Intersection of regular signal-event (timed) languages

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

[1] LAMSADE, Université Paris Dauphine & CNRS,
Place du Maréchal de Lattre de Tassigny, F-75775 Paris Cedex 16, France
[2] LSV, ENS de Cachan & CNRS,
61 av. du Président Wilson, F-94235 Cachan Cedex, France

Abstract. We propose in this paper a construction for a "well known" result: regular signal-event languages are closed by intersection. In fact, while this result is indeed trivial for languages defined by Alur and Dill's timed automata (the proof is an immediate extension of the one in the untimed case), it turns out that the construction is much more tricky when considering the most involved model of signal-event automata. While several constructions have been proposed in particular cases, it is the first time, up to our knowledge, that a construction working on finite and infinite signal-event words and taking into account signal stuttering, unobservability of zero-duration τ-signals and Zeno runs is proposed.

1 Introduction

In order to accept timed words, the model of timed automata was first proposed in [1, 2]. It has been widely studied for the last fifteen years and successfully applied to industrial cases. For this model, an observation, called a time-event word, may be viewed as an alternating sequence of waiting times and instantaneous actions. A timed automaton is a finite automaton extended with variables called clocks, designed to recognize time-event words: time elapses while the control stays in a given node and an event is observed when a discrete transition occurs.

Another model was introduced by [3], and further studied in [10, 4, 11] with the aim of describing hardware systems. In this case, an observation is a signal word, i.e. a sequence of factors a^d, where a is a signal and d is its duration. The original model of timed automata was then modified to fit this setting: a signal is emitted while the automaton stays in some state and no event is produced when a discrete transition is fired. In this framework, when a transition occurs between two states with the same signal a, we obtain a^{d_1} followed by a^{d_2}, which are merged in $a^{d_1+d_2}$. This phenomenon is called stuttering.

It was noticed in [4] that both approaches are complementary and can be combined in an algebraic formalism to obtain the so-called signal-event monoid. Timed automata can be easily adapted to take both signals and events into account, thus yielding signal-event automata: states emit signals and transitions produce events.

We consider in this paper both finite and infinite behaviors of signal-event automata and we also include unobservable events (ε-transitions) and hidden signals (τ-labeled states). These features can be very useful and even necessary, for instance for handling abstractions [6]. They also allow us to get as special cases the initial models of timed automata and signal automata.

We study in this paper a construction for the intersection of languages accepted by signal-event automata. Surprisingly, it turns out that this closure property is rather difficult to obtain. Usually, the construction for intersection relies on a basic synchronized product. In [1], which deals with infinite time-event words only (no signal is involved), a Büchi-like product is performed. The situation is more complex for signal words due to stuttering of signals and unobservability of zero-duration signals. In [3], a construction is given for the intersection of signal automata, but neither signal stuttering nor unobservability of zero-duration signal is taken into account, and only finite runs are considered. Note that the full version [4] of [3] deals with the intersection of usual timed automata only. In [10], in order to obtain a determinization result, a construction is proposed to remove stuttering and zero-duration signals on signal automata using a single clock but intersection is not considered directly. In [11], stuttering is handled but intersection is done for signal automata acting on finite sequences only and without zero-duration signals. To cope with stuttering, intermediate states and ε-transitions are added to the automaton, thus introducing all possible ways of splitting some signal a^d into a finite concatenation $a^{d_1} \dots a^{d_n}$. When dealing with ω-sequences, this approach would produce additional Zeno runs leading to another difficulty arising with the possibility to accept a finite signal-event word of finite duration with either a finite run or an infinite Zeno run.

We provide a general construction for the intersection of signal-event timed automata working on finite and infinite signal-event words. We solve the main difficulties of signal stuttering, unobservability of zero-duration τ-signals and Zeno runs. Note that, although Zeno behaviours have been studied (see for instance [12, 9]), it has been sometimes argued that excluding Zeno runs directly from the semantics of timed automata is a realistic assumption, since they do not appear in "real" systems. However, the semantics of timed automata used by model-checking tools like UPPAAL do include Zeno runs while performing forward reachability analysis (this can be easily checked with an example). Hence, we think that the general theory should include Zeno runs as well.

We first give in Section 2 precise definitions of finite and infinite signal-event languages, with the corresponding notion of signal-event automata. Section 3 establishes a normal form for signal-event automata so that no infinite run accepts a finite word with finite duration and no finite run accepts a word with infinite duration. This normal form is useful for the general construction of intersection of signal-event automata dealing both with signal stuttering and with finite and infinite sequences proposed in Section 4.

For lack of space, this paper does not contain the proofs of the correctness of the different automata constructions we propose. These proofs are available in the technical report [7].

2 Signal-event words and signal-event automata

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. The set Z^∞ is equipped with the usual partial concatenation defined from $Z^* \times Z^\infty$ to Z^∞.

Throughout this paper, we consider a time domain $\mathbb{T}$ which can be either the set $\mathbb{N}$ of natural numbers, the set $\mathbb{Q}_+$ of non-negative rational numbers or the set $\mathbb{R}_+$ of non-negative real numbers and we set $\overline{\mathbb{T}} = \mathbb{T} \cup \{\infty\}$.

2.1 Signal-event words

We now describe the most general class of systems where both piecewise-constant signals and discrete events can occur, based on the signal-event monoid defined in [4]. We consider two finite alphabets Σ_e and Σ_s, with $\Sigma = \Sigma_e \cup (\Sigma_s \times \mathbb{T})$: an element in Σ_e is the label of an instantaneous event, while a pair $(a, d) \in \Sigma_s \times \mathbb{T}$, written a^d, associates a duration d with a signal a. Moreover, Σ_s includes the special symbol τ for an internal (or hidden) signal, the purpose of which is to represent a situation where no signal can be observed.

Intuitively, *signal-event words* (SE-words for short and sometimes called *timed words*) correspond to sequences obtained from Σ^∞ by merging consecutive identical signals and removing internal τ-signals with duration 0. But note that signals different from τ may have a null duration.

Formally, the partial monoid of signal-event words is the quotient $\Sigma^\infty / \approx$ where $\approx$ is the congruence (with respect to the partial concatenation on Σ^∞) generated by

$$\begin{cases} \tau^0 \approx \varepsilon & \text{and} \\ \prod_{i \in I} a^{d_i} \approx \prod_{j \in J} a^{d'_j} & \text{if } \sum_{i \in I} d_i = \sum_{j \in J} d'_j \end{cases}$$

where the index sets I and J above may be infinite. The partial monoid $\Sigma^\infty / \approx$ will be denoted $SE(\Sigma, \mathbb{T})$ or simply $SE(\Sigma)$ or SE when there is no ambiguity. We write a^∞ for the equivalence class of any sequence of the form $\prod_{i \geq 1} a^{d_i}$, where $\sum_{i \geq 1} d_i = \infty$. Note that for two words of the forms ua^d and $a^{d'}v$ with $d < \infty$, the concatenation is $ua^{d+d'}v$.

A finite or infinite sequence in $\Sigma^\infty \cup \Sigma^* \cdot (\Sigma_s \times \{\infty\})$ which does not contain τ^0 and such that two consecutive signals are distinct is said to be in *normal form* (NF). SE-words are often identified with sequences in normal form. A SE-word is *finite* if its normal form is a finite sequence (even if it ends with a^∞).

A duration can be associated with each element of Σ by: $\|a\| = 0$ if $a \in \Sigma_e$ and $\|a^d\| = d$ if $a \in \Sigma_s$ and $d \in \overline{\mathbb{T}}$, so that the duration of a sequence $w = s_1 s_2 \cdots$ in Σ^∞ is $\|w\| = \sum_{i \geq 1} \|s_i\| \in \overline{\mathbb{T}}$. Note that the duration restricted to finite SE-words with finite durations is a morphism from Σ^* into $(\mathbb{T}, +)$. A *Zeno* word is a SE-word with finite duration and whose normal form is infinite. A *signal-event language* (or *timed language*) is a set of SE-words.

Example 1. Let $\Sigma_e = \{f, g\}$ and $\Sigma_s = \{a, b\}$. The SE-word $w = a^3 ffg \tau^{4.5} a^1 b^5$ can be viewed as the following sequence of observations: first, the signal a during 3 time units, then a sequence of three instantaneous events ffg, then some unobservable signal during 4.5 time units, again the signal a during 1 time unit and then the signal b during 5 time units. The total duration of w is 13.5. For infinite SE-words, we have for instance: $a^3gfa^1\prod_{i\geq 1}a^2 \approx a^1a^2gf\prod_{i\geq 1}a^4$ and the normal form is written a^3gfa^∞. Note also that an infinite timed sequence in Σ^ω may be a finite *SE*-word with finite duration: $\prod_{i\geq 0}a^{1/2^i} \approx a^2$.

2.2 Signal-event (timed) automata

Our model of *signal-event automata* (also called *timed automata* in the sequel) is a variant of the basic models proposed in the literature, integrating both instantaneous and durational semantics: signals are associated with the control states, while instantaneous events occur when the system switches between two states.

Clocks and guards. Let X be a set of variables with values in $\mathbb{T}$, called clocks. The set $\mathcal{C}(X)$ of guards or clock constraints over X consists of conjunctions of atomic formulas $x \bowtie c$, for a clock x, a constant $c \in \mathbb{T}$ and a binary operator $\bowtie$ in $\{<, \leq, =, \geq, >\}$.

A clock valuation $v : X \rightarrow \mathbb{T}$ is a mapping that assigns to each clock x a time value $v(x)$. The set of all clock valuations is $\mathbb{T}^X$. We write $v \models g$ when the clock valuation v satisfies the clock constraint g. If t is an element of $\mathbb{T}$ and α a subset of X, the valuations $v + t$ and $v[\alpha]$ are defined respectively by $(v + t)(x) = v(x) + t$, for each clock x in X and $(v[\alpha])(x) = 0$ if $x \in \alpha$, and $(v[\alpha])(x) = v(x)$ otherwise.

Signal-event (timed) automata. A Büchi signal-event automaton over the time domain $\mathbb{T}$ is a tuple $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q_0, F, R, I, \ell, \Delta)$, where Σ_e and Σ_s are alphabets of events and signals, X is a finite set of $\mathbb{T}$-valued clocks, Q is a finite set of control states, $Q_0 \subseteq Q$ is a subset of initial states, $F \subseteq Q$ is a subset of final states and $R \subseteq Q$ corresponds to a Büchi acceptance condition. The mapping $I : Q \rightarrow \mathcal{C}(X)$ associates with a state $q \in Q$ an *invariant* $I(q)$ being a conjunction of constraints of the form $x \bowtie c$, with $\bowtie \in \{<, \leq\}$, and $\ell : Q \rightarrow \Sigma_s$ associates a signal with each state.

The set of transitions is $\Delta \subseteq Q \times \mathcal{C}(X) \times \Sigma_e \cup \{\varepsilon\} \times \mathcal{P}(X) \times Q$. A transition, also written $q \xrightarrow{g,a,\alpha} q'$, is labeled by a guard g, an instantaneous event in Σ_e or the unobservable event ε, and the subset α of clocks to be reset. When $a = \varepsilon$, it is called an ε-transition or a silent transition. Recall that, contrary to the untimed case, ε-transitions increase the expressive power of timed automata [5].

First examples of signal-event automata are given in Figure 1 (where double-circled nodes correspond to Büchi repeated states). The semantics of *SE*-automata will be given below. But intuitively,

- A *SE*-word is accepted by $\mathcal{A}_1$ if it is of the form $a^{d_1}b^{d_2}$ with $d_1 \geq 1$.
- A *SE*-word is accepted by $\mathcal{A}_2$ if it is of the form $a^{d_1}ca^{d_2}c\ldots$ with $d_i < 1$ for any i.
- Since the concatenation merges consecutive identical signals ($a^{d_1}a^{d_2} = a^{d_1+d_2}$), the language accepted by $\mathcal{A}_3$ consists of the signal a emitted for a duration $d \geq 1$.
- $\mathcal{A}_4$ accepts the signal a emitted for a duration $d \leq 1$ (note that a^1 is accepted by an infinite run with successive durations $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\ldots$ for instance).

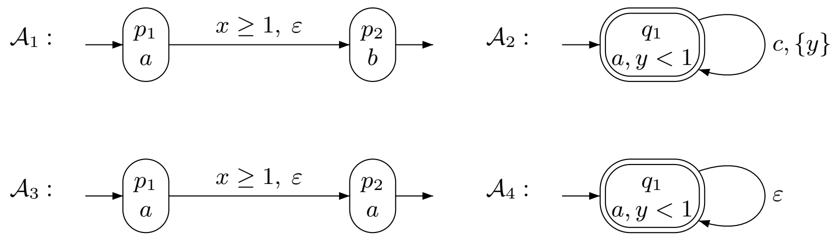

Fig. 1. Some signal automata.

Semantics. In order to define the semantics of SE-automata, we recall the notions of path and timed run through a path. A path in $\mathcal{A}$ is a finite or infinite sequence of consecutive transitions:

$$P = q_0 \xrightarrow{g_1,a_1,\alpha_1} q_1 \xrightarrow{g_2,a_2,\alpha_2} q_2 \ldots, \text{ where } (q_{i-1}, g_i, a_i, \alpha_i, q_i) \in \Delta,\ \forall i > 0$$

The path is said to be *accepting* if it starts in an initial state ($q_0 \in Q_0$) and *either* it is finite and ends in a final state, *or* it is infinite and visits infinitely often a *repeated* state $q \in R$. A *run* of the automaton through the path P is a sequence of the form:

$$\langle q_0, v_0 \rangle \xrightarrow{d_0} \langle q_0, v_0 + d_0 \rangle \xrightarrow{a_1} \langle q_1, v_1 \rangle \xrightarrow{d_1} \langle q_1, v_1 + d_1 \rangle \xrightarrow{a_2} \langle q_2, v_2 \rangle \ldots$$

where

- $d_i \in \mathbb{T}$ for $i \geq 0$ and if P is finite with n transitions then the last step of the run must be $\langle q_n, v_n \rangle \xrightarrow{d_n} \langle q_n, v_n + d_n \rangle$, with $d_n \in \overline{\mathbb{T}}$,
- $(v_i)_{i \geq 0}$ are clock valuations such that $v_0(x) = 0$ for all $x \in X$, and for each $i \geq 0$, we have

$$\begin{cases} v_i + d \models I(q_i), & \forall d \in [0, d_i] \\ v_i + d_i \models g_{i+1} \\ v_{i+1} = (v_i + d_i)[\alpha_{i+1}] \end{cases}$$

  Note that if d_i is finite, the condition about invariant $I(q_i)$ can be replaced simply by $v_i + d_i \models I(q_i)$.

The signal-event (timed) word generated by this run is simply (the equivalence class of) $\ell(q_0)^{d_0} a_1 \ell(q_1)^{d_1} a_2 \ell(q_2)^{d_2} \ldots$. The signal-event (timed) language accepted by $\mathcal{A}$ over the time domain $\mathbb{T}$ and the alphabet Σ, written $\mathcal{L}(\mathcal{A})$, is the set of SE-words generated by (finite or infinite) accepting runs of $\mathcal{A}$. Two automata $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.

The set of all signal-event (timed) automata is denoted by SEA_ε and the family of signal-event (timed) languages generated by some automaton in SEA_ε is denoted by SEL_ε.

Remark 1. A *Zeno run* is an infinite run for which the time sequence defined by $t_i = \sum_{j \leq i} d_j$ for $i \geq 0$, is convergent (keeping the notations just above). We did not include the non Zeno condition for runs, requiring that each infinite accepting run has an infinite duration. Thus, Zeno runs accepting finite words with finite duration may occur. Note that they also appear in the semantics of model-checking tools like UPPAAL, as can easily be checked with an example.

3 Normal forms

We propose in this section two technical results on the existence of "normal forms" for timed runs and signal-event automata. These existences will be crucial for the proof of the main result in the next section.

Recall that the normal form, obtained by merging consecutive identical signals and removing factors of the form τ^0, contains only visible events b in Σ_e and visible blocks a^d with either $d > 0$ or $a \neq \tau$. The *alternating normal form* (ANF) of a SE-word insists on a strict alternation between events and signals at the expense of keeping some invisible events (ε) and some invisible signals (τ^0).

More precisely, a sequence $w = a_0^{d_0} b_1 a_1^{d_1} b_2 \cdots$ over $\Sigma \cup \{\epsilon\}$ is in ANF if for all k we have $b_k = \varepsilon$ implies $a_{k-1} \neq a_k$ and $a_{k-1}^{d_{k-1}} \neq \tau^0 \neq a_k^{d_k}$.

Example 2. Let $\Sigma_e = \{f, g\}$ and $\Sigma_s = \{a, b\}$.

- $a^1 f a^3 g a^{2.5}$ is both in ANF and in NF,
- $f a^3 g f a^{2.5} b^4$ is in NF but not in ANF. Its ANF is $\tau^0 f a^3 g \tau^0 f a^{2.5} \varepsilon b^4$,
- $\tau^0 f \tau^0 g (a^2 \varepsilon b^3 \varepsilon)^\omega$ is in ANF but not in NF. Its NF is $fg(a^2 b^3)^\omega$.

Note that the ANF of a SE-word is unique. Indeed, assume that $w = e_0^{f_0} c_1 e_1^{f_1} c_2 \cdots$ is also in ANF. Assume that $a_0^{d_0} \neq e_0^{f_0}$. Either $a_0 = e_0$ and $d_0 < f_0$ and we must have $b_1 = \varepsilon$. We deduce that $a_1 \neq a_0$ and $a_0^{d_0} \neq \tau^0 \neq a_1^{d_1}$. A contradiction since in this case w cannot start simultaneously by $e_0^{f_0}$ and by $a_0^{d_0} b_1 a_1^{d_1}$. Or $a_0 \neq e_0$ and for instance $a_0 = \tau \neq e_0$. Then we must have $d_0 = 0$ and $b_1 = \varepsilon$, a contradiction with the ANF. Hence we have $a_0^{d_0} = e_0^{f_0}$. Assume now that $b_1 \neq c_1$. Then for instance $c_1 \neq \varepsilon$ and we must have $b_1 = \varepsilon$. Once again, this implies $a_1 \neq a_0$ and $a_0^{d_0} \neq \tau^0 \neq a_1^{d_1}$ leading to a contradiction as above. By induction, this shows that the ANF is unique.

We extend the ANF to runs and paths of a SE-automaton $\mathcal{B}$ as follows. Let $\rho = \langle p_0, v_0 \rangle \xrightarrow{d_0} \langle p_0, v_0 + d_0 \rangle \xrightarrow{b_1} \langle p_1, v_1 \rangle \xrightarrow{d_1} \cdots$ be a run for a SE-word w through some path $P = p_0 \xrightarrow{g_1, b_1, \alpha_1} p_1 \xrightarrow{g_2, b_2, \alpha_2} \cdots$ of $\mathcal{B}$. For the sake of simplicity, we assume that ρ and P are infinite. Our construction is similar when ρ and P are finite. For $0 \leq i < k \leq \infty$, we define $\ell(i,k) = \{\ell(p_j) \mid i \leq j < k \text{ and } \ell(p_j)^{d_j} \neq \tau^0\}$.

We build inductively the ANF of ρ and P and we simultaneously fix some notation. We start with $j_0 = 0$. Assume that $j_k < \infty$ has been defined. Intuitively, from the state p_{j_k}, we look for the first state where either an event $b \neq \epsilon$ is performed or a different signal is produced. Formally, we let $j_{k+1} = \inf\{j > j_k \mid b_j \neq \varepsilon \text{ or } |\ell(j_k, j+1)| = 2\}$ with the convention $\inf(\emptyset) = \infty$. Let P_k be the subpath of P starting at p_{j_k} and ending at p_{n-1} if $n = j_{k+1} < \infty$ and similarly, let ρ_k be the subrun of ρ starting at

$\langle p_{j_k}, v_{j_k}\rangle$ and ending at $\langle p_{n-1}, v_{n-1}+d_{n-1}\rangle$ if $n = j_{k+1} < \infty$. Let also $a_k = \tau$ if $\ell(j_k, j_{k+1}) = \emptyset$ and $\{a_k\} = \ell(j_k, j_{k+1})$ otherwise. Finally, let $D_k = \sum_{j_k \le j < j_{k+1}} d_j$. For $j_k < j < j_{k+1}$ we have $b_j = \varepsilon$ and for $j_k \le j < j_{k+1}$ we have either $\ell(p_j) = a_k$ or $\ell(p_j) = \tau$ and $d_j = 0$. Hence, ρ_k is a run for $a_k^{D_k}$ through P_k.

By construction, we have $P = P_0 \xrightarrow{g_{j_1}, b_{j_1}, \alpha_{j_1}} P_1 \xrightarrow{g_{j_2}, b_{j_2}, \alpha_{j_2}} P_2 \cdots$, and $\rho = \rho_0 \xrightarrow{b_{j_1}} \rho_1 \xrightarrow{b_{j_2}} \rho_2 \cdots$ which are the ANF of P and ρ respectively. We also have $w = a_0^{D_0} b_{j_1} a_1^{D_1} b_{j_2} a_2^{D_2} \cdots$ and this is the ANF of w. Indeed, if $b_{j_k} = \varepsilon$ then $|\ell(j_{k-1}, j_k + 1)| = 2$ hence $a_{k-1}^{D_{k-1}} \neq \tau^0$, $\ell(p_{j_k})^{d_{j_k}} \neq \tau^0$ and $\ell(p_{j_k}) \neq a_{k-1}$. We deduce that $a_k = \ell(p_{j_k}) \neq a_{k-1}$ and $a_k^{D_k} \neq \tau^0$.

A normal form for signal-event automata. We show how to transform a signal-event automaton into an equivalent one, in which finite accepting runs correspond exactly to finite words with finite duration. The result is interesting in itself for implementation issues: when a finite word with finite duration is accepted by an infinite run, we build instead a finite accepting run for this word. Furthermore, conditions (†) and (‡) below will be used in the next section for the intersection construction.

Note that the transformation removes a particular type of Zeno runs, those which contain ultimately only ε-transitions and a single signal. But it keeps Zeno runs corresponding to infinite words of finite duration.

Theorem 1. *Let $\mathcal{A}$ be a SE-automaton. We can effectively construct an equivalent SE-automaton $\mathcal{A}'$ such that:*

(†) *no infinite run of $\mathcal{A}'$ accepts a finite word with finite duration, and*
(‡) *no finite run of $\mathcal{A}'$ accepts a word with infinite duration.*

We start from an automaton $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q_0, F, R, I, \ell, \Delta)$. We first construct an automaton satisfying condition (†). We need to distinguish whether time progresses infinitely often or not.

First step. We first suppress infinite runs where time does not progress infinitely often. Such a run generates a word of the form wa^h with $h < \infty$ and loops eventually through some repeated state r. The loop is constituted of ε-transitions and states of label a or τ, all crossed instantaneously. The intuitive idea is to detect such a loop and to replace it by a single transition to a new final state. More formally, for any signal a, we build an automaton $\mathcal{A}(a, 0)$ which contains the states and transitions of $\mathcal{A}$ together with some new states and transitions described below (see Fig. 2 for an intuitive view).

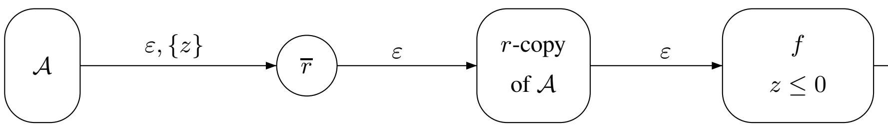

Fig. 2. The automaton $\mathcal{A}(a, 0)$

Let $z \notin X$ be a new clock. The automaton $\mathcal{A}(a,0)$ contains the states and transitions of $\mathcal{A}$. Moreover, for each $(p,g,\varepsilon,\alpha,q) \in \Delta$ with $\{\ell(p),\ell(q)\} \subseteq \{a,\tau\}$, we add the following states and transitions:

$$(1) \qquad \begin{array}{ll} (p,g,\varepsilon,\alpha \cup \{z\},\overline{q}) & \text{if } q \text{ is a repeated state of } \mathcal{A} \\ (\overline{p},g,\varepsilon,\alpha,(q,p)) & \\ ((p,r),g,\varepsilon,\alpha,(q,r)) & \end{array}$$

For the new states, we let $\ell(\overline{r}) = \ell(r)$, $I(\overline{r}) = I(r)$, $\ell(p,r) = \ell(p)$, $I(p,r) = I(p)$. These new transitions simulate a loop around some repeated state r: just before reaching r, we move into the copy and reach $\overline{r}$ instead. We remember r until reaching (r,r), where we know that the loop has been successfully completed. Therefore, we also add the transitions $((r,r),\varepsilon,f)$ where f is a new state which is the only final state of $\mathcal{A}(a,0)$, with $\ell(f) = \tau$ and $I(f) = z \leq 0$. The invariant of f ensures that the states of the form $\overline{r}$ or (p,q) have been crossed instantaneously.

The automaton $\mathcal{A}(a,0)$ has no repeated states and its initial states are those of $\mathcal{A}$.

Second step. We now treat the case of Zeno runs where time progresses infinitely often. As in the first step, the idea is to introduce copies of $\mathcal{A}$ in which we unfold once a loop around some repeated state to check if it can be taken infinitely often within finite time. But the construction is much more tricky since timing constraints have to be verified.

Let wa^d be a finite word with finite duration accepted by $\mathcal{A}$ with some infinite run ρ through some infinite path P and such that time progresses infinitely often in ρ. We denote by $\mathcal{L}_f(\mathcal{A},a)$ the set of such words. Let Y be the set of clocks that are not reset infinitely often in P. For each $y \in Y$, let $m_y = \sup\{c \mid y > c \text{ or } y \geq c \text{ occurs infinitely often in the guards of } P\}$ with the convention $\sup\emptyset = 0$. Similarly, for each $y \in Y$, let $M_y = \inf\{c \mid y < c \text{ or } y \leq c \text{ occurs infinitely often in the guards or invariants of } P\}$ with the convention $\inf\emptyset = \infty$. Finally, we let $m = (m_y)_{y \in Y}$ and $M = (M_y)_{y \in Y}$. Note that all transitions (p,g,b,α,q) used infinitely often in P satisfy

(2)
- $\ell(p),\ell(q) \in \{a,\tau\}$, $b = \varepsilon$ and $\alpha \cap Y = \emptyset$,
- $\forall y \in Y$, if $y > c$ or $y \geq c$ is a constraint in g then $m_y \geq c$,
- $\forall y \in Y$, if $y < c$ or $y \leq c$ is a constraint in g, $I(p)$ or $I(q)$ then $M_y \leq c$,
- $\forall x \in X \setminus Y$, if $x > c$ or $x \geq c$ is a constraint in g then $c = 0$.

The last condition holds since wa^d has a finite duration.

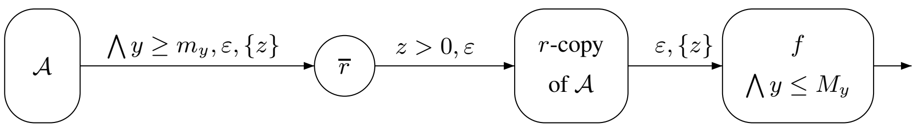

Fig. 3. The automaton $\mathcal{A}(a,Y,m,M)$

We build an automaton $\mathcal{A}(a,Y,m,M)$ depending only on (a,Y,m,M) which accepts wa^d with a finite run (note that the number of distinct tuples (a,Y,m,M) is finite). Let $z \notin X$ be a new clock. The automaton $\mathcal{A}(a, Y, m, M)$ contains the states and transitions of $\mathcal{A}$ together with some new states and transitions described below (see Fig. 3 for an intuitive view). It has no repeated states and its initial states are those of $\mathcal{A}$. Let $(p, g, \varepsilon, \alpha, q) \in \Delta$ be a transition of $\mathcal{A}$ satisfying (2). We add the following states and transitions:

$$
\begin{array}{lll}
 & (p, g \wedge \bigwedge_{y \in Y} y \geq m_y, \varepsilon, \alpha \cup \{z\}, \overline{q}) & \text{if } \ell(q) = a, \\
(3) & (\overline{p}, g \wedge z > 0, \varepsilon, \alpha \cup \{z\}, (q, p, \alpha, (q \in R))) & \text{if } \ell(p) = a, \\
 & ((p, r, \beta, \varphi), g, \varepsilon, \alpha \cup \{z\}, (q, r, \alpha \cup \beta, \varphi \vee (q \in R))) &
\end{array}
$$

For the new states, we let $\ell(\overline{r}) = a$, $I(\overline{r}) = I(r)$, $\ell(p, r, \beta, \varphi) = \ell(p)$, $I(p, r, \beta, \varphi) = I(p)$ if $\ell(p) = a$ and $I(p, r, \beta, \varphi) = I(p) \wedge z \leq 0$ otherwise. These new transitions simulate a loop around some repeated state r: just before reaching r, we move into the copy and reach $\overline{r}$ instead, while satisfying the lower constraints. We remember r until reaching $(r, r, X \setminus Y, \texttt{true})$, where we know that the loop has successfully terminated because we crossed some repeated state ($(q \in R)$). Therefore, we also add the transitions $((r, r, X \setminus Y, \texttt{true}), \texttt{true}, \varepsilon, \{z\}, f)$ where f is a new state which is the only final state of $\mathcal{A}(a, Y, m, M)$ and with $\ell(f) = a$ and $I(f) = \bigwedge_{y \in Y} y \leq M_y$ (with the convention that $y \leq \infty$ is $\texttt{true}$).

Third step. The resulting automaton $\mathcal{A}_1$ is the disjoint union of all automata $\mathcal{A}(a, 0)$ and $\mathcal{A}(a, Y, m, M)$. All finite words with finite duration accepted by $\mathcal{A}$ can be accepted by finite runs of $\mathcal{A}_1$ and $\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A})$. To obtain an equivalent *SE*-automaton $\mathcal{A}_2$ satisfying the condition (†), it remains to keep only the infinite runs of $\mathcal{A}$ which accept words with either an infinite duration or an infinite length. For this, we define an automaton $\mathcal{B}$ which goes to a repeated state whenever at least one time unit has elapsed or when a visible event is executed or when a *new* signal is emitted. The first two conditions are trivial to deal with. For the last one we need to keep track of the last signal that has been observed (a^d with $a \neq \tau$ or $d > 0$) so that we can enter a repeated state when a *new* signal is observed (b^δ with $b \neq a$ and $b \neq \tau$ or $\delta > 0$). The automaton $\mathcal{A}_2$ is obtained as a cartesian product of $\mathcal{A}_1$ and $\mathcal{B}$. Note that we cannot use an intersection operation since Theorem 1 is used in the proof of Theorem 2.

Fourth step. Finally, it remains to transform $\mathcal{A}_2$ into an automaton $\mathcal{A}'$ satisfying also condition (‡). The only problem comes from final states whose invariant is $\texttt{true}$. Words of the form wa^∞ accepted by finite runs ending in such states must now be accepted by infinite runs. The idea is to use again the new clock z to measure time intervals of length one. For each signal $a \in \Sigma_s$, we add a new repeated state r_a with label a and invariant $z \leq 1$. We also add loops $(r_a, z = 1, \varepsilon, \{z\}, r_a)$ and for each final state p with label a and invariant $\texttt{true}$ we add the transition $(p, \texttt{true}, \varepsilon, \{z\}, r_a)$ and p is not final anymore. This gives the automaton $\mathcal{A}'$ satisfying both conditions (†) and (‡) and concludes the construction. □

Remark 2. If Zeno runs are not allowed (see Remark 1), condition (†) is true by definition of an accepted run. Hence the construction of an automaton satisfying Theorem 1 reduces to Fourth step above and is therefore much simpler.

In the same way, if ε-transitions are not allowed, an infinite run can accept only an infinite word and Theorem 1 reduces to condition (‡).

4 Intersection

We present in this section the main construction of this paper.

Theorem 2. *The class SEL_ε is closed under intersection.*

Note that one of the problems arising in the construction of the intersection comes from the fact that a word can be accepted in two different automata by a finite and an infinite run respectively. For instance, consider the two automata $\mathcal{A}_3$ and $\mathcal{A}_4$ in Figure 1. We have $\mathcal{L}(\mathcal{A}_3) = \{a^d \mid d \geq 1\}$ and $\mathcal{L}(\mathcal{A}_4) = \{a^d \mid d \leq 1\}$, so that $\mathcal{L}(\mathcal{A}_3) \cap \mathcal{L}(\mathcal{A}_4) = \{a^1\}$. And this word a^1 is accepted in $\mathcal{A}_3$ by a finite run and in $\mathcal{A}_4$ by an infinite run. We will then use in a crucial way the normal form proposed by Theorem 1.

Before giving the general construction, let us point out some other difficulties. The treatment of visible events is easy and will be done like in the untimed case through a synchronized product. The case of signals is more tricky and needs more attention. Indeed, let us consider the following example:

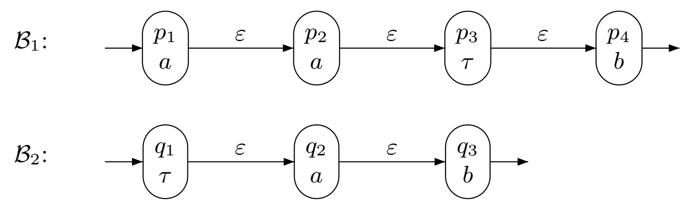

Fig. 4. Automata $\mathcal{B}_1$ and $\mathcal{B}_2$

If automaton $\mathcal{B}_1$ is in state p_1 and automaton $\mathcal{B}_2$ in state q_3, they can not compute anymore a SE-word which would be in the intersection of their languages. Indeed this word should have at the same time a factor a^d with $d \geq 0$ and a factor b^δ with $\delta \geq 0$, which is not possible since a and b are different from τ.

If automaton $\mathcal{B}_1$ is in state p_1 and automaton $\mathcal{B}_2$ in state q_2, they can both produce a signal a.

Now if automaton $\mathcal{B}_1$ is in state p_1 and automaton $\mathcal{B}_2$ in state q_1, whereas the labels of the two states are different, it is still possible to produce a word of the intersection. Indeed, it is sufficient to force $\mathcal{B}_2$ to leave immediately q_1 (i.e. to stay in q_1 0 time unit), $\mathcal{B}_2$ will thus produce a signal $\tau^0 \approx \varepsilon$ and thus not visible. This last case shows that we should allow in the intersection *asynchronous* moves where only one of the automata executes an ε-transition.

We now proceed to the construction of a SE-automaton accepting the intersection of the languages recognized by two automata

$$\mathcal{A}_j = (\Sigma_e, \Sigma_s, X_j, Q_j, Q_j^0, F_j, R_j, I_j, \ell_j, \Delta_j)$$

for $j = 1, 2$ on the same alphabet, satisfying the conditions (†) and (‡) of Theorem 1. We assume that Q_1 and Q_2 (respectively X_1 and X_2) are disjoint and, when no confusion can arise, we simply write ℓ for both labelling functions ℓ_1 and ℓ_2. We also assume that the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ do not contain a trivial loop of the form $(p, \mathtt{true}, \varepsilon, \emptyset, p)$.

We define the automaton $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q^0, F, R, I, \ell, \Delta)$ designed to accept $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$ as follows.

– We set $X = X_1 \cup X_2 \cup \{z\}$, where z is a new clock used to control if the time elapsed in a state of $\mathcal{A}$ is zero or not.
– The set $Q \subseteq \Sigma_s \times Q_1 \times Q_2 \times \{0, 1, 2\}$ consists of all tuples (a, p, q, i) satisfying
  • $\ell_1(p), \ell_2(q) \in \{a, \tau\}$ and
  • $i = 1$ if and only if $\ell_1(p) = \ell_2(q) = a$.

  Note that, the conjunction of these two constraints implies that if the first component is $a = \tau$ then the last component must be $i = 1$.
– For $(a, p, q, i) \in Q$, we set
  • $\ell(a, p, q, i) = a$ and $I(a, p, q, i) = I_1(p) \wedge I_2(q)$ if $i = 1$ and
  • $\ell(a, p, q, i) = \tau$ and $I(a, p, q, i) = I_1(p) \wedge I_2(q) \wedge z \leq 0$ otherwise.

The intuitive idea behind the fourth component of the states of $\mathcal{A}$ is the following:

– Value 0 means that one of the automata is ready to perform some signal $a \neq \tau$ and is waiting for the other to reach a state labelled a with ε-transitions and instantaneously traversing τ-labelled states. If a synchronization is not possible on signal a, then the whole computation will not produce any accepting SE-word of the intersection,
– Value 1 means that the two automata emit the same signals,
– Value 2 means that the two automata were producing the same signals but have "lost" their synchronisation (due an ε-transition performed by one of them). As in the case of value 0, they will try to re-synchronize. But the whole computation can still progress even if this synchronization is not possible anymore.

The transition relation Δ consists of *synchronous* moves where both automata progress simultaneously and of *asynchronous* moves where one automaton is idle while the second one performs an ε-transition.

A synchronous move is not possible in a state of the form $(a, p, q, 0)$ since a synchronization is expected first. Consider two states (a, p, q, i) and (a', p', q', i') in Q with $i \neq 0$ and $i' \neq 2$. For any two transitions $\delta_1 = (p, g, b, \alpha, p') \in \Delta_1$ and $\delta_2 = (q, h, b, \beta, q') \in \Delta_2$ with $b \neq \varepsilon$ if $a = a'$, we add in Δ the synchronous transition

$$\delta = (a, p, q, i) \xrightarrow{g \wedge h, b, \alpha \cup \beta \cup \{z\}} (a', p', q', i')$$

and we set $\pi_j(\delta) = \delta_j$ for $j = 1, 2$.

Consider now a state $(a, p, q, i) \in Q$. For any transition $\delta_1 = (p, g, \varepsilon, \alpha, p') \in \Delta_1$ with $\ell_1(p') \in \{a, \tau\}$ we add in Δ the asynchronous transition

$$\delta = (a, p, q, i) \xrightarrow{g, \varepsilon, \alpha \cup \{z\}} (a, p', q, i')$$

where i' is updated so that (a, p', q, i') is a legal state and the choice between values 0 and 2 is made according to the abstract description in Fig. 5. Formally, if $a = \tau$ then $i' = 1$ is the only possibility. Now, if $a \neq \tau$ we have the following cases:

- $i' = 1$ if $\ell_1(p') = \ell_2(q) = a$: synchronization on a is active,
- $i' = 0$ if $i = 0$ and ($\ell_1(p') = \tau$ or $\ell_2(q) = \tau$): synchronization on a has not yet been achieved,
- $i' = 2$ if $i \neq 0$ and ($\ell_1(p') = \tau$ or $\ell_2(q) = \tau$): synchronization on a has been lost.

We set $\pi_1(\delta) = \delta_1$ and $\pi_2(\delta) = \varepsilon$. We proceed symmetrically for asynchronous transitions of $\mathcal{A}_2$.

In the construction above, the subset of states with first component a is designed to handle maximal blocks of the form a^d. This part of the intersection is represented for $a \neq \tau$ by the abstract automaton in Figure 5. Note that all the transitions are asynchronous ε-transitions which reset the clock z.

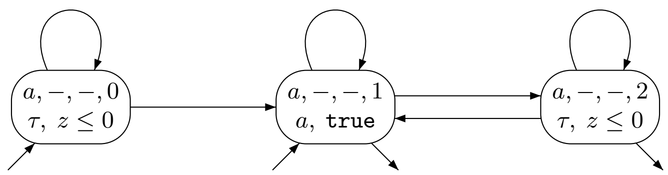

Fig. 5. Handling blocks a^d, for $a \neq \tau$

Since we have assumed that $\mathcal{A}_1$ and $\mathcal{A}_2$ do not contain a trivial loop of the form $(p, \mathtt{true}, \varepsilon, \emptyset, p)$, the projections $\pi_j(\delta)$ for $j = 1, 2$ are well-defined. Indeed, if $\delta = ((a, p_1, p_2, i), g, b, \alpha \cup \{z\}, (a', q_1, q_2, i')) \in \Delta$ then g is of the form $g_1 \wedge g_2$ where g_j involves clocks of $\mathcal{A}_j$ only. Hence, if we let $\alpha_j = \alpha \cap X_j$ we get $\pi_j(\delta) = \varepsilon$ if ($g_j = \mathtt{true}$ and $b = \varepsilon$ and $\alpha_j = \emptyset$ and $q_j = p_j$) and $\pi_j(\delta) = (p_j, g_j, b, \alpha_j, q_j)$ otherwise.

A path P of $\mathcal{A}$ can be seen as a sequence $\delta_1\delta_2\cdots$ of transitions in Δ. Clearly, the projection $\pi_j(P) = \pi_j(\delta_1)\pi_j(\delta_2)\cdots$ is a path of $\mathcal{A}_j$.

The initial and final states are defined by $Q^0 = Q \cap (\Sigma \times Q_1^0 \times Q_2^0 \times \{0, 1\})$ and $F = Q \cap (\Sigma \times F_1 \times F_2 \times \{1, 2\})$. We will not define the repeated states R explicitely. Instead, an infinite run P of $\mathcal{A}$ will be accepting if and only if each projection $\pi_j(P)$ is infinite and accepting in $\mathcal{A}_j$. It is well-known how to turn this intersection of Büchi conditions into a Büchi condition using some additional information [13]. For the sake of simplicity, we skip this easy construction here.

Note that, since conditions (†) and (‡) hold for $\mathcal{A}_1$ and $\mathcal{A}_2$, they also hold for $\mathcal{A}$.

Examples The next easy examples illustrate the construction and the usefulness of the additional components a and i. Consider the two automata $\mathcal{B}_1$ and $\mathcal{B}_2$ in Figure 4, which have only finite runs and thus satisfy condition (†). We could easily ensure that they also satisfy the condition (‡) by adding invariants in the final states, which is omitted for simplicity. Recall that in our model, the signal τ^0 is equivalent to the empty word ε. Consequently, the language accepted by $\mathcal{B}_1$ is $\{a^{d_1}\tau^{d_2}b^{d_3} \mid d_1, d_3 \geq 0, d_2 >$

$0\} \cup \{a^{d_1}b^{d_3} \mid d_1, d_3 \geq 0\}$ while $\mathcal{B}_2$ accepts $\{\tau^{d_1}a^{d_2}b^{d_3} \mid d_1 > 0, d_2, d_3 \geq 0\} \cup \{a^{d_2}b^{d_3} \mid d_2, d_3 \geq 0\}$. Hence $L(\mathcal{B}_1) \cap L(\mathcal{B}_2) = \{a^{d_1}b^{d_3} \mid d_1, d_3 \geq 0\}$. A word is in the intersection (if and) only if the states labeled by τ are crossed instantaneously by an accepting run.

The automaton $\mathcal{B}$ constructed for the intersection is represented in Figure 6. All transitions are ε-transitions which reset the clock z.

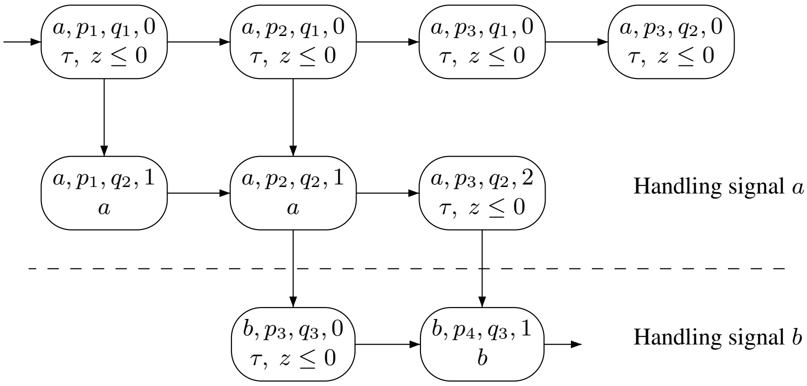

Fig. 6. Resulting automaton $\mathcal{B}$

We now modify automata $\mathcal{B}_1$ and $\mathcal{B}_2$ into $\mathcal{B}_1'$ and $\mathcal{B}_2'$ by adding loops, as represented in Figure 7. In this case, the words in the intersection may contain factors of the form τ^d, as can be seen on the resulting automaton $\mathcal{B}'$ in Figure 8.

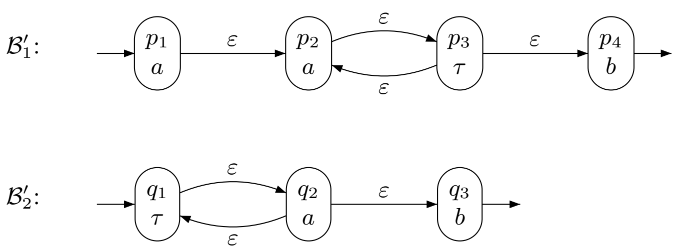

Fig. 7. Automata $\mathcal{B}_1'$ and $\mathcal{B}_2'$

We insist that the construction relies on condition (†). Consider again the two automata $\mathcal{A}_3$ and $\mathcal{A}_4$ in Figure 1. Condition (†) does not hold for $\mathcal{A}_4$, because a^1 is accepted by an infinite run. We have $\mathcal{L}(\mathcal{A}_3) \cap \mathcal{L}(\mathcal{A}_4) = \{a^1\}$. Note that the construction given in the proof of Theorem 2 would fail in this case since it would yield the automaton $\mathcal{A}$ in Figure 9. We have $\mathcal{L}(\mathcal{A}) = \emptyset$ for two reasons. First $\mathcal{A}$ contains no final state and it admits also no accepting infinite run since the first projection of the run cannot be

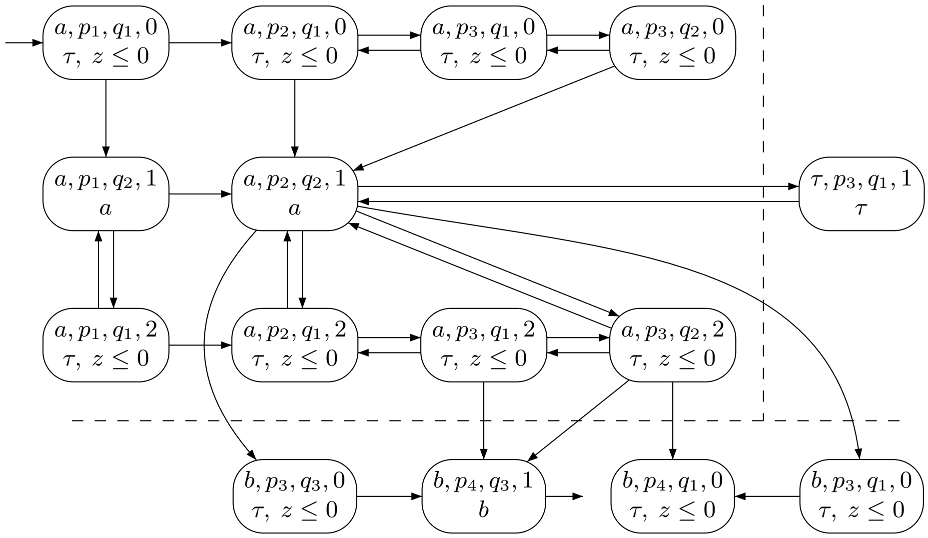

Fig. 8. Resulting automaton $\mathcal{B}'$

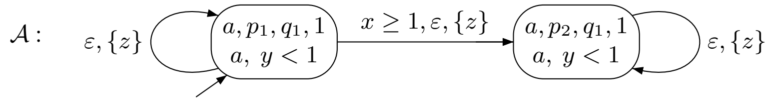

Fig. 9. Resulting automaton $\mathcal{A}$

infinite. This is not actually the main problem. We could have defined $\mathcal{A}$ so that a path P is accepting if and only if both projections $\pi_1(P)$ and $\pi_2(P)$ are accepting (finite or not). Then the argument above does not apply anymore. Still we would have $\mathcal{L}(\mathcal{A}) = \emptyset$ due to the invariant $y < 1$ and the guard $x \geq 1$.

Remark 3. If we consider signal-event automata where ε-transitions are not allowed, the treament of the intersection becomes much simpler. Indeed, the intersection of two SE-automata without ε-transitions can be done in a classical way, i.e., as a product of automata and a suitable treatment of Büchi conditions.

5 Conclusion

We proposed in this paper a construction for the intersection of two signal-event automata in the most general framework, working on finite and infinite signal-event words and taking into account signal stuttering, unobservability of zero-duration τ-signals and Zeno runs.

While constructions were proposed in the literature for important particular cases, it is the first time, up to our knowledge, that the general case is treated. There has been in the area of timed automata some examples of subtly erroneous constructions (e.g. with respect to forward analysis [8]) which should convince us of the importance to publish complete and proved constructions.

Moreover, it turns out that the closure of signal-event automata under intersection, and the normal form achieved in Theorem 1, are crucial to study the closure of SE-languages recognized by such automata under timed substitutions [6].

References

1. R. Alur and D.L. Dill. Automata for modeling real-time systems. In *Proceedings of ICALP'90*, number 443 in LNCS, pages 322–335. Springer, 1990.
2. R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
3. E. Asarin, P. Caspi, and O. Maler. A Kleene theorem for timed automata. In *Proceedings of LICS'97*, pages 160–171. IEEE Comp. Soc. Press, 1997.
4. E. Asarin, P. Caspi, and O. Maler. Timed regular expressions. *Journal of the ACM*, 49(2):172–206, 2002.
5. B. Bérard, V. Diekert, P. Gastin, and A. Petit. Characterization of the expressive power of silent transitions in timed automata. *Fundamenta Informaticae*, 36:145–182, 1998.
6. B. Bérard, P. Gastin, and A. Petit. Refinements and abstractions of signal-event (timed) languages. In *Proceedings of FORMATS'06*, number ??? in LNCS, pages ??–??. Springer, 2006.
7. B. Bérard, P. Gastin, and A. Petit. Timed substitutions for regular signal-event languages. Research Report LSV-06-04, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2006.
8. P. Bouyer. Forward analysis of updatable timed automata. *Formal Methods in System Design*, 24(3):281–320, May 2004.
9. P.J.L. Cuijpers, M.A. Reniers, and A.G. Engels. Beyond zeno-behaviour. Technical Report CSR 01-04, Department of Computing Science, University of Technology, Eindhoven, 2001.
10. C. Dima. Real-Time Automata and the Kleene Algebra of Sets of Real Numbers. In *Proceedings of STACS'2000*, number 1770 in LNCS, pages 279–289. Springer, 2000.
11. J. Durand-Lose. A Kleene theorem for splitable signals. *Information Processing Letters*, 89:237–245, 2004.
12. M.R. Hansen, P.K. Pandya, and C. Zhou. Finite divergence. *Theoretical Computer Science*, 138:113–139, 1995.
13. D. Perrin and J.-E. Pin. *Infinite words*. Elsevier, 2004.