ELSEVIER

Information Processing Letters 65 (1998) 313–318

Information Processing Letters

Timed automata and recognizability

Philippe Herrmann [1]

LIAFA, Université Paris 7, 2 place Jussieu, F-75251 Paris Cedex 05, France

Received 17 October 1997; revised 8 December 1997
Communicated by L. Boasson

Keywords: Real-time systems; Formal languages; Timed automata

1. Introduction

Although the theory of timed automata introduced in [1] has been studied quite extensively from the viewpoint of formal languages, only a few results are known concerning the recognizability of timed languages in this model. This may come as a surprise, considering the fact that in the underlying untimed theory it is easy to establish a so-called pumping lemma, providing a straightforward way to prove that certain languages cannot be recognized. In most papers, the authors answer the questions of nonrecognizability by the use of informal arguments, either because no formal proof is known or because it is deemed as sufficient evidence. Moreover, the only formal result we came across, presented in [3], has a rather complicated proof, and can only be used with timed automata using only rationally dependent constants. The aim of this paper is to provide a simple yet powerful way of proving formally that a variety of timed languages cannot be recognized by timed automata. The presented result has a wide range of applicability, since it can for instance be used for timed automata with ε-transitions or tube acceptance (see [4] for this last notion) while the proofs remain very similar.

[1] Email: herrmann@litp.ibp.fr.

2. Presentation of the model

A *timed automation* is a tuple $\mathcal{A} = (Q, \Sigma, E, Q_0, F, R, X, C)$ where Q is a finite set of states, Σ is a finite alphabet (the empty word of Σ^* will be denoted by ε), E is a finite set of transitions (see the details hereafter), $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, $R \subseteq Q$ is the set of repeated states, X is a finite set of clocks, $C \subseteq \mathbb{R}$ is a finite set of real constants.

A *constraint* is a formula defined by the grammar (with $x \in X$ and $c \in C$)

$$\Phi ::= \mathit{true} \mid x < c \mid x = c \mid x > c \mid \Phi \wedge \Phi.$$

A transition of E has the form $p \xrightarrow{A,a,\alpha} q$ where A is a constraint, $a \in \Sigma \cup \{\varepsilon\}$ an event, $\alpha \subseteq X$ a set of clock resets, and $p, q \in Q$ the source and goal states. Transitions for which $a = \varepsilon$ are called ε-transitions.

A *clock-valuation* ν is a function that assigns to each clock variable $x \in X$ a nonnegative real. A constraint A is satisfied by ν if it evaluates to true when each clock $x \in X$ takes the value $\nu(x)$. For $t \in \mathbb{R}^+$, the clock-valuation $\nu + t$ is defined by $(\nu + t)(x) = \nu(x) + t$.

A *time sequence* $\tau = (\tau_i)_{i \geqslant 1}$ is a finite or infinite increasing sequence of nonnegative real numbers. For any such sequence, we define $\tau_0 = 0$. Given a time se-

0020-0190/98/$19.00
PII S0020-0190(97)00217-2

quence τ and $\sigma \in (\Sigma \cup \{\varepsilon\})^\infty$ of same length as τ, we define the *ε-timed word* $(\sigma, \tau) = (a_1, \tau_1)(a_2, \tau_2)\cdots$. The date of the event a_i is the time τ_i.

Let $\tau = (\tau_i)_{1 \leqslant i < l}$ ($l = +\infty$ if τ is infinite) be a time sequence. A *run* of the ε-timed word (σ, τ) through the timed automaton $\mathcal{A}$ is a sequence of pairs of states and clock-valuations $(q_i, \nu_i)_{0 \leqslant i < l}$ satisfying

- $q_0 \in Q_0$ and for all $x \in X$, $\nu_0(x) = 0$;
- for $1 \leqslant i < l$, an edge of the form $q_{i-1} \xrightarrow{A_i, \sigma_i, \alpha_i} q_i$ belongs to E, A_i is satisfied by $\nu_{i-1} + \tau_i - \tau_{i-1}$, and if $x \in \alpha_i$, $\nu_i(x) = 0$, otherwise $\nu_i(x) = \nu_{i-1}(x) + \tau_i - \tau_{i-1}$;
- if the run is finite, then $q_{l-1} \in F$, otherwise it is required that at least one of the states of R appears infinitely often in the run.

The path $\pi = q_0 \xrightarrow{A_1, \sigma_1, \alpha_1} q_1 \xrightarrow{A_2, \sigma_2, \alpha_2} q_2 \cdots$ can be associated with such a run. For a given path π of the timed automaton, we define $TS(\pi)$ as the set of all time sequences τ such that there is a run of (σ, τ) associated with the path π. As a shorthand, we define $x(i) = \nu_{i-1}(x) + \tau_i - \tau_{i-1}$ (the valuation of clock x during transition i before the reset takes place) for $x \in X$ and $1 \leqslant i < l$, and $x(0) = \nu_0(x) = 0$.

We say that the ε-timed word is accepted by $\mathcal{A}$ when there is a run of (σ, τ) through $\mathcal{A}$. Since ε-transitions correspond to invisible transitions, the corresponding *timed word* accepted by the timed automaton is obtained by removing all the pairs (a_i, τ_i) for which $a_i = \varepsilon$, thus getting a sequence of the form $(a_{i_1}, \tau_{i_1})(a_{i_2}, \tau_{i_2})\cdots \in (\Sigma \times \mathbb{R}^+)^\infty$ (it is still an ε-timed word).

A *timed language* is a set of timed words. The timed language $L(\mathcal{A})$ recognized or accepted by the timed automaton $\mathcal{A}$ is the set of timed words accepted by $\mathcal{A}$. For any timed language L on the alphabet Σ, the language $\mathrm{Untime}(L)$ is defined as the set of words $\sigma \in \Sigma^\infty$ satisfying $(\sigma, \tau) \in L$ for some time sequence τ.

Note. The grammar used to express constraints is not as restrictive as it may appear. Since it is possible to have any finite number of transitions between states, the grammar

$$\Phi ::= x < c \mid x = c \mid x > c \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi$$

is in fact equivalent, though a few constructions presented afterwards would be more awkward if we were to use it.

3. Main result

Let $\mathcal{A}$ be a timed automaton (with ε-transitions) for which, without loss of generality, we suppose that $C = -C$. The normal form of $\mathcal{A}$ is obtained by:

(1) adding to X a clock x_0 which didn't already belong to it,
(2) adding x_0 to every reset set,
(3) transforming every edge with constraint A into two edges with the same label, source and goal, then changing the constraint of one of these edges to $A \wedge x_0 > 0$ and the other one to $A \wedge X_0 = 0$.

In particular, the normalized timed automaton still recognizes $L(\mathcal{A})$, and all its constraints contain either $x_0 > 0$ or $x_0 = 0$. We will use the notation $x\#c \in A$ to denote that $x\#c$ is one of the atomic formulas of A.

Let $\mathcal{A}$ be a normalized timed automaton, and π a finite or infinite path. We now define the graph $G(\pi)$ associated with this path. Its set of nodes V is equal to $\mathbb{N}$ if π is infinite, to $\{0, 1, \ldots, n\}$ where n is the number of transitions of π otherwise. Let $x \in X$ and $c \in C$ for which there are indices $i < j$ such that $x \in \alpha_i$, $x\#c \in A_j$ and $x \notin (\alpha_{i+1} \cup \cdots \cup \alpha_{j-1})$, assuming that $\alpha_0 = X$. If $x = c \in A_j$, $G(\pi)$ contains *hard edges* $i \xrightarrow{c} j$ (source i, goal j, weight c) and $j \xrightarrow{-c} i$. If $x < c \in A_j$ (respectively $x > c \in A_j$), $G(\pi)$ contains a *soft edge* $i \xrightarrow{c} j$ (respectively a soft edge $j \xrightarrow{-c} i$).

One of the atomic formulas $x_0 > 0$ or $x_0 = 0$ being part of any constraint, there is always a (soft or hard) edge $i + 1 \xrightarrow{-0} i$ for all nodes $i \in V$. There may be more than one edge between two given nodes of $G(\pi)$. Since we supposed $C = -C$, the weight of any edge is an element of C. No difference is made between ε-transitions and other transitions. All these notions are introduced in [3].

The following lemma is the main result of the article, and is valid for any normalized timed automaton. Despite being very simple, it can be applied in proofs of nonrecognizability for a variety of languages and models.

Lemma 1. *Let $\pi = q_0 \xrightarrow{A_1, \sigma_1, \alpha_1} q_1 \xrightarrow{A_2, \sigma_2, \alpha_2} q_2 \cdots$ be an accepting path of $\mathcal{A}$ and let $\tau = (\tau_i)_{i \geqslant 1}$ be a sequence of reals of the same length (we let $\tau_0 = 0$). Then we have $\tau \in TS(\pi)$ if and only if for every hard edge $i \xrightarrow{c} j$ of $G(\pi)$, we have $\tau_j - \tau_i = c$, and for every soft edge $i \xrightarrow{c} j$ of $G(\pi)$, we have $\tau_j - \tau_i < c$.*

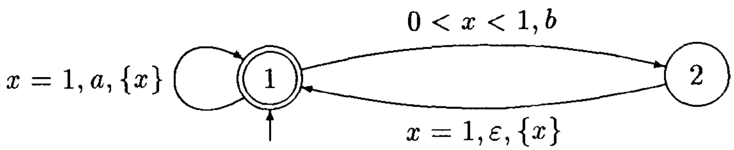

Fig. 1.

Proof. By definition, $\tau \in TS(\pi)$ if and only if $\tau = (\tau_i)_{1 \leqslant i < l}$ is a time sequence and there is a run $(q_i, \nu_i)_{0 \leqslant i < l}$ of (σ, τ) with associated path π through the normalized timed automaton. Since the constraints A_i are conjunctions of atomic formulas and π is an accepting path, this last requirement is equivalent to: for $1 \leqslant j < l$, $x \in X$ and $c \in C$, if $x\#c \in A_j$, then $x(j)\#c$ is true.

Assume that $\tau \in TS(\pi)$ and consider a hard (respectively soft) edge $i \xrightarrow{c} j$ in $G(\pi)$. By construction of $G(\pi)$, and defining $\alpha_0 = X$, we have:

- either $i < j$ and we find $x \in \alpha_i$ with $x \notin (\alpha_{i+1} \cup \cdots \cup \alpha_{j-1})$ and $x = c \in A_j$ (respectively $x < c \in A_j$). In that case, since $\tau \in TS(\pi)$, we have $x(j) = c$ (respectively $x(j) < c$). Now $x(j) = \tau_j - \tau_i$.
- or $i > j$ and we find $x \in \alpha_j$ with $x \notin (\alpha_{j+1} \cup \cdots \cup \alpha_{i-1})$ and $x = -c \in A_i$ (respectively $x > -c \in A_i$). In that case, since $\tau \in TS(\pi)$, we have $x(i) = -c$ (respectively $x(i) > -c$). Now $x(i) = \tau_i - \tau_j$.

Conversely, suppose that for every hard (respectively soft) edge $i \xrightarrow{c} j$ in $G(\pi)$, we have $\tau_j - \tau_i = c$ (respectively $\tau_j - \tau_i < c$). If $i \in V$, $G(\pi)$ contains an edge (hard or soft) $i + 1 \xrightarrow{-0} i$, hence $\tau_{i+1} \geqslant \tau_i$. Thus τ is increasing and since $\tau_0 = 0$, its consists only of nonnegative integers, hence it is a time sequence. Let $1 \leqslant j < l$ with $x\#c \in A_j$, and let $0 \leqslant i < j$ satisfying $x \in \alpha_i$ and for all $i < k < j$, $x \notin \alpha_k$: then $x(j) = \tau_j - \tau_i$. If # is equal to = (respectively <, >), $G(\pi)$ contains a hard edge $i \xrightarrow{c} j$ (respectively soft edge $i \xrightarrow{c} j$, $j \xrightarrow{-c} i$) thus $\tau_j - \tau_i = c$ (respectively $\tau_j - \tau_i < c$, $\tau_j - \tau_i > c$), and $x(j)\#c$ evaluates to true. □

The following lemma provides a formal way for stating that at a given time, only the dates of a finite number of events can be stored.

Lemma 2. *Let π be a path in a normalized timed automaton with n clocks (x_0 included), and $\tau \in TS(\pi)$. Let S be a set of at least $n + 1$ nodes, and let $t \in \mathbb{R}$ be such that for all $i \in S$, $t > \tau_i$. Then there exists $j \in S$ such that there is no edge between j and any other node k satisfying $\tau_k \geqslant t$.*

Proof. Let m be the number of nodes of S being the end of an edge whose other end is a node k such that $\tau_k \geqslant t$. Let s be one of these m nodes. By construction of $G(\pi)$, α_s must contain a clock which isn't reset again before time t, and thus by no node of S greater than s. Hence there are at least m distinct clocks, and $m < n + 1 \leqslant |S|$. □

4. A few proofs of nonrecognizability

In all the following examples, we will consider normalized timed automata, in order to be able to use Lemma 1 freely. The way to proceed is by contradiction: we suppose that the given language can be recognized by a timed automaton, then use Lemma 1, first to study the graph associated with a particular run of the timed automaton, and then to prove that a certain unwanted word belongs to the language of the timed automaton.

4.1. Example 1

In [3], the authors prove that the language of the timed automaton $\mathcal{A}_\varepsilon$ in Fig. 1 cannot be recognized without ε-transitions (at least in the case where only rationally dependent constants are used for the timed automaton). The following proof is simpler and more general, since any finite number of real constants may be used.

Suppose there exists a timed automaton $\mathcal{A}$ without ε-transitions recognizing $L(\mathcal{A}_\varepsilon)$. Let $C \subseteq \mathbb{R}$ be a finite set containing the constants used by $\mathcal{A}$ and such that $C = -C$, let $0 < \delta < 1$ be chosen such that for all $i \in \mathbb{Z}$, $C \cap (i-\delta, i+\delta) \subseteq \{i\}$, and let $n > 0$ be an integer which is strictly greater than $c_{\max}$, the maximum of C. Let $\tau = (\tau_i)_{i \geqslant 1}$ be an infinite time sequence defined by $\tau_n = n$, and for $i \neq n$, $\tau_i = i - 1 + \frac{1}{2}\delta$. The timed word $(b^{n-1}ab^\omega, \tau)$ is accepted by $\mathcal{A}_\varepsilon$. Hence there exists a path π of $\mathcal{A}$ such that $\tau \in TS(\pi)$. Let τ' be the time sequence defined by $\tau'_n = n + \frac{1}{4}\delta$ and $\tau'_i = \tau_i$ for $i \neq n$.

In $G(\pi)$, there cannot be any edge $0 \xrightarrow{c} n$, since in that case we would have $c_{\max} < n = \tau_n \leqslant c$ with $c \in C$. If there is an edge $i \xrightarrow{c} n$ with $i > 0$, since $\tau_n - \tau_i = n - i + 1 - \frac{1}{2}\delta \leqslant c$ with $c \in C$ and since $C \cap (n - i + 1 - \delta, n - i + 1) = \emptyset$, this edge is soft and $c \geqslant n - i + 1$. Therefore we still have for $i > 0$, $\tau'_n - \tau'_i \leqslant c$. Any edge $n \xrightarrow{c} i$ with $i \geqslant 0$ being soft, $-c < \tau_n - \tau_i < \tau'_n - \tau'_i$. Thus $\tau' \in TS(\pi)$, and the timed automaton $\mathcal{A}$ would accept a word which has two events in the interval $(n, n+1)$ and none in $(n-1, n]$. This contradicts the fact that $\mathcal{A}$ and $\mathcal{A}_\varepsilon$ recognize the same language. This proof remains valid whatever the label of the loop is (that is, even b).

The method based on precise times developed in [3] does not allow us to prove that the language of the timed automaton obtained by removing the loop from $\mathcal{A}_\varepsilon$ cannot be recognized by a timed automaton without any ε-transition. A proof similar to the one before yields this result easily.

4.2. Example 2

In [1], the authors give evidence for the fact that the timed automaton of the figure hereafter cannot be complemented (even if we allow ε-transitions for the complement).

Suppose there is a timed automaton $\mathcal{A}_\varepsilon$ (ε-transitions are allowed) whose language is the complement of the language of the automaton in Fig. 2.

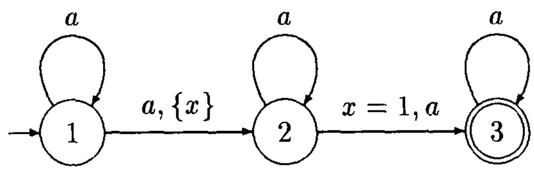

Fig. 2.

Let X be the set of clocks used by $\mathcal{A}_\varepsilon$ (x_0 included) and let n be an integer strictly greater than $|X| + 1$. Let C be a finite set satisfying $C = -C$ and containing all the constants used by $\mathcal{A}_\varepsilon$, let $0 < \delta < 1$ be such that for all $i \in \mathbb{Z}$, $C \cap (i - \delta, i + \delta) \subseteq \{i\}$.

Let the finite sequence $\tau = (\tau_i)_{1 \leqslant i \leqslant 2n}$ satisfy $0 < \tau_1 < \cdots < \tau_n < \delta < 1 + \tau_1 < \tau_{n+1} < 1 + \tau_2 < \cdots < 1 + \tau_n < \tau_{2n} < 1 + \delta$. The timed word (a^{2n}, τ) is accepted by $\mathcal{A}_\varepsilon$: let π be the path of the associated run (which may include any number of ε-transitions, even infinitely many), and let $\tau' \in TS(\pi)$ be the time sequence of the corresponding finite or infinite ε-timed word $(\sigma, (\tau'_i)_{i \geqslant 1})$. Let $(i_j)_{1 \leqslant j \leqslant 2n}$ be the strictly increasing sequence of integers satisfying $\sigma_{i_j} = a$ (hence $\tau'_{i_j} = \tau_j$). For $1 \leqslant j \leqslant n - 1$, let S_j be the nonempty set of nodes of $G(\pi)$ corresponding to the transitions of π taking place in the interval $[\tau_j, \tau_{j+1})$. The corresponding dates are all strictly lower than δ. Since $n - 1 > |X|$ and by the argument of Lemma 2, there exists $1 \leqslant k \leqslant n - 1$ such that there is no edge between any node of S_k and any node associated with a time greater than or equal to δ.

Let f be the strictly increasing affine mapping defined by $f(\tau_k) = \tau_{n+k} - 1 < \tau_{k+1}$ and $f(\tau_{k+1}) = \tau_{k+1} < \delta$. Let τ'' be the sequence of reals of same length as τ' defined by $\tau''_i = f(\tau'_i)$ if $\tau'_i \in [\tau_k, \tau_{k+1})$ and $\tau''_i = \tau'_i$ otherwise. Note that if $\tau'_i < \tau'_j$ then $\tau''_i < \tau''_j$ and if $\tau'_j = \tau'_i$ then $\tau''_j = \tau''_i$. To show that $\tau'' \in TS(\pi)$, we only have to check that the constraints of $G(\pi)$ involving nodes of S_k are still satisfied. Let $i \xrightarrow{c} j$ be an edge with i or j in S_k. Thus $-\delta < \tau'_j - \tau'_i < \delta$ and since $C \cap (-\delta, \delta) \subseteq \{0\}$, we deduce that $c = 0$ if the edge is hard and either $c = 0$ or $c > \delta$ if it is soft. In the first case $\tau'_j - \tau'_i = 0$ thus $\tau'_j - \tau'_i = 0$. In the second case, if $c = 0$ then $\tau''_j - \tau''_i < 0$ since $\tau'_j - \tau'_i < 0$ and if $c > \delta$ then $\tau''_j - \tau''_i \leqslant f(\tau'_j) < \delta < c$. Thus $\tau'' \in TS(\pi)$. But τ'' is such that $\tau''_{i_{n+k}} = \tau''_{i_k} + 1$, and the corresponding timed word doesn't belong to the required language. This ends the proof by contradiction.

4.3. Example 3

We give a formal proof of the fact that timed automata do not in general have a tube complement: the question was raised in [4], where the

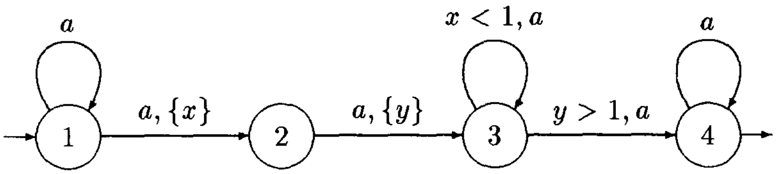

Fig. 3.

idea of the proof is given, and the reader should refer to this paper for detailed definitions and results about tube acceptance. An easy consequence of the proof is the fact that timed automata are not tube-determinizable, a proposition which was already proven in [4]. Also, we use a timed automaton equivalent to the one presented in [4] for this question, though it is not an open timed automaton anymore.

We recall that the distance $d_{\max}$ between the finite timed words (or *trajectories*) (u, τ) and (u', τ') is equal to $\max\{|\tau'_i - \tau_i|\}$ if $u = u'$ and $+\infty$ otherwise. Tube is defined as the set of all open sets of trajectories for this metric. The *tube language* $[L]$ induced by a timed language L is equal to $\{\mathcal{O} \in \text{Tube} \mid \mathcal{O} \subseteq \bar{L}\}$ where $\bar{L}$ denotes the closure of L. The complement $[L]^c$ of a tube language $[L]$ is defined as the set $\{\mathcal{O} \in \text{Tube} \mid \forall \mathcal{O}' \in [L],\ \mathcal{O} \cap \mathcal{O}' = \emptyset\}$.

Suppose there is a normalized timed automaton without ε-transitions $\mathcal{A}$ being a tube complement of $\mathcal{A}'$, the timed automaton in Fig. 3, that is $[L(\mathcal{A})] = [L(\mathcal{A}')]^c = \{\mathcal{O} \in \text{Tube} \mid \forall \mathcal{O}' \in [L(\mathcal{A})],\ \mathcal{O} \cap \mathcal{O}' = \emptyset\}$.

We define X, C, δ in the same way they have been defined in the last example. Let n be an integer strictly greater than $|X| + 2$. Let $\tau = (\tau_i)_{1 \leqslant i \leqslant 2n-1}$ be a finite time sequence satisfying $0 < \tau_1 < \cdots < \tau_n < \delta < 1 + \tau_1 < \tau_{n+1} < 1 + \tau_2 < \cdots < 1 + \tau_{n-1} < \tau_{2n-1} < 1 + \tau_n$, and let $\sigma = a^{2n-1}$. We check that the word (σ, τ) is rejected by $\mathcal{A}'$, since there are no consecutive events i and $i + 1$ for which there is no event in $[\tau_i + 1, \tau_{i+1} + 1]$ and at least one event afterwards. This word being finite, it is the center of a tube whose trajectories all have a time component satisfying the same specification as τ, and are therefore all rejected by $\mathcal{A}'$. Consequently, since $[L(\mathcal{A})] = [L(\mathcal{A}')]^c = [L(\mathcal{A}')^c]$ (see [4]), $\mathcal{A}$ must accept (σ, τ) robustly: in other words, there exists a timed word which is arbitrarily close to (σ, τ) (for distance $d_{\max}$ for instance) accepted by $\mathcal{A}$. Since we can choose the time component of this word so that it still satisfies the same specification as τ, we may consider that $\mathcal{A}$ accepts (σ, τ). There exists a path π in $\mathcal{A}$ associated with the corresponding run, and we define $G(\pi)$ in the usual way. For all distinct pairs of nodes $0 \leqslant i \neq j \leqslant 2n - 1$, the value of $\tau_i - \tau_j$ is either in $(-\delta, \delta) - \{0\}$, or in $(1 - \delta, 1 + \delta) - \{1\}$, or in $(-1 - \delta, -1 + \delta) - \{-1\}$. The intersection of C with each of these intervals being empty, there cannot be any hard edge in $G(\pi)$. Therefore, the path π being finite, one can check as a consequence of Lemma 1 that every timed word accepted through π will be accepted robustly as the center of an accepted tube.

By Lemma 2, since n is strictly greater than $|X| + 2$, there exists a node $2 \leqslant i \leqslant n - 1$ such that there is no edge between i and $j \geqslant n + 1$. Each edge having i as an end has its other end amongst the nodes $0 \leqslant k \neq i \leqslant n$. Such an edge would have to be soft and labelled with a constant whose value is either 0 or greater than or equal to δ since $C \cap (-\delta, \delta) \subseteq \{0\}$. The constraints remain satisfied if the date of event i takes any value in (τ_{i-1}, τ_{i+1}), in particular a value τ'_i in $(\tau_{i-1}, \tau_{n+i-1} - 1)$. The other dates staying invariant, we define this way a time sequence τ' such that (σ, τ') is still accepted through the path π, hence accepted robustly as seen before. The timed word we obtain is accepted by $\mathcal{A}'$ since $1 + \tau'_{i-1} < 1 + \tau'_i < \tau_{n+i-1} = \tau'_{n+i-1}$: there is no event in $(1 + \tau'_{i-1}, 1 + \tau'_i)$ and at least one afterwards. But every timed word accepted by $\mathcal{A}'$ is accepted robustly since all the constraints of this timed automaton are strict and its behaviour is finite. Therefore we obtain a tube which belongs to $[L(\mathcal{A})]$ and to $[L(\mathcal{A}')]$ at the same time, and this leads us to a contradiction.

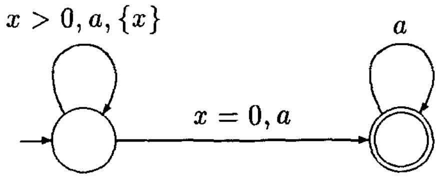

Fig. 4.

5. Related results

We present a few propositions that were established during our investigation of tube languages. The proofs can be found in [5]. For distance $d_{\max}$, we extend the definitions of [4] to infinite behaviours in the obvious way.

It can be proved that for a timed automaton with finite behaviours, edges carrying a constraint with equality may be removed when only the tube language is considered (provided that the constraints are conjunctions of atomic formulas of course). The proof of this result uses the fact that given any time sequence τ and any finite set of reals C, it is possible to find an arbitrarily close time sequence τ' for which no date and no difference between any two dates are in C. It must also be shown that if such a time sequence τ' is finite, there is a tube centered on τ' containing only time sequences satisfying the same property as τ'.

When infinite behaviours are considered, edges carrying a constraint with equality cannot in general be removed without changing the tube language. For instance, let $\mathcal{A}$ be the timed automaton depicted in Fig. 4. Its tube language $[L(\mathcal{A})]$ is not empty (consider the set of all infinite trajectories for which the number of events in a unit of time is unbounded), while if we remove the edge with the constraint $x = 0$, we obtain a timed automaton $\mathcal{A}'$ such that $[L(\mathcal{A}')] = \emptyset$.

Moreover, it can be shown with a method related to the previous sections that despite $\mathcal{A}$ being deterministic, there is no timed automaton $\mathcal{B}$ such that $[L(\mathcal{A})]^c = [L(\mathcal{B})]$. This is in sharp contrast with the case of finite behaviours, where for any timed language L we have $[L]^c = [L^c]$: thus if $\mathcal{A}$ is deterministic and has a finite behaviour, there exists $\mathcal{B}$ such that $L(\mathcal{A})^c = L(\mathcal{B})$ and thus $[L(\mathcal{A})]^c = [L(\mathcal{A})^c] = [L(\mathcal{B})]$. In general, the relationship between the tube complement and the trajectory complement is the following: $\mathcal{O} \in$ Tube belongs to $[L]^c$ if and only if $\mathcal{O}$ belongs to $[L^c]$ and $\mathcal{O}$ doesn't contain any element of $[L]$.

Acknowledgements

The author thanks Paul Gastin for his constant help and support, as well as Béatrice Bérard and Antoine Petit for their constructive criticism.

References

[1] R. Alur, D.L. Dill, Automata for modelling real-time systems, in: Proc. ICALP 90, Lecture Notes in Computer Science, vol. 443, Springer, Berlin, 1990, pp. 322–335.

[2] R. Alur, D.L. Dill, A theory of timed automata, Theoret. Comput. Sci. 126 (1994) 183–235.

[3] V. Diekert, P. Gastin, A. Petit, Removing ε-transitions in timed automata, in: Proc. STACS 97, Lecture Notes in Computer Science, vol. 1200, Springer, Berlin, 1997, pp. 583–594.

[4] V. Gupta, T.A. Henzinger, R. Jagadeesan, Robust timed automata, in: Proc. HART 97, Lecture Notes in Computer Science, vol. 1201, Springer, Berlin, 1997, pp. 331–345.

[5] P. Herrmann, Timed automata and recognizability, Tech. Rept., LIAFA, University Paris 7, 1997.