Removing ε-Transitions in Timed Automata

Volker Diekert[1], Paul Gastin[2] and Antoine Petit[3]

[1] Inst. für Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart
[2] LITP, Université Paris 7, 2, place Jussieu, F-75251 Paris Cedex 05
[3] LSV, ENS de Cachan, 61, av. du Prés. Wilson, F-94235 Cachan Cedex

Abstract. Timed automata are among the most widely studied models for real-time systems. Silent transitions, i.e., ε-transitions, have already been proposed in the original paper on timed automata by Alur and Dill [2]. In [7] it is shown that ε-transitions can be removed, if they do not reset clocks; moreover ε-transitions strictly increase the power of timed automata, if there is a self-loop containing ε-transitions which reset some clocks. The authors of [7] left open the problem about the power of the ε-transitions which reset clocks, if they do not lie on any cycle.
The present paper settles this open question. Precisely, we prove that a timed automaton such that no ε-transition with nonempty reset set lies on any directed cycle can be effectively transformed into a timed automaton without ε-transitions. Interestingly, this main result holds under the assumption of non-Zenoness and it is false otherwise.
Besides, we develop a promising new technique based on a notion of *precise time* which allows to show that some timed languages are not recognizable by any ε-free timed automaton.

1 Introduction

A number of "real-life systems" demand time requirements which cannot easily be treated with the classical models based on transition systems. Therefore, new timed models have been introduced for the specification and verification of systems with quantitative properties. A natural way to define such a model is to consider some usual untimed model and to add a suitable notion of time.
We focus in this paper on the basic and natural model of so-called timed automata, proposed by Alur and Dill [2,3]. Since its introduction, this model has been intensively studied under several aspects: determinization [4], minimization [1], power of clocks [9], extensions of the model [5,6] and logical characterization [11] have been considered in particular. Moreover, this model has been used for verification and specification of real-time systems successfully, [8,10,12].
In the original paper [2] silent or internal actions (ε-transitions) of timed automata have been considered, but, somewhat surprisingly, they disappeared in most of the following papers on timed automata (even in the extended version [3]), until the recent work of [7]. It is shown there that ε-transitions strictly increase the power of timed automata, only if these ε-transitions are allowed to reset clocks (called ε-reset transitions in the following). The emptiness problem

of the class of timed automata with ε-transitions is still decidable, and its language class is more robust (e.g. closed under projection) than the class where ε-transitions are forbidden. Thus, the natural question to characterize the "useful" ε-transitions arises.
In [7], it is left as an open question to find when ε-reset transitions can be removed. The present paper settles this problem. Precisely, we prove that a timed automaton with ε-transitions can be effectively transformed into a timed automaton without ε-transitions, if no ε-reset transition lies on any directed cycle of the automaton. Moreover and surprisingly, this result holds only under the assumption of non-Zenoness, otherwise it becomes false. Our main construction is quite involved and leads to some huge state explosion. However, this is just a serious argument in favor of ε-transitions. We may use them in order to have a compact and concise specification for languages recognized by automata without ε-reset transitions, although no quantitative assertion about this statement can be given at this moment.
The example of [7], showing that ε-reset transitions increase the power, was based on the idea to have a self-loop of some ε-reset transition and the proof considered a path which uses several consecutive ε-reset transitions. We exhibit here a very simple timed automaton with some cycle containing an ε-reset transition and in which no path uses two consecutive ε-transitions and whose language cannot be accepted without ε-transition. To this purpose, we develop a new technique based on a notion of *precise time*, which appears to be very promising in its own right. This new notion yields a formal tool in order to show that some timed languages are not recognizable by any timed automaton (without ε-reset transitions).

2 Preliminaries

A *timed automaton* (over $\mathbb{R}$) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F, R, X, C)$, where

Q is a finite set of states,
Σ is a finite alphabet and ε denotes the empty word of Σ^*,
δ is the transition relation explained below,
$Q_0 \subseteq Q$ is a subset of initial states,
$F \subseteq Q$ is a subset of final states,
$R \subseteq Q$ is a subset of repeated states,
X is a finite set of clocks, and
$C \subseteq \mathbb{R}$ is a finite set of constants.

A *constraint* is a propositional formula using the logical connectives $\{\vee, \wedge, \neg\}$ over atomic formulae of the form $x \# c$ or $x - y \# c$, for $x, y \in X$, $c \in C$, and $\# \in \{<, =, >\}$.
A transition of δ has the form $p \xrightarrow{A,a,\alpha} q$ where A is a constraint, $a \in \Sigma \cup \{\varepsilon\}$, $\alpha \subseteq X$, and $p, q \in Q$. If $a = \varepsilon$ and $\alpha = \emptyset$, then it is called an ε-transition without reset. If $a = \varepsilon$ and $\alpha \neq \emptyset$, then it is called an ε-reset transition. For the global time and the time values of clocks we shall use non-negative real numbers. For

a clock $x \in X$ and a time $t \in \mathbb{R}_+$, we denote by $x(t) \in \mathbb{R}_+$ the clock value of x. Initially, we are in some state $q_0 \in Q_0$ and the (global) time is $t_0 = 0$ with $x(0) = 0$ for all $x \in X$.

In the course of time all clocks run synchronously. However, executing a transition may reset some clocks to zero. Formally, assuming that the automaton has entered state p at time t with clock values $x(t)$, $x \in X$, then it may execute the transition $p \xrightarrow{A,a,\alpha} q$ at time $t' \geq t$, if the constraint A is satisfied with the clock value $x(t') = x(t) + (t' - t)$ for all clocks x. The execution switches to state q and enters this state at time t' with clock value $x(t') = x(t) + (t' - t)$ for all $x \in X \setminus \alpha$ and resets clock values $x(t') = 0$ for all $x \in \alpha$.

This leads to the notion of a run of the automaton:

$$p_0 \xrightarrow[t_1]{A_1, a_1, \alpha_1} p_1 \xrightarrow[t_2]{A_2, a_2, \alpha_2} p_2 \longrightarrow \cdots$$

The semantics is that $p_{i-1} \xrightarrow{A_i, a_i, \alpha_i} p_i$ has been executed at time t_i with $t_{i-1} \leq t_i$ for all $i \geq 1$ and $t_0 = 0$. In particular, for $i \geq 1$, the constraint A_i has been satisfied for the clock values $x(t_i)$ at time t_i of the execution of the i-th transition (before performing its reset operation).

The finite (infinite resp.) run is *accepted,* if both, $p_0 \in Q_0$ and it ends in a final state (and there are infinitely many states from R on the path resp.). Thus, we accept infinite runs by some Büchi condition. With every finite (infinite resp.) run we can associate in a natural way a finite (infinite resp.) timed ε-word

$$(a_1, t_1)(a_2, t_2) \cdots \in ((\Sigma \cup \{\varepsilon\}) \times \mathbb{R})^\infty.$$

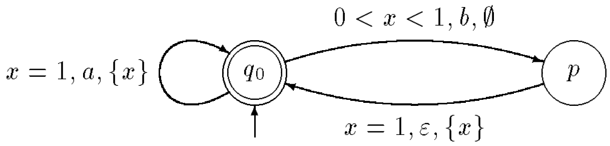
Since an ε-transition is viewed as an invisible action we may cancel all pairs (a_i, t_i) where $a_i = \varepsilon$. In this way we obtain a timed word (which might be finite even if the underlying timed ε-word has been infinite):

$$(a_{i_1}, t_{i_1})(a_{i_2}, t_{i_2}) \cdots \in (\Sigma \times \mathbb{R})^\infty.$$

The timed language $L(\mathcal{A}) \subseteq (\Sigma \times \mathbb{R})^\infty$ accepted by the automaton $\mathcal{A}$ is the set of timed words associated with accepting runs.

A timed (ε-)word $(a_1, t_1)(a_2, t_2) \cdots$ is called a *Zeno word*, if it is infinite but the sequence $t_1, t_2, \ldots$ remains bounded. Let NZ be the set of non Zeno words. For some applications Zeno words are not wanted. It is possible to transform a timed automaton $\mathcal{A}$ into a timed automaton $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A}) \cap NZ$. Note that this transformation is not as easy as it may appear because a finite timed word, which is indeed non Zeno, can be accepted by a run ending in a loop of ε-transitions where the underlying timed ε-word is Zeno.

Example 1. Consider the following automaton $\mathcal{A}_0$ where q_0 is both initial and repeated.



The accepted language can be described as follows: In each open time interval $(i, i+1)$, $i \geq 0$, there occurs at most one b. Moreover, there is an a at time $i+1$ if and only if there is no b in $(i, i+1)$. For the automaton $\mathcal{A}_0$, by construction, no infinite run yields any Zeno word.

In [7] it is shown that all ε-transitions without reset can be removed from the automaton. The technique is to shift the constraint of an ε-transition either to the previous or to the following (visible) transition. A priori, one could expect a similar technique to work in the example above. However we will see below in Corollary 9 that $L(\mathcal{A}_0)$ cannot be accepted by a timed automaton without ε-transition.

3 The main result

The main result of this paper is a construction how to remove all ε-transitions, if there is no directed cycle of the automaton including an ε-transition with reset.

Theorem 2. *Let $\mathcal{A}$ be a timed automaton such that no ε-reset transition lies on any directed cycle. Then we can effectively construct a timed automaton $\mathcal{A}'$ without any ε-transition such that*

$$L(\mathcal{A}) \subseteq L(\mathcal{A}')$$
$$L(\mathcal{A}) \cap NZ = L(\mathcal{A}') \cap NZ$$

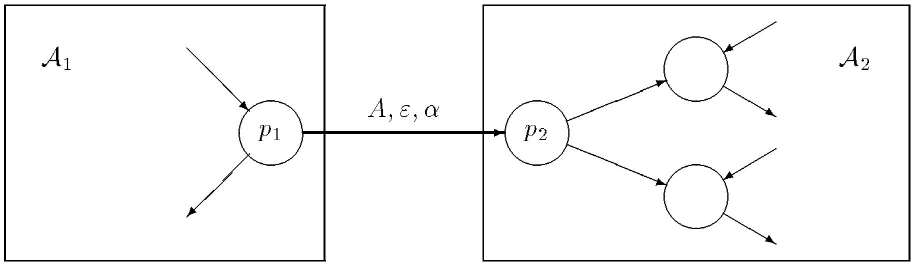
The *ε-depth* of a timed automaton is defined as the maximal number of ε-reset transitions which can be found on some directed path through the automaton. An ε-reset transition is of *maximal depth*, if it is the last one on such a directed path.
The proof of Theorem 2 will be done by induction on the ε-depth of the automaton. Our strategy is as follows. First (Steps 1 to 5), we transform $\mathcal{A}$ into some normal form without increasing the ε-depth. Then we explain how to remove all ε-reset transitions of maximal depth (this is the crucial part). We end up with an automaton $\mathcal{A}'$ (being not in normal form anymore) where the ε-depth is decreased by one. The result follows by induction.

Step 1: Remove all ε-transitions without reset by the procedure of [7]. Note that if the ε-depth of $\mathcal{A}$ is zero then the proof is done.

Step 2: Remove all constraints of the form $x - y \# c$ where x, y are clocks and c is some constant. Note that Step 1 introduces such constraints. By duplicating some transitions, we may also assume that all constraints are conjunction of atomic formulae of type $y \# c$ with $\# \in \{<, =, >\}$.

Note that Steps 1 and 2 do not increase the ε-depth of the automaton. Also, for the proof below, they may be restricted to the part of the automaton which follows ε-reset transitions of maximal depth.

Step 3: Using copies of the automaton, we may now assume that every ε-reset transition of maximal depth $p_1 \xrightarrow{A,\varepsilon,\alpha} p_2$ divides the automaton into two disjoint parts and the only bridge between these parts is this ε-reset transition (See the figure below). Moreover, we may assume that p_2 has no other in-going transition. Note that $\mathcal{A}_2$ contains no ε-transition and we may assume that it contains no initial state either.



Step 4: Reduction of α to a single clock x which is never reset in $\mathcal{A}_2$. Again, this is an easy construction on timed automata which does not change $\mathcal{A}_1$ and hence does not change the ε-depth.

Step 5: Using copies of the clocks used in the constraint A which are reset with their originals inside $\mathcal{A}_1$ and are substituted to their originals in A, we may assume that the clocks used in A are different from x and are not reset in $\mathcal{A}_2$.

Important note: In order to make the following construction more readable we shall assume that the clock constraints of A and inside the subautomaton $\mathcal{A}_2$ are of the form $y < c$ or $y > c$, only. Thus, we do not consider the case $y = c$. The inclusion of such constraints would multiply the case distinction without giving any new insight. In fact, using a constraint $y = c$ would make life even easier, since then we know the exact value of clock y. In the same spirit we consider (and allow) *strictly increasing* time sequences, only.

Step 6: This is the main step of the construction. We will replace in $\mathcal{A}$ the automaton $\mathcal{A}_2$ by a new one $\mathcal{A}_2'$ which will not use the clock x and such that there will be a correspondence between the legal paths of the new automaton $\mathcal{A}'$ and the legal ones of $\mathcal{A}$. Then we will replace the ε-reset transition $p_1 \xrightarrow{A,\varepsilon,\alpha} p_2$ by the ε-transition without reset $p_1 \xrightarrow{A,\varepsilon,\emptyset} (p_2, L_0, U_0)$.

• The states of $\mathcal{A}_2'$ will be triples (p, L, U) where p is a state of $\mathcal{A}_2$ and L, U are its lower and upper attributes. Intuitively, these attributes will keep track of the possible interval for the clock x in the corresponding run of $\mathcal{A}$. More precisely, when we reach a state p of $\mathcal{A}_2$ with a legal run of $\mathcal{A}$, then the corresponding run of $\mathcal{A}'$ will reach the state (p, L, U) with the value $x(t)$ in the open interval $(L(t), U(t))$. Conversely, for each legal run of $\mathcal{A}'$ leading to the state (p, L, U)

and for each value in $(L(t), U(t))$, there exists a corresponding run in $\mathcal{A}$ leading to state p with this value for $x(t)$.
We will use two new clocks x_ℓ and x_u. The second one is reset on each transition which enters state p_1. Since we only consider *strictly increasing* time sequences, we can assume that the constraint A contains $x_u > 0$ and that each transition from p_2 contains $x > 0$ in its constraint.
We can write $A = \bigwedge_r m_r < x_r < m'_r$ with $m_r \in C \cup \{-\infty\}$ and $m'_r \in C \cup \{+\infty\}$. Let $L_0 = \max_r (x_r - m'_r)$ and $U_0 = \min_r (x_r - m_r)$ be the initial attributes given to the state p_2. Apart from the initial values L_0 and U_0, the possible values for L and U are $\{x_\ell + c \mid c \in C\}$ and $\{x_u + c \mid c \in C\}$ respectively. Hence, there are finitely many possible values and the automaton $\mathcal{A}'_2$ remains finite.
Moreover, a state (p, L, U) of $\mathcal{A}'_2$ is final (repeated resp.) if and only if the state p of $\mathcal{A}_2$ is final (repeated resp.).

• For each transition $p \xrightarrow{(a<x<b)\wedge B,\sigma,\beta} q$ of $\mathcal{A}_2$ where B does not contain the clock x and $a \in C \cup \{-\infty\}$ and $b \in C \cup \{+\infty\}$, we add the following transitions to the automaton $\mathcal{A}'_2$.

$$
\begin{aligned}
(p, L, U) &\xrightarrow{(a\le L<U\le b)\wedge B,\sigma,\beta} (p, L, U) \\
&\xrightarrow{(a\le L<b<U)\wedge B,\sigma,\beta\cup\{x_u\}} (p, L, x_u + b) \\
&\xrightarrow{(L<a<U\le b)\wedge B,\sigma,\beta\cup\{x_\ell\}} (p, x_\ell + a, U) \\
&\xrightarrow{(L<a<b<U)\wedge B,\sigma,\beta\cup\{x_\ell,x_u\}} (p, x_\ell + a, x_u + b)
\end{aligned}
\tag{1}
$$

where $L_0 < a$ is an abbreviation for $\bigwedge_r (x_r - m'_r) < a$ and similarly for $a \le L_0$, $b < U_0$ and $U_0 \le b$.

Claim: All timed words (Zeno or non-Zeno) accepted by $\mathcal{A}$ are also accepted by $\mathcal{A}'$. Conversely, all non-Zeno timed words accepted by $\mathcal{A}'$ are also accepted by $\mathcal{A}$.

Note that, thanks to Step 3, we can do Steps 4 to 6 simultaneously on all ε-reset transitions of maximal depth. Hence we have reduced the ε-depth by one and Theorem 2 follows by induction. We will now prove the claim.

Proof. Let $\pi = q_0 \xrightarrow{A_1,\sigma_1,\alpha_1} q_1 \xrightarrow{A_2,\sigma_2,\alpha_2} q_2 \cdots$ be a path of $\mathcal{A}$ and let $t_1 t_2 \cdots$ be a timed sequence such that the timed ε-word $w = (\sigma_1, t_1)(\sigma_2, t_2)\cdots$ is accepted by π. Assume that the i-th transition of π is $p_1 \xrightarrow{A,\varepsilon,\{x\}} p_2$, hence the path from q_0 to $q_{i-1} = p_1$ runs in $\mathcal{A}_1$ while the path from $q_i = p_2$ runs in $\mathcal{A}_2$.
We will construct a path π' of $\mathcal{A}'$ which accepts precisely the same timed ε-word w. The path π' starts as the path π and its i-th transition enters the "initial" state of $\mathcal{A}'_2$:

$$q_0 \xrightarrow{A_1,\sigma_1,\alpha_1} q_1 \cdots \xrightarrow{A_{i-1},\sigma_{i-1},\alpha_{i-1}} q_{i-1} = p_1 \xrightarrow{A,\varepsilon,\emptyset} (p_2, L_0, U_0) = (q_i, L_i, U_i)$$

Since the constraint A has been satisfied at time t_i, we have $L_0(t_i) < 0 < U_0(t_i)$. Hence, we have the time invariant $L_0(t) < x(t) < U_0(t)$ for all $t \ge t_i$.

Assume that the path π' has been constructed up to state $(q_{j-1}, L_{j-1}, U_{j-1})$ and that the time invariant $L_{j-1}(t) < x(t) < U_{j-1}(t)$ for all $t \geq t_{j-1}$ holds. Assume also that the constraint $A_j = (a_j < x < b_j) \wedge B_j$ where B_j does not contain the clock x. Thanks to the time invariant, we can see that among the transitions described in (1), there is exactly one transition $(q_{j-1}, L_{j-1}, U_{j-1}) \xrightarrow{A'_j, \sigma_j, \alpha'_j} (q_j, L_j, U_j)$ whose constraint A'_j is true at time t_j. This transition is used to extend the path π'. One can easily verify that the time invariant $L_j(t) < x(t) < U_j(t)$ for all $t \geq t_j$ holds. We have thus obtain the desired path π' and we have proved the first part of the claim. Note that we do not need to assume that the time sequence $t_1 t_2 \cdots$ diverges for this part of the proof.

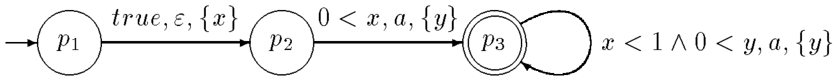
Conversely, let π' be a path of $\mathcal{A}'$ whose i-th transition is $p_1 \xrightarrow{A, \varepsilon, \emptyset} (p_2, L_0, U_0)$. Thus the path π' has the form

$$q_0 \xrightarrow{A_1, \sigma_1, \alpha_1} q_1 \cdots q_{i-1} \xrightarrow{A, \varepsilon, \emptyset} (q_i, L_i, U_i) \xrightarrow{A'_{i+1}, \sigma_{i+1}, \alpha'_{i+1}} (q_{i+1}, L_{i+1}, U_{i+1}) \cdots$$

Let $w = (\sigma_1, t_1)(\sigma_2, t_2) \cdots$ be a timed ε-word accepted by π'. For all $j > i$, let $q_{j-1} \xrightarrow{(a_j < x < b_j) \wedge B_j, \sigma_j, \alpha_j} q_j$ be the transition of $\mathcal{A}_2$ from which the corresponding transition of π' was obtained. We have thus constructed a path π of $\mathcal{A}$. The timed ε-word w is not necessarily accepted by the path π but we will see that we can find a time t'_i such that the corresponding timed ε-word w' is accepted by the path π. Since w and w' differs only by the time of the i-th action which is ε, this will prove the second part of the claim.
Let $I = \bigcap_{j \geq i} (t_j - U_j(t_j), t_j - L_j(t_j))$. By construction of $\mathcal{A}'_2$, since the constraint A'_j was satisfied at time t_j, we have $L_j(t_j) < U_j(t_j)$. Hence the open intervals considered in this intersection are nonempty. Moreover, we have $L_j(t_j) = \max(L_{j-1}(t_j), a_j)$. Hence, $L_j(t_j) \geq L_{j-1}(t_j) = L_{j-1}(t_{j-1}) + t_j - t_{j-1}$. Using the same argument for the upper attribute, we deduce that these intervals are decreasing.
We will see now that $I \neq \emptyset$. This is clear if the path π' is finite. Assume now that π is infinite and that the time sequence $t_1 t_2 \cdots$ diverges. Note that this is the only point where we need to use non-Zenoness.
By construction, $x > 0$ is part of the constraint A_{i+1}. Hence, $a_{i+1} \geq 0$ and $L_{i+1}(t_{i+1}) = \max(L_i(t_{i+1}), a_{i+1}) \geq 0$. Moreover, for all $j > i$ we have $L_j(t_j) \geq L_{i+1}(t_{i+1}) + t_j - t_{i+1} \geq t_j - t_{i+1}$. Let j_0 be such that $t_{j_0} - t_{i+1} > \max C$ where we recall that $C \subseteq \mathbb{R}$ is the set of (finite) constants which are used in the constraints of the automaton $\mathcal{A}$. We deduce that $L_j = L_{j_0}$ for all $j \geq j_0$. Now, for all $j \geq j_0$, we have $\max C < L_j(t_j) < U_j(t_j) = \min(U_{j-1}(t_j), b_j)$ and then $b_j = +\infty$. It follows that $U_j = U_{j_0}$ for all $j \geq j_0$. Finally, we have proved that $I = (t_{j_0} - U_{j_0}(t_{j_0}), t_{j_0} - L_{j_0}(t_{j_0}))$ which is non empty.
To complete the proof we will show that for all $t'_i \in I$, the timed ε-word w' obtained from w by replacing time t_i by t'_i is accepted by the path π.
First, $t'_i \in (t_i - U_i(t_i), t_i - L_i(t_i))$ therefore, $L_0(t'_i) = L_0(t_i) + t'_i - t_i < 0 < U_0(t_i) + t'_i - t_i = U_0(t'_i)$ and the constraint A is satisfied at time t'_i. Note that since by construction $x_u > 0$ is part of the constraint A and $x_u \in \alpha_{i-1}$, we deduce that $t_{i-1} < t'_i$.

Second, let $j > i$, we have $t'_i \in (t_j - U_j(t_j), t_j - L_j(t_j))$ therefore, $a_j \leq L_j(t_j) < t_j - t'_i < U_j(t_j) \leq b_j$ and the constraint $(a_j < x < b_j) \wedge B_j$ is satisfied at time t_j. Note that since by construction $x > 0$ is part of the constraint A_{i+1}, we deduce that $t'_i < t_{i+1}$. This concludes the proof of the claim.

We have seen that the non-Zenoness is needed in the proof above. One might wonder whether this hypothesis is really necessary. We will show that indeed, Theorem 2 becomes false if we allow Zeno-words. Consider the following automaton $\mathcal{A}$.



One can verify that the language accepted by this automaton is

$$L = \{(a, t_2)(a, t_3) \cdots \mid \exists \gamma > 0 \text{ such that } \forall i \geq 2, t_i < t_{i+1} < t_2 + 1 - \gamma\}$$

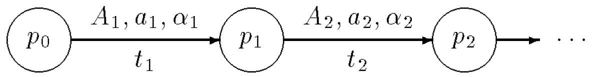
We will show that L is not recognizable by any timed automaton without ε-transitions, even if we allow for the automaton any set of constants which is a discrete subspace of $\mathbb{R}$.
Assume by contradiction that L is recognized by some ε-free automaton $\mathcal{A}'$. Let $C \subseteq \mathbb{R}$ be its set of constants assuming that C is a discrete subspace of $\mathbb{R}$ (e.g., C is finite). Let $t_2 > 0$ be an arbitrary positive real and choose $0 < \delta < 1$ such that $(t_2 + 1 - \delta, t_2 + 1) \cap (C \cup (t_2 + C)) = \emptyset$ and $\delta < \min\{c \in C \mid c > 0\}$.
Consider a timed word $w = (a, t_2)(a, t_3)(a, t_4) \cdots$ with $t_2 + 1 - \delta < t_i < t_{i+1}$ for all $i > 2$ and $\lim_{i \to \infty} t_i = t_2 + 1 - \gamma$ for some $0 < \gamma < \delta$.
Then we have $w \in L(\mathcal{A}')$ and there exists some accepting path π for the timed word w. Due to the choice of δ one can verify that the same path accepts also the timed word $w' = (a, t_2)(a, t_3 + \gamma)(a, t_4 + \gamma) \cdots$. However, since $\lim_{i \to \infty} t_i + \gamma = t_2 + 1$, we have $w' \notin L$ and therefore a contradiction.

4 A notion of precise time

Let $\mathcal{A}$ be a timed automaton which uses a set $C \subseteq \mathbb{Q}$ of *rational* constants. Let $C_{\max}$ be the maximum value of C. Let $\delta > 0$ be such that $C \subseteq \delta\mathbb{Z}$. Consider a finite or infinite path

$$\pi = p_0 \xrightarrow{A_1, a_1, \alpha_1} p_1 \xrightarrow{A_2, a_2, \alpha_2} p_2 \cdots$$

through $\mathcal{A}$ such that all constraints are conjunctions of atomic formulae $x \# c$ with $\# \in \{<, =, >\}$ for some clocks x and constants c. Note that one can always transform a timed automaton in such a way that this is true for all transitions of the automaton.
By $TS(\pi)$ we denote the set of possible time sequences $t_0 t_1 t_2 \cdots$ such that $0 = t_0 \leq t_1 \leq t_2 \leq \cdots$ and



defines a π-run of $\mathcal{A}$. Let us assume that $TS(\pi) \neq \emptyset$. By $TS_n(\pi)$ we denote the set of values $r_n \in \mathbb{R}$ such that there exists $t_0t_1t_2 \cdots t_{n-1}t_n \cdots \in TS(\pi)$ with $r_n = t_n$. Assuming that n is not greater than the length of π, we have $TS_n(\pi) \neq \emptyset$.

Definition 3. A time $t \in \mathbb{R}$ is called a *precise time* of π, if $TS_n(\pi) = \{t\}$ for some $n \geq 1$.

The aim of the present section is to show the following theorem.

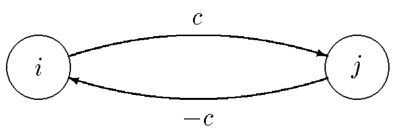
Theorem 4. *Let π, δ and $C_{\max}$ be as above. Assume that $TS(\pi) \neq \emptyset$ and let $t \in TS_n(\pi)$. If t is a precise time of π then $t \in \delta\mathbb{N}$ and if $t > 0$ then the half-open interval $[t - C_{\max}, t)$ contains another precise time. Otherwise, $TS_n(\pi)$ is a non-empty open interval (r, s) with $r \in \delta\mathbb{N}$ and $s \in \delta\mathbb{N} \cup \{+\infty\}$.*

The main idea of the proof is to associate with the path π a directed graph G with edge weights such that the interval can be calculated from the graph. We first normalize the automaton in two simple steps as follows:

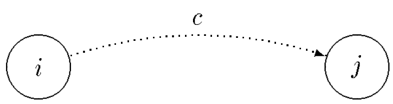
Step 1: We may assume that there is some clock x_0 such that $x_0 \geq 0$ is a constraint of A_i and $x_0 \in \alpha_i$ for all $i \geq 0$.

Step 2: We may assume that the path π is infinite. (If the path π is finite, enter a new state performing any action with the only constraint $x_0 > 0$. Loop in this state. This modification does not touch $TS_n(\pi)$, since $TS_n(\pi) \neq \emptyset$.)

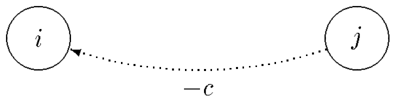
We are now ready to define the graph G. For convenience we put $\alpha_0 = X$. The vertex set of the graph is $V = \{0, 1, 2, \ldots\} = \mathbb{N}$. There are two types of directed edges. Let $x \in X$ be a clock and $c \in C$ be a constant such that $x \in \alpha_i$ and $(x \# c) \in A_j$ for some $i < j$ and $x \notin (\alpha_{i+1} \cup \ldots \cup \alpha_{j-1})$.
If $(x = c) \in A_j$, then we define *strong arcs* with weights c and $-c$ respectively:



If $(x < c) \in A_j$, then we define a *soft arc* with weight c:



If, finally, $(x > c) \in A_j$ we define a soft arc with weight $-c$:



We define $i \sim j$, if i and j are connected by some path using strong edges, only. Clearly $\sim$ is an equivalence relation on V. An induction on the length of the path yields:

Lemma 5. *Let $t_0 t_1 t_2 \cdots \in TS(\pi)$ be the time sequence of some π-run; $i, j \in V$ and γ_{ij} the weight of some directed path of G from i to j.*

1. *If the path uses strong edges only, then we have $t_j - t_i = \gamma_{ij}$.*
2. *If the path uses at least one soft edge, then we have $t_j - t_i < \gamma_{ij}$.*

Let $i, j \in V$, assuming $\inf \emptyset = +\infty$, we define

$$c_{ij} = \inf\{\gamma_{ij} \mid \gamma_{ij} \text{ is the weight of a directed path from } i \text{ to } j\}$$

In order to prove the theorem, we need some few technical results:

Lemma 6. *Let $t_0 t_1 t_2 \cdots \in TS(\pi)$ be the time sequence of some π-run and let $i, j, m \in V$. Then,*

1. $-c_{ji} \leq t_j - t_i \leq c_{ij}$.
2. $c_{ij} \in \delta\mathbb{Z} \cup \{+\infty\}$ *and if $c_{ij} \neq +\infty$ then there exists a directed path in G from i to j of weight c_{ij}.*
3. $i \sim j \iff t_j - t_i = c_{ij} \iff c_{ij} + c_{ji} = 0$.
4. $c_{ij} \leq c_{im} + c_{mj}$.
5. $c_{ij} = c_{im} + c_{mj}$ *if $i \sim m$ or $m \sim j$.*
6. $c_{ij} < c_{im} + c_{mj}$ *if $i \sim j$ and $i \not\sim m$.*

Lemma 7. *We have*

$$TS(\pi) = \{t_0 t_1 t_2 \cdots \mid \forall i, j \in V : t_j - t_i = c_{ij} \text{ if } i \sim j \text{ and}$$
$$t_j - t_i < c_{ij} \text{ otherwise}\}.$$

We can now state the decisive lemma.

Lemma 8. *We have*

$$TS_n(\pi) = \begin{cases} \{c_{0n}\} & \text{if } n \sim 0 \\ (-c_{n0}, c_{0n}) & \text{otherwise.} \end{cases}$$

Proof. If $n \sim 0$ then the result is a consequence of Lemma 6 (3) since $t_0 = 0$. Assume now that $n \not\sim 0$. For each $k \geq 0$ we will define inductively in stage k a subset $s(k) \subseteq V$ satisfying the following properties:

1. The $s(k)$ is a finite union of the equivalence classes.
2. For all $i \in s(k)$ a time t_i is defined.
3. For all $i, j \in s(k)$ we have

$$t_i - t_j = \begin{cases} c_{ij} & \text{if } i \sim j \\ t_j - t_i < c_{ij} & \text{otherwise.} \end{cases}$$

To begin with let $s(0) = [0] = \{i \in V \mid 0 \sim i\}$ the equivalence class of 0. For $i \in [0]$ define $t_i = c_{0i}$. Clearly 1), 2), and 3) are satisfied. To see 3) observe that for $i \sim j \sim 0$ we have by Lemma 6 $c_{ij} = c_{i0} + c_{0j}$ and $c_{0i} = -c_{i0}$.
Assume that $s(k)$ has been defined, $k \geq 0$. If $s(k) = V$ we stop the procedure. Otherwise, for each $m \in V \setminus s(k)$ define an open interval

$$I_m^{(k)} = (r_m^{(k)}, s_m^{(k)})$$

by $r_m^{(k)} = \max\{t_j - c_{mj} \mid t_j \in s(k)\}$ and $s_m^{(k)} = \min\{t_i + c_{im} \mid t_i \in s(k)\}$.
Note that for $i \sim j \in s(k)$ and $m \in V$, by Lemma 6, we have $t_i - c_{mi} = t_j - c_{mj}$ and $t_i + c_{im} = t_j + c_{jm}$. Hence the maximum and the minimum are taken over finite sets. The interval $I_m^{(k)}$ is not empty since $t_j - c_{mj} < t_i + c_{im}$ for all $i, j \in s(k), m \notin s(k)$: Indeed, for $i \sim j$ we have $t_j - t_i = c_{ij} < c_{im} + c_{mj}$; and for $i \not\sim j$ we have $t_j - t_i < c_{ij} \leq c_{im} + c_{mj}$. Hence the claim in both cases.
For $k = 0$ and $m \not\sim 0$ we have $I_m = (c_{m0}, c_{0m})$. Now consider $k+1$. If $k+1 = 1$ then choose $m = n$. For $k+1 > 1$ choose any $m \notin s(k)$. Define $s(k+1) = s(k) \cup \{j \in V \mid j \sim m\}$ so that (1) holds for $s(k+1)$. Let t_m be any value of the open interval $I_m^{(k)}$. For all $j \sim m$ define $t_j = t_m + c_{mj}$. Now (2) holds for $s(k+1)$ as well. Finally, (3) is a direct consequence of $t_j \in I_j^{(k)}$ for all $j \sim m$ which can be easily checked using $t_m \in I_m^{(k)}$, the definition of t_j and Lemma 6.
Since $0 \not\sim n$, we can choose a sequence $m_0 = 0, m_1 = n, m_2, m_3, \ldots$ such that $s(k) = [0] \cup [n] \cup [m_2] \cup \cdots \cup [m_k]$ and $\bigcup_{k \geq 0} s(k) = V$. This defines values t_m for all $m \geq 0$ and thereby a sequence $t_0 t_1 t_2 \cdots$.
By property 3) above and Lemma 7 we see that $t_0 t_1 t_2 \cdots \in TS(\pi)$. The result follows since $m_1 = n$ and the only condition on t_n has been $t_n \in I_n^{(0)} = (-c_{n0}, c_{0n})$.

The proof of Theorem 4 is now easy. From Lemma 8 we deduce that the precise times of π are $\{c_{0n} \mid n \sim 0\} \subseteq \delta\mathbb{N}$. Moreover, assume that $t = c_{0n} > 0$ is a precise time. Since $n \sim 0$ there is a path in G composed of strong arcs from n to 0: $n = n_0 \longrightarrow n_1 \longrightarrow \cdots \longrightarrow n_k = 0$. Let $i = \inf\{j \mid c_{0n_j} < c_{0n}\}$. We have $c_{0n} - c_{0n_j} \leq c_{0n_{j-1}} - c_{0n_j} \leq C_{\max}$. Hence the precise time $c_{0n_j} \in [c_{0n} - C_{\max}, c_{0n})$. Now if $n \not\sim 0$, then by Lemma 8 we have $TS_n(\pi) = (-c_{n0}, c_{0n})$ and by Lemma 6 we obtain $-c_{n0} \in \delta\mathbb{N}$ and $c_{0n} \in \delta\mathbb{N} \cup \{+\infty\}$ which concludes the proof.

The theorem above yields a tool to prove that certain languages are not recognizable by timed automata (without ε-reset transitions resp.). The application we have in mind is the following simple consequence.

Corollary 9. *Every timed automaton recognizing the language $L(\mathcal{A}_0)$ from Ex. 1 has an ε-reset transition lying on some directed cycle.*

Proof. By contradiction. By Theorem 2 we may assume $L(\mathcal{A}_0)$ is recognized by some automaton without any ε-transition. Applying Theorem 4 we let $\delta > 0$ and $C_{\max}$ be the constants introduced at the beginning of Section 4. We find some $d \in \mathbb{N}$, $d \geq C_{\max}$ and an accepted word of the form

$$(b, \delta_1)(b, \delta_2) \cdots (b, \delta_{d-1})(a, d)(a, d+1) \cdots$$

such that $\delta_i \in (i-1, i) \setminus \delta\mathbb{N}$ for all $0 < i < d$. Let π be a path accepting this timed word. The time d must be precise contradicting Theorem 4.

Acknowledgment: We thank the anonymous referees for useful remarks which yield to a simplification of our original construction.

References

1. R. Alur, C. Courcoubetis, D.L. Dill, N. Halbwachs, and H. Wong-Toi. Minimization of timed transition systems. In *Proceedings of CONCUR'92*, number 630 in Lecture Notes in Computer Science. Springer Verlag, 1992.
2. R. Alur and D.L. Dill. Automata for modeling real-time systems. In *Proceedings of ICALP'90*, number 443 in Lecture Notes in Computer Science, pages 322–335. Springer Verlag, 1990.
3. R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
4. R. Alur, L. Fix, and T.A. Henzinger. A determinizable class of timed automata. In *Proceedings of CAV'94*, number 818 in Lecture Notes in Computer Science, pages 1–13. Springer Verlag, 1994.
5. R. Alur and T.A. Henzinger. Back to the future: towards a theory of timed regular languages. In *Proceedings of FOCS'92*, Lecture Notes in Computer Science, pages 177–186. Springer Verlag, 1992.
6. B. Bérard. Untiming timed languages. *Information Processing Letters*, 55:129–135, 1995.
7. B. Bérard, P. Gastin, and A. Petit. On the power of non observable actions in timed automata. In *Proceedings of STACS'96*, number 1046 in Lecture Notes in Computer Science, pages 257–268. Springer Verlag, 1996.
8. C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. In *Proceedings of CAV'91*, number 575 in Lecture Notes in Computer Science, pages 399–409. Springer Verlag, 1991.
9. T.A. Henzinger, P.W. Kopke, and H. Wong-Toi. The expressive power of clocks. In *Proceedings of ICALP'95*, number 944 in Lecture Notes in Computer Science, pages 335–346. Springer Verlag, 1995.
10. T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
11. T. Wilke. Specifying timed state sequences in powerful decidable logics and timed automata. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.
12. H. Wong-Toi and G. Hoffmann. The control of dense real-time discrete event systems. In *Proceedings of the 30th IEEE Conf. on Decision and Control*, pages 1527–1528, 1991.