
6. or there are two dotted letters or more between two control states:



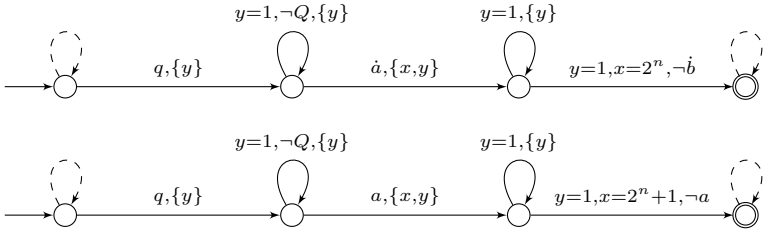
7. or there are not $2^n + 1$ time units between two consecutive control states in u:



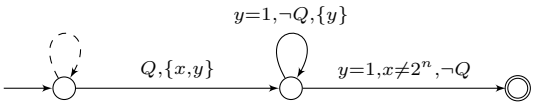
8. or a rule of the form $(q, a, b, \mathsf{right}, q')$ is not faithfully mimicked: either the dotted letter is not transferred properly, or the rest of the configuration is not copied properly, yielding the two following automata.



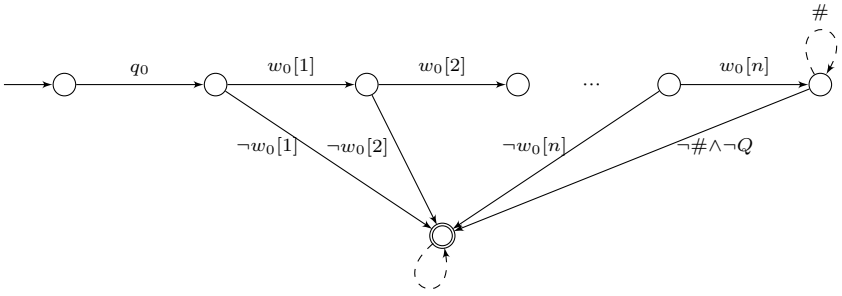
9. or a rule of the form $(q, a, b, \mathsf{left}, q')$ is not faithfully mimicked: either the dotted letter is not transferred properly, or the rest of the configuration is not copied properly, yielding the two following automata.



10. or the last configuration has not the correct length:



11. or the initial configuration does not encode properly the initial content of the tape, w_0:



It is tedious but easy to verify that the timed automaton we have just constructed, called $\mathcal{A}_{\mathcal{M},w_0}$, is universal iff there is no halting computation in $\mathcal{M}$ on the timed word w_0.

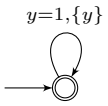
So far, the timed automaton does not verify the p-assumption, because of the loops depicted in dashed lines. However it is worth noting that all these loops could reset the two clocks of this timed automaton, without affecting the timed language which is accepted. This new automaton satisfies the p-assumption for $p = 2^n + 3$.

We can also notice that thanks to module 4, we can assume that all edges of the other modules reset clock y and test whether $y = 1$, without changing the timed language which is accepted. This gives a new timed automaton which has integral resets.

This concludes the proof of the EXPSPACE lower bound. □

Proposition U *The inclusion problem for strongly non-Zeno timed automata is EXPSPACE-hard.*

Proof. We do the very same proof as the previous one, except that we add the constraint $y = 1$ and the reset of y on every transition. We consider the timed automaton $\mathcal{C}$:



One is easily convinced that $\mathcal{L}(\mathcal{C}) \subseteq \mathcal{L}(\mathcal{A}_{\mathcal{M},w_0})$ iff $\mathcal{M}$ does not halt on w_0. □