Timed substitutions for regular signal-event 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. In the classical framework of formal languages, a refinement operation is modeled by a substitution and an abstraction by an inverse substitution. These mechanisms have been widely studied, because they describe a change in the specification level, from an abstract view to a more concrete one, or conversely. For timed systems, there is up to now no uniform notion of substitution. In this paper, we study timed substitutions in the general framework of signal-event languages, where both signals and events are taken into account. We prove that regular signal-event languages are closed under substitution and inverse substitution.

To obtain these results, we use in a crucial way a "well known" result: regular signal-event languages are closed under intersection. In fact, while this result is indeed easy for languages defined by Alur and Dill's timed automata, it turns out that the construction is much more tricky when considering the most involved model of signal-event automata. We give here 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. Note that if several constructions have been proposed in particular cases, it is the first time that a general construction is provided.

1 Introduction

Refinements and abstractions. Operations of refinements and abstractions are essential tools for the design and study of systems, real-time or not. They allow to consider a given system at different abstraction levels. For instance, some procedure or function can simply be viewed at some abstract level as a single action, and can be later expanded into all its possible behaviours at some more concrete level. Or conversely, a set of behaviours are merged together and replaced by a single action, in order to obtain a more abstract description.

These operations can be formally described by substitutions and inverse substitutions respectively. This is the reason why substitutions have been extensively studied in the untimed framework, with the underlying idea that interesting classes of languages have to be closed under these operations.

Timed languages. In order to represent timed observations, 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 events (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 in [3], and further studied in [11, 4, 12] 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 silent transition occurs between two states with the same signal a, we obtain a^{d_1} followed by a^{d_2}, which are merged into $a^{d_1+d_2}$. 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*. Again, 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 (silent transitions, with label ε) and hidden signals (τ-labeled states). It turns out that these features are very useful, for instance for handling abstractions. They also allow us to get as special cases the initial models of timed automata and signal automata.

We also include Zeno runs in our general theory. Examples are given with automata $\mathcal{A}_2$ and $\mathcal{A}_4$ in Figure 1, where it can be seen that, with ε-transitions, open constraints in a timed automaton can lead to a closed constraint in the corresponding language. Although Zeno behaviours have been studied (see for instance [13, 10]), 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, experiments with the well known model-checking tool UPPAAL show that Zeno runs are included when forward reachability analysis is performed. Thus, omitting them from the semantics could lead to erroneous results for verification. Hence, we think that Zeno runs have to be taken into account when developping the theory of signal-event automata.

Timed substitutions. When time is added to the picture, the global duration of a sequence must be taken into account when it replaces a signal. For instance, considering an operation of connection to a server, the signal $ConnectToServer^2$ (with a duration of 2 time units) can be replaced by the signal-event word of same duration $Req \cdot Wait^2 \cdot Ack$, and $ConnectToServer^{4.5}$ (with duration 4.5 t.u.) by $Req \cdot Wait^1 \cdot Nack \cdot Wait^{0.5} \cdot Retry \cdot Wait^3 \cdot Ack$.

Timed substitutions were studied for regular transfinite time-event languages in [8] where, although no signal appear explicitely, actions are handled in a way similar (but not identical) to signals, *without* stuttering. Here we restrict the study of substitutions to finite and ω-sequences, excluding transfinite sequences, but we do handle signal stuttering which is a major difficulty. The principal aim of this paper is to investigate the closure under substitutions and inverse substitutions of the families SEL_ε and SEL of languages accepted by signal-event automata, respectively with or without ε-transitions. We prove that the class SEL_ε is closed under arbitrary substitutions and under arbitrary inverse substitutions. These closure properties do not hold for the class SEL. Nevertheless, we show that SEL is closed under inverse substitutions acting on events only, i.e., leaving signals unchanged, and we give a sufficient condition for its closure under substitution. These results again show the robustness of the class SEL_ε, which is in favour of signal-event automata including ε-transitions.

Intersection. The proofs of closure under substitution and inverse substitution relies on the closure under intersection of the class 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 is based on a cartesian product of automata, where constraints on the transitions are added to enforce some synchronization. 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 the stuttering of signals and the unobservability of hidden signals with null duration. In [3], a construction is given for the intersection of signal automata, but neither signal stuttering nor the unobservability of zero-duration signal are 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 [11], in order to obtain a determinization result, a construction is proposed to remove stuttering and zero-duration on signal automata using a single clock but intersection is not considered directly. In [12], 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.

Outline of the paper. We first give in Section 2 precise definitions of finite and infinite signal-event languages, with the corresponding notion of signal-event automata. In Section 3, we give a general construction for the intersection of signal-event automata dealing both with signal stuttering and with finite and infinite sequences. To establish this result in full generality, we show that signal-event automata can be put in a normal form where no infinite run accepts a finite word with finite duration and no finite run accepts a word with infinite duration. In Section 4, we define timed substitutions which are duration preserving mappings. We then study in Section 5 the closure of the classes SEL and SEL_ε under substitutions and inverse substitutions. Section 6 gives concluding remarks.

Extended abstracts of this work appeared in [6, 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 : we assume that they can be observed, even if they are instantaneous, while the unobservable signal τ cannot be observed in this case.

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 finite or infinite. The partial monoid $\Sigma^\infty/\approx$ will be denoted $\mathit{SE}(\Sigma, \mathbb{T})$ or simply $\mathit{SE}(\Sigma)$ 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). Thus, a word in normal form is a particular representant of a SE-word, which is the whole equivalence class. Therefore, 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: $a^3 gfa^1 \prod_{i \geq 1} a^2 \approx a^1 a^2 gf \prod_{i \geq 1} a^4$ and the normal form is written $a^3 gfa^\infty$. 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 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 $v : X \to \mathbb{T}$ 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

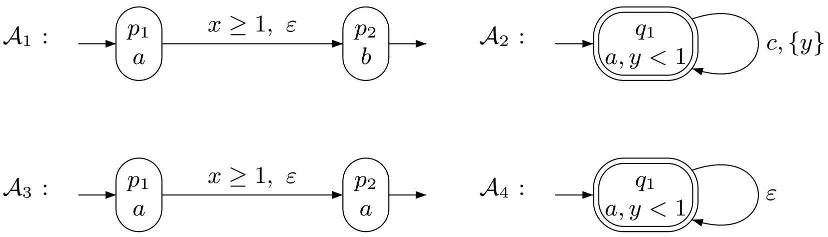


Fig. 1. Some signal automata.

of final states and $R \subseteq Q$ corresponds to a Büchi acceptance condition[3]. The mapping $I : Q \to \mathcal{C}(X)$ 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 $\ell : Q \to \Sigma_s$ 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].

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 $a^{d_1} c a^{d_2} c \ldots$ with $d_i < 1$ for any i.
- Since 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). Note that, due to the ε-transition and Zeno runs, the open constraint of the invariant $y < 1$ is changed into a closed constraint for the language.

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

[3] We also use freely conjunctions of Büchi conditions and/or conditions on transitions instead of states. It is well-known how to turn these generalized Büchi conditions into classical ones [14].

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 $v_i + d_i \models I(q_i)$.

Remark 1. The condition about the last step of a finite run fits the semantics of signal-event automata : it is symmetric with the first step which is always a duration step. Of course, this condition is not the same as in a time-event automaton, where the time spent in a state is considered as a waiting time with respect to the following action.

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 $\mathbb{T}$, 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 whereas *SEA* is the set of all signal-event automata using transitions with observable events only, i.e., with labels in Σ_e instead of $\Sigma_e \cup \{\varepsilon\}$. The family of signal-event (timed) languages generated by some signal-event automaton in SEA_ε (respectively *SEA*) is denoted by SEL_ε (respectively *SEL*).

Remark 2. We believe that the semantics associating signals with control states and instantaneous events with transitions is natural for timed automata. Of course, considering only timed words, we could easily identify zero-duration signals with events. However, this would not be consistent with the semantics chosen here.

Remark 3. 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 above notations).

As mentioned in the introduction, 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. Again note that such runs appear in the semantics of a timed model-checking tool like Uppaal, as can be easily checked on examples.

3 Intersection

We present in this section the construction of a timed automaton recognizing the intersection of the languages accepted by two given timed automata, thus giving a constructive proof to the following theorem.

Theorem 1. *The class SEL_ε is closed under intersection.*

This result might seem easy but this is not the case in its full generality. So we start by explaining informally the natural construction then we turn to the difficulties that arise with signals and ε-transitions. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two *SE*-automata with $\mathcal{A}_j = (\Sigma_e, \Sigma_s, X_j, Q_j, Q_j^0, F_j, R_j, I_j, \ell_j, \Delta_j)$ for $j = 1, 2$. Without loss of generality, we assume that the set of clocks are disjoint: $X_1 \cap X_2 = \emptyset$. In order to recognize the intersection $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$ we usually build a product automaton. This works fine if the automata $\mathcal{A}_j$ have no ε-transitions. In this case, we let $Q = \{(q_1, q_2) \in Q_1 \times Q_2 \mid \ell_1(q_1) = \ell_2(q_2)\}$ and for $(q_1, q_2) \in Q$ we let $\ell((q_1, q_2)) = \ell_1(q_1) = \ell(q_2)$ and $I((q_1, q_2)) = I_1(q_1) \wedge I_2(q_2)$. The transitions of $\mathcal{A}$ are obtained by synchronizing transitions of $\mathcal{A}_j$. More precisely, if $(p_1, p_2), (q_1, q_2) \in Q$ and $(p_j, g_j, e, \alpha_j, q_j) \in \Delta_j$ then we introduce in Δ the synchronized transition $((p_1, p_2), g_1 \wedge g_2, e, \alpha_1 \cup \alpha_2, (q_1, q_2))$. We define the initial states, the final states and the accepting condition of $\mathcal{A}$ in such a way that a path P of $\mathcal{A}$ is accepting iff each projections P_j on $\mathcal{A}_j$ is accepting (for $j = 1, 2$). If an *SE*-word w is accepted by $\mathcal{A}_j$ through some path P_j for $j = 1, 2$ then the synchronization of P_1 and P_2 is an accepting path of $\mathcal{A}$ and w admits a run through this path. Conversely, if w admits a run through some accepting path P of $\mathcal{A}$ then it also admits a run through the projections P_j of P which are accepting paths of $\mathcal{A}_j$. Therefore, $\mathcal{A}$ accepts the intersection $\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.

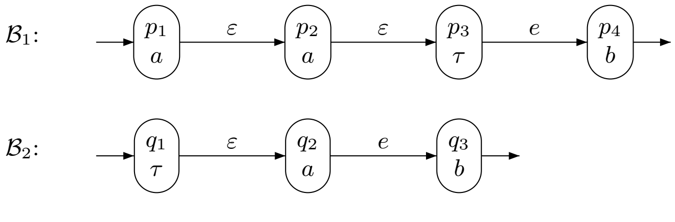


Fig. 2. Automata $\mathcal{B}_1$ and $\mathcal{B}_2$

We explain now the difficulties that arise with signal stuttering when the automata contain ε-transitions. Consider for instance the automata $\mathcal{B}_1$ and $\mathcal{B}_2$ of Figure 2. We have

$$\mathcal{L}(\mathcal{B}_1) = \{a^{d_1} \tau^{d_2} e b^{d_3} \mid d_1, d_2, d_3 \geq 0\}$$
$$\mathcal{L}(\mathcal{B}_2) = \{\tau^{d_1} a^{d_2} e b^{d_3} \mid d_1, d_2, d_3 \geq 0\}$$

Recall that in our model, the signal τ^0 is equivalent to the empty word ε. Hence, we have $L(\mathcal{B}_1) \cap L(\mathcal{B}_2) = \{a^{d_1} e b^{d_3} \mid d_1, d_3 \geq 0\}$. The word $a^2 e b^1$ in this intersection is accepted by $\mathcal{B}_1$ and $\mathcal{B}_2$ with the runs

$$\langle p_1, 0\rangle \xrightarrow{1} \langle p_1, 1\rangle \xrightarrow{\varepsilon} \langle p_2, 1\rangle \xrightarrow{1} \langle p_2, 2\rangle \xrightarrow{\varepsilon} \langle p_3, 2\rangle \xrightarrow{0} \langle p_3, 2\rangle \xrightarrow{e} \langle p_4, 2\rangle \xrightarrow{1} \langle p_4, 3\rangle$$

$$\langle q_1, 0\rangle \xrightarrow{0} \langle q_1, 0\rangle \xrightarrow{\varepsilon} \langle q_2, 0\rangle \xrightarrow{2} \langle q_2, 2\rangle \xrightarrow{e} \langle q_3, 2\rangle \xrightarrow{1} \langle q_3, 3\rangle.$$

Note that the ε-transitions in these runs are not synchronized, whereas the e-transitions are synchronized. Hence, in order to deal with signal stutttering and ε-transitions, we have to use asynchronous transitions for blocks of the form a^d and synchronous transitions to switch between blocks. The automaton that we will build for the intersection $\mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$ is shown in Figure 3 where the synchronous transition is depicted with a strong line. We will explain later the meaning of the fourth component of a state, which is from the set $\{0, 1, 2\}$. Note that, switching between blocks might involve a visible action like in the example above but it might also involve an ε-transition. This is the case for instance if the normal form of a word in the intersection is $a^{d_1} b^{d_2}$. The automata $\mathcal{B}_1'$ and $\mathcal{B}_2'$ of Figure 6 will provide an example for this situation.

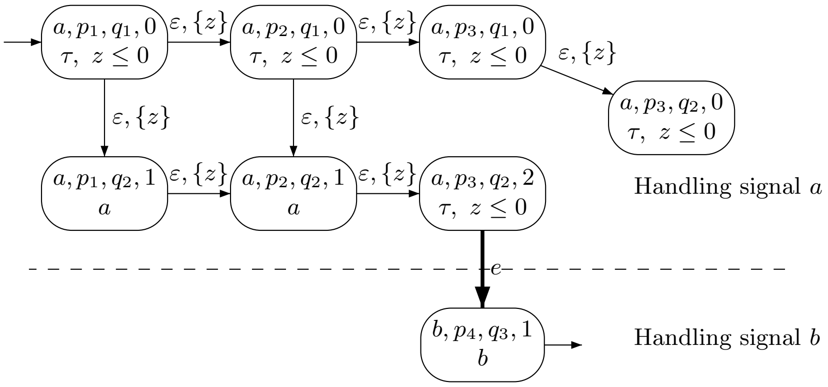


Fig. 3. Automaton $\mathcal{B}$ for the intersection of $\mathcal{B}_1$ and $\mathcal{B}_2$

The next problem arising in the construction of the intersection comes from the fact that a word can be accepted in one automaton by a finite run and in the other automaton by an infinite run. 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. The asynchronous product of $\mathcal{A}_3$ and $\mathcal{A}_4$ is given in Figure 4. We have $\mathcal{L}(\mathcal{A}) = \emptyset$ due to the invariant $y < 1$ and the guard $x \geq 1$. Therefore, this product cannot be used in this case.

Indeed, this problem is linked to Zeno runs and would not occur if Zeno runs were ruled out by the semantics. But we need not exclude Zeno runs to get

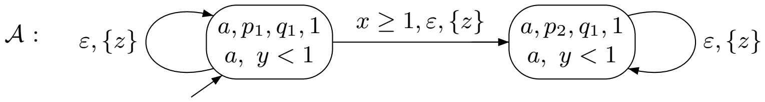


Fig. 4. Asynchronous product of $\mathcal{A}_3$ and $\mathcal{A}_4$

Theorem 1. We show that it is possible to preprocess the automata so that the above situation does not occur anymore. The problem comes from finite words with finite durations which are accepted by Zeno runs through infinite paths.

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 durations. 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.

Theorem 2. *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.*

Ensuring (‡) is easy. Intuitively this happens when the invariant of some final state is `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 a new clock z to measure time intervals of length 1. For each signal $a \in \Sigma_s$, we add a new state r_a with label a and invariant $z \leq 1$. State r_a is both repeated and final. We also add loops $(r_a, z = 1, \varepsilon, \{z\}, r_a)$ and for each final state p with label a and invariant `true` we add the transition $(p, \texttt{true}, \varepsilon, \{z\}, r_a)$ and p is not final anymore. This gives an automaton satisfying (‡).

The difficulty is therefore to ensure the first condition. Note that, if Zeno runs are not allowed by the semantics, then all accepting infinite runs have an infinite duration, hence condition (†) is automatically fulfilled. Similarly, if the automaton does not have ε-transitions, an accepting infinite run yields an infinite *SE*-word and again condition (†) is automatically fulfilled. The proof for the general case of automata with ε-transitions and where the semantics allow Zeno runs is postponed to Section 3.1. Note that the transformation only 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.

We now continue with the proof of Theorem 1. We start with two *SE*-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. 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 $(p, \texttt{true}, \varepsilon, \emptyset, p)$.

We define a product automaton $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q^0, F, R, I, \ell, \Delta)$. We will show that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$ (Lemma 1) and that the converse holds when the two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy conditions (†) and (‡) of Theorem 2.

- 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
  - $\ell(a, p, q, i) = \tau$ and $I(a, p, q, i) = I_1(p) \wedge I_2(q) \wedge z \leq 0$ if $i \neq 1$ and
  - $\ell(a, p, q, 1) = a$ and $I(a, p, q, 1) = I_1(p) \wedge I_2(q)$.

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 using ε-transitions and instantaneous traversings of τ-labelled states. If a synchronization is not possible on signal a, then we cannot extend the computation to an accepting run.
- 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 to 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 by switching to another block 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 $(a, p_1, p_2, 0)$ since a synchronization on signal a is expected first. Consider two states $(a, p_1, p_2, i) \in Q$ and $(a', q_1, q_2, i') \in Q$ with $i \neq 0$ and $i' \neq 2$. Then, for any two transitions $\delta_j = (p_j, g_j, b, \alpha_j, q_j) \in \Delta_j$ $(j = 1, 2)$ with $b \neq \varepsilon$ if $a = a'$, we add in Δ the synchronous transition

$$\delta = (a, p_1, p_2, i) \xrightarrow{g_1 \wedge g_2, b, \alpha_1 \cup \alpha_2 \cup \{z\}} (a', q_1, q_2, i')$$

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

Consider now a state $(a, p_1, p_2, i) \in Q$ of $\mathcal{A}$. Then, for any transition $\delta_1 = (p_1, g_1, \varepsilon, \alpha_1, q_1)$ in Δ_1 with $\ell_1(q_1) \in \{a, \tau\}$ we add in Δ the asynchronous transition

$$\delta = (a, p_1, p_2, i) \xrightarrow{g_1, \varepsilon, \alpha_1 \cup \{z\}} (a, q_1, p_2, i')$$

where i' is updated so that (a, q_1, p_2, i') is a legal state and follows the abstract description in Figure 5. Formally, if $a = \tau$ then $i' = 1$ is the only possibility. Now, if $a \neq \tau$ we have the following cases:

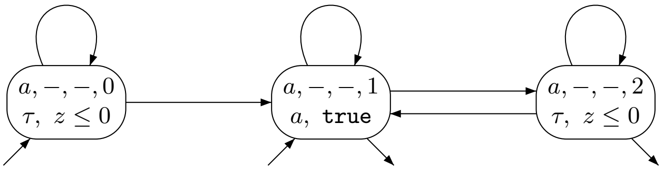


Fig. 5. Handling blocks a^d, for $a \neq \tau$

- $i' = 1$ if $\ell_1(q_1) = \ell_2(p_2) = a$: synchronization on a is active,
- $i' = 0$ if $i = 0$ and $\neg(\ell_1(q_1) = \ell_2(p_2) = a)$: synchronization on a has not yet been achieved,
- $i' = 2$ if $i \neq 0$ and $\neg(\ell_1(q_1) = \ell_2(p_2) = a)$: 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, but they may also have guards and other clocks to reset.

Since we have assumed that $\mathcal{A}_1$ and $\mathcal{A}_2$ do not contain a trivial loop of the form $(p, \texttt{true}, \varepsilon, \emptyset, p)$, 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 $(g_j = \texttt{true} \wedge b = \varepsilon \wedge \alpha_j = \emptyset \wedge q_j = p_j)$ and $\pi_j(\delta) = (p_j, g_j, b, \alpha_j, q_j)$ otherwise.

A path P of $\mathcal{A}$ is 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 $Q^0 = Q \cap (\Sigma_s \times Q_1^0 \times Q_2^0 \times \{0, 1\})$ and $F = Q \cap (\Sigma_s \times F_1 \times F_2 \times \{1, 2\})$. 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 $\mathcal{A}_j$.

Remark 4. Note that the construction would be simpler if we consider a subclass of the model where we keep ε-transitions, but require that a non-zero duration is spent in any state. In this case, since no signal can have a null duration, there is no need for the synchronization part in Figure 5 and the 4^{th} component of the states could be dropped.

Examples. We have already seen an easy example with the two automata $\mathcal{B}_1$ and $\mathcal{B}_2$ in Figure 2, 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. The automaton $\mathcal{B}$ constructed for the intersection is represented in Figure 3.

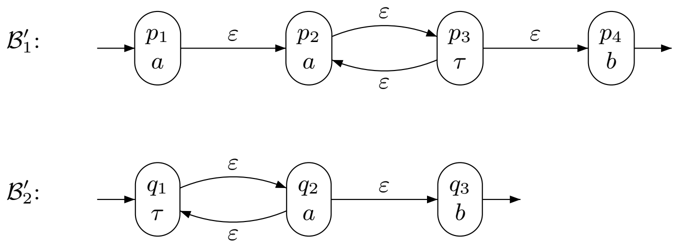


Fig. 6. Automata $\mathcal{B}_1'$ and $\mathcal{B}_2'$

We now modify automata $\mathcal{B}_1$ and $\mathcal{B}_2$ into $\mathcal{B}_1'$ and $\mathcal{B}_2'$ by adding loops and transforming the e-transitions to ε-transitions, as represented in Figure 6. In this case, we have

$$\mathcal{L}(\mathcal{B}_1') \cap \mathcal{L}(\mathcal{B}_2') = \{a^{d_1}\tau^{d_2} \dots \tau^{d_{2n}} a^{d_{2n+1}} b^{d_{2n+2}} \mid n \geq 0 \wedge d_i \geq 0\}.$$

The product automaton $\mathcal{B}'$ obtained with our construction is represented in Figure 7.

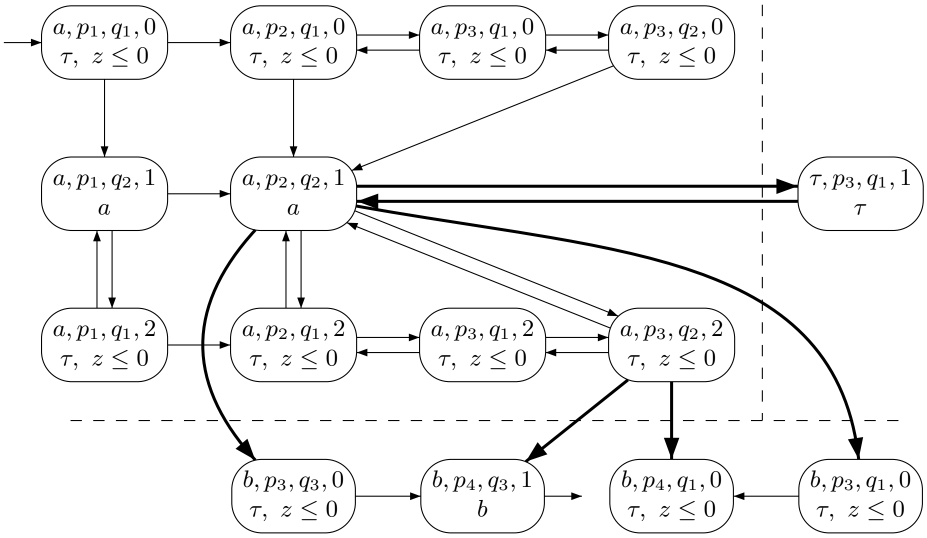

$a, p_1, q_1, 0$ $\tau,\ z \leq 0$
$a, p_2, q_1, 0$ $\tau,\ z \leq 0$
$a, p_3, q_1, 0$ $\tau,\ z \leq 0$
$a, p_3, q_2, 0$ $\tau,\ z \leq 0$
$a, p_1, q_2, 1$ a
$a, p_2, q_2, 1$ a
$\tau, p_3, q_1, 1$ τ
$a, p_1, q_1, 2$ $\tau,\ z \leq 0$
$a, p_2, q_1, 2$ $\tau,\ z \leq 0$
$a, p_3, q_1, 2$ $\tau,\ z \leq 0$
$a, p_3, q_2, 2$ $\tau,\ z \leq 0$
$b, p_3, q_3, 0$ $\tau,\ z \leq 0$
$b, p_4, q_3, 1$ b
$b, p_4, q_1, 0$ $\tau,\ z \leq 0$
$b, p_3, q_1, 0$ $\tau,\ z \leq 0$

Fig. 7. Resulting automaton $\mathcal{B}'$

We insist again that the construction relies on condition (†). As explained above, the construction for the two automata $\mathcal{A}_3$, $\mathcal{A}_4$ in Figure 1 yields the automaton $\mathcal{A}$ in Figure 4. Condition (†) does not hold for $\mathcal{A}_4$. Due to the invariant $y < 1$ and the guard $x \geq 1$ we have $\mathcal{L}(\mathcal{A}) = \emptyset$ whereas $\mathcal{L}(\mathcal{A}_3) \cap \mathcal{L}(\mathcal{A}_4) = \{a^1\}$.

Correctness of the construction. We keep of course the notations of the construction given above. We start with one inclusion which does not need the conditions (†) and (‡) of Theorem 2.

Lemma 1. *We have $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.*

Proof. Let $w \in \mathcal{L}(\mathcal{A})$ and let ρ be a run for w through some accepting path P of $\mathcal{A}$. We write

$$P = r_0 \xrightarrow{g_1,b_1,\alpha_1} r_1 \xrightarrow{g_2,b_2,\alpha_2} r_2 \cdots$$

with $r_k = (a_k, p_k, q_k, i_k)$ for $k \geq 0$ and

$$\rho = \langle r_0, v_0\rangle \xrightarrow{d_0} \langle r_0, v_0 + d_0\rangle \xrightarrow{b_1} \langle r_1, v_1\rangle \xrightarrow{d_1} \langle r_1, v_1 + d_1\rangle \xrightarrow{b_2} \langle r_2, v_2\rangle \cdots$$

Let $0 < j_1 < j_2 < \cdots$ be the indices of the transitions of P whose first projection are nonempty. We also set $j_0 = 0$. For $k \geq 0$, we have $p_n = p_{j_k}$ for all $j_k \leq n < j_{k+1}$ hence we may write

$$\pi_1(P) = p_{j_0} \xrightarrow{g^1_{j_1},b_{j_1},\alpha^1_{j_1}} p_{j_1} \xrightarrow{g^1_{j_2},b_{j_2},\alpha^1_{j_2}} p_{j_2} \cdots$$

By definition of $\mathcal{A}$, since P is accepting, we get that $\pi_1(P)$ is accepting, too. We assume that P is infinite, hence $\pi_1(P)$ is infinite, too. The slight modifications for the case P finite are left to the reader. For $k \geq 0$, let $D_k = d_{j_k} + \cdots + d_{j_{k+1}-1}$ and let v^1_k be the restriction of v_k to X_1. Then define the first projection $\pi_1(\rho)$ of ρ by

$$\langle p_{j_0}, v^1_{j_0}\rangle \xrightarrow{D_0} \langle p_{j_0}, v^1_{j_0} + D_0\rangle \xrightarrow{b_{j_1}} \langle p_{j_1}, v^1_{j_1}\rangle \xrightarrow{D_1} \langle p_{j_1}, v^1_{j_1} + D_1\rangle \xrightarrow{b_{j_2}} \langle p_{j_2}, v^1_{j_2}\rangle \cdots$$

We claim that $\pi_1(\rho)$ is a run for w through the accepting path $\pi_1(P)$ of $\mathcal{A}$. We fix some $k \geq 0$ and we first show that

$$(1) \qquad \begin{cases} v^1_{j_k} + D_k \models I(p_{j_k}) \\ v^1_{j_k} + D_k \models g^1_{j_{k+1}} \\ v^1_{j_{k+1}} = (v^1_{j_k} + D_k)[\alpha^1_{j_{k+1}}] \end{cases}$$

With $n = j_{k+1} - 1$, we have $v^1_{j_k} + D_k = v^1_n + d_n$. Since $v^1_n + d_n \models I(r_n)$ we deduce that $v^1_{j_k} + D_k = v^1_n + d_n \models I(p_n) = I(p_{j_k})$. Similarly, $v^1_{j_k} + D_k = v^1_n + d_n \models g^1_{j_{k+1}}$. Finally, $v^1_{j_{k+1}} = (v^1_n + d_n)[\alpha^1_{j_{k+1}}] = (v_{j_k} + D_k)[\alpha^1_{j_{k+1}}]$ and (1) is shown. We deduce by induction that $\pi_1(\rho)$ is a run through $\pi_1(P)$.

It remains to prove that this run $\pi_1(\rho)$ generates the word w. For $k \geq 0$ and $j_k < n < j_{k+1}$ we have $b_n = \varepsilon$, hence $w = w_0 b_{j_1} w_1 b_{j_2} w_2 \cdots$ where $w_k = \ell(r_{j_k})^{d_{j_k}} \ell(r_{j_k+1})^{d_{j_k+1}} \cdots \ell(r_{j_{k+1}-1})^{d_{j_{k+1}-1}}$. We have to show that $w_k = \ell(p_{j_k})^{D_k}$. For $k \geq 0$ and $j_k \leq n < j_{k+1}$, we have either $i_n = 1$ and $\ell(r_n) = \ell(p_n) = \ell(p_{j_k})$ or $i_n \neq 1$ and $\ell(r_n) = \tau$ and $d_n = 0$ since $z \in \alpha_n$ and $z \leq 0$ is in the invariant $I(r_n)$. Therefore, $w_k = \ell(p_{j_k})^{D_k}$ and we deduce that $\pi_1(\rho)$ is a run for w through $\pi_1(P)$. We have already noticed that $\pi_1(P)$ is accepting in $\mathcal{A}_1$ hence we get $w \in \mathcal{L}(\mathcal{A}_1)$.

In a symmetric way, $w \in \mathcal{L}(\mathcal{A}_2)$ and we have proved the first inclusion $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. □

We finally come to the converse inclusion. Here we need some hypotheses which are explicited in the next lemma.

Lemma 2. *Let $w \in \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. Assume that w is accepted by $\mathcal{A}_j$ through some accepting path P^j for $j = 1, 2$. If*

- *either both P^1, P^2 are finite,*
- *or both P^1, P^2 are infinite and w is either infinite or has infinite duration,*

then $w \in \mathcal{L}(\mathcal{A})$.

Proof. For $j = 1, 2$, let ρ^j be a run for w through the accepting path P^j of $\mathcal{A}_j$. We assume in the following that both paths are infinite. The slight modifications to deal with the case where both paths are finite are left to the reader. Write

$$P^j = p_0^j \xrightarrow{g_1^j, b_1^j, \alpha_1^j} p_1^j \xrightarrow{g_2^j, b_2^j, \alpha_2^j} p_2^j \cdots$$

and

$$\rho^j = \langle p_0^j, v_0^j \rangle \xrightarrow{d_0^j} \langle p_0^j, v_0^j + d_0^j \rangle \xrightarrow{b_1^j} \langle p_1^j, v_1^j \rangle \xrightarrow{d_1^j} \langle p_1^j, v_1^j + d_1^j \rangle \xrightarrow{b_2^j} \langle p_2^j, v_2^j \rangle \cdots$$

We will define a block decomposition of the paths. We first introduce a notation. For $i < k$, let $w^j(i,k) = \ell(p_i^j)^{d_i^j} b_{i+1}^j \cdots b_{k-1}^j \ell(p_{k-1}^j)^{d_{k-1}^j}$. Now, we define inductively a sequence of integers. We let $n_0^j = 0$ and assuming that n_k^j has been defined then we let $n_{k+1}^j = \max\{i > n_k^j \mid w^j(n_k^j, i) \in \Sigma_s \times \mathbb{T}\} \in \mathbb{N} \cup \{\infty\}$. We have $w^j(n_k^j, n_{k+1}^j) = (a_k^j)^{D_k^j}$ for some signal $a_k^j \in \Sigma_s$ and $D_k^j = \sum_{n_k^j \leq i < n_{k+1}^j} d_i^j$. Note that $b_i^j = \varepsilon$ for all $n_k^j < i < n_{k+1}^j$ and for $n_k^j \leq i < n_{k+1}^j$ we have $\ell(p_i^j) \in \{\tau, a_k\}$ and $d_i^j = 0$ if $\ell(p_i^j) \neq a_k$. Also, if $n_{k+1}^j = \infty$ then w must be finite and from the hypotheses of Lemma 2 we deduce that w has an infinite duration, which implies $D_k^j = \infty$.

To simplify the notation, we let $e_k^j = b_{n_k^j}^j$ for $k > 0$. Since ρ^j is a run for w through P^j we have

$$\begin{aligned} w &\approx (a_0^1)^{D_0^1} e_1^1 (a_1^1)^{D_1^1} e_2^1 (a_2^1)^{D_2^1} e_3^1 \cdots \\ &\approx (a_0^2)^{D_0^2} e_1^2 (a_1^2)^{D_1^2} e_2^2 (a_2^2)^{D_2^2} e_3^2 \cdots \end{aligned}$$

We show by induction that in fact $a_k^1 = a_k^2$, $D_k^1 = D_k^2$ and $e_{k+1}^1 = e_{k+1}^2$ for all $k \geq 0$.

Assume that $a_0^1 \neq a_0^2$. Since $(a_0^1)^{D_0^1}$ and $(a_0^2)^{D_0^2}$ are both prefixes of w, we must have for instance $a_0^1 = \tau$ and $D_0^1 = 0$ and $e_1^1 = \varepsilon$. This is a contradiction with the definition of n_1^1. Therefore, $a_0^1 = a_0^2$. Assume now that $D_0^1 < D_0^2$. To simplify the notation, we let $m = n_1^1$. Since $(a_0^1)^{D_0^1} e_1^1 \ell(p_m^1)^{d_m^1}$ and $(a_0^2)^{D_0^2}$ are both prefixes of w, we must have $e_1^1 = \varepsilon$ and if $\ell(p_m^1) \neq a_0^1 = a_0^2$ then $\ell(p_m^1) = \tau$ and $d_m^1 = 0$. Again, this is a contradiction with the definition of n_1^1. Therefore, we have also $D_0^1 = D_0^2$. Assume finally that $e_1^1 \neq e_1^2$. Again, from the fact that $(a_0^1)^{D_0^1} e_1^1$ and $(a_0^2)^{D_0^2} e_1^2$ are both prefixes of w we deduce that for instance $e_1^1 = \varepsilon \neq e_1^2$. Then, with $m = n_1^1$ we also have $(a_0^1)^{D_0^1} \ell(p_m^1)^{d_m^1}$ prefix of w and we deduce that $d_m^1 = 0$ and either $\ell(p_m^1) = a_0^1$ or $\ell(p_m^1) = \tau$. Again, this is a contradiction with the definition of n_1^1. Therefore, $e_1^1 = e_1^2$.

Repeating the above arguments, we can prove by induction that $a_k^1 = a_k^2 = a_k$, $D_k^1 = D_k^2 = D_k$ and $e_{k+1}^1 = e_{k+1}^2 = e_{k+1}$ for all $k \geq 0$. Therefore, we have

$$w \approx a_0^{D_0} e_1 a_1^{D_1} e_2 a_2^{D_2} e_3 \cdots$$

which is called the *alternating normal form* (ANF) of w since it insists on a strict alternation between events and signals at the expense of keeping some invisible events (ε) and some invisible signals (τ^0). Note that the ANF is different from the normal form introduced in Section 2. For instance, if $\Sigma_e = \{f, g\}$ and $\Sigma_s = \{a, b\}$ then we have

- $a^1 f a^3 g a^{2.5}$ 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 $fg(a^2 b^3)^\omega$.

We build now an accepting path P of $\mathcal{A}$ and the corresponding run ρ for w through P. Intuitively, for each block of the form $a_k^{D_k}$ we use an asynchronous shuffle of the transitions in P^1 and P^2 used for this block. The difficulty is to choose a correct order for this shuffle. To switch from block $a_k^{D_k}$ to block $a_{k+1}^{D_{k+1}}$, we synchronize the transitions n_{k+1}^1 and n_{k+1}^2 of $\mathcal{A}_1$ and $\mathcal{A}_2$. The formal construction of P is given below.

The initial state of P is $r_0 = (a_0, p_0^1, p_0^2, i_0)$ with $i_0 \neq 2$. Assume now that the path P has been built up to state $r_m = (a_k, p_{m^1}^1, p_{m^2}^2, i_m)$ which was reached at time t_m with clock valuation v_m such that

i) we have used the prefixes of P^j of length m^j with $n_k^j \leq m^j < n_{k+1}^j$ for $j = 1, 2$,
ii) $t_m = \max\{d_0^j + \cdots + d_{m^j-1}^j \mid j = 1, 2\} \leq \min\{d_0^j + \cdots + d_{m^j}^j \mid j = 1, 2\}$,
iii) $v_m(z) = 0$ and $v_m(x) = v_{m^j}^j(x) + t_m - (d_0^j + \cdots + d_{m^j-1}^j)$ for $x \in X^j$.

By convention, an empty sum is 0 so that the above holds for $m = 0$.

Let $\delta^j = d_0^j + \cdots + d_{m^j}^j$. The next transition of P will be taken at time $t_{m+1} = \min(\delta^1, \delta^2)$. We start by checking that the invariant $I(r_m)$ is satisfied by $v_m + d$ for $0 \leq d \leq t_{m+1} - t_m$.

- The conjunct $I_j(p_{m^j}^j)$ of the invariant is satisfied since the restriction of $v_m + d$ to X^j is $v_{m^j}^j + d'$ with $0 \leq d' = t_m - (d_0^j + \cdots + d_{m^j-1}^j) + d \leq d_{m^j}^j$ and ρ^j is a run through P^j.
- If $i \neq 1$ then $a_k \neq \tau$ and either $\ell(p_{m^1}^1) = \tau$ or $\ell(p_{m^2}^2) = \tau$ (or both). Assume without loss of generality that $\ell(p_{m^1}^1) = \tau$. Then, by definition of n_{k+1}^1 we deduce that $d_{m^1}^1 = 0$. It follows that $t_{m+1} = t_m$ and $d = 0$. Using *(iii)* we infer that the conjunct $z \leq 0$ of the invariant is also satisfied.

Before we continue, note that $\delta^j \leq D_0 + \cdots + D_k$ and we have equality if $m^j + 1 = n^j_{k+1}$.

1. If $t_{m+1} = \delta^1 < \delta^2$ then from the remark above we must have $m^1 + 1 < n^1_{k+1}$. In this case, we add to the path P the asynchronous transition
$$r_m \xrightarrow{g^1_{m^1+1}, \varepsilon, \alpha^1_{m^1+1} \cup \{z\}} r_{m+1} = (a_k, p^1_{m^1+1}, p^2_{m^2}, i_{m+1}) \tag{2}$$
where i_{m+1} is uniquely determined by the transition relation of $\mathcal{A}$. The restriction of $v_m + t_{m+1} - t_m$ to X^1 is $v^1_{m^1} + d^1_{m^1}$ hence the guard $g^1_{m^1+1}$ is satisfied when we take this transition and after the transition the clock valuation v_{m+1} satisfies (iii). It is also clear that (i) and (ii) are still satisfied.
2. If $t_{m+1} = \delta^2 < \delta^1$ the situation is symmetric and we add to the path P the asynchronous transition
$$r_m \xrightarrow{g^2_{m^2+1}, \varepsilon, \alpha^2_{m^2+1} \cup \{z\}} r_{m+1} = (a_k, p^1_{m^1}, p^2_{m^2+1}, i_{m+1}) \tag{3}$$
where i_{m+1} is uniquely determined by the transition relation of $\mathcal{A}$.
3. We assume now that $t_{m+1} = \delta^1 = \delta^2$. The difficult case is when $i_m = 0$. In this case, the synchronization between the two paths P^1 and P^2 in order to generate the signal $a_k^{D_k}$ has not yet been achieved. We have $a_k \neq \tau$ and either $\ell(p^1_{m^1}) = \tau$ or $\ell(p^2_{m^2}) = \tau$ (or both).
   - If $\ell(p^1_{m^1}) = \tau \neq a_k$ then $w^1(n^1_k, m^1 + 1) = \tau^0 \neq a_k^{D_k}$ and we deduce that $m^1 + 1 < n^1_{k+1}$. The automaton $\mathcal{A}_1$ still needs to perform ε-transitions in order to reach a state with signal a_k. In this case, we extend P with the asynchronous transition (2) and we check as above that (*i*-*iii*) are still satisfied.
   - If $\ell(p^1_{m^1}) = a_k$ then the automaton $\mathcal{A}_1$ has reached a state with signal a_k and it is waiting for $\mathcal{A}_2$ to do the same. We must have $\ell(p^2_{m^2}) = \tau$ and $m^2 + 1 < n^2_{k+1}$. We extend P with the asynchronous transition (3) and we check as above that (*i*-*iii*) are still satisfied.
4. We assume now that $t_{m+1} = \delta^1 = \delta^2$ and $i_m \neq 0$. Then both automata have already successfully synchronized on a_k and the order in which we take asynchronous transitions which occured at the same date in P^1 and P^2 is not important. There are three cases:
   - If $m^1 + 1 < n^1_{k+1}$ then extend P with the asynchronous transition (2).
   - If $m^2 + 1 < n^2_{k+1}$ then extend P with the asynchronous transition (3).
   - If $m^1+1 = n^1_{k+1}$ and $m^2+1 = n^2_{k+1}$ then we have completed the block for $a_k^{D_k}$ in both automata and we extend P with the synchronous transition
$$r_m \xrightarrow{g^1_{m^1+1} \wedge g^2_{m^2+1}, e_{k+1}, \alpha^1_{m^1+1} \cup \alpha^2_{m^2+1} \cup \{z\}} r_{m+1} = (a_{k+1}, p^1_{m^1+1}, p^2_{m^2+1}, i_{m+1})$$
     with $i_{m+1} \neq 2$ to reach the next block $a_{k+1}^{D_{k+1}}$. As above we check that both guards are satisfied when we take this transition and that (*i*-*iii*) still hold for $m + 1$.

We have build a path P and simultaneously a run ρ for w through P. By construction of P, for $j = 1, 2$ the projection $\pi_j(P)$ is a prefix of P^j. Actually, we have $\pi_j(P) = P^j$. Indeed, assume on the contrary that $\pi_2(P)$ is a strict prefix of P^2 of length m^2. Then, we eventually use transition of type (2) only within some block $a_k^{D_k}$. In particular, w must be finite and its duration must be infinite. Using transitions of type (2) only implies that $d_0^1 + \cdots + d_m^1 \leq d_0^2 + \cdots + d_{m^2}^2$ for all m, which is a contradiction with the infinite duration of w.

We have proved that $\pi_j(P) = P^j$ for $j = 1, 2$ and we deduce that the path P of $\mathcal{A}$ is accepting. Therefore, $w \in \mathcal{L}(\mathcal{A})$. □

This concludes also the proof of Theorem 1. Indeed, by Theorem 2 we may assume that the two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy conditions (†) and (‡). We deduce that the hypotheses of Lemma 2 are automatically fulfiled. Using also Lemma 1 we get $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$.

3.1 Proof of Theorem 2

We start from an automaton $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q_0, F, R, I, \ell, \Delta)$. We have to construct an automaton satisfying condition (†). Let $\mathcal{A}_F$ be a copy of $\mathcal{A}$ in which R is replaced by the empty set. Similarly, let $\mathcal{A}_R$ be a copy of $\mathcal{A}$ in which F is replaced by the empty set. We have $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_F) \cup \mathcal{L}(\mathcal{A}_R)$ and the automaton $\mathcal{A}_F$ clearly satisfies (†). So we only have to concentrate on infinite runs of $\mathcal{A}$. In the following, we simply assume that $F = \emptyset$ in $\mathcal{A}$.

We first analyse how a finite word with finite duration wa^d can be accepted by a Zeno run ρ through some infinite path P. If this is the case then we can write $P = P_1 P_2$ where all transitions used in P_2 occurs infinitely often in P_2. Then, P_2 consists only of ε-transitions and states with label a or τ. The states labelled $\tau \neq a$ must be traversed instantaneously. Now, if a clock x is reset on some transition of P_2 and a constraint of the form $x > c$ or $x \geq c$ is in the guard of some transition of P_2 then $c = 0$ (otherwise we would not have $d < \infty$). Indeed, P_2 must also contain at least one accepting state.

The ituitive idea is to replace P_2 by a finite loop P_2' satisfying the above conditions. Once the loop P_2' is completed we will use an ε-transition to some new final state with a suitable invariant. The difficulty is to make sure that any word accepted through the new path could have been accepted with a Zeno run through some infinite path of $\mathcal{A}$.

It will be convenient to distinguish two cases depending on whether time progresses infinitely often or not in P_2.

First case. For any signal a and subset of clocks $Z \subseteq X$, we build an automaton $\mathcal{A}(a, Z)$ which accepts a word wa^d iff this word can be accepted by a run ρ through an infinite path P of $\mathcal{A}$ such that Z is the set of clocks that are reset infinitely often in P and eventually time does not progress anymore in ρ.

Let $z \notin X$ be a new clock. The automaton $\mathcal{A}(a, Z)$ contains (a copy of) the states and transitions of $\mathcal{A}$ together with some new states and transitions (see

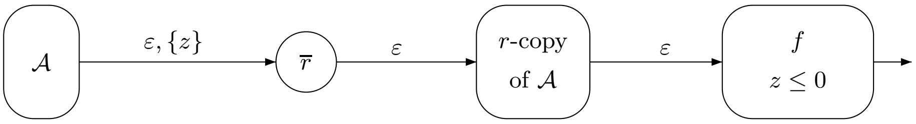


Fig. 8. The automaton $\mathcal{A}(a, Z)$

Figure 8 for an intuitive view). For each $(p, g, \varepsilon, \alpha, q) \in \Delta$ with

$$\begin{array}{l} \ell(p), \ell(q) \in \{a, \tau\} \text{ and } \alpha \subseteq Z \text{ and } \forall x \in Z, \\ \text{if } x \geq c \text{ is a constraint in } g \text{ then } c = 0 \\ \text{and there is no constraint in } g \text{ of the form } x > c \end{array} \tag{4}$$

we add the following states and transitions:

$$\begin{array}{ll} (p, g, \varepsilon, \alpha \cup \{z\}, \overline{q}) & \text{if } \ell(q) = a, \\ (\overline{p}, g, \varepsilon, \alpha, (q, p, \alpha, (q \in R))) & \text{if } \ell(p) = a, \\ ((p, r, \beta, \varphi), g, \varepsilon, \alpha, (q, r, \alpha \cup \beta, \varphi \vee (q \in R))) & \text{if } \ell(r) = a. \end{array} \tag{5}$$

For the new states, we let $\ell(\overline{r}) = \ell(r) = a$, $I(\overline{r}) = I(r) \wedge z \leq 0$, $\ell(p, r, \beta, \varphi) = \ell(p)$ and $I(p, r, \beta, \varphi) = I(p) \wedge z \leq 0$. These new transitions simulate an instantaneous loop around some a-labelled state r: just before reaching r, we move into the copy and reach $\overline{r}$ instead. We remember r until reaching $(r, r, Z, \texttt{true})$, where we know that the loop has been successfully completed and all clocks in Z has been reset and a repeated state has been visited. Therefore, we also add the transitions $((r, r, Z, \texttt{true}), \varepsilon, f_a)$ where f_a is a new state which is the only final state of $\mathcal{A}(a, Z)$, with $\ell(f_a) = a$ and $I(f_a) = z \leq 0$. The automaton $\mathcal{A}(a, Z)$ has no repeated states and its initial states are those of $\mathcal{A}$. The constraint $z \leq 0$ in the invariants ensures that, if a word is accepted by $\mathcal{A}(a, Z)$ then it can be accepted by $\mathcal{A}$ with an ω-iteration of the instantaneous loop going from $\overline{r}$ to $(r, r, Z, \texttt{true})$.

Second case. We now build a collection of automata $\mathcal{A}(a, Y, m, M)$ which accept a finite word with finite duration wa^d iff this word can be accepted through an infinite path of $\mathcal{A}$ by a Zeno run in which time progresses infinitely often (even though time converges). The construction is more tricky than in the first case. As explained above, the intuition is that $\mathcal{A}(a, Y, m, M)$ detects loops of $\mathcal{A}$ that can be iterated infinitely often using a Zeno run for which time progresses infinitely often.

Let $a \in \Sigma_s$ be a signal and $Y \subseteq X$ be a subset of clocks. Intuitively, Y is the set of clocks that are *not* reset along the loop. Let $m = (m_y)_{y \in Y}$ be a tuple of *lower* constants for clocks in Y, i.e., for each y, either $m_y = 0$ or there exists a constraint $y > m_y$ or $y \geq m_y$ occuring in some guards of $\mathcal{A}$. Here the intuition is that m_y should be the maximal lower constant for y on the loop. Similarly, let $M = (M_y)_{y \in Y}$ be a tuple of *upper* constants for clocks in Y, i.e., for each y, either $M_y = \infty$ or there exists a constraint $y < M_y$ or $y \leq M_y$ occuring in some

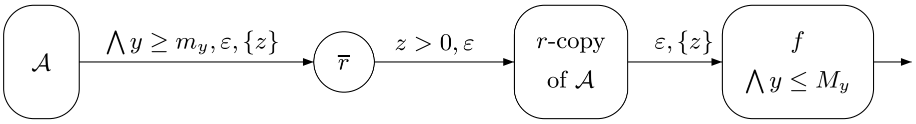


Fig. 9. The automaton $\mathcal{A}(a, Y, m, M)$

guard or invariant of $\mathcal{A}$. Similarly, M_y should be the minimal upper constant for y along the loop. Note that the number of such tuples (a, Y, m, M) is finite.

Let z be a new clock. The automaton $\mathcal{A}(a, Y, m, M)$ contains (a copy of) the states and transitions of $\mathcal{A}$ together with some new states and transitions described below (see Figure 9 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

$$(6)\quad \begin{array}{l} \ell(p), \ell(q) \in \{a, \tau\} \text{ and } \alpha \cap Y = \emptyset, \\ \forall y \in Y, \text{ if } y > c \text{ or } y \geq c \text{ is a constraint in } g \text{ then } m_y \geq c, \\ \forall y \in Y, \text{ if } y < c \text{ or } y \leq c \text{ is a constraint in } g, I(p) \text{ or } I(q) \text{ then } M_y \leq c, \\ \forall x \in X \setminus Y, \text{ if } x > c \text{ or } x \geq c \text{ is a constraint in } g \text{ then } c = 0. \end{array}$$

We add the following states and transitions to $\mathcal{A}(a, Y, m, M)$:

$$(7)\quad \begin{array}{ll} (p, g \wedge \bigwedge_{y \in Y} y \geq m_y, \varepsilon, \alpha \cup \{z\}, \overline{q}) & \text{if } \ell(q) = a, \\ (\overline{p}, g \wedge z > 0, \varepsilon, \alpha \cup \{z\}, (q, p, \alpha, (q \in R))) & \text{if } \ell(p) = a, \\ ((p, r, \beta, \varphi), g, \varepsilon, \alpha \cup \{z\}, (q, r, \alpha \cup \beta, \varphi \vee (q \in R))) & \text{if } \ell(r) = a. \end{array}$$

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 $I(p, r, \beta, \varphi) = I(p) \wedge z \leq 0$ if $\ell(p) = \tau \neq a$. These new transitions simulate a loop around some a-labelled 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 $\texttt{true}$).

Let $\mathcal{A}_1$ be the disjoint union of all automata $\mathcal{A}(a, Z)$ and $\mathcal{A}(a, Y, m, M)$. We will prove below that $\mathcal{A}_1$ accepts precisely all finite words with finite durations accepted by $\mathcal{A}$ with Zeno runs through infinite paths. In order to get the automaton $\mathcal{A}'$ satisfying (†) we still have to build an automaton $\mathcal{A}_2$ accepting $\mathcal{L}(\mathcal{A}) \setminus \mathcal{L}(\mathcal{A}_1)$. This second step is much easier. We have 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 $\mathcal{B}$ with no final state and 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 a

that has been observed ($a^d \neq \tau^0$) so that we can enter a repeated state when a *new* signal b is observed ($b^\delta \neq \tau^0$ with $b \neq a$). Let $\mathcal{A}_2$ be the product of the automata $\mathcal{A}$ and $\mathcal{B}$ as defined for the proof of Theorem 1. By Lemma 1 we have $\mathcal{L}(\mathcal{A}_2) \subseteq \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$. Now the accepting runs of $\mathcal{A}$ and $\mathcal{B}$ are infinite since these automata have no final states. Moreover, a word is accepted by $\mathcal{B}$ iff it is infinite or has an infinite duration. Hence, we can apply Lemma 2 and we get the converse inclusion: $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A}_2)$. Note that there is no vicious circle in this proof. We are not using Theorem 1 here, only Lemmas 1 and 2 which were stated carefully so that they do not depend on Theorem 2.

Correctness of the construction. The most difficult part of the proof takes place in the second case of the construction above. We thus prove this case in details. Let wa^d be some finite word with finite duration ending with signal a and which is accepted by $\mathcal{A}$ with some infinite run ρ through some infinite path P and such that time progresses infinitely often in ρ. Let also Y be the set of clocks that are not reset infinitely often on P, and for each $y \in Y$, let m_y (resp. M_y) be the maximal lower constant (resp. the minimal upper constant) for clock y which is used infinitely often in P.

Lemma 3. *$wa^d \in \mathcal{L}(\mathcal{A}(a, Y, m, M))$.*

Proof. We first show that wa^d is accepted by $\mathcal{A}(a, Y, m, M)$. We may write $P = P_0P_1P_2P_3$ and $\rho = \rho_0\rho_1\rho_2\rho_3$ accordingly where

- ρ_0 is a run through P_0 for wa^{d_0} and ρ_i is a run through P_i for a^{d_i} for $i \in \{1, 2, 3\}$,
- for each $i \in \{1, 2, 3\}$, the set of transitions used in P_i is exactly the set of transitions that are used infinitely often in P,
- P_2 is a loop around some state r and the time spent in state r between P_1 and P_2 is strictly positive.

Since some time is spent in state r, we have $\ell(r) = a$ and we can change the last transition $(p, g, \varepsilon, \alpha, r)$ of P_1 to $(p, g \wedge \bigwedge_{y \in Y} y \geq m_y, \varepsilon, \alpha \cup \{z\}, \overline{r})$ to obtain a new path P'_1. We then *lift* P_2 to P'_2 so that $P_0P'_1P'_2$ is a path of $\mathcal{A}(a, Y, m, M)$. The last state of P'_2 is $(r, r, X \setminus Y, \texttt{true})$ hence we obtain an accepting path P' of $\mathcal{A}(a, Y, m, M)$ by adding the transition $((r, r, X \setminus Y, \texttt{true}), \texttt{true}, \varepsilon, \{z\}, f)$.

For $i \in \{0, 1, 2\}$, let ρ'_i be obtained from ρ_i by adding the valuation of the new clock z. Then, $\rho'_0\rho'_1\rho'_2$ is a run through $P_0P'_1P'_2$. Indeed, the additional constraint $\bigwedge_{y \in Y} y \geq m_y$ on the last transition of P'_1 is satisfied since P_1 activates at least once each transition used infinitely often in P and clocks in Y are not reset in P_1. The additional constraint $z > 0$ on the first transition of P'_2 is also satisfied since the time spent in state r between P_1 and P_2 is strictly positive. Moreover, if P_2 uses a state with label $\ell(p) = \tau \neq a$ then the time spent by ρ_2 in p is 0 since ρ_2 is a run for a^{d_2}. Hence, the additional invariants $z \leq 0$, if any, are also satisfied.

Let $v \in \mathbb{T}^X$ be the clock valuations at the end of $\rho_0\rho_1\rho_2$ (including the time spent in the last state r of P_2). Then, for each $y \in Y$, we have $v(y) + d_3 \leq M_y$

since if $M_y < \infty$ then some constraint $y < M_y$ or $y \leq M_y$ is activated infinitely often in P_3 and y is never reset. Therefore, we obtain a run ρ' for wa^d through the path P': $\rho' = \rho'_0\rho'_1\rho'_2 \xrightarrow{\varepsilon} (f, v, 0) \xrightarrow{d_3} (f, v + d_3, d_3)$ (the last component is the valuation of the new clock z) and wa^d is accepted by $\mathcal{A}(a, Y, m, M)$. □

Conversely, let (a, Y, m, M) be defined as in case 2. Hence, $a \in \Sigma_s$ is a signal and $Y \subseteq X$ is a subset of clocks. For each $y \in Y$, m_y and M_y are lower constant and upper constant for clock y.

Lemma 4. *$\mathcal{L}(\mathcal{A}(a, Y, m, M)) \subseteq \mathcal{L}(\mathcal{A})$.*

Proof. Let ρ be a run through some accepting path P of $\mathcal{A}(a, Y, m, M)$. We write $P = P_1P_2P_3$ where the first state of P_2 is some $\overline{r}$, the last state of P_2 is $(r, r, X \setminus Y, \mathtt{true})$ and P_3 consists of a single transition leading to the state f. With $p_0 = p_n = r$, we may write

$$\begin{aligned}
\rho = \rho_1 &\xrightarrow{\varepsilon} (\overline{p}_0, v_0, 0) \xrightarrow{d_0} (\overline{p}_0, v_0 + d_0, d_0) \\
&\xrightarrow{\varepsilon} ((p_1, r, \alpha_1, \varphi_1), v_1, 0) \xrightarrow{d_1} ((p_1, r, \alpha_1, \varphi_1), v_1 + d_1, d_1) \\
&\cdots \\
&\xrightarrow{\varepsilon} ((p_n, r, \alpha_n, \varphi_n), v_n, 0) \xrightarrow{d_n} ((p_n, r, \alpha_n, \varphi_n), v_n + d_n, d_n) \\
&\xrightarrow{\varepsilon} (f, v_{n+1}, 0) \xrightarrow{d_{n+1}} (f, v_{n+1} + d_{n+1}, d_{n+1})
\end{aligned}$$

where the last components of the tuples describe the valuations of z. The timed word accepted by ρ is of the form $w = w_1a^\delta$ where w_1 is associated with ρ_1 through P_1 and $\delta = d_0 + \cdots + d_{n+1}$. Note that $d_0 > 0$.

Let $P'_1P'_2$ be the *projection* of the path P_1P_2 (the projection of each added transition presented in (7) is $(p, g, \varepsilon, \alpha, q)$). Note that P'_2 is a loop from r to r containing a repeated state since the flag in the last state of P_2 is $\mathtt{true}$. We show that $w = w_1a^\delta$ is accepted by some run through the infinite path $P'_1P'^{\omega}_2$.

For this, we first exhibit a sufficient condition under which the path P'_2 can be taken. Fix $d = d_0 + \cdots + d_{n-1} > 0$ and assume that $v'_0 \in \mathbb{T}^X$ and $\gamma > 0$ satisfy

$$(8) \qquad \begin{aligned}
&\gamma d \leq d_0/2 \\
&v'_0(x) \leq v_0(x) + d_0/2 \text{ for all } x \notin Y \\
&m_y \leq v'_0(y) < M_y - \gamma d \text{ for all } y \in Y.
\end{aligned}$$

Define $d'_i = \gamma d_i$ for $0 \leq i < n$. Then,

$$\begin{aligned}
\rho'_2(v'_0, \gamma) = (p_0, v'_0) &\xrightarrow{d'_0} (p_0, v'_0 + d'_0) \xrightarrow{\varepsilon} (p_1, v'_1) \\
&\cdots \\
&\xrightarrow{d'_{n-1}} (p_{n-1}, v'_{n-1} + d'_{n-1}) \xrightarrow{\varepsilon} (p_n, v'_n)
\end{aligned}$$

is a run through P'_2 with duration γd.

Indeed, for all $y \in Y$ and $0 \leq i < n$, we have $v'_i(y) = v'_0(y) + d'_0 + \ldots + d'_{i-1}$ since y is not reset. Hence $m_y \leq v'_0(y) \leq v'_i(y)$ and $v'_i(y) + d'_i \leq v'_0(y) + \gamma d < M_y$ and all guards or invariants involving y are satisfied. Now, for $x \in X \setminus Y$, we show by induction that for all $0 \leq i < n$ we have $v'_i(x) + d'_i \leq v_i(x) + d_i$ and $v'_i(x) + d'_i = 0$ if and only if $v_i(x) + d_i = 0$. First, we have $0 < v'_0(x) + d'_0 \leq v_0(x) + d_0/2 + d_0/2$. Now, for $i > 0$, using the induction hypothesis we get $v'_{i-1} + d'_{i-1} = 0$ iff $v_{i-1} + d_{i-1} = 0$. Hence, we have either $v'_i(x) = 0 = v_i(x)$ and the result follows since $0 < \gamma < 1$ and $d'_i = \gamma d_i$, or $v'_i(x) = v'_{i-1}(x) + d'_{i-1} > 0$ and $v_i(x) = v_{i-1}(x) + d_{i-1} > 0$ and the result follows by induction since $d'_i \leq d_i$. Therefore, all the constraints involving x are satisfied (recall that these constraints are of the form $x < c$ or $x \leq c$ or $x > 0$ or $x \geq 0$, see (6)).

Now, we can choose a sequence $(\gamma_k)_{k \geq 0}$ such that $0 < \gamma_k d \leq d_0/2$ for all $k \geq 0$ and $\delta = \sum_{k \geq 0} \gamma_k d$.

The pair (v_0, γ_0) satisfies condition (8). Indeed, for all $y \in Y$, we have $m_y \leq v_0(y)$ since the last transition of P_1 carries the constraint $m_y \leq y$. Moreover, $v_0(y) + \gamma_0 d < v_0(y) + \delta \leq M_y$ due to the invariant $I(f)$. Hence $\rho'_2(v_0, \gamma_0)$ is a run through P'_2.

Let u_1 be the valuation at the end of $\rho'_2(v_0, \gamma_0)$. Note that all clocks $x \in X \setminus Y$ have been reset in P'_2 since the last state of P_2 is $(r, r, X \setminus Y, \mathtt{true})$. Hence, we have $u_1(x) \leq \gamma_0(d_1 + \cdots + d_{n-1}) \leq \gamma_0 d \leq d_0/2 \leq v_0(x) + d_0/2$ for all $x \in X \setminus Y$. Also, for all $y \in Y$, we have $m_y < u_1(y)$ and $u_1(y) + \gamma_1 d = v_0(y) + \gamma_0 d + \gamma_1 d \leq v_0(y) + \delta < M_y$. Hence, the pair (u_1, γ_1) satisfies condition (8) and $\rho'_2(u_1, \gamma_1)$ is a run through P'_2.

By induction, we construct a sequence of valuations $(u_k)_{k \geq 0}$ (with $u_0 = v_0$) such that the pair (u_k, γ_k) satisfies condition (8) and u_{k+1} is the valuation at the end of $\rho'_2(u_k, \gamma_k)$. We obtain a run $\rho' = \rho_1 \rho'_2(u_0, \gamma_0) \rho'_2(u_1, \gamma_1) \cdots$ for $w = w_1 a^\delta$ through the accepting infinite path $P'_1 P'^{\omega}_2$ of $\mathcal{A}$. Therefore, $w_1 a^\delta \in \mathcal{L}(\mathcal{A})$. □

4 Signal-event (timed) substitutions

Recall that substitutions are a suitable model for refinements. In the untimed framework, the image of each letter $a \in \Sigma$ is a given language L_a over another alphabet Σ' and a substitution is a morphism extending this mapping.

Dealing with timed words requires to preserve durations. Therefore, an instantaneous event must be replaced by *SE*-words with null duration, while a signal a with duration d must be replaced by *SE*-words of the same duration d. Formally, the new alphabet is also of the form $\Sigma' = \Sigma'_e \cup (\Sigma'_s \times \mathbb{T})$ and a substitution σ is defined by a family of *SE*-languages $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$ such that:

- $L_a \subseteq (\Sigma'_e \cup (\Sigma'_s \times \{0\}))^*$ if $a \in \Sigma_e$,
- L_a is a language of *non Zeno SE*-words over Σ' if $a \in \Sigma_s \setminus \{\tau\}$.
  Using Zeno-words in substitutions may give rise to transfinite sequences, therefore we have excluded them from the languages L_a.
- $L_\tau = \{\tau\} \times \overline{\mathbb{T}}$: we assume that the internal signal τ is never modified.

Then, for each $a \in \Sigma_e$, $\sigma(a) = L_a$ and for each $a^d \in \Sigma_s \times \overline{\mathbb{T}}$, $\sigma(a^d) = \{w \in L_a \mid \|w\| = d\}$. A substitution is thus a duration preserving mapping.

For a *SE*-word $v = v_1 v_2 \cdots$ in *normal form* over Σ, $\sigma(v)$ is the set of *SE*-words obtained from $\sigma(v_1)\sigma(v_2)\cdots$ by merging consecutive identical signals. Note that $w \in \sigma(v)$ can be written $w_1 w_2 \cdots$ with $w_i \in \sigma(v_i)$ for each $i \geq 1$. However this decomposition of w may not be in normal form. Finally, for a timed language L over Σ, we set $\sigma(L) = \cup_{v \in L} \sigma(v)$.

Example 2.

- Choosing $L_f = \{f\}$ for $f \in \Sigma_e$ and $L_a = \{\tau^0\} \cup \{a\} \times (\overline{\mathbb{T}} \setminus \{0\})$ for $a \in \Sigma_s$ leads to a substitution that removes all signals with a null duration.
- Hiding some signal a is simulated by a substitution where $L_a = \{\tau\} \times \overline{\mathbb{T}}$.

SE-substitutions and morphisms. It should be noticed that, while substitutions are morphisms in the untimed framework, this is not the case with our definition. For instance, assume that a is a signal such that $L_a = \{b^2\}$, then $\sigma(a^1) = \emptyset$ and $\sigma(a^2) = \{b^2\} \neq \sigma(a^1)\sigma(a^1)$ while $a^2 = a^1 \cdot a^1$.

We call *m-substitution* a *SE*-substitution which is a morphism with respect to the partial concatenation. We have the following characterization:

Proposition 1. *Let σ be a SE-substitution, given by a family $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$. Then, σ is a morphism if and only if for each signal $a \in \Sigma_s$ we have*

1. *L_a is closed under concatenation: for all $u, v \in L_a$ with $\|u\| < \infty$, we have $uv \in L_a$,*
2. *L_a is closed under decomposition: for each $v \in L_a$ with $\|v\| = d$, for all $d_1 \in \mathbb{T}$, $d_2 \in \overline{\mathbb{T}}$ such that $d = d_1 + d_2$, there exist $v_i \in L_a$ with $\|v_i\| = d_i$ such that $v = v_1 v_2$.*

Remark 5. This proposition is easy to prove and shows that rather restrictive conditions should be added on the languages L_a to obtain morphisms. Moreover, considering the questions of closure under substitutions, we can see that the proofs would be much simpler for m-substitutions. However, morphisms do not represent natural operations in our framework. Consider for instance the automaton of Figure 10 in the next Section, which can be used to refine the signal *Connect* with some implementation details. Clearly, this substitution is not a morphism. More generally, substitions that arise naturally to describe abstractions or refinements are not morphisms. Therefore, it is important to study the closure of recognizable *SE*-languages under substitutions that are not necessarily morphisms.

Inverse substitutions. The notion of abstraction is fundamental in the study of systems, and in particular in their verification. It consists in replacing a set of behaviors with a single action in order to obtain a smaller system, simpler to study and to understand. The abstraction operation is thus the inverse of refinement. As in the untimed case, inverse substitutions provide a suitable model for abstractions in our framework.

For a substitution σ, the inverse substitution σ^{-1} is the operation defined for a language $L' \subseteq SE(\Sigma')$ by $\sigma^{-1}(L') = \{v \in SE(\Sigma) \mid \sigma(v) \cap L' \neq \emptyset\}$.

5 Recognizable substitutions

We now focus on recognizable substitutions, whose associated languages are defined by *SE*-automata. Formally, a substitution σ defined by a family of languages $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$ is a SEL_ε-substitution (a *SEL*-substitution resp.) if $L_a \in SEL_\varepsilon$ (resp. $L_a \in SEL$) for each $a \in \Sigma_e \cup \Sigma_s$.

For instance, if the automaton of Figure 10 is associated with the signal *Connect*, then a connection operation of 14 time units, $Connect^{14}$, can be replaced by the word $Idle^3 \cdot Req \cdot Wait^2 \cdot TimeOut^0 \cdot Req \cdot Wait^1 \cdot Ack \cdot Connected^8$.

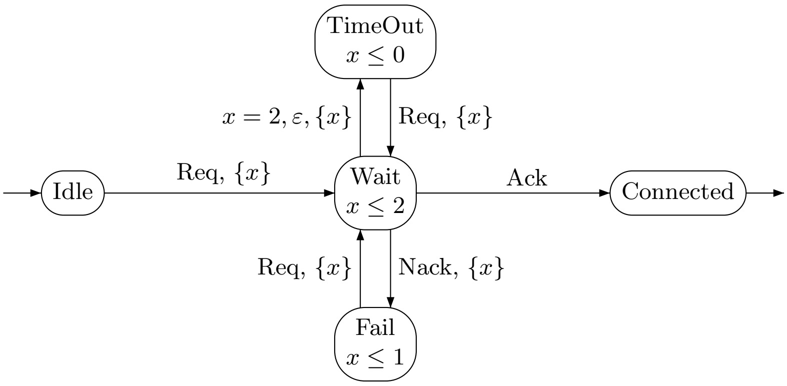


Fig. 10. A signal-event automaton to replace signal *Connect*

The aim of this paper is precisely to investigate the closure of the two classes of languages *SEL* and SEL_ε under recognizable *SE*-substitutions and their inverse.

We first consider the special cases of renaming and event-hiding. Formally, a renaming is simply a substitution such that, for each $f \in \Sigma_e$, $L_f = \{g\}$ for some $g \in \Sigma'_e$, and for each $a \in \Sigma_s \setminus \{\tau\}$, $L_a = \{b\} \times \overline{\mathbb{T}}$ for some $b \in \Sigma'_s$. An event-hiding is a substitution such that $L_f = \{f\}$ or $L_f = \{\varepsilon\}$ for $f \in \Sigma_e$ and $L_a = \{a\} \times \overline{\mathbb{T}}$ for $a \in \Sigma_s$. We have:

Proposition 2.

1. *The classes SEL and SEL_ε are closed under renaming,*
2. *The class SEL_ε is closed under event-hiding whereas SEL is not,*

Proof. Point 1 is straightforward. For the second point, the closure of SEL_ε under event-hiding was already noticed in [1] and the result easily extends to our framework: in order to hide the event a, one replaces a-labelled transitions by ε-transitions.

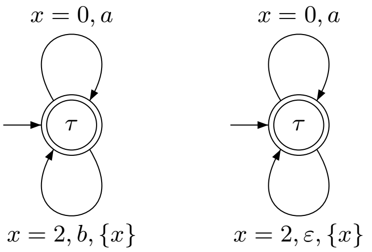
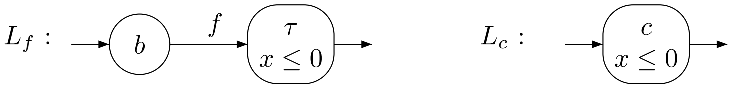


Fig. 11. Timed automata for L_1 and $h(L_1)$.

For the class *SEL*, recall from [5] that the language $\mathit{Even} = ((\tau^2)^* a)^\infty$ cannot be accepted by a timed automaton without ε-transitions. Consider the language $L_1 = ((\tau^2 b)^* a)^\omega \cup ((\tau^2 b)^* a)^* (\tau^2 b)^\omega$ accepted by the automaton on the left of Figure 11, with no final state, where the unique state is labeled τ and is both initial and repeated. Hiding the b's in L_1 yields the language $h(L_1)$ accepted by the automaton in SEA_ε on the right of Figure 11. Since $h(L_1)$ is the analogous of *Even* in our framework, it is not in the class *SEL*. □

The class *SEL* is not closed under *SEL*-substitution. Indeed, take $L_f = \{b^0 f\}$ and $L_c = \{c^0\}$ which are accepted by the automata below and are therefore both in *SEL*. Consider also $L = \{c^0 f\} \in \mathit{SEL}$. Then $\sigma(L) = \{c^0 b^0 f\}$ is not in *SEL* since it contains a *SE*-word with two consecutive distinct signals. The next result gives a sufficient condition on a substitution for the closure property of *SEL* to hold.



Theorem 3. *Let L be a language in SEL and σ a SEL-substitution such that for each $f \in \Sigma_e$ the language L_f contains only SE-words starting and ending with events from Σ'_e. Then $\sigma(L)$ belongs to SEL.*

Proof. We prove the theorem for a substitution acting only on events or acting only on signals. The general case is obtained using compositions of such elementary substitutions and of renaming.

We first show how to handle substitution of events. So let σ be a *SEL*-substitution satisfying the condition of the theorem and which is defined by a family $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$ of *SEL*-languages. Since σ acts only on events, we have $L_a = \{a\} \times \overline{\mathbb{T}}$ for all $a \in \Sigma_s$. For $f \in \Sigma_e$, the language L_f is accepted by some *SEA*-automaton $\mathcal{A}_f$ with only one clock x_f which is never tested or reset, with no repeated states, where all the guards are `true` and where all final states carry the invariant $x_f \leq 0$ in order to ensure that each accepted *SE*-word has a null duration. Moreover, from the hypothesis on σ, we can assume that all initial and final states are labelled τ and no state is simultaneously initial and final.

Now, let $L \subseteq SE(\Sigma)$ be accepted by a *SEA*-automaton $\mathcal{A}$. We build from $\mathcal{A}$ a *SEA*-automaton $\mathcal{A}'$ accepting $\sigma(L)$ as follows. For each state q, we consider a new copy $\mathcal{A}_f^q$ of $\mathcal{A}_f$ and we replace any transition (p, g, f, α, q) by the following set of transitions (see Figure 12):

– for each transition $(q_0, \mathtt{true}, b, \emptyset, q_1)$ in $\mathcal{A}_f^q$ with q_0 initial, we add the transition $(p, g, b, \alpha \cup \{x_f\}, q_1)$ to $\mathcal{A}'$,

– for each transition $(q_1, \mathtt{true}, b, \emptyset, q_2)$ in $\mathcal{A}_f^q$ with q_2 final, we add the transition $(q_1, x_f = 0, b, \emptyset, q_2)$ to $\mathcal{A}'$,

– for each transition $(q_0, \mathtt{true}, b, \emptyset, q_2)$ in $\mathcal{A}_f^q$ with q_0 initial and q_2 final, we add the transition (p, g, b, α, q) to $\mathcal{A}'$,

– all states and transitions of $\mathcal{A}_f^q$ are kept unchanged in $\mathcal{A}'$.

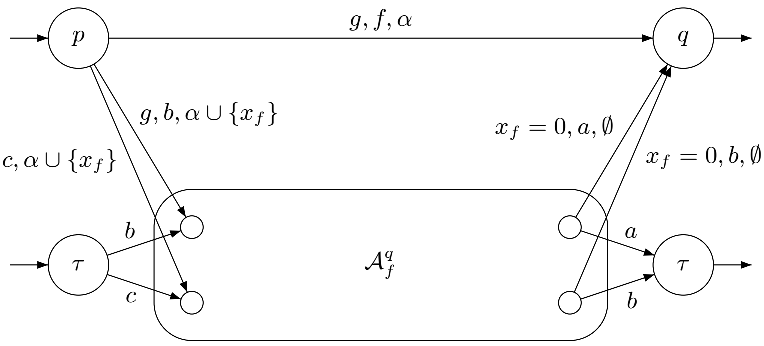


Fig. 12. Event substitution for *SEL*

The clock operations on x_f ensure an instantaneous traversal of $\mathcal{A}_f^q$. The initial, final and repeated states remain those of $\mathcal{A}$. Clearly $\mathcal{A}'$ is a *SEA*-automaton.

Correctness of the construction. We show that $\mathcal{A}'$ accepts $\sigma(L)$.

First, let $w \in \sigma(v)$ with $v \in L$. Consider an accepting path $P = p_0 \xrightarrow{g_1, f_1, \alpha_1} p_1 \xrightarrow{g_2, f_2, \alpha_2} p_2 \cdots$ for v in the automaton $\mathcal{A}$ and let $a_i = \ell(p_i)$ for $i \geq 0$. Then we have $v = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ and $w = a_0^{d_0} w_1 a_1^{d_1} w_2 \cdots$ with $w_i \in L_{f_i}$ for $i > 0$. For each $i > 0$, consider an accepting path $P_i = q_0 \xrightarrow{b_1} q_1 \cdots \xrightarrow{b_k} q_k$ in $\mathcal{A}_{f_i}^{p_i}$ (omiting true guards and empty resets) and define the corrresponding path $P_i' = p_{i-1} \xrightarrow{g_i, b_1, \alpha_i \cup \{x_{f_i}\}} q_1 \cdots \xrightarrow{x_{f_i} = 0, b_k, \emptyset} p_i$ in $\mathcal{A}'$ (if $|P_i| = 1$ then we let $P_i' = p_{i-1} \xrightarrow{g_i, b_1, \alpha_i} p_i$). Then, $P' = P_1' P_2' \cdots$ is an accepting path of $\mathcal{A}'$ and w admits a run through P'. Therefore, $w \in \mathcal{L}(\mathcal{A}')$.

Conversely, let $w \in \mathcal{L}(\mathcal{A}')$ be a word accepted through some path P'. Let $p_0, p_1, p_2, \ldots$ be the sequence of states along P' that are also states of $\mathcal{A}$. We split $P' = P_1' P_2' \cdots$ accordingly where P_i' is the subpath of P' going from p_{i-1} to p_i. By construction of $\mathcal{A}'$, there is a transition $p_{i-1} \xrightarrow{g_i, f_i, \alpha_i} p_i$ in $\mathcal{A}$ and an accepting path $P_i = q_0 \xrightarrow{b_1} q_1 \cdots \xrightarrow{b_k} q_k$ in $\mathcal{A}_{f_i}^{p_i}$ such that $P'_i = p_{i-1} \xrightarrow{g_i, b_1, \alpha_i \cup \{x_{f_i}\}} q_1 \cdots \xrightarrow{x_{f_i}=0, b_k, \emptyset} p_i$ in $\mathcal{A}'$ (or $P'_i = p_{i-1} \xrightarrow{g_i, b_1, \alpha_i} p_i$ if $|P_i| = 1$). Therefore, $P = p_0 \xrightarrow{g_1, f_1, \alpha_1} p_1 \xrightarrow{g_2, f_2, \alpha_2} p_2 \cdots$ is an accepting path of $\mathcal{A}$. Moreover, with $a_i = \ell(p_i)$, we have $w = a_0^{d_0} w_1 a_1^{d_1} w_2 \cdots$ for some d_i and w_i accepted through the path P_i in $\mathcal{A}_{f_i}^{p_i}$. Finally, with $v = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ we have $w \in \sigma(v)$ and v is accepted by $\mathcal{A}$ through the path P.

We now handle substitution of signals. Let σ be a *SEL*-substitution defined by a family $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$ of *SEL*-languages. We assume that σ acts only on signals, i.e., $L_f = \{f\}$ for each $f \in \Sigma_e$. For $a \in \Sigma_s$, the language L_a is accepted by some *SEA*-automaton $\mathcal{A}_a = (\Sigma'_e, \Sigma'_s, X_a, Q_a, Q_a^0, F_a, R_a, I_a, \ell_a, \Delta_a)$. We assume that all the X_a's are pairwise disjoints.

Let $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q_0, F, R, I, \ell, \Delta)$ be a *SEA*-automaton accepting a language $L \subseteq SE(\Sigma)$. We assume that X is disjoint from the X_a's. We build from $\mathcal{A}$ a *SEA*-automaton $\mathcal{A}'$ accepting $\sigma(L)$ as follows. For each state p of $\mathcal{A}$ with label a, we consider a new copy $\mathcal{A}_p$ of $\mathcal{A}_a$ in which the invariant $I(p)$ is added to all states: $I_p(r_p) = I_a(r) \wedge I(p)$ if r_p is the copy in $\mathcal{A}_p$ of state $r \in Q_a$. To build $\mathcal{A}'$, we start with the disjoint union of all the $\mathcal{A}_p$'s and we add *switching* transitions:

> If (p, g, f, α, q) is a transition of $\mathcal{A}$ then for each final state r_p of $\mathcal{A}_p$ and each initial state s_q of $\mathcal{A}_q$ we add the transition $(r_p, g, f, \alpha \cup X_{\ell(q)}, s_q)$ to $\mathcal{A}'$.

The initial states of $\mathcal{A}'$ are the initial states of all $\mathcal{A}_p$ such that $p \in Q_0$. The final states of $\mathcal{A}'$ are the final states of all $\mathcal{A}_p$ such that $p \in F$. An infinite path of $\mathcal{A}'$ is accepting if

- either it uses infinitely many switching transitions $(r_p, g, f, \alpha \cup X_{\ell(q)}, s_q)$ with $p \in R$,
- or else, it stays ultimately in some $\mathcal{A}_p$ with $p \in F$ and it visits infinitely often R_p, the copy of $R_{\ell(p)}$ in $\mathcal{A}_p$.

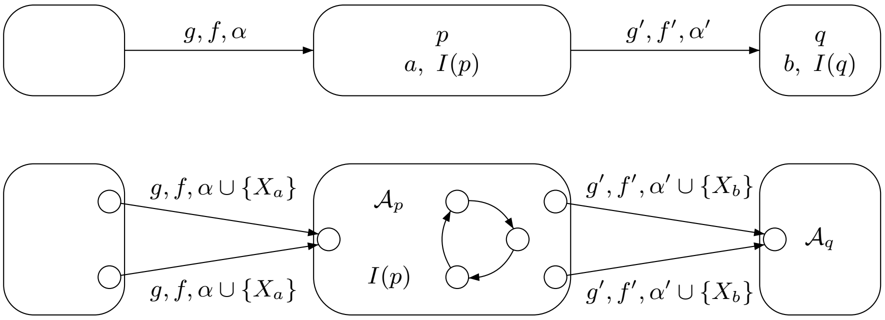
The construction is illustrated in Figure 13 for the general case where the original state p is neither initial nor final. The small state on the left of $\mathcal{A}_p$ is its initial state and the two small states on the right of $\mathcal{A}_p$ are its final states.

Correctness of the construction. We show that $\mathcal{A}'$ accepts $\sigma(L)$.

First, let $w \in \sigma(v)$ with $v \in L$ accepted through the accepting path

$$P = p_0 \xrightarrow{g_1, f_1, \alpha_1} p_1 \xrightarrow{g_2, f_2, \alpha_2} p_2 \cdots$$

of $\mathcal{A}$. We have $v = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ with $a_i = \ell(p_i)$ and $d_i \geq 0$. Since $w \in \sigma(v)$ we deduce $w = w_0 f_1 w_1 f_2 \cdots$ with $w_i \in \sigma(a_i^{d_i})$ for $i \geq 0$. Let P_i be an accepting path for w_i in $\mathcal{A}_{p_i}$. Note that, since $\mathcal{A}_{p_i}$ does not use ε-transitions we have w_i and P_i either both finite or both infinite. Moreover, if $d_i < \infty$ then w_i is finite since



g, f, α
p
$a,\ I(p)$
g', f', α'
q
$b,\ I(q)$
$g, f, \alpha \cup \{X_a\}$
$\mathcal{A}_p$
$I(p)$
$g, f, \alpha \cup \{X_a\}$
$g', f', \alpha' \cup \{X_b\}$
$\mathcal{A}_q$
$g', f', \alpha' \cup \{X_b\}$

Fig. 13. Signal substitution for *SEL*

by definition the language L_{a_i} does not contain Zeno words. Adding switching transitions to the P_i we obtain the path

$$P' = P_0 \xrightarrow{g_1, f_1, \alpha_1 \cup X_{a_1}} P_1 \xrightarrow{g_2, f_2, \alpha_2 \cup X_{a_2}} P_2 \cdots$$

of $\mathcal{A}'$ and it is easy to check that P' is an accepting path of $\mathcal{A}'$ and that w admits a run through P'.

Conversely, let $w \in \mathcal{L}(\mathcal{A}')$ and consider an accepting path P' of $\mathcal{A}'$ for w. Splitting P' according to switching transitions we get

$$P' = P_0 \xrightarrow{g_1, f_1, \alpha_1 \cup X_{a_1}} P_1 \xrightarrow{g_2, f_2, \alpha_2 \cup X_{a_2}} P_2 \cdots$$

where for $i \geq 0$, P_i is a path in $\mathcal{A}_{p_i}$ for some state p_i with $a_i = \ell(p_i)$. By definition of $\mathcal{A}'$ and since P' is accepting, we deduce that each P_i starts in some initial state of $\mathcal{A}_{p_i}$ and either is finite and ends in some final state of $\mathcal{A}_{p_i}$ or is infinite and visits infinitely often R_{p_i} and $p_i \in F$. Moreover, there are in $\mathcal{A}$ transitions $p_{i-1} \xrightarrow{g_i, f_i, \alpha_i} p_i$ for $i > 0$ so that

$$P = p_0 \xrightarrow{g_1, f_1, \alpha_1} p_1 \xrightarrow{g_2, f_2, \alpha_2} p_2 \cdots$$

is a path in $\mathcal{A}$. Since P' is accepting, we deduce easily that P is also accepting. The factorization of w associated with the splitting of P' is $w_0 f_1 w_1 f_2 w_2 \cdots$ where each w_i is accepted by $\mathcal{A}_{p_i}$ through the path P_i. We deduce $w_i \in \sigma(a_i^{d_i})$ with $d_i = \|w_i\|$. Consider $v = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ so that we get $w \in \sigma(v)$. Using the fact that for each $i \geq 0$, the invariant $I(p_i)$ has been added to all states in $\mathcal{A}_{p_i}$, it is easy to check that v is accepted by $\mathcal{A}$ through the path P. Therefore, $v \in L$ and $w \in \sigma(L)$. □

We will now show that the class SEL_ε is closed under SEL_ε-substitutions. The construction for the substitution of signals given in the previous proof does not work. Indeed, by definition, a substitution must be applied to a word in normal form. The difficulty comes from the fact that in the automaton for L, a factor a^d

of some normal form may be generated by a path with several a-labelled states and even τ-labelled states that are crossed instantaneously if all these states are linked by ε-transitions. An example can be seen on the automaton of Figure 14, for which a possible run gives :

$$b^2 f a^{0.3} a^{0.6} \tau^0 a^{0.5} \tau^1 a^{0.6} \tau^0 a^{0.5} \tau^0 b^3 \approx b^2 f a^{1.4} \tau^1 a^{1.1} b^3.$$

So we cannot simply replace each a-labelled state by a copy of $\mathcal{A}_a$.

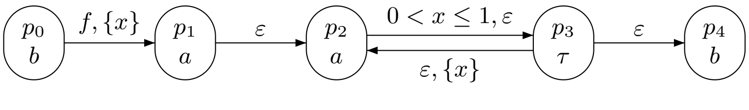


Fig. 14. The case of *SEL*$_\varepsilon$

To circumvent this difficulty, we use a proof technique inspired from rational transductions and that can be applied to establish the closure of the class *SEL*$_\varepsilon$ both under *SEL*$_\varepsilon$-substitutions and their inverse. Hence, we state and prove both results simultaneously. It should be noted that these closure properties hold for arbitrary substitutions, showing once again the robustness of the class *SEL*$_\varepsilon$.

Theorem 4. *The class SEL*$_\varepsilon$ *is closed under SEL*$_\varepsilon$*-substitution and inverse SEL*$_\varepsilon$*-substitution.*

Proof. Let σ be a *SEL*$_\varepsilon$-substitution from $SE(\Sigma)$ to $SE(\Sigma')$ given by a family of automata $(\mathcal{A}_a)_{a\in\Sigma_e\cup\Sigma_s}$, with $\mathcal{A}_a = (\Sigma'_e, \Sigma'_s, X_a, Q_a, Q^0_a, F_a, R_a, I_a, \ell_a, \Delta_a) \in SEA_\varepsilon$. We assume that these automata satisfy condition (†) of Proposition 2.

We show that $\sigma(L)$ and $\sigma^{-1}(L)$ can both be expressed as a projection of the intersection of a *SEL*$_\varepsilon$-language with an inverse projection of L. This is in the spirit of rational transductions for classical word languages.

Consider a new alphabet $\hat{\Sigma} = \hat{\Sigma}_e \cup \hat{\Sigma}_s \times \mathbb{T}$ with $\hat{\Sigma}_e = \Sigma_e \uplus \Sigma'_e$ ($\uplus$ is the disjoint union) and $\hat{\Sigma}_s = \Sigma_s \times \Sigma'_s$ (we identify (τ,τ) with the unobservable signal τ). The projections π_1 and π_2 are the morphisms defined by:

$\pi_1(f) = f$ and $\pi_2(f) = \varepsilon$ if $f \in \Sigma_e$,
$\pi_1(f) = \varepsilon$ and $\pi_2(f) = f$ if $f \in \Sigma'_e$,
$\pi_1((a,b)^d) = a^d$ and $\pi_2((a,b)^d) = b^d$ if $(a,b)^d \in \Sigma_s \times \Sigma'_s \times \overline{\mathbb{T}}$.

With this definition we have $\pi_i((a,b)^{d_1+d_2}) = \pi_i((a,b)^{d_1})\pi_i((a,b)^{d_2})$. Note that projection π_i is a composition of an event-hiding and a signal-renaming. By Proposition 2 we deduce that the projection by π_i of a *SEL*$_\varepsilon$ language is again in the class *SEL*$_\varepsilon$.

For $L \subseteq SE(\Sigma)$, we let $\pi_1^{-1}(L) = \{w \in SE(\hat{\Sigma}) \mid \pi_1(w) \in L\}$. We define similarly $\pi_2^{-1}(L)$ for $L \subseteq SE(\Sigma')$. We will show later that if L is recognizable then so is $\pi_i^{-1}(L)$. We will also define a recognizable language $M \subseteq SE(\hat{\Sigma})$ with the following properties:

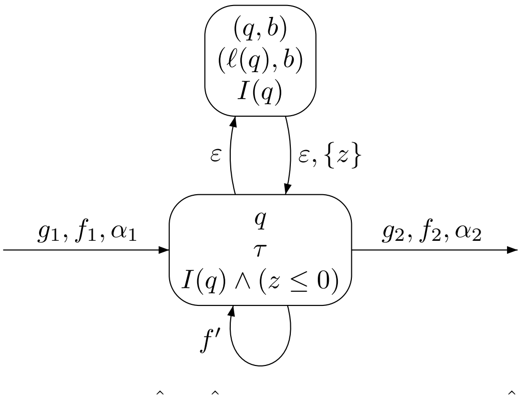
1. for each $w \in M$, we have $\pi_2(w) \in \sigma(\pi_1(w))$,
2. for each $u \in SE(\Sigma)$ and $v \in \sigma(u)$, there exists $w \in M$ such that $u = \pi_1(w)$ and $v = \pi_2(w)$.

Then, for $L \subseteq SE(\Sigma)$, we have $\sigma(L) = \pi_2(\pi_1^{-1}(L) \cap M)$. Indeed, Let $v \in \sigma(L)$ and let $u \in L$ with $v \in \sigma(u)$. Using property 2 of M we find $w \in M$ with $\pi_1(w) = u$ and $\pi_2(w) = v$. Then, $w \in \pi_1^{-1}(L) \cap M$ and $v \in \pi_2(\pi_1^{-1}(L) \cap M)$. Conversely, let $v \in \pi_2(\pi_1^{-1}(L) \cap M)$ and let $w \in \pi_1^{-1}(L) \cap M$ with $\pi_2(w) = v$. By definition, we have $u = \pi_1(w) \in L$ and using property 1 of M, we get $v \in \sigma(u) \subseteq \sigma(L)$.

Similarly, for $L \subseteq SE(\Sigma')$, we have $\sigma^{-1}(L) = \pi_1(\pi_2^{-1}(L) \cap M)$. Indeed, Let $u \in \sigma^{-1}(L)$ and let $v \in \sigma(u) \cap L$. Using property 2 of M we find $w \in M$ with $\pi_1(w) = u$ and $\pi_2(w) = v$. Then, $w \in \pi_2^{-1}(L) \cap M$ and $u \in \pi_1(\pi_2^{-1}(L) \cap M)$. Conversely, let $u \in \pi_1(\pi_2^{-1}(L) \cap M)$ and let $w \in \pi_2^{-1}(L) \cap M$ with $\pi_1(w) = u$. By definition, we have $v = \pi_2(w) \in L$ and using property 1 of M, we get $v \in \sigma(u)$. Hence, $\sigma(u) \cap L \neq \emptyset$ and $u \in \sigma^{-1}(L)$.

We already know that SEL_ε-languages are closed under intersection (Theorem 1) and projections π_i. To conclude the proof of Theorem 4, it remains to show that they are also closed under inverse projections π_i^{-1} and to define the SEL_ε-language M with the properties above.

We show first that SEL_ε-langagues are closed under inverse projections π_i^{-1}. Let L be recognized by some automaton $\mathcal{A} = (\Sigma_e, \Sigma_s, X, Q, Q^0, F, R, I, \ell, \Delta) \in SEA_\varepsilon$. We build an automaton $\hat{\mathcal{A}}$ accepting $\pi_1^{-1}(L)$. Intuitively, we start with the automaton $\mathcal{A}$ where for each state the label has been replaced by τ and $z \leq 0$ has been added to the invariant so that no signal can be observed while we are in the states of $\mathcal{A}$. We also add loops to each states so that events $f' \in \Sigma'_e$ can be inserted and signals in $\Sigma_s \times \Sigma'_s$ can be emitted.



Formally, the set of states of $\hat{\mathcal{A}}$ is $\hat{Q} = Q \uplus (Q \times \Sigma'_s)$, with $\hat{Q}^0 = Q^0$ for initial states and $\hat{F} = F$ for final states. The set of clocks is $X \uplus \{z\}$. The labels and invariants are defined by $\hat{\ell}(q) = \tau$ and $\hat{I}(q) = I(q) \wedge (z \leq 0)$ for $q \in Q$, and $\hat{\ell}((q,b)) = (\ell(q), b)$ and $\hat{I}((q,b)) = I(q)$ for $(q,b) \in Q \times \Sigma'_s$. The set of transitions $\hat{\Delta}$ is defined by:

1. All transitions $(p, g, f, \alpha, q) \in \Delta$ are kept in $\hat{\Delta}$.

2. For all $f' \in \Sigma'_e$ and $q \in Q$, we put $(q, \mathtt{true}, f', \emptyset, q)$ in $\hat{\Delta}$.
3. For all $(q, b) \in Q \times \Sigma'_s$, we put $(q, \mathtt{true}, \varepsilon, \emptyset, (q, b))$, $((q, b), \mathtt{true}, \varepsilon, \{z\}, q)$ in $\hat{\Delta}$.

We use a generalized acceptance condition for infinite paths. By transforming the automaton we can get a classical Büchi condition if needed. An infinite path is accepting if it uses

- either infinitely many transitions (p, g, f, α, q) of type 1 with $q \in R$,
- or ultimately transitions of type 2 and 3 only around some state $q \in F$.

Correctness of the construction. We show that $\mathcal{L}(\hat{\mathcal{A}}) = \pi_1^{-1}(L)$.

Let w be a word over $\hat{\Sigma}$ accepted by $\hat{\mathcal{A}}$. We consider a run of $\hat{\mathcal{A}}$ for w through an accepting path $\hat{P}$. Erasing from this path all transitions of type 2 and 3 above, we obtain a path P of $\mathcal{A}$. If $\hat{P}$ is finite then it ends in a final state $q \in \hat{F} = F$ and P is also accepting since it ends in state q. If $\hat{P}$ uses infinitely many transitions (p, g, f, α, q) of type 1 with $q \in R$, then P is infinite and visits R infinitely often, hence it is also accepting. Finally, if $\hat{P}$ is infinite and uses ultimately transitions of type 2 and 3 only around some state $q \in F$, then P is finite and accepting since it ends in state q. In all cases, P is an accepting path of $\mathcal{A}$. Splitting the paths according to transitions of type 1 we get

$$P = q_0 \xrightarrow{g_1, f_1, \alpha_1} q_1 \xrightarrow{g_2, f_2, \alpha_2} q_2 \cdots$$
$$\hat{P} = \hat{P}_0 \xrightarrow{g_1, f_1, \alpha_1} \hat{P}_1 \xrightarrow{g_2, f_2, \alpha_2} \hat{P}_2 \cdots$$

where $\hat{P}_i$ is a path consisting of transitions of type 2 and 3 only around state q_i. We consider the corresponding factorization $w = w_0 f_1 w_1 f_2 \cdots$ and for each $i \geq 0$ we let $d_i = \|w_i\|$ and $a_i = \ell(q_i)$ so that $\pi_1(w_i) = a_i^{d_i}$. Hence, we have $\pi_1(w) = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ and it is easy to verify that $\pi_1(w)$ admits a run through P. Therefore, $\pi_1(w) \in L$.

Conversely, let $w \in SE(\hat{\Sigma})$ and assume that $\pi_1(w) \in L$ is accepted by a run $\langle q_0, v_0 \rangle \xrightarrow{d_0} \langle q_0, v_0 + d_0 \rangle \xrightarrow{f_1} \langle q_1, v_1 \rangle \xrightarrow{d_1} \langle q_1, v_1 + d_1 \rangle \xrightarrow{f_2} \langle q_2, v_2 \rangle \cdots$ through an accepting path $P = q_0 \xrightarrow{g_1, f_1, \alpha_1} q_1 \xrightarrow{g_2, f_2, \alpha_2} q_2 \cdots$ of $\mathcal{A}$. Then, we have $\pi_1(w) \approx a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ with $a_i = \ell(q_i)$ for $i \geq 0$. We deduce that $w \approx w_0 f_1 w_1 f_1 w_2 \cdots$ with $\pi_1(w_i) = a_i^{d_i}$. Now, if w_i is finite with finite duration then we find a path $\hat{P}_i$ following the normal form of w_i, starting and ending in q_i, and using transitions of type 2 and 3 only. If w_i is infinite or with infinite duration then the path P must be finite ending in state $q_i \in F$ since P is accepting. Then, we find an *infinite* path $\hat{P}_i$ for w_i using only transitions of type 2 and 3 around q_i. Note that if w_i is finite with infinite duration, then it ends with $(a_i, b)^\infty$ for some $b \in \Sigma'_s$ and we still need an infinite path ultimately alternating between states q_i and (q_i, b). Finally, $\hat{P} = \hat{P}_0 \xrightarrow{g_1, f_1, \alpha_1} \hat{P}_1 \xrightarrow{g_2, f_2, \alpha_2} \hat{P}_2 \cdots$ is a path in $\hat{\mathcal{A}}$ and it is easy to see that $\hat{P}$ is accepting. Moreover, we can show that w admits a run through $\hat{P}$ and therefore, w is accepted by $\hat{\mathcal{A}}$.

We turn now to the definition of M. For $f \in \Sigma_e$ and $a \in \Sigma_s \setminus \{\tau\}$, we define

$$\begin{aligned} M_f &= \{w \in SE(\hat{\Sigma}) \mid w = (\tau, b_0)^0 f_1 (\tau, b_1)^0 f_2 \cdots (\tau, b_n)^0 \\ &\qquad\qquad\qquad\qquad \text{with } b_0^0 f_1 b_1^0 f_2 \cdots b_n^0 \in \sigma(f)\} \cdot f \\ M_a &= \{w \in SE(\hat{\Sigma}) \mid w = (a, b_0)^{d_0} f_1 (a, b_1)^{d_1} f_2 \cdots \\ &\qquad\qquad\qquad\qquad \text{with } b_0^{d_0} f_1 b_1^{d_1} f_2 \cdots \in \sigma(a^{d_0+d_1+\cdots})\} \end{aligned}$$

We also let $M_\tau = \{(\tau,\tau)^d \mid d \in \overline{\mathbb{T}} \setminus \{0\}\}$. Note that for $w \in M_a$ with $a \in \Sigma_e \cup \Sigma_s$ we have $\pi_2(w) \in \sigma(\pi_1(w))$ as required by property 1 of M. Moreover, if $f \in \Sigma_e$ and $v \in \sigma(f)$ then there exists $w \in M_f$ such that $\pi_1(w) = f$ and $\pi_2(w) = v$. Similarly, if $a^d \in \Sigma_s \times \overline{\mathbb{T}}$ (with $d > 0$ if $a = \tau$) and $v \in \sigma(a^d)$ then there exists $w \in M_a$ such that $\pi_1(w) = a^d$ and $\pi_2(w) = v$.

Intuitively, M consists of finite or infinite products of words in $\bigcup_{a \in \Sigma_e \cup \Sigma_s} M_a$ except that, in order to ensure that the first projection is in normal form, we should not allow consecutive factors associated with the same signal. Formally, we define

$$M = \{w_1 w_2 \cdots \mid \exists a_1, a_2, \ldots \in \Sigma_e \cup \Sigma_s \text{ with } w_i \in M_{a_i} \text{ and } a_i \in \Sigma_s \Rightarrow a_{i+1} \neq a_i\}.$$

We show that M satisfies property 1. Let $w = w_1 w_2 \cdots \in M$ and let $a_1, a_2, \ldots \in \Sigma_e \cup \Sigma_s$ be such that $w_i \in M_{a_i}$ and $a_i \in \Sigma_s \Rightarrow a_{i+1} \neq a_i$. Then, $\pi_1(w) = \pi_1(w_1)\pi_1(w_2)\cdots$ is in normal form and we have seen above that $\pi_2(w_i) \in \sigma(\pi_1(w_i))$ for each $i \geq 1$. Therefore, $\pi_2(w) \in \sigma(\pi_1(w))$ and property 1 is proved.

We show that M satisfies property 2. Let $u \in SE(\Sigma)$ and $v \in \sigma(u)$. Write $u = u_1 u_2 \cdots$ in normal form and $v = v_1 v_2 \cdots$ with $v_i \in \sigma(u_i)$. Let $a_i = u_i$ if $u_i \in \Sigma_e$ and $a_i = a$ if $u_i = a^d \in \Sigma_s \times \overline{\mathbb{T}}$. Since the product $u = u_1 u_2 \cdots$ is in normal form, we have $a_i \in \Sigma_s \Rightarrow a_{i+1} \neq a_i$ and $u_i = \tau^d \Rightarrow d > 0$. Since $v_i \in \sigma(u_i)$, we have seen above that there exists $w_i \in M_{a_i}$ such that $\pi_1(w_i) = u_i$ and $\pi_2(w_i) = v_i$. Then, $w = w_1 w_2 \cdots \in M$, $\pi_1(w) = u$ and $\pi_2(w) = v$ as required by property 2.

It remains to show that M is recognizable by some SEA_ε-automaton. For each $a \in \Sigma_e \cup \Sigma_s$ we first show that M_a is accepted by some automaton $\hat{\mathcal{A}}_a \in SEA_\varepsilon$ derived from $\mathcal{A}_a$. This is clear for $a = \tau$. Note that $\hat{\mathcal{A}}_\tau$ needs two states to ensure the positive duration required by M_τ. For $a \in \Sigma_s \setminus \{\tau\}$, the automaton $\hat{\mathcal{A}}_a$ is simply a copy of $\mathcal{A}_a$, with new label $(a, \ell_a(q))$ for $q \in Q_a$. For $f \in \Sigma_e$, the set of states is $\hat{Q}_f = Q_f \uplus \{q_f\}$, where q_f is a new state, which is also the only final state. The label of $q \in Q_f$ is $(\tau, \ell_f(q))$, the label of q_f is (τ,τ) and its invariant is $x_f \leq 0$ where x_f is the clock ensuring instantaneous traversal of $\mathcal{A}_f$. The transitions are those in $\mathcal{A}_f$, to which we add (q, f, q_f) for any state q which was final in $\mathcal{A}_f$. Note that, since $\mathcal{A}_a$ satisfies (†) for $a \in \Sigma_e \cup \Sigma_s$, then so does $\hat{\mathcal{A}}_a$.

Since M is essentially the iteration of the languages M_a, it should be clear that $M \in SEL_\varepsilon$. The SEA_ε-automaton $\mathcal{B}$ recognizing M is the disjoint union of the automata $\hat{\mathcal{A}}_a$ to which we add ε-transitions allowing to switch between automata: if p is a final state of $\hat{\mathcal{A}}_a$ and q is an initial state of $\hat{\mathcal{A}}_b$ and $a \in \Sigma_s \Rightarrow b \neq a$ then we add the switching ε-transition $(p, \texttt{true}, \varepsilon, X_b, q)$. All initial (resp. final, repeated) states of the $\hat{\mathcal{A}}_a$'s are initial (resp. final, repeated) in $\mathcal{B}$. But we also need to accept runs that switch infinitely often between the $\hat{\mathcal{A}}_a$'s, i.e., taking infinitely many switching ε-transitions. If needed, it is easy to transform the automaton so that it uses classical Büchi condition to accept also these runs.

We show that any *SE*-word w accepted by $\mathcal{B}$ is in M. Consider an acceting path P in $\mathcal{B}$ such that w admits a run through P. We split P according to the switching ε-transitions:

$$P = P_1 \xrightarrow{\varepsilon, X_{a_2}} P_2 \xrightarrow{\varepsilon, X_{a_3}} \cdots$$

where each P_i is a path in $\hat{\mathcal{A}}_{a_i}$. Note that, since P is accepting, using the definition of $\mathcal{B}$ we deduce that each P_i must be accepting. Let $w = w_1 w_2 \cdots$ be the corresponding factorization so that each w_i is accepted by $\hat{\mathcal{A}}_{a_i}$ through the path P_i. Hence, we have $w_i \in M_{a_i}$ and by definition of $\mathcal{B}$ we also get that $a_i \in \Sigma_s$ implies $a_{i+1} \neq a_i$. Therefore, $w \in M$.

Condition (†) is needed to prove the converse. Indeed, a finite *SE*-word $v \in M_a$ with $\|v\| < \infty$ could appear as an internal factor of some *SE*-word $uvw \in M$ with $w \neq \varepsilon$. If v could only be accepted by an infinite run in $\hat{\mathcal{A}}_a$, then we would not be able to build a (non transfinite) run for uvw in $\mathcal{B}$. Formally, let $w = w_1 w_2 \cdots \in M$ and $a_1, a_2, \ldots \in \Sigma_e \cup \Sigma_s$ with $w_i \in M_{a_i}$ and such that $a_i \in \Sigma_s$ implies $a_{i+1} \neq a_i$. For each $i \geq 1$, consider an accepting path P_i for w_i in $\hat{\mathcal{A}}_{a_i}$. If $\|w_i\| < \infty$ then w_i must be finite since by definition of a substitution, the automaton $\mathcal{A}_{a_i}$ does not accept any Zeno word, and by construction the same holds for $\hat{\mathcal{A}}_{a_i}$. Since $\hat{\mathcal{A}}_{a_i}$ satisfies (†) we deduce that $\|w_i\| < \infty$ implies P_i finite. Note that if $\|w_i\| = \infty$ then w_i must be the last factor of w. From the above, we can link the paths P_i with switching ε-transitions and we get a path

$$P = P_1 \xrightarrow{\varepsilon, X_{a_2}} P_2 \xrightarrow{\varepsilon, X_{a_3}} \cdots$$

of $\mathcal{B}$ which is clearly accepting. It is now easy to show that w admits a run through P and we obtain $w \in \mathcal{L}(\mathcal{B})$. □

The class *SEL* is not closed under inverse *SEL*-substitutions. Indeed, assume $\Sigma_s = \Sigma'_s = \{a, b\}$, $\Sigma_e = \Sigma'_e = \{f\}$ and let σ be the *SEL*-substitution defined by $L_a = \{a^0 f\}$, $L_b = \{b^0\}$ and $L_f = \{f\}$. Then the inverse image by σ of the *SEL*-language $\{a^0 f b^0\}$ is the language $\{a^0 b^0\}$ which is not in the class *SEL* since it contains a *SE*-word with two consecutive and distinct signals. Nevertheless sufficient conditions on the substitutions can be proposed to ensure the closure of the class *SEL* under inverse substitutions.

Theorem 5. *The class SEL is closed under inverse SEL-substitutions acting on events only.*

Proof. Let σ be a substitution defined by a family $(L_a)_{a \in \Sigma_e \cup \Sigma_s}$ of *SEL*-languages. We assume that σ acts only on events, i.e., $L_a = \{a\} \times \overline{\mathbb{T}}$ for all $a \in \Sigma_s \subseteq \Sigma'_s$. For $f \in \Sigma_e$, let $\mathcal{A}_f = (\Sigma'_e, \Sigma'_s, \{x_f\}, Q_f, Q^0_f, F_f, \emptyset, I_f, \ell_f, \Delta_f)$ be an automaton without ε-transitions accepting L_f. Since the guards and resets on the transitions of $\mathcal{A}_f$ are always `true` and $\emptyset$ respectively, we write a transition of $\mathcal{A}_f$ simply (r, f', s)

to simplify the notation. We consider now a language $L \in SEL$, recognized by some automaton without ε-transition $\mathcal{A}_2 = (\Sigma_e', \Sigma_s, X, Q, Q^0, F, R, I, \ell, \Delta) \in SEA$.

We build an automaton $\mathcal{A}_1$ in SEA accepting $\sigma^{-1}(L)$ essentially by changing the transitions of $\mathcal{A}_2$. If there is a path P in $\mathcal{A}_2$ from p to q having some instantaneous run for some word in $\sigma(f)$ then we add to $\mathcal{A}_1$ a transition (p, g, f, α, q) with a suitable guard g and reset α. The difficulty is to compute a suitable pair (g, α) for each triple (p, q, f).

Given a guard g and a subset of clocks α, we define the restriction of g by α, written $g[\alpha]$, as the guard g where all clocks from α have been replaced by 0. For instance, if g is $(x < 3) \wedge (y > 2)$ and $\alpha = \{x, z\}$ then $g[\alpha]$ is (equivalent to) $(y > 2)$ and if g is $(x > 3) \wedge (y < 2)$ and $\alpha = \{x, z\}$ then $g[\alpha]$ is (equivalent to) `false`. We let $\mathcal{G}$ be the smallest set of guards including all guards of $\mathcal{A}_2$ and closed under conjunctions and restrictions. Formally $\mathcal{G}$ is not a finite set, but it can be identified with its finite quotient under equivalence of formulae: two formulae φ and ψ are equivalent if $v \models \varphi$ iff $v \models \psi$ for all valuations v. The set $\Gamma = \mathcal{G} \times \mathcal{P}(X)$ is thus a finite monoid, with $(True, \emptyset)$ as neutral element, for the associative composition:

$$(g_1, \alpha_1) \cdot (g_2, \alpha_2) = (g_1 \wedge g_2[\alpha_1], \alpha_1 \cup \alpha_2).$$

Finally, we define a morphism $\gamma : \Delta^* \mapsto \Gamma$ by $\gamma((p, g, f, \alpha, q)) = (g, \alpha)$. Note that, for $Z \subseteq \Gamma$, the sequence $Z^{\leq n} = \bigcup_{k=0}^{n} Z^k$ is increasing and since Γ is finite it must be eventually stationary. Moreover, if $Z^{\leq n} = Z^{\leq n+1}$ then $Z^* = Z^n$. Therefore, we can effectively compute the finite set Z^* for any $Z \subseteq \Gamma$. We deduce that we can effectively compute the image by γ of any rational expression over Δ. This fact will be used below.

Let $P = q_0 \xrightarrow{g_1, b_1, \alpha_1} q_1 \xrightarrow{g_2, b_2, \alpha_2} \cdots \xrightarrow{g_n, b_n, \alpha_n} q_n$ be a path in $\mathcal{A}_2$. We define $\mathcal{W}(P) = \{\varepsilon, \ell(q_0)^0\} b_1 \ell(q_1)^0 b_2 \cdots b_n \{\varepsilon, \ell(q_n)^0\}$. Note that $\mathcal{W}(P)$ contains 1, 2 or 4 SE-words depending on the whether the labels of q_0 and q_n are different from τ or not. Now, given a triple $(p, q, f) \in Q^2 \times \Sigma_e$, we denote by $L_{p,q}^f$ the set of paths P from p to q in $\mathcal{A}_2$ with $\mathcal{W}(P) \cap \sigma(f) \neq \emptyset$. We build an automaton $\mathcal{B}_{p,q}^f$ recognizing the language $L_{p,q}^f \subseteq \Delta^*$. This is not difficult since we are dealing with automata without ε-transitions, hence we can perform a simple synchronized product as follows. The set of states of $\mathcal{B}_{p,q}^f$ is $Q' = \{p, q\} \uplus \{(r, s) \in Q_f \times Q \mid \ell_f(r) = \ell(s)\}$. The only initial state is p and the only final state is q. The transitions of $\mathcal{B}_{p,q}^f$ are of the three following forms:

1. $(p, (p, g, b, \alpha, s_2), (r_2, s_2))$ if $(p, g, b, \alpha, s_2) \in \Delta$ and there exists an initial state $r_1 \in Q_f$ with $\ell(r_1) \in \{\tau, \ell(p)\}$ and such that $(r_1, b, r_2) \in \Delta_f$,
2. $((r_1, s_1), (s_1, g, b, \alpha, s_2), (r_2, s_2))$ if $(r_1, b, r_2) \in \Delta_f$ and $(s_1, g, b, \alpha, s_2) \in \Delta$,
3. $((r_1, s_1), (s_1, g, b, \alpha, q), q)$ if $(s_1, g, b, \alpha, q) \in \Delta$ and there is a final state $r_2 \in Q_f$ with $\ell(r_2) \in \{\tau, \ell(q)\}$ and such that $(r_1, b, r_2) \in \Delta_f$.

We show that $L_{p,q}^f \subseteq \mathcal{L}(\mathcal{B}_{p,q}^f)$. Let $P = p \xrightarrow{g_1, b_1, \alpha_1} q_1 \cdots q_{n-1} \xrightarrow{g_n, b_n, \alpha_n} q \in L_{p,q}^f$. Since $\mathcal{W}(P) \cap \sigma(f) \neq \emptyset$, there is in $\mathcal{A}_f$ an accepting path $r_0 \xrightarrow{b_1}$

$r_1 \cdots r_{n-1} \xrightarrow{b_n} r_n$ for some word in $\mathcal{W}(P)$. By definition of $\mathcal{W}(P)$, we deduce that $\ell_f(r_i) = \ell(q_i)$ for all $0 < i < n$. Therefore,

$$p \xrightarrow{(p,g_1,b_1,\alpha_1,q_1)} (r_1,q_1) \cdots (r_{n-1},q_{n-1}) \xrightarrow{(q_{n-1},g_n,b_n,\alpha_n,q)} q$$

is an accepting path for P in $\mathcal{B}^f_{p,q}$ and we get $P \in \mathcal{L}(\mathcal{B}^f_{p,q})$ as desired. The converse inclusion can be shown similarly.

We let $\Delta_R \subseteq \Delta^*$ be the set of sequences containing at least a transition ending in some repeated state of R. From the automaton $\mathcal{B}^f_{p,q}$, we can effectively compute a rational expression for the languages $L^f_{p,q} \cap \Delta_R$. We deduce that we can effectively compute the finite set $\gamma(L^f_{p,q} \cap \Delta_R)$. Similarly, we can effectively compute the finite set $\gamma(L^f_{p,q} \setminus \Delta_R)$. These two sets are used to define the transitions of a new automaton $\mathcal{A}_1 = (\Sigma_e, \Sigma_s, X, Q \uplus \overline{Q}, Q^0, F \uplus \overline{F}, \overline{Q}, I, \ell, \Delta_1)$ in *SEA*. The set $\overline{Q}$ is a disjoint copy of Q and for $q \in Q$, we let $I(\bar{q}) = I(q)$ and $\ell(\bar{q}) = \ell(q)$. The transitions are defined by:

$$\begin{aligned} \Delta_1 = \quad & \{(p,g,f,\alpha,q), (\overline{p},g,f,\alpha,q) \mid (g,\alpha) \in \gamma(L^f_{p,q} \setminus \Delta_R)\} \\ & \cup \{(p,g,f,\alpha,\overline{q}), (\overline{p},g,f,\alpha,\overline{q}) \mid (g,\alpha) \in \gamma(L^f_{p,q} \cap \Delta_R)\}. \end{aligned}$$

The automaton $\mathcal{A}_1$ can therefore be effectively computed from the automata $\mathcal{A}_2$ and $(\mathcal{A}_f)_{f \in \Sigma_e}$.

Correctness of the construction. We show that $\mathcal{L}(\mathcal{A}_1) = \sigma^{-1}(L)$.

Let $u \in \mathcal{L}(\mathcal{A}_1)$ be accepted through a path $P = q'_0 \xrightarrow{g_1,f_1,\alpha_1} q'_1 \xrightarrow{g_2,f_2,\alpha_2} q'_2 \cdots$ in $\mathcal{A}_1$. For each $i \geq 0$, let $q_i \in Q$ such that $q'_i \in \{q_i, \bar{q}_i\}$ and let $a_i = \ell(q_i)$. Hence, we have $u = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ for suitable $d_i \in \overline{\mathbb{T}}$. We can build from each transition $(q'_{i-1}, g_i, f_i, \alpha_i, q'_i)$ of $\mathcal{A}_1$ a path $P_i \in L^{f_i}_{q_{i-1},q_i}$ such that $(g_i, \alpha_i) = \gamma(P_i)$ and $P_i \in \Delta_R$ if $q'_i = \bar{q}_i$. This gives a path $P' = P_1 P_2 \cdots$ in $\mathcal{A}_2$. Since P is accepting in $\mathcal{A}_2$, we deduce easily that P' is accepting in $\mathcal{A}_1$. By definition of $L^{f_i}_{q_{i-1},q_i}$, we find a word $w_i \in \mathcal{W}(P_i) \cap \sigma(f_i)$. Note that w_i starts either with a^0_{i-1} or by en event in Σ_e and it ends either with a^0_i or by an event in Σ_e. Since the factorization $a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ is the normal form of u, we obtain $w = a_0^{d_0} w_1 a_1^{d_1} w_2 \cdots \in \sigma(u)$. To prove that w is accepted by $\mathcal{A}_1$ through the path P', we use the following

Claim. Let $r_0 \xrightarrow{h_1,e_1,\beta_1} r_1 \xrightarrow{h_2,e_2,\beta_2} \cdots \xrightarrow{h_n,e_n,\beta_n} r_n$ be a path in $\mathcal{A}_2$ whose image by γ is (h, β), and let v_0 be a clock valuation such that $v_0 \models h$. Then, there is an instantaneous run $(r_0, v_0) \xrightarrow{e_1} (r_1, v_1) \xrightarrow{0} (r_1, v_1) \xrightarrow{e_2} \cdots \xrightarrow{e_n} (r_n, v_n)$ through this path and $v_n = v_0[\beta]$.

The proof of the claim is based on the following property of restriction: let h be a guard, β a subset of clocks and v a valuation, then $v \models h[\beta]$ if and only if $v[\beta] \models h$. Now, for $i \geq 0$, we let $Y_i = \beta_1 \cup \cdots \cup \beta_i$ (note that $Y_0 = \emptyset$). By definition of γ, we have $\beta = Y_n$ and $h = \bigwedge_{i=1}^n h_i[Y_{i-1}]$. Therefore, $v_0 \models h_1 = h_1[Y_0]$. By induction, we get $v_i = v_{i-1}[\beta_i] = v_0[Y_i]$ and since $v_0 \models h_{i+1}[Y_i]$, using the property above we get $v_i = v_0[Y_i] \models h_{i+1}$ and the claim is proved.

Using this claim, we deduce easily by induction that w admits a run through P'. Therefore, $w \in L$ and $u \in \sigma^{-1}(L)$.

Conversely, let $u \in \sigma^{-1}(L)$. By definition, we find $w \in \sigma(u) \cap L$. In the normal form of u we cannot have two consecutive signals, otherwise, since the substitution σ acts on events only, the same would be true in the normal form of w and w could not be accepted by $\mathcal{A}_2$ which has no ε-transitions. Therefore, we can write $u = a_0^{d_0} f_1 a_1^{d_1} f_2 \cdots$ with $f_i \in \Sigma_e$ and $a_i \in \Sigma_s$. It follows that $w = a_0^{d_0} w_1 a_1^{d_1} w_2 \cdots$ with $w_i \in \sigma(f_i)$.

An accepting path in $\mathcal{A}_2$ for w can be written as $P = P_1 P_2 \cdots$, where P_i is a path such that $w_i \in \mathcal{W}(P_i)$ and if we let q_0 be the first state of P (hence also of P_1) and q_i be the last state of P_i (which is also the first state of P_{i+1}) then $\ell(q_i) = a_i$ for all $i \geq 0$. Moreover, P_i is a path in $\mathcal{A}_2$ from q_{i-1} to q_i and $w_i \in \mathcal{W}(P_i) \cap \sigma(f_i) \neq \emptyset$, so that $P_i \in L^{f_i}_{q_{i-1},q_i}$ for all $i > 0$. Now, we define inductively a path P' of $\mathcal{A}_1$. We start with $q'_0 = q_0$. Assuming that q'_{i-1} has been defined, we let $(g_i, \alpha_i) = \gamma(P_i)$ and $q'_i = q_i$ if $P_i \notin \Delta_R$, or $q'_i = \overline{q}_i$ otherwise. We obtain in this way a transition $(q'_{i-1}, g_i, f_i, \alpha_i, q'_i)$ in Δ_1. Since P is accepting, the new path P' is also accepting. To show that u admits a run through P' we use the following

Claim. Let $r_0 \xrightarrow{h_1,e_1,\beta_1} r_1 \xrightarrow{h_2,e_2,\beta_2} \cdots \xrightarrow{h_n,e_n,\beta_n} r_n$ be a path in $\mathcal{A}_2$ whose image by γ is (h, β), and let $v_0, v_1, \ldots, v_n$ be clock valuations such that $v_i = v_{i-1}[\beta_i]$ and $v_{i-1} \models h_i$ for each $0 < i \leq n$. Then, $v_0 \models h$ and $v_n = v_0[\beta]$.

For $i \geq 0$, we let $Y_i = \beta_1 \cup \cdots \cup \beta_i$ (note that $Y_0 = \emptyset$) so that we get by induction $v_i = v_0[Y_i]$. By definition of γ, we have $\beta = Y_n$ and $h = \bigwedge_{i=1}^{n} h_i[Y_{i-1}]$. Therefore, $v_n = v_0[\beta]$ and using the property of restriction we get $v_0[Y_{i-1}] = v_{i-1} \models h_i$ hence $v_0 \models h_i[Y_{i-1}]$ for all $i > 0$. Therefore, $v_0 \models h$ which concludes the proof of the claim.

Using this claim, we show easily by induction that u admits a run through P'. Therefore, $u \in \mathcal{L}(\mathcal{A}_1)$ and Theorem 5 is proved. □

6 Conclusion

We have shown in this paper that the class *SEL* of signal-event languages is not closed under arbitrary *SEL*-substitutions and inverse *SEL*-substitutions but that natural sufficient conditions ensure closure properties for this class.

But our main contribution is to propose effective constructions to prove the closure of the larger class SEL_ε under arbitrary SEL_ε-substitutions and inverse SEL_ε-substitutions. We give these constructions in the general framework of signal-event automata and languages. The usual cases of event languages [1, 2] or signal languages [3, 11, 4, 12] are particular cases for which the interested reader will verify that simplified constructions can be given.

Our proofs use a preliminary technical result showing that the intersection of two signal-event automata can be effectively constructed. While constructions were proposed in the literature for important particular cases, it is the first time, up to our knowledge, that 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 - is treated. There has been in the area of timed automata some examples of subtly erroneous constructions (e.g. with respect to forward analysis [9]) which should convince us of the importance to publish complete and proved constructions.

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. Intersection of regular signal-event (timed) languages. In Eugène Asarin and Patricia Bouyer, editors, *Proceedings of FORMATS'06*, number 4202 in LNCS, pages 52–66. Springer, 2006.
7. B. Bérard, P. Gastin, and A. Petit. Refinements and abstractions of signal-event (timed) languages. In Eugène Asarin and Patricia Bouyer, editors, *Proceedings of FORMATS'06*, number 4202 in LNCS, pages 67–81. Springer, 2006.
8. B. Bérard and C. Picaronny. Accepting Zeno words: a way toward timed refinements. *Acta Informatica*, 37(1):45–81, 2000.
9. P. Bouyer. Forward analysis of updatable timed automata. *Formal Methods in System Design*, 24(3):281–320, May 2004.
10. 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.
11. 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.
12. J. Durand-Lose. A Kleene theorem for splitable signals. *Information Processing Letters*, 89:237–245, 2004.
13. M.R. Hansen, P.K. Pandya, and C. Zhou. Finite divergence. *Theoretical Computer Science*, 138:113–139, 1995.
14. D. Perrin and J.-E. Pin. *Infinite words.* Elsevier, 2004.