On the power of non observable actions in timed automata

Béatrice Bérard[1], Paul Gastin[2] and Antoine Petit[1]

[1] ENS de Cachan, LIFAC, 61 av. du Prés. Wilson, F-94235 Cachan Cedex, France
[2] Université Paris 7, LITP - IBP, 2, place Jussieu, F-75251 Paris Cedex 05, France

Abstract. Timed finite automata, introduced by Alur and Dill, are one of the most widely studied models for real-time systems. We focus in this paper on the power of silent transitions, i.e. ε-transitions, in such automata. We show that ε-transitions strictly increase the power of timed automata and that the class of timed languages recognized by automata with ε-transitions is much more robust than the corresponding class without ε-transitions. Our main result shows that these transitions increase the power of these automata only if they reset clocks.

1 Introduction

Finite transition systems form one of the most intensively studied models for the specification and verification of systems, see e.g. [?, ?, ?, ?]. By essence, these classical transition systems are suitable for reasoning about qualitative properties. Time requirements cannot be easily described by these automata which do not contain any notion of time. Several attempts have been made to propose models based on transition systems together with a notion of time. For instance, timed Petri nets [?], timed transition systems [?, ?], timed graphs [?] or timed I/O automata [?] have been discussed. We focus in this paper on the simple and natural model of so-called timed automata proposed by Alur and Dill [?, ?]. Since its introduction, this model has been intensively studied under several aspects very close from classical automata: the determinization problem [?], the minimization problem [?], the power of clocks [?, ?], and extensions of the model [?, ?]. Moreover this model has been used successfully for verification and specification of real-time systems [?, ?, ?].

We are interested in this paper on the basic model of timed automata as proposed in the original paper [?]. In this paper, transitions of timed automata can be labeled with a non observable action, that is the empty word, and are therefore often called silent transitions or, shortly, ε-transitions. Such a possibility has curiously disappeared in most of the other papers on timed automata, even in the extended version [?]. It is well known that in the untimed case, such transitions do not increase the power of (classical) finite automata and that these transitions are just a technical, even though very useful, tool.

We show in this paper that in the case of timed automata, the situation is completely different. Namely, ε-transitions strictly increase the power of timed automata. Moreover, we show that the class of timed languages recognized by such automata with ε-transitions is much more robust than when ε-transitions are forbidden. For instance, any recognizable timed language over the time domain $\mathbf{N}$ remains recognizable if the time domain is $\mathbf{R}$. Note that it is surprisingly not the case for timed automata without ε-transitions. Following similar ideas, we exhibit some very natural timed languages which become recognizable with the help of ε-transitions.

In a second part, we investigate precisely the power of ε-transitions in timed automata. Our main result is that these transitions increase the power of these automata, only if they reset clocks. More precisely, we prove that any timed language recognized by a timed automaton where no ε-transition resets clocks is also recognized with a timed automaton without ε-transition.

This extended abstract does not contain any proof. They can be found in [?].

2 Timed languages with non observable actions

In this section, we briefly recall the basic definitions and notations for timed words and finite automata with timing constraints, essentially from [?] and [?]. The only difference concerns the non Zeno property as we explain below.

2.1 Timed words and clocks

Let Z be any set. We write Z^* (respectively Z^ω) the set of finite (respectively infinite) sequences of elements in Z, with ε for the empty sequence, and $Z^\infty = Z^* \cup Z^\omega$ the set of all sequences of elements in Z.

Throughout this paper, we consider a time domain $\mathbf{T}$ which will be either the set of finite integers $\mathbf{N}$ or the set of non-negative reals $\mathbf{R}_+$. A time sequence over $\mathbf{T}$ is a finite or infinite increasing sequence $\tau = (\tau_i)_{i\geq 0}$ of elements in $\mathbf{T}$. The time sequence τ is strict if all its elements are distinct and is divergent if the sequence is unbounded.

The observable actions of the processes we consider are given by a finite set Σ. A timed word is an element of $(\Sigma \times \mathbf{T})^\infty$, $w = (a_i, \tau_i)_{i\geq 0}$, also written as a pair $w = (a, \tau)$, where $a = (a_i)_{i\geq 0}$ is a word in Σ^∞ and $\tau = (\tau_i)_{i\geq 0}$ is a timed sequence in $\mathbf{T}^\infty$ of same length than a. Note that, contrary to the usual definition of a time sequence, we do not include the non Zeno property, i.e. the divergence of the sequence. We will see in Section 3 that, if needed, this condition can be ensured through the intersection with an appropriate regular timed language.

Let X be a set of variables with values in $\mathbf{T}$, called clocks. The clock constraints over X are the formulas defined by the grammar:

$$A ::= x \# t \mid x - y \# t \mid \neg A \mid A \vee A \mid A \wedge A \mid \mathit{true} \mid \mathit{false},$$

where x and y are clocks, $t \in \mathbf{T}$ and $\#$ is a binary operator in $\{<, \leq, =, >, \geq\}$. The set of all clocks constraints over X is denoted by $\mathcal{C}(X)$. A clock valuation $v : X \mapsto \mathbf{T}$ is a mapping that assigns to each clock x a time value $v(x)$. The set of all clock valuations is denoted $\mathbf{T}^X$. We say that $A(v)$ holds and we also write $v \models A$ when the clock valuation v satisfies the clock constraint A. If t is an element of $\mathbf{T}$ and α a subset of X, the valuations $v+t$ and $v[\alpha \leftarrow 0]$ are defined by $(v+t)(x) = v(x)+t$, for each clock x in X and $(v[\alpha \leftarrow 0])(x) = 0$ if $x \in \alpha$, $v(x)$ otherwise. In the same way the valuation $v - t$ is defined only if $v(x) \geq t$, for each clock $x \in X$ and in this case, $(v-t)(x) = v(x) - t$, for each clock x in X.

2.2 Timed automata with ε-transitions

A (Büchi) timed automaton over $\mathbf{T}$ is a tuple $\mathcal{A} = (\Sigma, Q, X, T, I, F, R)$, where Σ is a finite alphabet, S is a finite set of states, $I \subseteq S$ is the subset of initial states, X is the set of $\mathbf{T}$-valued clocks and $T \subseteq S \times [\mathcal{C}(X) \times (\Sigma \cup \{\varepsilon\}) \times \mathcal{P}(X)] \times S$ is the subset of transitions, $F \subseteq S$ is the subset of final states and $R \subseteq S$ is the subset of repeated states. The set of all such automata will be denoted by $\mathbf{TA}_\varepsilon$ whereas the set of all timed automata using transitions with observable actions only will be denoted by $\mathbf{TA}$.

In order to define how a timed word is recognized by such a timed automaton, we recall the notion of path and of timed run through a path. A path in $\mathcal{A}$ is a finite or infinite sequence of consecutive transitions :

$$P = s_0 \xrightarrow{A_1,\sigma_1,\alpha_1} s_1 \xrightarrow{A_2,\sigma_2,\alpha_2} s_2 \cdots$$

where $s_{i-1} \xrightarrow{A_i,\sigma_i,\alpha_i} s_i \in T$ for all $i > 0$. P is an accepting path if it starts in an initial state and either is finite and ends in a final state or is infinite and goes infinitely often through a repeated state.

Intuitively, a transition $s \xrightarrow{A,a,\alpha} s'$ can be performed only if the clock constraint A is satisfied by the current clock valuation and the clocks in α are reset at that time. A run through this path P corresponds to a time sequence $(t_i)_{i>0}$ which satisfies $v_i + (t_{i+1} - t_i) \models A_{i+1}$ for all $i \geq 0$, where the sequence of clock valuations $(v_i)_{i>0}$ is defined by $v_0 = 0$ and $v_{i+1} = (v_i + (t_{i+1} - t_i))[\alpha_{i+1} \leftarrow 0]$ for all $i \geq 0$. The timed word generated by this run is simply the subsequence of $(\sigma_i, t_i)_{i>0}$ obtained by erasing all non observable actions. Note that the timed word generated by a finite run is always finite whereas an infinite run may generate either a finite or an infinite timed word.

The timed language accepted by the timed automaton $\mathcal{A}$ over the time domain $\mathbf{T}$ is the set $L(\mathcal{A}, \mathbf{T})$ of timed words of $(\Sigma \times \mathbf{T})^\infty$ generated by (finite or infinite) accepting runs of $\mathcal{A}$. The family of timed languages generated by some timed automaton of $\mathbf{TA}_\varepsilon$ (resp. of $\mathbf{TA}$) is denoted $\mathbf{TL}_\varepsilon(\mathbf{T})$ (resp. $\mathbf{TL}(\mathbf{T})$) or simply $\mathbf{TL}_\varepsilon$ (resp. $\mathbf{TL}$) if the time domain has not to be stated explicitly.

Above, we use a Büchi type of acceptance for infinite paths. Equivalently, Muller or extended Büchi (see e.g. [?]) conditions could also be used.

3 Expressiveness and Robustness

We study in this section the expressiveness of timed automata with non observable actions and the robustness of the corresponding classes of languages. We

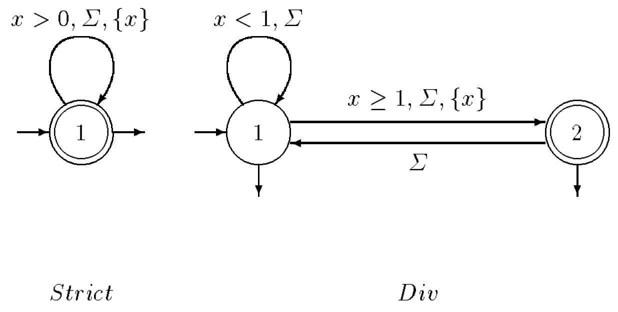

Fig. 1. Timed automata for *Strict* and *Div*

give some examples of timed languages accepted by timed automata and we use some of these languages to compare classes of languages.

Note that in some papers the timed sequences are required to be strictly increasing and/or the infinite timed sequences must be divergent (non Zeno). We did not include these requirements in our definition since they can be enforced respectively by intersection with the language *Strict* of timed words with strict timed sequences and *Div* of timed words with divergent timed sequences. Note that *Strict* and *Div* are accepted by the automata in Figure 1 and that the classes $\mathbf{TL}$ and $\mathbf{TL}_\varepsilon$ are closed under intersection ([?, ?]).

$$\begin{aligned} Strict &= \{(w_i, \tau_i)_{i\geq 0} \in (\Sigma \times \mathbf{T})^\infty \mid \tau_i < \tau_{i+1} \text{ for all } i \geq 0\} \\ Div &= \{(w_i, \tau_i)_{i\geq 0} \in (\Sigma \times \mathbf{T})^\omega \mid \tau_i \leq \tau_{i+1} \text{ for all } i \geq 0 \text{ and } \forall t \in \mathbf{T} \exists i : t < \tau_i\} \end{aligned}$$

We now give a parametrized family of languages of the class $\mathbf{TL}_\varepsilon$ which cannot be accepted without ε-transitions. Let $k < p$ be two values of the time domain $\mathbf{T}$. We consider the language $Delay(k,p)$ which consists in timed words in which the delay between two consecutive actions is always in the set $k + p\mathbf{N}$:

$$Delay(k,p) = \{(w_i, \tau_i)_{i\geq 0} \in (\Sigma \times \mathbf{N})^\infty \mid \tau_{i+1} - \tau_i \in k + p\mathbf{N} \text{ for all } i \geq 0\}$$

The language $Delay(k,p)$ is accepted by the timed automaton of Figure 2. When $k = 0$ and $p = 2$, we obtain as a special case the language *Even* of finite or infinite timed words with even integer dates (see Figure 2).

$$Even = \{(w_i, \tau_i)_{i\geq 0} \in (\Sigma \times \mathbf{N})^\infty \mid \tau_i \in 2\mathbf{N} \text{ and } \tau_i \leq \tau_{i+1} \text{ for all } i \geq 0\}$$

This simple language can be used to show that silent transitions strictly increases the power of timed automata.

Proposition 1. *The class* $\mathbf{TL}(\mathbf{T})$ *is strictly contained in the class* $\mathbf{TL}_\varepsilon(\mathbf{T})$.

Note that, although the expressive power of the class $\mathbf{TA}_\varepsilon(\mathbf{T})$ is greater, it is known from [?] that emptiness is decidable when the constants appearing in the clock constraints are rational. Also, if only rational constants were allowed in the timing constraints of timed automata, we could use the powerful notion of clock regions [?] to prove this proposition. Of course, the proposition above is correct whatever the timing constraints are and it does not need the deep result on clock regions.

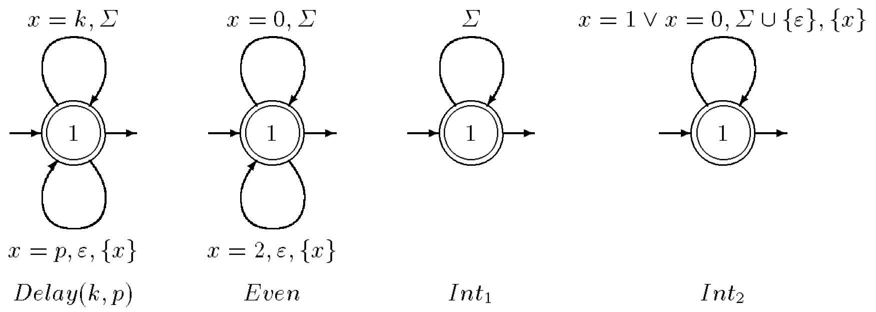

Fig. 2. Timed automata for $Delay(k, p)$, $Even$ and Int

Finally, the set Int of timed words with integer times,

$$Int = \{(w_i, \tau_i)_{i \geq 0} \in (\Sigma \times \mathbf{N})^\infty \mid \tau_i \leq \tau_{i+1} \text{ for all } i \geq 0\}$$

is accepted by the timed automaton Int_1 in $\mathbf{TA}$, over $\mathbf{N}$, in Figure 2. It is also accepted by the timed automaton Int_2 in $\mathbf{TA}_\varepsilon$, over $\mathbf{T} = \mathbf{N}$ or $\mathbf{R}$, in Figure 2. With an argument similar to the one used in the proof of Proposition 1, it can be proved that Int does not belong to $\mathbf{TL}(\mathbf{R})$. Therefore, we obtain the strange result that the class $\mathbf{TL}(\mathbf{N})$ is not included in the class $\mathbf{TL}(\mathbf{R})$. The case of the classes $\mathbf{TL}_\varepsilon(\mathbf{N})$ and $\mathbf{TL}_\varepsilon(\mathbf{R})$ is much more satisfactory. Namely, for any timed automaton $\mathcal{A}$ in $\mathbf{TA}_\varepsilon$, the equality $L(\mathcal{A}, \mathbf{N}) = L(\mathcal{A}, \mathbf{T}) \cap Int$ holds. Since Int is in the class $\mathbf{TL}_\varepsilon$ and this class is closed under intersection, we deduce:

Proposition 2. *The class* $\mathbf{TL}_\varepsilon(\mathbf{N})$ *is contained in the class* $\mathbf{TL}_\varepsilon(\mathbf{R})$.

4 ε-transition without clock reset

We have proved in the previous section that transitions with non observable actions add power to timed automata (Proposition 1). The aim of this section is to show that ε-transitions add power to timed automata only if they reset clocks. More precisely, if $\mathcal{A} \in \mathbf{TA}_\varepsilon$ is a timed automaton such that no ε-transitions reset clocks then there exists a timed automaton without ε-transitions $\mathcal{B} \in \mathbf{TA}$ which accepts the same timed language.

Throughout this section we will denote by $\mathbf{RTA}_\varepsilon$ (for restricted $\mathbf{TA}_\varepsilon$) the class of timed automata with ε-transitions without clock reset and we will denote by $\mathbf{RTL}_\varepsilon$ the class of timed languages accepted by automata of $\mathbf{RTA}_\varepsilon$. Using these notations, the main theorem of this section is

Theorem 3. $\mathbf{RTL}_\varepsilon = \mathbf{TL}$

The proof of this theorem makes use of several involved constructions on timed automata and on clock constraints. In the next subsection, we define and study closures of clock constraints. This allows us to explain intuitively how ε-transitions without clock resets are removed from timed automata. Then we give the constructions on timed automata which deal with the general case.

4.1 Closures of clock constraints

The purpose of an ε-transition without clock reset is only to check that the clock constraint carried by the ε-transition is satisfied at some point of time. More precisely, let us consider the timed automaton $\mathcal{A}$ of $\mathbf{RTA}_\varepsilon$ in Figure 4. The ε-transition ensures that there exists some time between the execution of a and the execution of c where the clock constraint B holds. The main idea to remove this ε-transition is to shift this verification either forward to the c-transition or backward to the a-transition. In order to shift this verification, we will define and study closures of clock constraints. We start with the definition of forward closures of clock constraints. When n clocks are considered, a constraint can be represented by a subset of $\mathbf{R}_+^n$, called a region in this section. An example of a region and its forward closure is given in Figure 3.

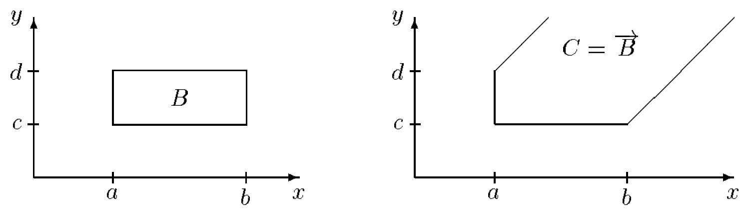

Fig. 3. A region and its forward closure

Definition 4. Let B be a clock constraint. The forward closure of B, denoted by $\overrightarrow{B}$, is a formula which is satisfied by a clock valuation v if the formula B is satisfied by the clock valuation $v - \delta$ for some $\delta \geq 0$:

$$v \models \overrightarrow{B} \quad \text{if} \quad \exists \delta \geq 0, v - \delta \models B$$

In order to remove the ε-transition of $\mathcal{A}$ we consider a new clock x_0 and the timed automaton $\mathcal{B}$ of Figure 4. In this simple case, it is not hard to check that $L(\mathcal{A}) = L(\mathcal{B})$. The general case will be proved in Proposition 12. Note that, the new clock x_0 and the associated constraint are necessary. The automaton $\mathcal{C}$ of

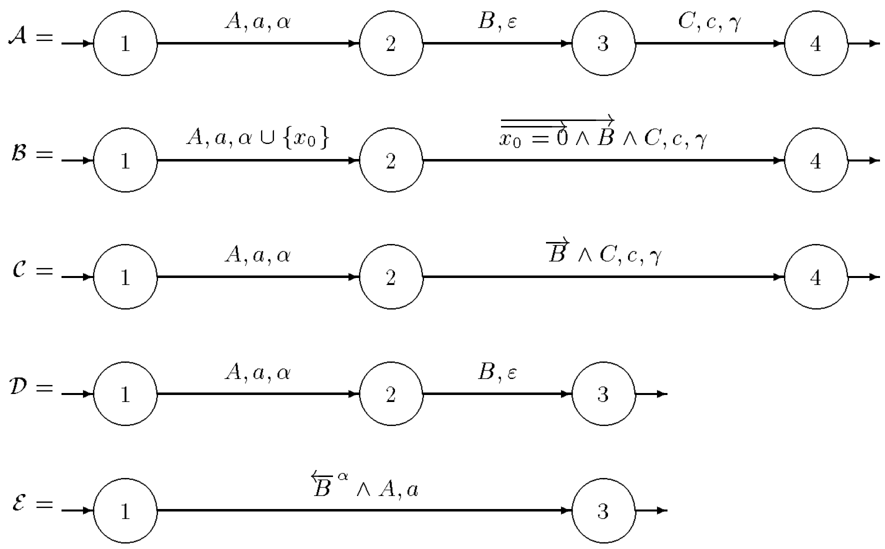

Fig. 4. Two automata in $\mathbf{RTA}_\varepsilon$ and their equivalent in **TA**.

Figure 4 may accept more timed word than the automaton $\mathcal{A}$. Indeed, assume that $A = (x > 3)$, $B = (x < 2)$, $C = true$ and $\alpha = \gamma = \emptyset$. Then, $\mathcal{A}$ do not accept any timed word while $L(\mathcal{C}) = \{(a, t_1)(c, t_2) \mid t_2 \geq t_1 > 3\}$ since the forward closure of B is $\overrightarrow{B} = true$.

The problem with the automaton $\mathcal{B}$ is that it uses a constraint which does not look like a clock constraint as defined in Section 2.1. Fortunately, by Proposition 5, this constraint can be replaced by an equivalent "regular" clock constraint, hence $\mathcal{B}$ can be viewed as a timed automaton without ε-transition which accepts the same timed language as the timed automaton $\mathcal{A}$ in $\mathbf{RTA}_\varepsilon$.

Proposition 5. *Let B be a clock constraint. There exists a clock constraint C equivalent to $\overrightarrow{B}$, i.e. such that for all clock valuations v,*

$$v \models \overrightarrow{B} \quad \textit{if and only if} \quad v \models C$$

Note that, if we reduce to clock constraints with rational constants, we could use the notion of clock regions [?] to prove this proposition. We give in [?] a general proof which holds whatever the clock constraints are and which does not depend on the classical region construction.

The forward closure of a clock constraint allows to shift the verification of the clock constraint carried by an ε-transition to the next transition. This method cannot be used when the ε-transition is the last one of a run. For instance, let $\mathcal{D}$ be the $\mathbf{RTA}_\varepsilon$ of Figure 4. In order to remove the ε-transition of $\mathcal{D}$ we must shift the verification of the constraint B backward to the a-transition. A region and its backward closure are presented in Figure 5.

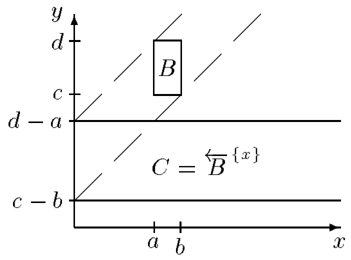

Fig. 5. A region and its backward closure

Definition 6. Let B be a clock constraint and α be a set of clocks. The backward closure of B with respect to α, denoted by $\overleftarrow{B}^{\alpha}$, is a formula which is satisfied by a clock valuation v if the formula B is satisfied by the clock valuation $v[\alpha \leftarrow 0]+\delta$ for some $\delta \geq 0$:

$$v \models \overleftarrow{B}^{\alpha} \quad \text{if} \quad \exists \delta \geq 0, v[\alpha \leftarrow 0] + \delta \models B$$

As above, it is not very hard to check in this simple case that the automaton $\mathcal{E}$ of Figure 4 recognizes the same timed language as the automaton $\mathcal{D}$. Again, by Proposition 7, the backward closure of a clock constraint with respect to a set of clocks is equivalent to a "regular" clock constraint. Hence, $\mathcal{E}$ is a timed automaton without ε-transitions which accepts the same timed language as the timed automaton $\mathcal{D}$ in $\mathbf{RTA}_{\varepsilon}$.

Proposition 7. *Let B be a clock constraint and let α be a set of clocks. There exists a clock constraint C equivalent to $\overleftarrow{B}^{\alpha}$, i.e. such that for all clock valuations v,*

$$v \models \overleftarrow{B}^{\alpha} \quad \textit{if and only if} \quad v \models C$$

4.2 First step : ε-circuits

The first step to remove the ε-transitions without reset in a timed automata consists in the suppression of ε-circuits. In the case of (infinite) untimed words, any ε-circuit can be simply erased provided it does not contain any repeated state. If such a circuit contains a repeated state, the property "repeated" has to be shifted to some suitable state before erasing the circuit. In the case of timed words, the situation is not so simple. For instance in the automaton $\mathcal{A}$ of Figure 6, it is not possible to transfer the "repeated" property from state 2 to state 1. Namely, the automaton $\mathcal{B}$ of Figure 6 accepts for instance the timed word $(a, i)_{i \geq 0}$ which does not belong to $L(\mathcal{A})$. In order to keep the conditions enforced by the ε-circuit, we unfold this circuit and we obtain the automaton $\mathcal{C}$ of Figure 6 for which it is easy to prove that $L(\mathcal{A}) = L(\mathcal{C})$. A generalization of this example will allow us to suppress all ε-circuits constituted of ε-transitions without reset in any timed automaton. Therefore, we will obtain

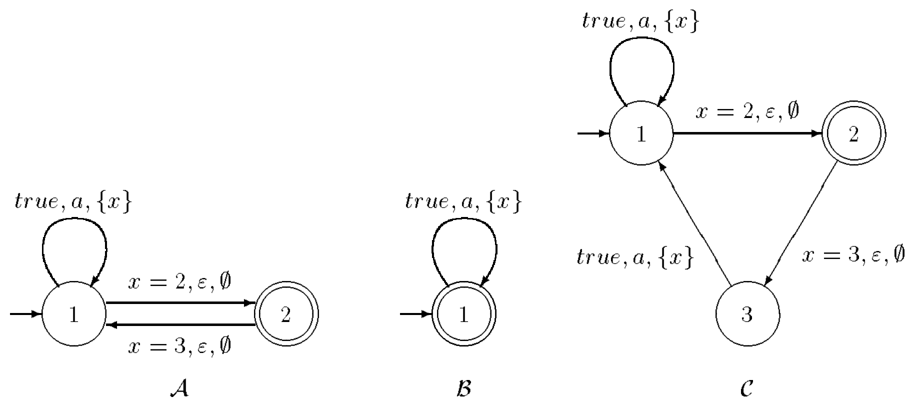

Fig. 6. Suppression of ε-circuit

Proposition 8. *Let $\mathcal{A}$ be an automaton in $\mathbf{RTA}_\varepsilon$. There exists an automaton $\mathcal{A}'$ in $\mathbf{RTA}_\varepsilon$ without ε-circuit which accepts the same timed language : $L(\mathcal{A}) = L(\mathcal{A}')$.*

4.3 Second step : ε-transitions

In this section, we give several constructions on timed automata which allow us to remove ε-transitions without clock resets. As sketched in Section 4.1, we will shift forward or backward the verification of clock constraints carried by ε-transitions and then we will remove ε-transitions from the automaton. But we first need to deal with a problem about repeated states. An infinite accepting run may reach repeated states only with ε-transitions. If we remove the ε-transitions, then all repeated states may become unreachable. Hence we will shift repeated states to states which are entered by visible transitions only.

To this purpose, the first construction will add two flags to the states of a timed automaton. The first flag will be set if the state is reached by an ε-transition and reset otherwise. The second flag will be set if a repeated state was taken since the previous visible transition. Furthermore, we will add new clocks associated with each state. Each clock will be reset when entering the corresponding state with a visible transition.

Throughout this section, $\mathcal{A} = (Q, \Sigma, X, T, I, F, R) \in \mathbf{RTA}_\varepsilon$ will denote an ε-circuit free (i.e. without ε-circuit) restricted timed automaton. Let $X_Q = \{x_q \mid q \in Q\}$ be a set of new clocks. We define a new automaton $\mathcal{A}_1 = (Q_1, \Sigma, X_1, T_1, I_1, F_1, R_1) \in \mathbf{RTA}_\varepsilon$ as follows. $Q_1 = Q \times \{0,1\}^2$, $X_1 = X \cup X_Q$, $I_1 = I \times \{(0,0)\}$, $F_1 = F \times \{0,1\}^2$, $R_1 = Q \times \{(0,1)\}$. If $p \xrightarrow{A,a,\alpha} q \in T$ is a visible transition of $\mathcal{A}$ then for all $e, r \in \{0,1\}$ we put in T_1 the transition $(p, e, r) \xrightarrow{A,a,\alpha\cup\{x_q\}} (q, 0, s)$ where $s = 1$ if $q \in R$ or $e = r = 1$ and $s = 0$ otherwise. Finally, if $p \xrightarrow{A,\varepsilon} q \in T$ is an ε-transition of $\mathcal{A}$ then for all $e, r \in \{0,1\}$ we put in T_1 the ε-transition $(p,e,r) \xrightarrow{A,\varepsilon} (q,1,s)$ where $s = 1$ if $q \in R$ or $e = r = 1$ and $s = 0$ otherwise.

Proposition 9. *$\mathcal{A}_1$ is an ε-circuit free restricted timed automaton and*

$$L(\mathcal{A}_1) = L(\mathcal{A}).$$

Note that this proposition remains true if $\mathcal{A} \in \mathbf{TA}_\varepsilon$ is an ε-circuit free timed automaton with ε-transitions. Indeed, in this case, we have to keep untouched the set of clocks reset by ε-transitions.

The next construction will shift the verification of clock constraints carried by ε-transitions to the next visible transition.

Let $p, q \in Q$ and $r, s \in \{0,1\}$, and let

$$P = (p,0,r) \xrightarrow{B_1,\varepsilon} (p_1,1,r_1) \cdots (p_{n-1},1,r_{n-1}) \xrightarrow{B_n,\varepsilon} (q,1,s) \tag{1}$$

be an ε-path from $(p,0,r)$ to $(q,1,s)$ in $\mathcal{A}_1$. Note that the length of this path is necessarily at least 1. In order to shift the verification of the constraints $B_1, \ldots, B_n$ to the next visible transition we will use the formula

$$\varphi(P) = \overrightarrow{\overrightarrow{\overrightarrow{x_p = 0} \wedge B_1} \cdots \wedge B_n}.$$

Since there may be several ε-pathes from $(p,0,r)$ to $(q,1,s)$ we will actually use a disjunction of such formulas. We denote by $\mathcal{P}(p,r,q,s)$ the set of ε-pathes from $(p,0,r)$ to $(q,1,s)$ in $\mathcal{A}_1$ and we define

$$\varphi(p,r,q,s) = \bigvee_{P \in \mathcal{P}(p,r,q,s)} \varphi(P).$$

Note that since $\mathcal{A}_1$ is ε-circuit free, the set $\mathcal{P}(p,r,q,s)$ is finite and $\varphi(p,r,q,s)$ is a finite disjunction.

We are now ready for the next construction which will only add new transitions to the automaton $\mathcal{A}_1$. Let $\mathcal{A}_2 = (Q_1, \Sigma, X_1, T_2, I_1, F_1, R_1)$ where

$$\begin{aligned} T_2 = T_1 \cup \{(p,0,r) &\xrightarrow{\varphi(p,r,q,s) \wedge A, a, \alpha} (m,0,t) \\ &\mid (p,0,r) \in Q_1 \text{ and } (q,1,s) \xrightarrow{A,a,\alpha} (m,0,t) \in T_1\}. \end{aligned}$$

Proposition 10. *$\mathcal{A}_2$ is an ε-circuit free restricted timed automaton and*

$$L(\mathcal{A}_2) = L(\mathcal{A}_1).$$

We will now shift the verification of the clock constraints of an ε-path which ends in a final state backward to the previous visible transition. As above, we need first to define auxiliary formulas. Let P be the path defined in Equation (1) and let α be a set of clocks, we define

$$\psi(P,\alpha) = \overleftarrow{\overrightarrow{\overrightarrow{\overrightarrow{x_p = 0} \wedge B_1} \cdots \wedge B_{n-1}} \wedge B_n}^{\alpha}$$

and

$$\psi(p, r, q, s, \alpha) = \bigvee_{P \in \mathcal{P}(p,r,q,s)} \psi(P, \alpha).$$

Again, since $\mathcal{A}_1$ is ε-circuit free, the formula $\psi(p, r, q, s, \alpha)$ is a finite disjunction.

We consider now a new dead final state σ and we will construct a new automaton from $\mathcal{A}_2$ by adding transitions to this new state. More precisely, let $\mathcal{A}_3 = (Q_3, \Sigma, X_1, T_3, I_1, F_3, R_1)$ where $Q_3 = Q_1 \cup \{\sigma\}$, $F_3 = F_1 \cup \{\sigma\}$ and

$$T_3 = T_2 \cup \{(m, 0, t) \xrightarrow{\psi(p,r,q,s,\sigma) \wedge A, a} \sigma \mid (q, 1, s) \in F_1 \text{ and } (m, 0, t) \xrightarrow{A, a, \alpha} (p, 0, r) \in T_2\}.$$

Proposition 11. $\mathcal{A}_3$ *is an* ε*-circuit free restricted timed automaton and*

$$L(\mathcal{A}_3) = L(\mathcal{A}_2).$$

It remains now to remove ε-transitions from $\mathcal{A}_3$ in order to get a timed automaton which recognizes the same language. Note that if we simply remove these transitions, then the states with a 1 in the second component will not be reachable any more.

In fact, there is still a last little technical problem. Namely, the empty word may be accepted by an ε-path in $\mathcal{A}_3$ and if we remove ε-transitions the empty word may not be accepted anymore. Fortunately, it is decidable whether or not the empty word is accepted by a timed automaton with ε-transitions. Thus, if the empty word is accepted by $\mathcal{A}_3$ we will put the dead state σ in the set of initial states.

Therefore, we define $\mathcal{A}_4 = (Q_4, \Sigma, X_1, T_4, I_4, F_4, R_1)$ where $Q_4 = Q \times \{0\} \times \{0, 1\} \cup \{\sigma\}$, T_4 is the restriction of T_3 to the new set of states Q_4, $I_4 = I_1 \cup \{\sigma\}$ if $\varepsilon \in L(\mathcal{A})$ and $I_4 = I_1$ otherwise, and $F_4 = F_3 \cap Q_4 = F \times \{0\} \times \{0, 1\} \cup \{\sigma\}$.

Proposition 12. $\mathcal{A}_4$ *is a timed automaton without* ε*-transitions and*

$$L(\mathcal{A}_4) = L(\mathcal{A}).$$