Comparison of Expressiveness for Timed Automata and Time Petri Nets

Béatrice Berard, F. Cassez, Serge Haddad, D. Lime, Olivier Henri Roux

Comparison of Expressiveness for Timed Automata and Time Petri Nets

B. Bérard[*], F. Cassez[†], S. Haddad[*], D. Lime[‡], O.H. Roux[†]

Abstract

In this paper we consider the model of Time Petri Nets (TPN) "à la Merlin" where a time interval is associated with the firing of a transition, but we extend it with open intervals. We also consider Timed Automata (TA) as defined by Alur & Dill. We investigate some questions related to expressiveness for these models : we study the impact of slight variations of semantics for TPN and we compare the expressive power of TA and TPN, with respect to both time language acceptance and weak time bisimilarity. We prove that TA and bounded TPNs (enlarged with strict constraints) are equivalent w.r.t. timed language equivalence, providing an efficient construction of a TPN equivalent to a TA. We then exhibit a TA $\mathcal{A}$ such that no TPN (even unbounded) is weakly bisimilar to $\mathcal{A}$. Because of this last result, it is natural to try and identify the (strict) subclass of TA that is equivalent to TPN w.r.t. weak timed bisimilarity. Thus we give some further results: 1) we characterize the subclass TA^- of TA that is equivalent to the original model of TPN as defined by Merlin, *i.e.* restricted to closed intervals, 2) we show that the associated membership problem for TA^- is $PSPACE$-complete and 3) we prove that the reachability problem for TA^- is also $PSPACE$-complete.

Key words : Time Petri Nets, Timed Automata, Timed Languages, Timed Bisimilarity, Expressiveness.

1 Introduction

Petri Nets with Time. The two main extensions of Petri Nets with time are Time Petri Nets (TPNs) [15] and Timed Petri Nets [17]. For TPNs a transition can fire within a time

[*]LAMSADE, Paris, France. {beatrice.berard,serge.haddad}@lamsade.dauphine.fr
[†]IRCCyN, Nantes, France. {Franck.Cassez,Olivier-h.Roux}@irccyn.ec-nantes.fr
[‡]CISS, Aalbork, Denmark, Didier@cs.aau.dk

interval whereas for Timed Petri Nets it fires as soon as possible. Among Timed Petri Nets, time can be considered relative to places or transitions [18, 16]. The two corresponding subclasses namely P-Timed Petri Nets and T-Timed Petri Nets are expressively equivalent [18, 16]. The same classes are defined for TPNs i.e. T-TPNs and P-TPNs, but both classes of Timed Petri Nets are included in both P-TPNs and T-TPNs [16]. P-TPNs and T-TPNs are proved to be incomparable in [13]. Finally TPNs form a subclass of Time Stream Petri Nets [10] which were introduced to model multimedia applications. Timed Arc Petri Nets are also studied in more recent work [1, 9].

Timed Automata. Timed Automata (TA) were introduced by Alur & Dill [3] and have since been extensively studied. This model is an extension of finite automata with (dense time) *clocks* and enables one to specify real-time systems. Theoretical properties of various classes of TA have been considered in the last decade. For instance, a class of determinizable TA such as *Event Clock Automata* are investigated in [4] and form a strict subclass of TA. More general models of TA like *Rectangular Automata* or *Linear Hybrid Automata* have also been considered and their expressive power compared.

However, not much is known about the expressive power of TPN compared to TA.

Related Work. In a previous work [8] we have proved that TPN forms a subclass of TA in the sense that every TPN can be simulated by a TA (weak timed bisimilarity). A similar result can be found in [14] with a completely different approach.

In another line of work [12], Haar, Kaiser, Simonot & Toussaint compare Timed State Machines and Time Petri Nets. They give a translation from one model to another that preserves timed languages. Nevertheless, in the translation from TSM to TPN they use a *weak* semantics for TPN and consider only the constraints with bounded and closed intervals.

Our Contribution. In this article, we compare precisely the expressive power of TA vs. TPN using the notions of *Timed Language Acceptance* and *Timed Bisimilarity*. This extends previous results in this area in the following directions: *i)* we consider general types of constraints (strict, large, bounded, unbounded); *ii)* we then show that there is a TA $\mathcal{A}$ s.t. no TPN is (even weakly) timed bisimilar to $\mathcal{A}$; *iii)* this leads us to consider weaker notions of equivalence and we focus on Timed Language Acceptance. We prove that TA (with general types of constraints) and TPN are equally expressive w.r.t. Timed Language Acceptance; *iv)* to conclude we characterize the subclass of TA that is equally expressive to TPN without strict constraints w.r.t. Timed Bisimilarity, and show that the membership problem for this class is $PSPACE$-complete as well the reachability problem. The results of the paper are summarized in Table 1: all the results are new except the one on the first line obtained in [8]. We use the following notations: B-$\mathcal{TPN}_\varepsilon$ for the set of bounded TPNs with ε-transitions; 1-B-$\mathcal{TPN}_\varepsilon$ for the subset of B-$\mathcal{TPN}_\varepsilon$ with at most one token in each place (one safe TPN); B-$\mathcal{TPN}(\leq, \geq)$ for the subset of B-$\mathcal{TPN}_\varepsilon$ where only closed

	Timed Language Acceptance	**Timed Bisimilarity**
B-$\mathcal{TPN}_\varepsilon$	$\leq_{\mathcal{L}} \mathcal{TA}_\varepsilon$ ([8])	$\leq_{\mathcal{W}} \mathcal{TA}_\varepsilon$ ([8])
	$=_{\mathcal{L}}$ 1-B-$\mathcal{TPN}_\varepsilon$ $=_{\mathcal{L}} \mathcal{TA}_\varepsilon$	$<_{\mathcal{W}} \mathcal{TA}_\varepsilon$
B-$\mathcal{TPN}(\leq,\geq)$	—	$\approx_{\mathcal{W}} \mathcal{TA}^-$

	Emptyness Problem	**Universal Problem**
B-$\mathcal{TPN}_\varepsilon$	Decidable	Undecidable

	Membership Problem	**Reachability Problem**
$\mathcal{TA}^-$	$PSPACE$-complete	

Table 1: Summary of the Results

intervals are used; $\mathcal{TA}_\varepsilon$ for TA with ε-transitions; $\mathcal{TA}^-$ for the class of TA (to be defined precisely in section 6) that is equivalent to B-$\mathcal{TPN}(\leq,\geq)$.

Outline of the paper. Section 2 introduces the semantics of TPNs and TA, Timed Languages and Timed Bisimilarity and section 3 is devoted to the comparison between various semantics for TPNs. In section 4, we prove negative results: we exhibit some timed automata for which there exist no (weakly) timed bisimilar TPN. In section 5 we focus on Timed Language Acceptance and prove that TA and TPNs are equally expressive w.r.t. this equivalence. Section 6 is devoted to a characterization of the subclass of TA that is equivalent to TPN w.r.t. Timed Bisimilarity. Finally we give some hints on further work in section 7.

2 Time Petri Nets and Timed Automata

Notations. Let Σ be a set (or alphabet). Σ^* (resp. Σ^ω) denotes the set of finite (resp. infinite) sequences of elements (or words) of Σ and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. By convention if $w \in \Sigma^\omega$ then the *length* of w denoted $|w|$ is ω; otherwise if $w = a_1 \cdots a_n$, $|w| = n$. We also use $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ with $\varepsilon \notin \Sigma$, where ε is the empty word. B^A stands for the set of mappings from A to B. If A is finite and $|A| = n$, an element of B^A is also a vector in B^n. The usual operators $+, -, <$ and $=$ are used on vectors of A^n with $A = \mathbb{N}, \mathbb{Q}, \mathbb{R}$ and are the point-wise extensions of their counterparts in A. The set $\mathbb{B}$ denotes the boolean values $\{\mathsf{tt}, \mathsf{ff}\}$ and $\mathbb{R}_{\geq 0}$ denotes the set of positive reals. A *valuation* ν over a set of variables X is an element of $\mathbb{R}_{\geq 0}^X$. For $\nu \in \mathbb{R}_{\geq 0}^X$ and $d \in \mathbb{R}_{\geq 0}$, $\nu + d$ denotes the valuation defined by $(\nu + d)(x) = \nu(x) + d$, and for $X' \subseteq X$, $\nu[X' \mapsto 0]$ denotes the valuation ν' with $\nu'(x) = 0$ for $x \in X'$ and $\nu'(x) = \nu(x)$ otherwise. $\mathbf{0}$ denotes the valuation s.t. $\forall x \in X, \nu(x) = 0$. An *atomic constraint* is a formula of the form $x \bowtie c$ for $x \in X$,

$c \in \mathbb{Q}_{\geq 0}$ and $\bowtie \in \{<, \leq, \geq, >\}$. We denote $\mathcal{C}(X)$ the set of *constraints* over a set of variables X which consists of the conjunctions of atomic constraints. Given a constraint $\varphi \in \mathcal{C}(X)$ and a valuation $\nu \in \mathbb{R}_{\geq 0}^X$, we denote $\varphi(\nu) \in \mathbb{B}$ the truth value obtained by substituting each occurrence of x in φ by $\nu(x)$. Accordingly each constraint $\varphi \in \mathcal{C}(X)$ defines a set of valuations $[\![\varphi]\!]$ defined by $[\![\varphi]\!] = \{\nu \in \mathbb{R}_{\geq 0}^X \mid \varphi(\nu) = \mathsf{tt}\}$.

A set I is a $\mathbb{Q}_{\geq 0}$-*interval* of $\mathbb{R}_{\geq 0}$ if there is a constraint φ of the form $a \prec_1 x \prec_2 b$ with $a \in \mathbb{Q}_{\geq 0}$, $b \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$ and $\prec_1, \prec_2 \in \{<, \leq\}$, such that $I = [\![\varphi]\!]$. We let $I^{\downarrow} = [\![0 \leq x \prec_2 b]\!]$ be the *downward closure* of I and $I^{\uparrow} = [\![a \prec_1 x]\!]$ be the *upward closure* of I. We denote by $\mathcal{I}(\mathbb{Q}_{\geq 0})$ the set of $\mathbb{Q}_{\geq 0}$-intervals of $\mathbb{R}_{\geq 0}$. Let $g \in \mathbb{N}_{>0}$, we write $\mathbb{N}_g = \{\frac{i}{g} \mid i \in \mathbb{N}\}$. A vector $\mathbf{v} \in \mathbb{Q}^n$ belongs to the *g-grid* if $\mathbf{v}(k) \in \mathbb{N}_g$ for all $1 \leq k \leq n$.

2.1 Timed Transition Systems and Equivalence Relations

Let Σ be a fixed finite alphabet s.t. $\varepsilon \notin \Sigma$.

Definition 1 (Timed Words) *A* timed word w *over* Σ_ε *is a finite or infinite sequence* $w = (a_0, d_0)(a_1, d_1) \cdots (a_n, d_n) \cdots$ *s.t. for each* $i \geq 0$, $a_i \in \Sigma_\varepsilon$, $d_i \in \mathbb{R}_{\geq 0}$ *and* $d_{i+1} \geq d_i$.

A timed word w over Σ_ε can be viewed as a pair $(v, \tau) \in \Sigma_\varepsilon^\infty \times \mathbb{R}_{\geq 0}^\infty$ s.t. $|v| = |\tau|$. The value d_k gives the absolute time (from the initial instant 0) of action a_k. We write *Untimed*$(w) = v$ for the untimed part of w, and *Duration*$(w) = \sup_{d_k \in \tau} d_k$ for the duration of the timed word w. For a timed word $(a_i, d_i)_{i \geq 0}$ we define the *relative time stamp* *Rstamp*(a_i) of a_i as *Rstamp*$(a_i) = d_i - d_{i-1}$ with the convention that $d_{-1} = 0$ and extend this notion to timed words by defining *Rstamp*$(w) = (a_i, \textit{Rstamp}(a_i))_{i \geq 0}$. Note that, conversely, from such a sequence, we can retrieve a timed word with *absolute time stamps* by cumulating the successive delays.

Since ε-transitions correspond to the empty word and are not visible, we can remove from each timed word $w \in \Sigma_\varepsilon^\infty \times \mathbb{R}_{\geq 0}^\infty$ all the ε-actions and obtain a timed word in $\Sigma^\infty \times \mathbb{R}_{\geq 0}^\infty$.

Definition 2 (Timed Languages) *We denote by* $\mathcal{TW}^*(\Sigma)$ *(resp.* $\mathcal{TW}^\omega(\Sigma)$*) the set of finite (resp. infinite) timed words over* Σ *and* $\mathcal{TW}^\infty(\Sigma) = \mathcal{TW}^*(\Sigma) \cup \mathcal{TW}^\omega(\Sigma)$. *A* timed language L *over* Σ *is any subset of* $\mathcal{TW}^\infty(\Sigma)$.

Timed transition systems describe systems which combine discrete and continuous evolutions.

Definition 3 (Timed Transition Systems) *A* timed transition system (TTS) *over the set of actions* Σ_ε *is a tuple* $S = (Q, Q_0, \Sigma_\varepsilon, \longrightarrow, F, R)$ *where* Q *is a set of states,* $Q_0 \subseteq Q$ *is the set of initial states,* Σ_ε *is a finite set of actions disjoint from* $\mathbb{R}_{\geq 0}$ *and* $\longrightarrow \subseteq Q \times (\Sigma_\varepsilon \cup$

$\mathbb{R}_{\geq 0}) \times Q$ *is a set of edges. If* $(q, e, q') \in \longrightarrow$*, we also write* $q \xrightarrow{e} q'$*. For a transition* $q \xrightarrow{d} q'$ *with* $d \in \mathbb{R}_{\geq 0}$*, the value* d *represents a relative time stamp. The sets* $F \subseteq Q$ *and* $R \subseteq Q$ *are respectively the sets of* final *and* repeated *states.*

We make the following common assumptions about TTS:

- 0-DELAY: $q \xrightarrow{0} q'$ if and only if $q = q'$,
- ADDITIVITY: if $q \xrightarrow{d} q'$ and $q' \xrightarrow{d'} q''$ with $d,\ d' \in \mathbb{R}_{\geq 0}$, then $q \xrightarrow{d+d'} q''$,
- CONTINUITY: if $q \xrightarrow{d} q'$, then for every d' and d'' in $\mathbb{R}_{\geq 0}$ such that $d = d' + d''$, there exists q'' such that $q \xrightarrow{d'} q'' \xrightarrow{d''} q'$,
- TIME-DETERMINISM: if $q \xrightarrow{d} q'$ and $q \xrightarrow{d} q''$ with $d \in \mathbb{R}_{\geq 0}$, then $q' = q''$.

A *run* ρ of length $n \geq 0$ is is a finite or infinite ($n = \omega$) sequence of transitions of the form

$$\rho = q_0 \xrightarrow{d_0} q'_0 \xrightarrow{a_0} q_1 \xrightarrow{d_1} q'_1 \xrightarrow{a_1} \cdots q_n \xrightarrow{d_n} q'_n \ldots$$

where discrete actions alternate with durations. We write $\mathit{first}(\rho) = q_0$ and if ρ is finite, we assume that it ends with an action transition and we set $\mathit{last}(\rho) = q_n$. We write $q \xrightarrow{*} q'$ if there is a run ρ s.t. $\mathit{first}(\rho) = q$, $\mathit{last}(\rho) = q'$.

A run is *initial* if $\mathit{first}(\rho) \in Q_0$. A run ρ is *accepting* if i) either ρ is a finite initial run and $\mathit{last}(\rho) \in F$ or ii) ρ is infinite and there is a state $q \in R$ that appears infinitely often on ρ. From the sequence $(a_0, d_0)(a_1, d_1) \ldots$ associated with ρ, we obtain a timed word w by considering the absolute time stamps of actions : $w = (a_0, d_0)(a_1, d_0 + d_1) \ldots$. This word is *accepted* by S if ρ is an accepting run.

The *timed language* $\mathcal{L}(S)$ accepted by S is the set of timed words accepted by S.

Definition 4 (Strong Timed Similarity) *Let* $S_1 = (Q_1, Q_0^1, \Sigma_\varepsilon, \longrightarrow_1, F_1, R_1)$ *and* $S_2 = (Q_2, Q_0^2, \Sigma_\varepsilon, \longrightarrow_2, F_2, R_2)$ *be two TTS and* $\preceq$ *be a binary relation over* $Q_1 \times Q_2$*. We write* $s \preceq s'$ *for* $(s, s') \in \preceq$*. The relation* $\preceq$ *is a* strong (timed) simulation relation *of* S_1 *by* S_2 *if:*

1. *if* $s_1 \in F_1$ *(resp.* $s_1 \in R_1$*) and* $s_1 \preceq s_2$ *then* $s_2 \in F_2$ *(resp.* $s_2 \in R_2$*),*
2. *if* $s_1 \in Q_0^1$ *there is some* $s_2 \in Q_0^2$ *s.t.* $s_1 \preceq s_2$*;*
3. *if* $s_1 \xrightarrow{d}_1 s'_1$ *with* $d \in \mathbb{R}_{\geq 0}$ *and* $s_1 \preceq s_2$ *then* $s_2 \xrightarrow{d}_2 s'_2$ *for some* s'_2*, and* $s'_1 \preceq s'_2$*;*
4. *if* $s_1 \xrightarrow{a}_1 s'_1$ *with* $a \in \Sigma_\varepsilon$ *and* $s_1 \preceq s_2$ *then* $s_2 \xrightarrow{a}_2 s'_2$ *and* $s'_1 \preceq s'_2$*;*

A TTS S_2 strongly simulates S_1 *if there is a strong (timed) simulation relation of* S_1 *by* S_2. *We write* $S_1 \preceq_{\mathcal{S}} S_2$ *in this case.*

When there is a strong simulation relation $\preceq$ of S_1 by S_2 and $\preceq^{-1}$ is also a strong simulation relation[1] of S_2 by S_1, we say that $\preceq$ is a *strong (timed) bisimultion relation* between S_1 and S_2 and use $\approx$ instead of $\preceq$. Two TTS S_1 and S_2 are *strongly (timed) bisimilar* if there exists a strong (timed) bisimulation relation between S_1 and S_2. We write $S_1 \approx_{\mathcal{S}} S_2$ in this case.

Let $S = (Q, Q_0, \Sigma_\varepsilon, \longrightarrow, F, R)$ be a TTS. We define the ε-abstract TTS $S^\varepsilon = (Q, Q_0^\varepsilon, \Sigma, \longrightarrow_\varepsilon, F, R)$ (with no ε-transitions) by:

- $s \xrightarrow{d}_\varepsilon s'$ iff there is a run $\rho = s \xrightarrow{*} s'$ with *Untimed*$(\rho) = \varepsilon$ and *Duration*$(\rho) = d$,
- $s \xrightarrow{a}_\varepsilon s'$ with $a \in \Sigma$ iff there is a run $\rho s \xrightarrow{*} s'$ with *Untimed*$(\rho) = a$ and *Duration*$(\rho) = 0$,
- $Q_0^\varepsilon = \{s \mid \exists s' \in Q_0 \mid \ s' \xrightarrow{w} s$ *and Duration*$(w) = 0 \wedge$ *Untimed*$(w) = \varepsilon\}$.

Definition 5 (Weak Time Similarity) *Let* $S_1 = (Q_1, Q_0^1, \Sigma_\varepsilon, \longrightarrow_1, F_1, R_1)$ *and* $S_2 = (Q_2, Q_0^2, \Sigma_\varepsilon, \longrightarrow_2, F_2, R_2)$ *be two TTS and* $\preceq$ *be a binary relation over* $Q_1 \times Q_2$. $\preceq$ *is a* weak (timed) simulation relation *of* S_1 *by* S_2 *if it is a strong timed simulation relation of* S_1^ε *by* S_2^ε. *A TTS* S_2 weakly simulates S_1 *if there is a weak (timed) simulation relation of* S_1 *by* S_2. *We write* $S_1 \preceq_{\mathcal{W}} S_2$ *in this case.*

When there is a weak simulation relation $\preceq$ of S_1 by S_2 and $\preceq^{-1}$ is also a weak simulation relation of S_2 by S_1, we say that $\preceq$ is a *weak (timed) bisimulation relation* between S_1 and S_2 and use $\approx$ instead of $\preceq$. Two TTS S_1 and S_2 are *weakly (timed) bisimilar* if there exists a weak (timed) bisimulation relation between S_1 and S_2. We write $S_1 \approx_{\mathcal{W}} S_2$ in this case.

Note that if $S_1 \preceq_{\mathcal{S}} S_2$ then $S_1 \preceq_{\mathcal{W}} S_2$ and if $S_1 \preceq_{\mathcal{W}} S_2$ then $\mathcal{L}(S_1) \subseteq \mathcal{L}(S_2)$. Moreover, proving that $S_1 \preceq_{\mathcal{W}} S_2$ usually amounts to proving that if $q_1 \preceq q_2$, then each move $q_1 \xrightarrow{e}_1 q_1'$ can be simulated by a set of moves $q_2 \xrightarrow{e}_{2,\varepsilon} q_2'$ s.t. $q_2 \preceq q_2'$.

Let $S = (Q, Q_0, \Sigma_\varepsilon, \longrightarrow, F, R)$ be a TTS. We define the *time-abstract* TTS $S^\Delta = (Q, Q_0, \Sigma_\varepsilon \cup \{\delta\} \longrightarrow_\Delta, F, R)$ with $\delta \notin \Sigma_\varepsilon$ by:

- $s \xrightarrow{\delta}_\Delta s'$ iff $s \xrightarrow{d} s'$ for some $d \in \mathbb{R}_{\geq 0}$,
- $s \xrightarrow{a}_\Delta s'$ with $a \in \Sigma$ iff $s \xrightarrow{a} s'$ for some $a \in \Sigma_\varepsilon$.

Notice that S^Δ has no transition $s \xrightarrow{d} s'$ with $d \in \mathbb{R}_{\geq 0}$.

[1] $s_2 \preceq^{-1} s_1 \iff s_1 \preceq s_2$.

2.2 Time Petri Nets

Time Petri Nets (TPN) were introduced in [15] and extend Petri Nets with timing constraints on the firings of transitions. In TPN, a time interval is associated with each transition. An implicit clock can then be associated with each enabled transition, and gives the elapsed time since it was last enabled. An enabled transition can be fired if its clock value belongs to the interval of the transition. Furthermore, time cannot progress beyond any upper bound of an interval associated with a transition. The following definitions formalize these principles. We consider here a generalized version[2] of TPN with accepting and repeated markings and prove our results for this general model.

Definition 6 (Labeled Time Petri Net) *A* Labeled Time Petri Net $\mathcal{N}$ *is a tuple* $(P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I, F, R)$ *where:* P *is a finite set of* places *and* T *is a finite set of* transitions *with* $P \cap T = \emptyset$*;* $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ *is a finite set of* actions *and* ε *the empty word* i.e. *the* silent *action;* ${}^\bullet(.) \in (\mathbb{N}^P)^T$ *is the* backward *incidence mapping;* $(.)^\bullet \in (\mathbb{N}^P)^T$ *is the* forward *incidence mapping;* $M_0 \in \mathbb{N}^P$ *is the* initial *marking;* $\Lambda : T \to \Sigma_\varepsilon$ *is the* labeling function*;* $I : T \to \mathcal{I}(\mathbb{Q}_{\geq 0})$ *associates with each transition a* firing interval*;* $R \subseteq \mathbb{N}^P$ *is the set of* final markings *and* $F \subseteq \mathbb{N}^P$ *is the set of* repeated markings*. An* unlabeled *TPN is a TPN s.t.* $\Sigma = T$ *and* $\Lambda(t) = t$ *for all* $t \in T$.

A TPN $\mathcal{N}$ is a g-TPN if for all $t \in T$, $I(t)$ is an interval with bounds in $\mathbb{N}_g$. We also use ${}^\bullet t$ (resp. $t^\bullet$) to denote the set of places ${}^\bullet t = \{p \in P \mid {}^\bullet t(p) > 0\}$ (resp. $t^\bullet = \{p \in P \mid t^\bullet(p) > 0\}$) as it is common is the literature[3].

Semantics of Time Petri Nets. The semantics of TPNs is given in terms of Timed Transition Systems. A *marking* M of a TPN is a mapping in $\mathbb{N}^P$ and $M(p)$ is the number of tokens in place p. A transition t is *enabled* in a marking M iff $M \geq {}^\bullet t$. We denote by $\mathit{En}(M)$ the set of enabled transitions in M. To decide whether a transition t can be fired, we need to know for how long it has been enabled: if this amount of time lies within the interval $I(t)$, t can actually be fired, otherwise it cannot. On the other hand time can progress only if the enabling duration still belongs to the downward closure of the interval associated with an enabled transition. Let $\nu \in (\mathbb{R}_{\geq 0})^{\mathit{En}(M)}$ be a *valuation* such that each value $\nu(t)$ is the time elapsed since transition t was last enabled. A *configuration* of the TPN $\mathcal{N}$ is a pair (M, ν). An *admissible configuration* of a TPN is a configuration (M, ν) s.t. $\forall t \in \mathit{En}(M), \nu(t) \in I(t)^\downarrow$. We let $\mathit{ADM}(\mathcal{N})$ be the set of admissible markings.

When defining the semantics of a TPN, three kinds of policies must be fixed.

[2] This is required to be able to define Büchi timed languages, which is not possible in the original version of TPN of [15].

[3] Whether ${}^\bullet t$ (resp. $t^\bullet$) stands for a vector of $(\mathbb{N}^P)^T$ or a subset of P will be unambiguously defined by the context.

The choice policy concerns the choice of the next event to be fired (scheduled). For TPNs (and also timed automata), this choice is non deterministic (possible alternatives use priorities, probabilities, etc.).

The service policy concerns the possibility of simultaneous instances of a same event to occur. In the context of Petri nets, this is formalized by the enabling *degree* of a transition. Here we adopt the *single-server policy* (at most one instance of a firing per transition in every state). The results presented are also valid for the other standard policies (multiple or infinite server) at least for the important case of bounded Petri nets. However taking them explicitely into account would lead to intricate notations.

The memory policy concerns the updating of timing informations when a discrete step occurs. The key issue[4] in the semantics is to define when we reset the clock measuring the time since a transition was last enabled. This can only happen when we fire a transition. We let $\uparrow enabled(t', M, t) \in \mathbb{B}$ be true if t' is *newly enabled* by the firing of transition t from marking M, and false otherwise.

Let M be a marking and $t \in \mathit{En}(M)$. The firing of t leads to a new marking $M' = M - {}^\bullet t + t^\bullet$. Three semantics are possible:

I: The *intermediate* semantics (I) considers that the firing of a transition is performed in two steps: consuming the input tokens in ${}^\bullet t$, and then producing output tokens in $t^\bullet$. The fact that a transition t' is newly enabled on the firing of a transition $t \neq t'$ is determined w.r.t. the intermediate marking $M - {}^\bullet t$. When a transition t is fired it is newly enabled whatever the intermediate marking. We denote by $\uparrow enabled_I(t', M, t)$ the newly enabled predicate in this case. This mapping is defined by:

$$\begin{aligned}\uparrow enabled_I(t', M, t) = (t' \in \mathit{En}(M - {}^\bullet t + t^\bullet) \\ \wedge \left(t' \notin \mathit{En}(M - {}^\bullet t) \vee (t = t')\right)\end{aligned} \quad (1)$$

A: The *atomic semantics* considers that the firing of a transition is obtained by an atomic step. The corresponding mapping $\uparrow enabled_A(t', M, t)$ is defined by:

$$\uparrow enabled_A(t', M, t) = (t' \in \mathit{En}(M - {}^\bullet t + t^\bullet)) \wedge \left(t' \notin \mathit{En}(M) \vee (t = t')\right) \quad (2)$$

PA: The *persistent atomic semantics* considers that the firing of a transition is also obtained by an atomic step. The difference with the A semantics in only on the value of $\uparrow enabled_A(t', M, t)$ when $t = t'$. The transition begin fired is not always newly enabled:

$$\uparrow enabled_{PA}(t', M, t) = t' \in \mathit{En}(M - {}^\bullet t + t^\bullet) \wedge (t' \notin \mathit{En}(M)) \quad (3)$$

[4]The new marking obtained after firing a transition t from a marking M is given by the untimed semantics of Petri Nets *i.e.* $M' = M - {}^\bullet t + t^\bullet$.

Note that we have the relation:

$$\uparrow enabled_{PA}(t, M, t') \Rightarrow \uparrow enabled_{A}(t, M, t') \Rightarrow \uparrow enabled_{I}(t, M, t')$$

The intermediate semantics I, based on [6, 5] is the most common one. However, depending on the systems to be modeled, another semantics may be more appropriate. The relative expressive power of the three semantics has not been investigated so far: we address this problem in section 3.

We now define the semantics of a TPN: this is a parameterized semantics that depends on the choice of the semantics for the $\uparrow enabled$ predicate.

Definition 7 (Semantics of TPN) *Let $s \in \{I, A, PA\}$. The s-semantics of a TPN $\mathcal{N} = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I, F, R)$ is a timed transition system $S_\mathcal{N} = (Q, \{q_0\}, T, \rightarrow, F', R')$ where: $Q = ADM(\mathcal{N})$, $q_0 = (M_0, \mathbf{0})$, $F' = \{(M, \nu) \mid M \in F\}$ and $R' = \{(M, \nu) \mid M \in R\}$, and $\longrightarrow \in Q \times (T \cup \mathbb{R}_{\geq 0}) \times Q$ consists of the discrete and continuous transition relations:*

- *the discrete transition relation is defined $\forall t \in T$ by:*

$$(M, \nu) \xrightarrow{\Lambda(t)} (M', \nu') \text{ iff } \begin{cases} t \in En(M) \wedge M' = M - {}^\bullet t + t^\bullet \\ \nu(t) \in I(t), \\ \forall t \in \mathbb{R}_{\geq 0}^{En(M')}, \nu'(t) = \begin{cases} 0 & \text{if } \uparrow enabled_s(t', M, t), \\ \nu(t) & \text{otherwise.} \end{cases} \end{cases}$$

- *the continuous transition relation is defined $\forall d \in \mathbb{R}_{\geq 0}$:*

$$(M, \nu) \xrightarrow{d} (M, \nu') \text{ iff } \begin{cases} \nu' = \nu + d \\ \forall t \in En(M), \nu'(t) \in I(t)^\downarrow \end{cases}$$

A run ρ of $\mathcal{N}$ is an initial run of $S_\mathcal{N}$. The timed language accepted by $\mathcal{N}$ is $\mathcal{L}(\mathcal{T}) = \mathcal{L}(S_\mathcal{N})$. An unlabelled TPN accepts a timed language in $(T \times \mathbb{R}_{\geq 0})^\infty$.

We simply write $(M, \nu) \xrightarrow{w}$ to emphasize that a sequence of transitions w can be fired in $S_\mathcal{N}$ from (M, ν). If *Duration*$(w) = 0$ we say that w is an *instantaneous firing sequence*. The set of *reachable markings* of $\mathcal{N}$ is $\mathsf{Reach}(\mathcal{N}) = \{M \in \mathbb{N}^P \mid \exists (M, \nu) \mid (M_0, \mathbf{0}) \xrightarrow{w} (M, \nu)\}$.

2.3 Timed Automata

Definition 8 (Timed Automaton) *A* Timed Automaton *$\mathcal{A}$ is a tuple $(L, \ell_0, X, \Sigma_\varepsilon, E, Inv, F, R)$ where: L is a finite set of* locations*; $\ell_0 \in L$ is the* initial location*; X is a finite*

set of non negative real-valued clocks*;* $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ *is a finite set of* actions *and* ε *is the* silent *action;* $E \subseteq L \times \mathcal{C}(X) \times \Sigma_\varepsilon \times 2^X \times L$ *is a finite set of* edges*,* $e = \langle \ell, \gamma, a, R, \ell' \rangle \in E$ *represents an edge from the location* ℓ *to the location* ℓ' *with the guard* γ*, the label* a *and the reset set* $R \subseteq X$*;* $Inv \in \mathcal{C}(X)^L$ *assigns an* invariant *to any location. We restrict the invariants to conjuncts of terms of the form* $x \preceq r$ *for* $x \in X$ *and* $r \in \mathbb{N}$ *and* $\preceq \in \{<, \leq\}$*.* $F \subseteq L$ *is the set of* final locations *and* $R \subseteq L$ *is the set of* repeated locations*.*

Definition 9 (Semantics of a Timed Automaton) *The semantics of a timed automaton* $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, Inv, F, R)$ *is a timed transition system* $S_\mathcal{A} = (Q, q_0, \Sigma_\varepsilon, \rightarrow, F', R')$ *with* $Q = L \times (\mathbb{R}_{\leq 0})^X$*,* $q_0 = (\ell_0, \mathbf{0})$ *is the initial state,* $F' = \{(\ell, v) \mid \ell \in F\}$ *and* $R' = \{(\ell, v) \mid \ell \in R\}$*, and* $\rightarrow$ *is defined by:*

$$(\ell, v) \xrightarrow{a} (\ell', v') \quad \text{iff} \quad \exists\, (\ell, \gamma, a, R, \ell') \in E \text{ s.t. } \begin{cases} \gamma(v) = \mathit{tt}, \\ v' = v[R \mapsto 0] \\ Inv(\ell')(v') = \mathit{tt} \end{cases}$$

$$(\ell, v) \xrightarrow{d} (\ell', v') \quad \text{iff} \quad \begin{cases} \ell = \ell' \qquad v' = v + d \quad \textit{and} \\ \forall\, 0 \leq d' \leq d,\ Inv(\ell)(v + d') = \mathit{tt} \end{cases}$$

A run ρ *of* $\mathcal{A}$ *is an initial run of* $S_\mathcal{A}$*. The timed language accepted by* $\mathcal{A}$ *is* $\mathcal{L}(\mathcal{A}) = \mathcal{L}(S_\mathcal{A})$*.*

Recall [3] that, if m is the maximal constant appearing in atomic formulas $x \bowtie c$ of $\mathcal{A}$, an equivalence relation with finite index can be defined on clock valuations, leading to a partition of $(\mathbb{R}_{\geq 0})^X$, with the following property: two equivalent valuations have the same behaviour under progress of time and reset operations, with respect to the constraints. Note that a partition using any $K \geq m$ would have the same property. Also, the construction can be extended to a g-grid, by taking all constants of the form $\frac{i}{g}$, $0 \leq i \leq Kg$ instead of $\{0, 1, \ldots, K\}$. Finally, taking $K = +\infty$ (as depicted in Figure 1 on the left) leads to a similar structure except for the fact that the partition is infinite. When it is possible, we will sometimes use such a partition in order to simplify some proofs. Indeed, with this partition, the extremal case where x is greater than K has not to be distinguished from the standard case.

In this paper, the elements of the partition are called *elementary zones* and we consider a slight variation for the definition of elementary zones: we take the constant $K = m + 1$ and with each clock $x \in X$, we associate an interval in the set $\{\{0\},]0, 1[, \{1\}, \ldots, \{K - 1\},]K - 1, K[, [K, +\infty[\}$, instead of keeping $\{K\}$ separately. As usual, we also specify the ordering on the fractional parts for all clocks x such that $x < K$. Such a partition is represented in Figure 1 (on the right) for the set of two clocks $X = \{x, y\}$ and $K = 3$. For this example, elementary zones Z_1 and Z_2 are described by the constraints: $Z_1 : (2 < x < 3) \wedge (1 < y < 2) \wedge (0 < frac(y) < frac(x))$ and $Z_2 : (x \geq 3) \wedge (1 < y < 2)$.

When considering diagonal constraints (also with constants up to K), another partition (Figure 1 in the middle, with $K = 2$) must be considered.

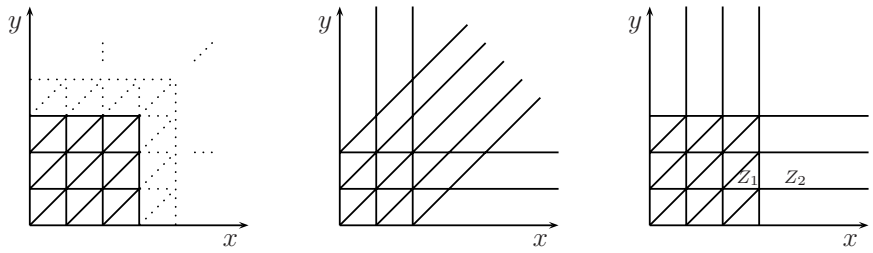


Figure 1: Partitions of $(\mathbb{R}^+)^2$ with $K = +\infty$, $K = 2$ (with diagonal constraints) and $K = 3$ (no diagonal constraints)

The future of a zone Z is defined by $fut(Z) = \{v + d \mid v \in Z, d \in \mathbb{R}_{\geq 0}\}$. If Z and Z' are elementary zones, Z' is a time successor of Z, written $Z \leq Z'$, if for each valuation $v \in Z$, there is some $d \in \mathbb{R}_{\geq 0}$ such that $v + d \in Z'$. For each elementary zone Z, there is at most one elementary zone such that (i) Z' is a time successor of Z, (ii) $Z \neq Z'$ and (iii) there is no time successor Z'' such that $Z \leq Z'' \leq Z'$. When it exists, this elementary zone is called the immediate successor of Z and is denoted by $succ(Z)$. Note that $fut(Z) \subseteq \cup_{Z \leq Z'} Z'$, with a strict inclusion when no diagonal constraints are permitted.

Finally recall that a finite automaton $R(\mathcal{A})$, called the *region automaton*, can be built from $\mathcal{A}$. This automaton is time abstract bisimilar to the original automaton $\mathcal{A}$. Its states, called here *regions*, are of the form (ℓ, Z), where ℓ is a location of $\mathcal{A}$ and Z an elementary zone of $(\mathbb{R}_{\geq 0})^X$. They are built from the initial region $(\ell_0, \mathbf{0})$ by transitions of the form $(\ell, Z) \xrightarrow{\delta} (\ell, Z')$ for a time successor Z' of Z, if $Inv(\ell)(Z) = \mathsf{tt}$ or $(\ell, Z) \xrightarrow{a} (\ell', Z')$ if there is a transition $(\ell, \gamma, a, R, \ell') \in E$ such that $\gamma(Z) = \mathsf{tt}$ and $Z' = Z[R \mapsto 0]$, with $Inv(\ell')(Z') = \mathsf{tt}$. A region (ℓ, Z) is said to be maximal in $R(\mathcal{A})$ with respect to ℓ if no δ-transition is possible from (l, Z). The automaton $R(\mathcal{A})$ is restricted to the regions reachable from the initial region $(\ell_0, \mathbf{0})$, and accepts the language

$$Untime(L(\mathcal{A})) = \{a_1 a_2 \ldots \mid (a_1, d_1)(a_2, d_2) \ldots \in L(\mathcal{A}) \text{ for some } d_1, d_2, \ldots \in \mathbb{R}_{\geq 0}\}.$$

We also consider another automaton, called *class automaton*, in which the states, called *classes*, are of the form $(l, fut(Z) \cap Inv(\ell))$, where Z is a zone. In this case, the second component is not an elementary zone anymore (but a general zone) and the automaton is build from the initial class $(\ell_0, fut(\mathbf{0}) \cap Inv(\ell_0))$ by the following transitions: $(\ell, Z_1) \xrightarrow{a} (\ell', Z_2)$ if there exists $(l, \gamma, a, R, l') \in E$ such that $Z_1 \cap [\![\gamma]\!] \neq \emptyset$, and

$Z_2 = fut((Z_1 \cap [\![\gamma]\!])[R \mapsto 0]) \cap Inv(\ell')$.
Note that the class automaton also accepts $Untime(L(\mathcal{A}))$. Moreover, since a class can be represented by a Difference Bounded Matrix [11], its size is at most $(4K+2)^{(|X|+1)^2}$, which is exponential in the size of $\mathcal{A}$, as for the region automaton.

2.4 Expressiveness and Equivalence Problems

If B, B' are either TPN or TA, we write $B \approx_{\mathcal{S}} B'$ (resp. $B \approx_{\mathcal{W}} B'$) for $S_B \approx_{\mathcal{S}} S_{B'}$ (resp. $S_B \approx_{\mathcal{W}} S_{B'}$). Let $\mathcal{C}$ and $\mathcal{C}'$ be two classes of TPNs or TA.

Definition 10 (Expressiveness w.r.t. Timed Language Acceptance) *The class $\mathcal{C}$ is* more expressive *than $\mathcal{C}'$ w.r.t. timed language acceptance if for all $B' \in \mathcal{C}'$ there is a $B \in \mathcal{C}$ s.t. $\mathcal{L}(B) = \mathcal{L}(B')$. We write $\mathcal{C}' \leq_{\mathcal{L}} \mathcal{C}$ in this case. If moreover there is some $B \in \mathcal{C}$ s.t. there is no $B' \in \mathcal{C}'$ with $\mathcal{L}(B) = \mathcal{L}(B')$, then $\mathcal{C}' <_{\mathcal{L}} \mathcal{C}$ (read "strictly more expressive"). If both $\mathcal{C}' \leq_{\mathcal{L}} \mathcal{C}$ and $\mathcal{C} \leq_{\mathcal{L}} \mathcal{C}'$ then $\mathcal{C}$ and $\mathcal{C}'$ are equally expressive w.r.t. timed language acceptance, and we write $\mathcal{C} =_{\mathcal{L}} \mathcal{C}'$.*

Definition 11 (Expressiveness w.r.t. Timed Bisimilarity) *The class $\mathcal{C}$ is* more expressive *than $\mathcal{C}'$ w.r.t. strong (resp. weak) timed bisimilarity if for all $B' \in \mathcal{C}'$ there is a $B \in \mathcal{C}$ s.t. $B \approx_{\mathcal{S}} B'$ (resp. $B \approx_{\mathcal{W}} B'$). We write $\mathcal{C}' \leq_{\mathcal{S}} \mathcal{C}$ (resp. $\mathcal{C}' \leq_{\mathcal{W}} \mathcal{C}$) in this case. If moreover there is a $B \in \mathcal{C}$ s.t. there is no $B' \in \mathcal{C}'$ with $B \approx_{\mathcal{S}} B'$ (resp. $B \approx_{\mathcal{W}} B'$), then $\mathcal{C}' <_{\mathcal{S}} \mathcal{C}$ (resp. $\mathcal{C}' <_{\mathcal{W}} \mathcal{C}$). If both $\mathcal{C}' <_{\mathcal{S}} \mathcal{C}$ and $\mathcal{C} <_{\mathcal{S}} \mathcal{C}'$ (resp. $<_{\mathcal{W}}$) then $\mathcal{C}$ and $\mathcal{C}'$ are equally expressive w.r.t. strong (resp. weak) timed bisimilarity, and we write $\mathcal{C} \approx_{\mathcal{S}} \mathcal{C}'$ (resp. $\mathcal{C} \approx_{\mathcal{W}} \mathcal{C}'$).*

In the sequel we will compare various subclasses of TPNs and TA. We denote $\mathcal{TPN}$ the class of TPNs and $\mathcal{TA}$ the class of TA, according to definitions 6 and 8. We recall the following theorem adapted from [8]:

Theorem 1 ([8]) *For any $\mathcal{N} \in$ B-$\mathcal{TPN}_\varepsilon$ there is a TA $\mathcal{A}$ s.t. $\mathcal{N} \approx_{\mathcal{W}} \mathcal{A}$, hence B-$\mathcal{TPN}_\varepsilon \leq_{\mathcal{W}} \mathcal{TA}_\varepsilon$.*

3 Comparison of semantics I, A and PA

In the first paragraph, we establish two relations between these semantics for TPN, which hold in the general case. In the second paragraph, we complete these results with a third one, restricted to bounded time Petri nets, with only closed intervals for transitions. Since we prove results concerning weak timed bisimulation, we consider unlabeled TPN, where all states are final and repeated states. For all figures in this section, a transition is filled in black when its firing interval is $[0, 0]$.

3.1 A first comparison between the different semantics of TPNs

Proposition 1 *Let $\mathcal{N}$ be a time Petri net with intermediate semantics. There exists a TPN $\overline{\mathcal{N}}$ with atomic semantics which is (weakly timed) bisimilar to $\mathcal{N}$.*

Proof. The construction is quite easy. The set of places of $\overline{\mathcal{N}}$ is obtained by adding to the set of places of $\mathcal{N}$ a new place for each transition t from $\mathcal{N}$: $\overline{P} = P \cup \{p_t, t \in T\}$. The transitions T of $\mathcal{N}$ are duplicated in $\overline{\mathcal{N}}$: $\overline{T} = T^+ \cup T^-$ and the construction follows Figure 2, from left to right.

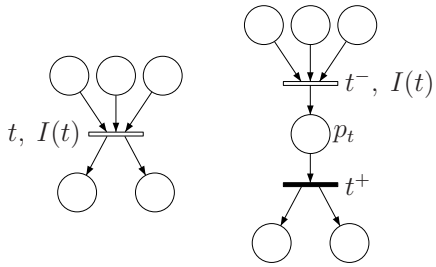


Figure 2: From I to A

We consider the equivalence relation $\mathcal{R}$ which contains all pairs $((M, \nu), (\overline{M}, \overline{\nu}))$ such that:

- for all $p \in P$, $M(p) = \overline{M}(p) + \Sigma_{t \in T} t^\bullet(p).\overline{M}(p_t)$
- for all $t \in En(M)$, $\nu(t) = \overline{\nu}(t^-)$ if t^- is enabled in $\overline{M}$ and 0 otherwise. The latter case corresponds in $\mathcal{N}$ to a newly enabled transition.

To prove that $\mathcal{R}$ is a bisimulation, we first note that, with the definition above for markings, from any configuration $(\overline{M}, \overline{\nu})$, we can reach instantaneously a configuration $(\overline{M}_1, \overline{\nu}_1)$ such that $\overline{M}_1(p_t) = 0$ for all t, with the firing of a (possibly empty) sequence of transitions in T^+. Moreover, the relation between valuations implies that $(\overline{M}_1, \overline{\nu}_1)$ is still equivalent to (M, ν).

Consider now a pair $((M, \nu), (\overline{M}, \overline{\nu})) \in \mathcal{R}$.

- if $(M, \nu) \xrightarrow{t} (M', \nu')$, then from the remark above, we first fire a sequence from $\overline{M}$ to empty all places p_t, leading instantaneously to $(\overline{M}_1, \overline{\nu}_1)$, which is equivalent to (M, ν). Then transition t^- can be fired from $(\overline{M}_1, \overline{\nu}_1)$, immediately followed by t^+, leading to $(\overline{M}', \overline{\nu}')$, where all places p_t are empty again. Moreover, the transitions

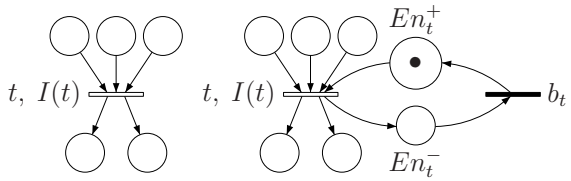


Figure 3: From A to PA

newly enabled by t^+ in $\overline{\mathcal{N}}$ are exactly those which were newly enabled by t in $\mathcal{N}$, so that $(M', \nu')\mathcal{R}(\overline{M}', \overline{\nu}')$.

- Conversely, suppose that a transition is fired from $(\overline{M}, \overline{\nu})$ in $\overline{\mathcal{N}}$. If the transition is some t^+, then the new configuration $(\overline{M}_1, \overline{\nu}_1)$ is still equivalent to (M, ν) (as above), thus no move at all is necessary in $\mathcal{N}$.
  If $(\overline{M}, \overline{\nu}) \xrightarrow{t^-} (\overline{M}', \overline{\nu}')$, then t can be fired from (M, ν) and the resulting marking is, (M', ν'), equivalent to $(\overline{M}', \overline{\nu}')$.

- if $(M, \nu) \xrightarrow{d} (M, \nu+d)$, for some delay d, then again we have to apply the emptying sequence from $(\overline{M}, \overline{\nu})$, to reach a configuration $(\overline{M}_1, \overline{\nu}_1)$ still equivalent to (M, ν), where time can elapse. The relation between ν and $\overline{\nu}_1$ implies that this is possible, leading to $(\overline{M}_1, \overline{\nu}_1 + d)$.

- Conversely, if $(\overline{M}, \overline{\nu}) \xrightarrow{d} (\overline{M}, \overline{\nu} + d)$, then all places p_t are empty in $\overline{M}$, so that the move $(M, \nu) \xrightarrow{d} (M, \nu + d)$ is also possible in $\mathcal{N}$.

Thus $\mathcal{R}$ is a bisimulation. ∎

Proposition 2 *Let $\mathcal{N}$ be a time Petri net with atomic semantics. There exists a TPN $\overline{\mathcal{N}}$ with persistent atomic semantics which is (weakly timed) bisimilar to $\mathcal{N}$.*

Proof. Here again, the construction is simple. Note that the only difference between the two semantics concerns the question wether a transition t can newly enable itself. With atomic semantics, this is the case as soon as t is enabled in the new marking while with persistent atomic semantics, this is never possible. In order to ensure that a transition t will be newly enabled if it is enabled in the new marking, we add an input place En_t^+ and an output place En_t^- to the transition, with an instantaneous loop b_t leading back to En_t^+, once the transition has been fired. The construction is represented in Figure 3, again from left to right.

We consider the equivalence relation $\mathcal{R}$ which contains all pairs $((M, \nu), (\overline{M}, \overline{\nu}))$ such that:

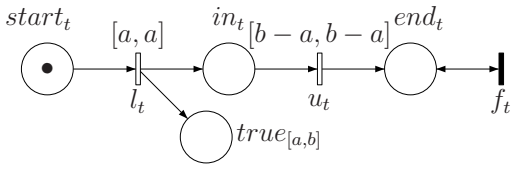


Figure 4: Time subnet for transition t with interval $[a, b]$

- $M(p) = \overline{M}(p)$ for all places p in P, and
- for a transition $t \in En(M)$, $\nu(t) = \overline{\nu}(t)$ if t is enabled in $\overline{M}$ and 0 otherwise. Again the latter case corresponds in $\mathcal{N}$ to a newly enabled transition.

Like in the previous proposition, the proof is mostly based on the fact that from any configuration $(\overline{M}, \overline{\nu})$, we can reach instantaneously a configuration $(\overline{M}_1, \overline{\nu}_1)$ such that $\overline{M}_1(En_t^+) = 1$ for all t, with the firing of a (possibly empty) sequence of transitions b_t, with again $(\overline{M}_1, \overline{\nu}_1)$ still equivalent to (M, ν). ∎

3.2 A second comparison for standard bounded TPN

We now restrict to bounded TPNs, with the standard definition, *i.e.* with closed intervals ($[a, b]$ or $[a, \infty[$) for the transitions. Thus, this third result holds only for the subclass B-$\mathcal{TPN}(\leq, \geq)$.

Proposition 3 *Let $\mathcal{N}$ be a TPN in B-$\mathcal{TPN}(\leq, \geq)$ with persistent atomic semantics. There exists a TPN $\overline{\mathcal{N}}$ with intermediate semantics which is (weakly timed) bisimilar to $\mathcal{N}$.*

In this case, the construction of $\overline{\mathcal{N}}$ is more involved. Like above, we show how to simulate a transition t equipped with interval $[a, b]$, for $a \leq b$, or $[a, +\infty[$. We first build a time subnet for t (Figure 4 below), to simulate time elapsing since a reset operation until reaching (and staying inside) interval $[a, b]$. The token is in place $start_t$ if the transition is enabled in the initial marking. The double arrow at the end indicates that the place $term_t$ is both an input and an output place for the corresponding transition: time cannot progress. Of course, the time subnet for a transition with interval $[a, +\infty[$ is reduced by removing u_t, end_t and f_t.

Now, using the fact that the TPN is bounded, we consider its upper bound B and we associate with each place p a complementary place $\overline{p}$ such that for any reachable marking M, $M(\overline{p}) = B - M(p)$. Figure 5 represents a part of the subnet (on the right) for transition t (on the left), where $test_1$ is the beginning of the test step for what timing updates are

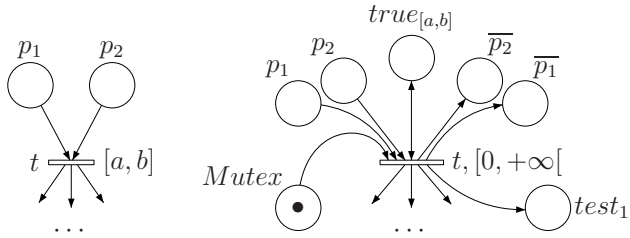


Figure 5: From PA to I

required by the firing of t, and $Mutex$ ensures that the updates are done (instantaneously) before anything else, as explained further.

The remaining part of $\overline{\mathcal{N}}$ is devoted to the test of the other transitions from the original TPN, including t itself. Consider a given transition (say t_i), with again two input places p_1^i and p_2^i. The corresponding subnet consists of 4 modules, one for each case, depending on wether t_i can be fired or not before and after t. For this, two additional places are associated with t_i: E_{t_i}, which contains a token if t_i was enabled before the firing of t and NE_{t_i} its complementary place. If t_i is initially enabled then E_{t_i} is initially marked otherwise NE_{t_i} is marked. This group of 4 modules has a common input place $test_i$ and a common output place $test_{i+1}$, which means that the tests are to be executed sequentially (and instantaneously), except for the last one where all outgoing transitions are linked to $Mutex$. These places are not shown in the following figures.

Case 1: transition t_i is enabled both before and after t. To test this case, we use the simple module on the left of Figure 6, where E_{t_i} (test before t) and p_1^i and p_2^i (test after t) are input and output places.
Case 2: t_i is not enabled before but enabled after t. The module is very similar to the previous one and is on the right of Figure 6. Note that, in this case only, because of the PA semantics, there must be a reset on the valuation of the transition, which explains why the initial place $start_{t_i}$ of the time module for t_i is an output place.

Case 3: t_i is enabled neither before nor after t. To test this, we must find an input place of t_i, where the current number of tokens disable t_i. Here is the point where the boundedness hypothesis is required. In order to perform this test, we check whether $B - {}^\bullet t_i(p) + 1$ tokens can be removed from a complementary place $\overline{p}$.

Case 4: t_i is enabled before but not after t. In this case, we have a module (see Figure 8) similar to the one above, except that we must also test for all the different configurations of the time subnet corresponding to t_i, to disable the transitions by removing the tokens.

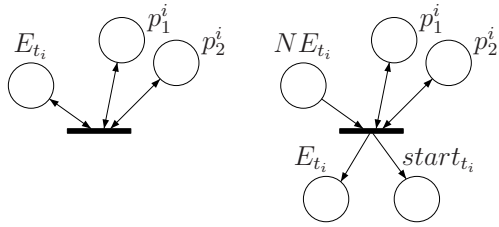


Figure 6: Testing transition t_i: cases 1 and 2

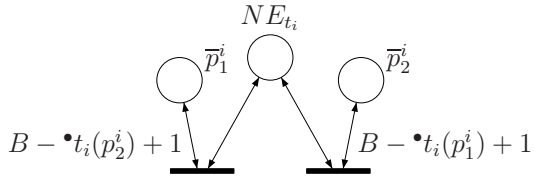


Figure 7: Testing transition t_i: case 3

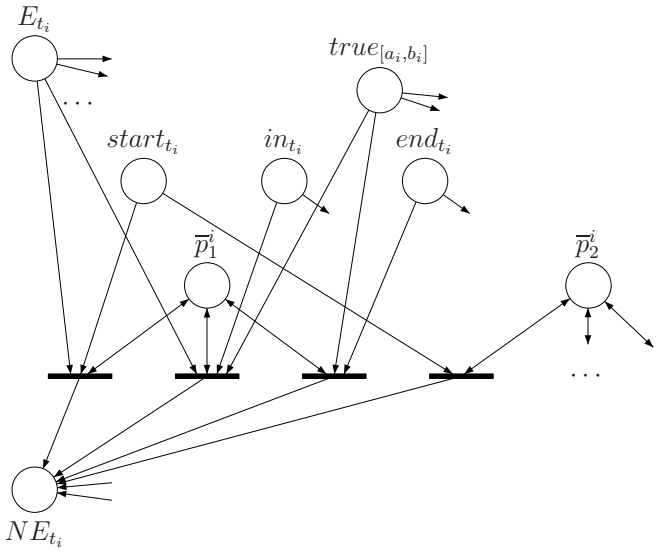


Figure 8: Testing transition t_i: case 4

It can be seen in Figure 8 that there is a transition for each pair $(\overline{p}, state)$, where p is an input place of t_i and $state$ may be either the place $start_{t_i}$, the pair of places $(in_{t_i},\ true_{[a_i,b_i]})$ or the pair $(end_{t_i},\ true_{[a_i,b_i]})$. Like above, an edge from $\overline{p}$ to a transition must be labeled with $B - {}^\bullet t_i(p) + 1$ (which is omitted in the figure).

We consider the equivalence relation $\mathcal{R}$ containing all pairs $((M,\nu),(\overline{M},\overline{\nu}))$ such that

- M in $\mathcal{N}$ is obtained by projection: $M(p) = \overline{M}(p)$ for each place $p \in P$,
- for a transition t in T enabled by M: $\nu(t) = 0$ if the time subnet of t is empty, $\nu(t) = \overline{\nu}(l_t)$ if the place $start_t$ contains a token, $\nu(t) = a + \overline{\nu}(u_t)$ if the place in_t contains a token and $\nu(t) = b$ if the place end_t contains a token. Note that in both latter cases, $true_{[a,b]}$ also contains a token and the transition t can be fired in $\overline{\mathcal{N}}$.

Also note that if $\overline{M}(start_t) = 1$ and $\overline{\nu}(u_t) = a$, then with instantaneously firing u_t, transition t can also be fired. By a development similar to the previous ones, we can show that $\mathcal{R}$ is a bisimulation relation. More precisely, the proof is mainly based on emptying sequences from a configuration $(\overline{M},\overline{\nu})$ of $\overline{\mathcal{N}}$: it is always possible to reach instantaneously a configuration $(\overline{M}_1,\overline{\nu}_1)$ such that the testing subnet is empty, with $(\overline{M}_1,\overline{\nu}_1)$ still equivalent to (M,ν). The details are omitted.

We can conclude this section with:

Corollary 1 *For the class B-$\mathcal{TPN}(\leq,\geq)$, the three semantics I, A and PA are equally expressive w.r.t. weak time bisimulation.*

4 Strict Ordering Results

In this section, we establish results proving that $\mathcal{TPN}$ are strictly less expressive w.r.t. weak timed bisimilarity than various classes of TA. For this, we consider the two automata $\mathcal{A}_0 \in \mathcal{TA}(<)$ and $\mathcal{A}_1 \in \mathcal{TA}(\leq)$ in Figure 9.

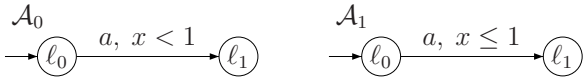


Figure 9: Timed automata $\mathcal{A}_0$ and $\mathcal{A}_1$

We will prove that no TPN can be weakly timed bisimilar to either $\mathcal{A}_0$ or $\mathcal{A}_1$. The proof relies on the following lemma, which states that in a TPN, waiting in some marking cannot disable transitions. The proof is easy and is thus omitted. Note that the results holds without modification for any semantics of $\uparrow enabled_s(t', M, t)$.

Lemma 1 (Waiting Cannot Disable Transitions) *Let (M, ν) be an admissible configuration of a TPN, $d \in \mathbb{R}_{\geq 0}$ and let $w = t_1 t_2 \cdots t_k$ be an instantaneous firing sequence. If $(M, \nu) \xrightarrow{w}$ then $(M, \nu + d) \xrightarrow{w}$.*

Theorem 2 *There is no TPN weakly timed bisimilar to $\mathcal{A}_0$.*

Proof. Assume there is a TPN $\mathcal{N}$ that is weakly timed bisimilar to $\mathcal{A}_0$ and let $\approx$ be a weak timed bisimulation between $S_{\mathcal{N}}$ and $S_{\mathcal{A}_0}$. Let $(M_0, \mathbf{0})$ be the initial state of $S_{\mathcal{N}}$ and $(\ell_0, v(x) = 0)$ the initial state of $S_{\mathcal{A}_0}$. In $S_{\mathcal{A}_0}$ there is a run of duration 1 leading to configuration $(\ell_0, 1)$ and thus there is a run $(M_0, \mathbf{0}) \xrightarrow{\varepsilon^{i_0} d_1 \varepsilon^{i_1} d_2 \varepsilon^{i_2} \cdots d_n \varepsilon^{i_n}} (M_1, \nu_1)$ in $\mathcal{N}$, with $i_k \geq 1$ for $1 \leq k \leq n-1$, $i_0 \geq 0$, $i_n \geq 0$ and $\sum_{1 \leq k \leq n} d_k = 1$. We can further assume $d_k > 0$ for all k, and also $i_n = 0$ because the configuration reached after d_n is also bisimilar to $(\ell_0, 1)$. Then $(M_0, \mathbf{0}) \xrightarrow{\varepsilon^{i_0} d_1 \varepsilon^{i_1} d_2 \varepsilon^{i_2} \cdots d_{n-1} \varepsilon^{i_{n-1}}} (M', \nu')$, where (M', ν') is bisimilar to a configuration (ℓ_0, d') with $d' = 1 - d_n < 1$. This entails that $(M', \nu') \xrightarrow{\varepsilon^* a}$. Since $(M', \nu') \xrightarrow{d_n} (M_1, \nu_1)$, if follows from lemma 1 that $(M_1, \nu_1) \xrightarrow{\varepsilon^* a}$ contradicting the fact that $(M_1, \nu_1) \approx (\ell_0, 1)$ from which no a can be fired. ∎

The result is also true with large constraints:

Theorem 3 *There is no TPN weakly timed bisimilar to $\mathcal{A}_1$.*

Proof. Again assume there is a TPN $\mathcal{N}$ that is weakly timed bisimilar to $\mathcal{A}_1$. Since $(\ell_0, 0) \xrightarrow{1} (\ell_0, 1)$, we have $(M_0, \mathbf{0}) \xrightarrow{1}_{\varepsilon} (M_1, \nu_1)$, where $(\ell_0, 1)$ and (M_1, ν_1) are weakly timed bisimilar. Since a can be fired from $(\ell_0, 1)$, a transition labeled a can also be fired from all the configurations (M_1', ν_1') reachable from (M_1, ν_1) in null duration (ε transitions). Also there must be one such configuration (M', ν') s.t. some duration $d > 0$ can elapse from (M', ν') reaching (M'', ν''). By lemma 1, some a can be fired from (M'', ν''). But (M'', ν'') is weakly timed bisimilar to the configuration $(\ell_0, 1 + d)$ which prevents a to be fired. Hence a contradiction.

From Theorems 1, 2 and 3, we immediately obtain:

Corollary 2 *$\mathcal{TPN} <_{\mathcal{W}} \mathcal{TA}(<)$ and $\mathcal{TPN} <_{\mathcal{W}} \mathcal{TA}(\leq)$.*

The next proposition shows that the expressive power of TPNs depends on the chosen semantic even in the bounded case.

Theorem 4 *There exists a bounded TPN $\mathcal{N}$ with persistent atomic semantics such that no TPN (even unbounded) with atomic semantics is bisimilar to $\mathcal{N}$.*

Proof. Consider the following (Zeno) timed automaton $\mathcal{A}_3$: 

It is bisimilar to the TPN with PA semantics composed by a single transition t labeled by ε with firing interval $[0, 1[$ (or any interval $]a, 1[$ or $[a, 1[$).

Suppose that there is a TPN $\mathcal{N}$ with atomic semantics bisimilar to $\mathcal{A}_3$ and let d_{min} be the minimum of the non null upper bounds occuring in the intervals associated with the transitions of $\mathcal{N}$ and 0.5 (in fact any value less than 1 would be convenient).

There must be a sequence $(M_0, \nu_0) \xrightarrow{d_0 t_1 \dots t_k d_k} (M, \nu)$ with $\Sigma_{i=0}^{k} d_i = 1 - d_{min}/2$ and (M, ν) bisimilar to $(\ell_0, 1 - d_{min}/2)$.
From (M, ν), we fire or disable the transitions enabled at this configuration, which leads to a new configuration (M', ν') bisimilar to some $(\ell_0, 1 - \delta')$ with $0 < \delta' \leq d_{min}/2$. Now since (M', ν') is bimilar to $(\ell_0, 1-\delta')$ there must be a sequence $(M', \nu') \xrightarrow{d'_0 t'_1 \dots t'_{k'} d'_{k'}}$ with $0 < \Sigma_{i=0}^{k'} d'_i < \delta'$.
Choose the first $d'_i > 0$ and let (M^*, ν^*) be the state reached before the duration d'_i. Since time may elapse in this state, all enabled transitions have non null upper bound for their interval, hence these bounds are greater than or equal to d_{min}. Since the transitions have been enabled at or after configuration (M', ν'), we have $\forall t, \nu^*(t) \leq d_{min}/2 - \delta' < d_{min}/2$, thus $(M^*, \nu^*) \xrightarrow{d_{min}/2}$. But (M^*, ν^*) is bisimilar to $(\ell_0, 1-\delta')$ which cannot let time elapse for a duration of $d_{min}/2$. This is a contradiction. ∎

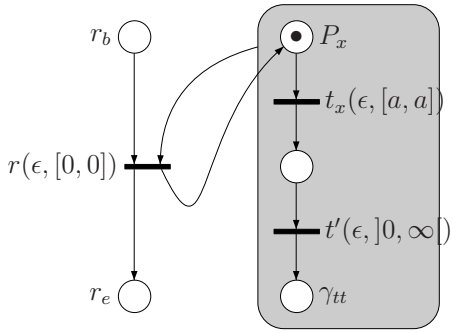
Following this negative results, we compare the expressiveness of TPNs and TA w.r.t. to Timed Language Acceptance and exhibit a subclass of TA that admits bisimilar TPNs.

5 Equivalence w.r.t. Timed Language Acceptance

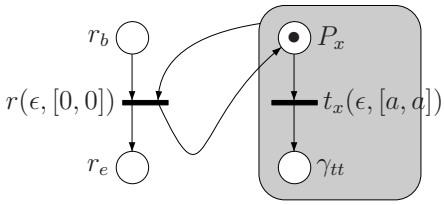
In this section, we prove that TA and labeled TPNs are equally expressive w.r.t. timed languages acceptance, and give an effective syntactical translation from TA to TPNs.

Let $\mathcal{A} = (L, l_0, X, \Sigma_\varepsilon, E, Inv, F, R)$ be a TA. Since we are concerned in this section with the langage accepted by $\mathcal{A}$ we assume the invariant function is uniformly true. Let $\mathcal{C}_x$ be the set of atomic constraints on clock x that are used in $\mathcal{A}$. The Time Petri Net resulting from our translation will be built from "elementary blocks" modeling the truth value of the constraints in $\mathcal{C}_x$. We next link them with blocks for resetting clocks. In the next subsection we show how to encode atomic constraints into TPNs.

As a consequence of corollary 1, the semantics I, A and *PA* for TPNs are equivalent w.r.t. language acceptance. In this section, we use the I semantics.



(a) Widget $\mathcal{T}_{x>a}$



(b) Widget $\mathcal{T}_{x\geq a}$ (assume $a > 0$)

Figure 10: ets for $\mathcal{T}_{x>a}$ and $\mathcal{T}_{x\geq a}$

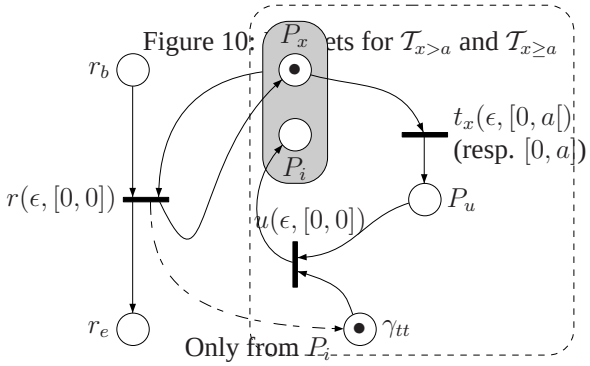


Figure 11: Widget $\mathcal{T}_{x<a}$ (resp. $\mathcal{T}_{x\leq a}$)

5.1 Encoding Atomic Constraints

Let $\varphi \in \mathcal{C}_x$ be an atomic constraint on x. From φ, we define the TPN $\mathcal{N}_\varphi$, given by the widgets of Fig. 10 ((a) and (b)) and Fig. 11. In the figures, a transition is written $t(\ell, I)$ where t is the name of the transition, $\ell \in \Sigma_\varepsilon$ and $I \in \mathcal{I}(\mathbb{Q}_{\geq 0})$.

To avoid drawing too many arcs, we have adopted the following semantics: the grey box is seen as a macro place; an arc from this grey box means that there are as many copies of the transition as places in the grey box. For instance the TPN of Fig. 10.(b) has 2 copies of the target transition r: one with input places P_x and r_b and output places r_e and P_x and another fresh copy of r with input places r_b and γ_{tt} and output places r_e and P_x. Note that in the widgets of Fig. 11 we put a token in γ_{tt} when firing r only on the copy of r with input place P_i (otherwise the number of tokens in place γ_{tt} could be unbounded). Also we assume that the automaton $\mathcal{A}$ has no constraint $x \geq 0$ (as it evaluates to true they can be safely removed) and thus that the widget of Fig. 10.(b) only appears with $a > 0$.

Each of these TPNs basically consists of a “constraint” subpart (in the grey boxes

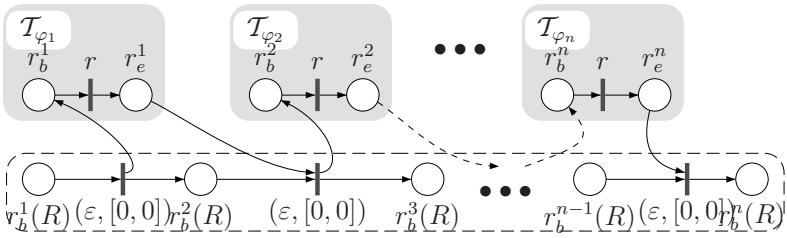


Figure 12: Widget $\mathcal{N}_{Reset(R)}$ to reset the widgets $\mathcal{N}_{\varphi_i}$, $1 \leq i \leq n$

for Fig. 10 and in the dashed box for Fig. 11) that models the truth value of the atomic constraint, and another "reset" subpart that will be used to update the truth value of the constraint when the clock x is reset. The "constraint" subpart features the place γ_{tt}: the intended meaning is that when a token is available in this place, the corresponding atomic constraint φ is true.

When a clock x is reset, all the grey blocks modeling an x-constraint must be set to their *initial* marking with has one token in P_x for Fig. 10 and one token in P_x and γ_{tt} for Fig. 11. Our strategy to reset a block modeling a constraint is to put a token in the r_b place (r_b stands for "reset begin"). Time cannot elapse from there on (strong semantics for TPNs), as there will be a token in one of the places of the grey block and thus transition r will be enabled.

We first prove three useful lemmas, the first one providing a structural invariant for the grey boxes of the widgets:

Lemma 2 *For each widget of Fig. 10, each reachable configuration (M, ν) (from the initial marking) has exactly one token in one of the places of the grey box.*

Lemma 3 *For the widgets of Fig. 11, each reachable configuration (M, ν) (from the initial marking) satisfies either i) $M(P_x) = 1$, $M(\gamma_{tt}) = 1$ and $M(P_i) = 0$ or ii) $M(P_u) = 1$ and $M(\gamma_{tt}) = 1$ or iii) $M(P_x) = 0$, $M(\gamma_{tt}) = 0$ and $M(P_i) = 1$.*

Proof. The proof is easy for the widgets of Fig. 10. For the widgets of Fig. 11, just notice that as soon as t_x is fired, the output transition u is enabled (there must be a token in γ_{tt} as it can only be removed by the firing of u). Later on, either the token remains in P_i forever, or if the copy of r from P_i is fired a token is put in γ_{tt} and P_x. From lemmas 2 and 3 we obtain the following:

Lemma 4 *If there is a token in r_b, exactly one (instance of a) copy of r is firable and due to the time constraint $[0, 0]$, time cannot progress until it is fired.*

5.2 Resetting Clocks

Assume $R \subseteq X$ is a non empty subset of clocks. Let $D(R)$ be the set of atomic constraints that are in the scope of R (the clock of the constraint is in R). We write $D(R) = \{\varphi_1, \varphi_1, \cdots, \varphi_n\}$. To update all the widgets $\mathcal{N}_{\varphi_i}$, we connect the places r_b and r_e of each widget $\mathcal{N}_\varphi$ as described on Fig. 12. The picture inside the dashed box denotes the widget $\mathcal{N}_{\mathit{Reset}(R)}$. We denote by $r_b^1(R)$ the first place of this widget and $r_b^n(R)$ the last one. To update the (truth value of the) widgets $\mathcal{N}_{\varphi_i}$ it then suffices to put a token in $r_b^1(R)$. In null duration it will go to $r_b^n(R)$ and have the effect of updating each widget $\mathcal{N}_{\varphi_i}$ on its way.

5.3 The Complete Construction

First we create fresh places P_ℓ for each $\ell \in L$, and another place *Firing* just for convenience: it will allow us to define a simulation relation more succintly. Then we build the widgets $\mathcal{N}_\varphi$, for each atomic constraint φ that appears in $\mathcal{A}$. Finally for each $R \subseteq X$ s.t. there is an edge $e = (\ell, \gamma, a, R, \ell') \in E$ we build a reset widget $\mathcal{N}_{\mathit{Reset}(R)}$.

Then for each edge $(\ell, \gamma, a, R, \ell') \in E$ with $\gamma = \wedge_{i=1,n}\varphi_i$ and $n \geq 0$ we proceed as follows:

1. assume $\gamma = \wedge_{i=1,n}\varphi_i$ and $n \geq 0$,
2. create a transition $f(a, [0, \infty[)$ and if $n \geq 1$ another one $r(\varepsilon, [0, 0])$,
3. connect them to the places of the widgets $\mathcal{N}_{\varphi_i}$ and $\mathcal{N}_{\mathit{Reset}(R)}$ as described on Fig. 13. In case $\gamma = \mathsf{tt}$ (or $n = 0$) there is only one input place to $f(a, [0, \infty[)$ which is P_ℓ. In case $R = \emptyset$ there is no transition $r(\varepsilon, [0, 0])$ and the output place of $f(a, [0, \infty[)$ is $P_{\ell'}$ instead of *Firing*.

The place *Firing* is just added for convenience: it has a token only during the reset phase of the TPN $\mathcal{N}_e$ and thus means "we are firing transitions in the reset widget $\mathcal{N}_{\mathit{Reset}(R)}$".

To complete the construction we just need to put a token in the place P_{ℓ_0} if ℓ_0 is the initial location of the automaton, and set each widget $\mathcal{T}_\varphi$ to its initial marking, for each atomic constraint φ that appears in $\mathcal{A}$, and this defines the initial marking M_0. The set of final markings is defined by the set of markings M s.t. $M(P_\ell) = 1$ for $\ell \in F$ and the set of repeated markings by by the set of markings M s.t. $M(P_\ell) = 1$ for $\ell \in R$. We denote by $\Delta(\mathcal{A})$ the TPN obtained as described previously. Notice that by construction 1) $\Delta(\mathcal{A})$ is 1-safe and moreover 2) in each reachable marking M of $\Delta(\mathcal{A})$ $\left(\sum_{\ell \in L} M(P_\ell)\right) + M(\mathit{Firing}) = 1$.

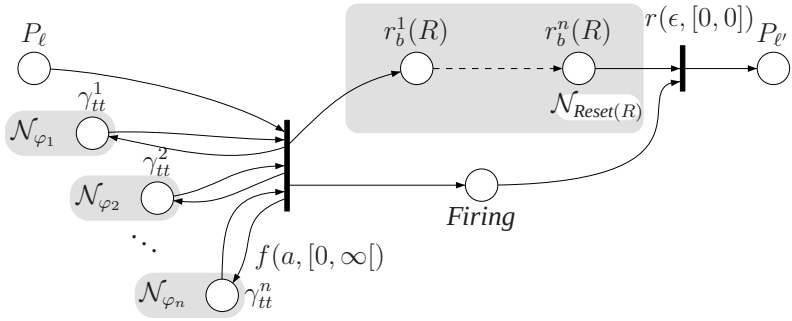


Figure 13: Widget $\mathcal{N}_e$ of an edge $e = (\ell, \gamma, a, R, \ell')$

5.4 $\Delta(\mathcal{A})$ and $\mathcal{A}$ accepts the same timed language

We now prove the following proposition:

Proposition 4 *If $\Delta(\mathcal{A})$ is defined as above, then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\Delta(\mathcal{A}))$.*

Proof. The proof works as follows: we first show that $\Delta(\mathcal{A})$ weakly simulates $\mathcal{A}$ which implies $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\Delta(\mathcal{A}))$. Then we show that we can define a TA $\mathcal{A}'$ s.t. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$ and $\mathcal{A}'$ weakly simulates $\Delta(\mathcal{A})$ which entails $\mathcal{L}(\Delta(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. It is sufficient to give the proof for the case where $\mathcal{A}$ has no ε transitions. In case $\mathcal{A}$ has ε transitions we rename them with a fresh letter $\mu \notin \Sigma_\varepsilon$ and obtain an automaton $\mathcal{A}_\mu$ with no ε transitions. We apply our construction to $\mathcal{A}_\mu$ and obtain a TPN in which we replace every label μ by ε.

Recall that $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, Inv, F, R)$ and $\Delta(\mathcal{A}) = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, M_0, \Lambda, I, F_\Delta, R_\Delta)$ and write $X = \{x_1, \cdots, x_k\}$, $P = \{p_1, \cdots, p_m\}$ and $T = \{t_1, \cdots, t_n\}$. We assume that the set of atomic constraints of $\mathcal{A}$ is $\mathcal{C}_\mathcal{A}$. The place γ_{tt} of a widget $\mathcal{N}_{x \bowtie c}$ (for $x \bowtie c$ an atomic constraint of $\mathcal{A}$) is written $\gamma_{tt}^{x \bowtie c}$.

Proof that $\Delta(\mathcal{A})$ simulates $\mathcal{A}$. We define the relation $\preceq \subseteq (L \times \mathbb{R}_{\geq 0}^n) \times (\mathbb{N}^p \times \mathbb{R}_{\geq 0}^m)$ by:

$$(\ell, v) \preceq (M, \nu) \iff \begin{cases} (1)\ M(P_\ell) = 1 \vee M(\textit{Firing}) = 1 \\ (2)\ \textit{for each } \varphi = x \bowtie c, \bowtie \in \{<, \leq\},\ M(P_u) = 0 \\ (3)\ \textit{for each } \varphi \in \mathcal{C}_\mathcal{A},\ v \in [\![\varphi]\!] \iff M(\gamma_{tt}^\varphi) = 1 \end{cases} \quad \text{(I)}$$

Now we prove that $\preceq$ is a weak simulation relation of $\mathcal{A}$ by $\Delta(\mathcal{A})$, and this by checking the 4 conditions of Def. 4:

1. final and repeated states: by definition of $\Delta(\mathcal{A})$ and lemmas 2 and 3 and the definition of $\preceq$;

2. initial states: it is clear that $(l_0, \mathbf{0}) \preceq (M_0, \mathbf{0})$;

3. continuous transitions: let $(\ell, v) \xrightarrow{d} (\ell, v+d)$. Take (M, ν) s.t. $(\ell, v) \preceq (M, \nu)$. As the widgets $\mathcal{N}_{\varphi_i}$ are non-blocking, time d can elapse from (M, ν), and there is a run $(M, \nu) \xrightarrow{\rho} (M', \nu')$ with *Duration*(*trace*$(\rho)) = d$ and *Untimed*(*trace*$(\rho)) = \varepsilon$. We can choose ρ without any transitions $f(a, [0, \infty[)$ so that a token remains in P_ℓ and $M'(P_\ell) = 1$. Thus to prove $(\ell, v+t) \preceq (M', \nu')$ it remains to prove items (2) and (3) of equation (I).

   Let $\varphi = x \bowtie c$ with $\bowtie \in \{<, \leq\}$.

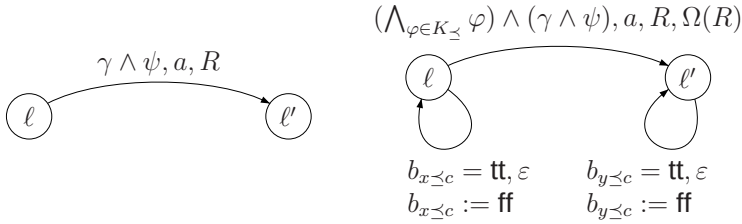
   - if $\varphi(v) = \mathsf{tt}$ and $\varphi(v+d) = \mathsf{ff}$, then there is some $d' \leq d$ s.t. transition t_x of widget $\mathcal{N}_\varphi$ is enabled and it must be fired before φ becomes false. Thus t_x is fired at d' (which is possible as there is no token in P_u and thus the token is in P_x) and subsequently u in the same widget, thus transfering the tokens from P_x, γ^φ_{tt} to P_i.
   - if $\varphi(v) = \mathsf{tt}$ and $\varphi(v+d) = \mathsf{tt}$, it is possible to do nothing in widget $\mathcal{N}_\varphi$ and let the token in P_x and γ^φ_{tt}.
   - if $\varphi(v) = \mathsf{ff}$ then $\varphi(v+d) = \mathsf{ff}$, then there must be a token in P_i and we let time elapse without firing any transition.

   Let $\varphi = x \bowtie c$ with $\bowtie \in \{>, \geq\}$.

   - if $\varphi(v) = \mathsf{tt}$ then $\varphi(v+d) = \mathsf{tt}$ and $M(\gamma^\varphi_{tt}) = 1$. We just let time elapse in $\mathcal{N}_\varphi$.
   - if $\varphi(v) = \mathsf{ff}$ and $\varphi(v+d) = \mathsf{tt}$, there is $t' \leq t$ s.t. transitions t_x must be fired (and t' can be fired at $d' + \xi$ with $\xi > 0$ for $\mathcal{N}_{x>c}$). We fire those transitions at d' and let $d - d'$ elapse.
   - if $\varphi(v) = \mathsf{ff}$ and $\varphi(v+d) = \mathsf{ff}$ we also let time elapse and leave a token in P_x.

   This way for each cosntraint $\varphi = x \bowtie c$, there is a run $\rho_\varphi = (M, \nu) \xrightarrow{d}_\varepsilon (M_\varphi, \nu_\varphi)$ s.t. (M_φ, ν_φ) satisfies requirements (2) and (3) of equation (I). Taken separately we have for each constraint $(\ell, v) \preceq (M_\varphi, \nu_\varphi)$. It is not difficult[5] to build a run ρ with an interleaving of the previous runs ρ_φ s.t. $\rho = (M, \nu) \xrightarrow{t}_\varepsilon (M', \nu')$ and (M', ν') satisfies requirements (2) and (3) of equation (I) for each constraint φ, and thus $(\ell, v) \preceq (M', \nu')$.

[5] Just find an ordering for all the date d' at which a transition must be fired and fire those transitions in this order with time elapsing between them.



(a) Edge $(\ell, \gamma \wedge \psi, a, R, \ell')$ in $\mathcal{A}$ (b) Extended edge in $\mathcal{A}'$.

Figure 14: From $\mathcal{A}$ to $\mathcal{A}'$.

4. discrete transitions: Let $(\ell, v) \xrightarrow{a} (\ell', v')$ and $(\ell, v) \preceq (M, \nu)$. Then there is an edge $e = (\ell, \gamma, a, R, \ell') \in E$ s.t. $\gamma = \wedge_{i=1,n} \varphi_i$, $n \geq 0$ and φ_i is an atomic constraint. By definition 9, $v \in [\![\varphi_i]\!]$ for $1 \leq i \leq n$. This implies $M(\gamma_{tt}^{\varphi_i}) = 1$ (definition of $\preceq$). Thus the transition $f(a, [0, \infty[)$ is fireable in the widget $\mathcal{N}_e$ leading to (M', ν'). From there on we do not change the marking of widgets $\mathcal{N}_{\varphi_i}$ for the constraints φ_i that do not need to be reset (the clock of φ_i is not in R). We also use the widget $\mathcal{T}_e$ to reset the constraints φ_i with a clock in R and finally put a token in $P_{\ell'}$. The new state (M'', ν'') obtained this way satisfies $(\ell', v') \preceq (M'', \nu'')$.

This completes the proof that $\Delta(\mathcal{A})$ simulates $\mathcal{A}$ and thus $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\Delta(\mathcal{A}))$.

Proof of $\mathcal{L}(\Delta(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A})$. To prove this, we cannot easily exhibit a simulation of $\Delta(\mathcal{A})$ by $\mathcal{A}$. Indeed, $\Delta(\mathcal{A})$, because of the widgets $\mathcal{N}_{x \bowtie c}$ with $\bowtie \in \{<, \leq\}$, has to make a decision at some point to fire transition t_x and immedialty after u, *i.e.* it is as if it decides that $x \bowtie c$ is now false and the transitions with this guard cannot be fired anymore (until they are reset). To use the simulation framework, we build first a TA $\mathcal{A}'$ that accepts the same language as $\mathcal{A}$ but has the capability to sometimes (non deterministically) decide it will not use a transition with a guard $x \bowtie c$ until it is reset. It is then possible to build a simulation relation of $\Delta(\mathcal{A})$ by $\mathcal{A}'$.

We denote $\preceq$ for either $\{<, \leq\}$ and $\succeq$ for $\{>, \geq\}$. Let $K_\preceq$ be the set of contraints $x \preceq c$ in $\mathcal{A}$. For each $x \preceq c \in K_\preceq$ we introduce a boolean variable $b_{x \preceq c}$. Each $b_{x \preceq c}$ is initially true.

We start with $\mathcal{A}' = \mathcal{A}$. The construction of the new features of $\mathcal{A}'$ is depicted on Fig. 14. Let $(\ell, \gamma \wedge \psi, a, R, \ell')$ be an edge of $\mathcal{A}'$ with $\gamma = \wedge_{x \preceq c \in K_\preceq} x \preceq c$ and $\psi = \wedge_{x \succeq c \in K_\succeq} x \succeq c$. For such an edge we strengthen[6] the guard $\gamma \wedge \psi$ to obtain γ' as follows: $\gamma' = \gamma \wedge \psi \wedge \bigwedge_{x \preceq c \in K_\preceq} b_{x \preceq c}$. This way the transition $(\ell, \gamma \wedge \psi, a, R, \ell')$ can be fired in $\mathcal{A}'$ only if the corresponding guard in $\mathcal{A}$ and the conjunction of the $b_{x \preceq c}$ is true as well. We

[6]We need an extended type of TA with boolean variables; this does not add any expressive power to the model.

also reset to true all the variables $b_{x\preceq c}$ s.t. $x \in R$ on a transition $(\ell, \gamma \wedge \psi, a, R, \ell')$ and $\Omega(R)$ corresponds to the reset of all $b_{x\preceq c}$ s.t. $x \in R$, $\Omega(R) = \wedge_{x\in R} b_{x\preceq c} := \mathsf{tt}$.

Now let ℓ be location of $\mathcal{A}'$. For each variable $b_{x\preceq c}$ we add a loop edge $(\ell, b_{x\preceq c} = \mathsf{tt}, \varepsilon, b_{x\preceq c} := \mathsf{ff}, \ell)$ in $\mathcal{A}'$, *i.e.* the automaton $\mathcal{A}'$ can decide non deterministically[7] to set $b_{x\preceq c}$ to false if it is true (see Fig. 14). There are as many loops on each location as the number of variables $b_{x\preceq c}$. The new non deterministic TA $\mathcal{A}'$ accepts exactly the same language as $\mathcal{A}$ *i.e.* $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.

We can now build a simulation relation of $\Delta(\mathcal{A})$ by $\mathcal{A}'$. We denote (ℓ, v, b) a configuration of $\mathcal{A}'$ with b the vector of b_φ variables. We define the relation $\preceq \subseteq (\mathbb{N}^p \times \mathbb{R}^m_{\geq 0}) \times (L \times \mathbb{R}^n_{\geq 0} \times \mathbb{B}^k)$ by:

$$(M, \nu) \preceq (\ell, v, b) \iff \begin{cases} (1)\, M(P_\ell) = 1 \vee M(\textit{Firing}) = 1 \\ (2)\, \forall \varphi = x > c \in K_>,\ v \in [\![\varphi]\!] \iff M(\gamma^\varphi_{tt}) = 1 \\ (3)\, \forall \varphi = x \geq c \in K_\geq,\ v \in [\![\varphi]\!] \iff M(\gamma^\varphi_{tt}) = 1 \vee \\ \qquad\qquad (M(P^\varphi_x) = 1 \wedge \nu(t^\varphi_x) = c) \\ (4) \forall \varphi \in K_\preceq, M(P^\varphi_i) = 1 \iff (b_\varphi = \mathsf{ff} \vee v \notin [\![\varphi]\!]) \end{cases} \quad \text{(II)}$$

Now we prove that $\preceq$ is a weak simulation relation of $\Delta(\mathcal{A})$ by $\mathcal{A}$.

- property on final and repeated states is satisfied by definition of $\mathcal{A}'$,
- for the initial configuration, it is clear that $(M_0, \mathbf{0}) \preceq (l_0, \mathbf{0}, b_0)$ (in b_0 all the variables b are true),
- continuous time transitions: let $(M, \nu) \xrightarrow{d} (M', \nu')$ with $d \geq 0$. Let $(M, \nu) \preceq (\ell, v, b)$. As there are no invariant in $\mathcal{A}'$ time d can elapse from (ℓ, v, b). If no ε transition fires in the TPN, then all the truth values of the constraints stay unchanged. Thus $(\ell, v, b) \xrightarrow{d} (\ell, v + d, b)$ and in $\mathcal{A}'$ s.t. $(M', \nu') \preceq (\ell, v + d, b)$.
- discrete transitions: let $(M, \nu) \xrightarrow{a} (M', \nu')$. We distinguish the cases $a = \varepsilon$ and $a \in \Sigma$.
  If $a \in \Sigma$ then we must fire a transition $f(a, [0, \infty[)$ of some widget $\mathcal{N}_e$ for $e = (\ell, \gamma, a, R, \ell')$. After firing f we end up in *Firing* and have left the input places γ_{tt} unchanged. By equation II and the definition of $\mathcal{A}'$ we can fire a matching transition in $\mathcal{A}'$ leading to a state (ℓ', v', b') and $(M', \nu') \preceq (\ell', v', b')$.
  If $a = \varepsilon$ then we are either updating some widgets $\mathcal{N}_\varphi$ or doing a reset.
  Assume we fire a ε transition that is not a reset transition ($M(P_\ell) = 1$). We split the cases according to the different types of widgets:

[7] This means we add ε transitions to $\mathcal{A}'$; nevertheless the restriction we made at the beginning that $\mathcal{A}$ has no ε transitions is useful when proving that $\Delta(\mathcal{A})$ simulates $\mathcal{A}$ and not required to prove that $\mathcal{A}'$ weakly simulates $\Delta(\mathcal{A})$.

- update of a widget $\mathcal{N}_{x>c}$: either t_x or t' is fired. If t_x is fired then the time elapsed since the x was last reset is equal to a. Thus $M(\gamma_{tt}) = 0$ and $v(x) \leq c$ and $v \notin [\![x > c]\!]$. This implies $(M', \nu') \preceq (\ell, v)$.
  If t' is fired on the contrary, $v'(x) > c$ but again $(M', \nu') \preceq (\ell, v, b)$.
- update of a widget $\mathcal{N}_{x\geq c}$: the same reasoning as before can be used and leads to $(M', \nu') \preceq (\ell, v, b)$.
- update of a widget $\mathcal{N}_{x<c}$: In this case either t_x or u is fired. Assume t_x is fired. Thus $M'(P_i) = 0$. The time elapsed since x was last reset is strictly less than c and $v \in [\![\varphi]\!]$. b_φ is true in (ℓ, v, b) as $M(P_i) = 0$. Thus $(M', \nu') \preceq (\ell, v, b)$. Now assume u is fired. Again $M(P_i) = 0$ and thus $v(x) < c$ and b_φ is true. This time $M'(P_i) = 1$. In the automaton $\mathcal{A}'$ we fire the transition setting b_φ to false and we end up in a state (ℓ, v, b') s.t. $(M', \nu') \preceq (\ell, v, b')$. The same reasoning applies for $\mathcal{N}_{x\geq c}$.

For the reset transitions and the last transition putting a token in $P_{\ell'}$ we can proceed similarly.

This completes the proof that $\mathcal{A}'$ simulates $\Delta(\mathcal{A})$ and thus $\mathcal{L}(\Delta(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}')$ and $\mathcal{L}(\Delta(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A})$.

We can thus conclude that $\mathcal{L}(\Delta(\mathcal{A})) = \mathcal{L}(\mathcal{A})$, which ends the proof of Proposition 4. ∎

5.5 Consequences of the Previous Results

Let k-$\mathcal{TPN}$ be the set of k-bounded TPNs (Note that boundedness is not decidable for TPNs). Let B-$\mathcal{TPN} = \{T \mid \exists k \geq 0 \mid T \in k\text{-}\mathcal{TPN}\}$, *i.e.* the set of bounded TPNs. From the previous proposition we can state the following corollaries:

Corollary 3 *The classes B-$\mathcal{TPN}$ and $\mathcal{TA}$ are equally expressive w.r.t. timed language acceptance, i.e. B-$\mathcal{TPN} =_{\mathcal{L}} \mathcal{TA}$.*

Proof. From Theorem 1, we know that B-$\mathcal{TPN} \leq_{\mathcal{L}} \mathcal{TA}$. Proposition 4 proves that $\mathcal{TA} \leq_{\mathcal{L}} \mathcal{TPN}$ and hence B-$\mathcal{TPN} =_{\mathcal{L}} \mathcal{TA}$. ∎

Corollary 4 k-$\mathcal{TPN} =_{\mathcal{L}} 1$-$\mathcal{TPN}$.

Proof. Let $T \in k$-$\mathcal{TPN}$. We use Theorem 1 and thus there is a TA A_T s.t. $\mathcal{L}(T) = \mathcal{L}(A_T)$. From A_T we use Proposition 4 and obtain $\Delta(A_T)$ which is a 1-safe TPN. ∎

6 Bisimulation of TA by TPNs

We now focus on the expressiveness of the models w.r.t. weak time bisimilarity. In the sequel, we often abbreviate weak timed bisimilarity by bisimilarity.

First, we recall two related results:

- There are unbounded TPNs which do not admit a bisimilar TA. This is a direct consequence of the following observation: the untimed language of a TA is regular which is not necessarily the case for PNs (and thus for TPNs).
- For any bounded TPN, there is a TA which is bisimilar to it (see Theorem 1 from [8]).

This last result was proved by the construction of a synchronized product of automata enlarged with a vector of bounded integers (a model equivalent to standard automata). The proposed construction is structural and linear w.r.t. to the size of the PN. It has the additional advantage that the available tools exploit the product in order to reduce the complexity of verification. Here we are mainly concerned with expressivity. So we can also give a straightforward construction based on the reachability space:

- With each transition t, we associate a clock x_t.
- With each reachable marking M, we associate a location ℓ_M. The invariant of ℓ_M is given by $\bigwedge_{t\in En(M)} x_t \in I(t)^{\downarrow}$, (recall that $I(t)$ is the interval associated with t).
- With each firing $M \xrightarrow{t} M'$ (in the untimed PN), we associate an edge $e = (\ell_M, \gamma, t, R, \ell_{M'})$ with $\gamma = x_t \in I(t)$ and R defined according to the chosen semantic. For instance, if PA semantics is chosen then

  $R = \{x_{t'} \mid t' \notin En(M) \wedge t' \in En(M')\}$.

In this section, we consider the TPNs originally defined by Merlin (*i.e.* without strict constraints) and labeled-free TA (*i.e.* where two different edges have different labels and no label is ε) and we develop the main result of the paper: a characterization of the subclass of TA which admit a bisimilar TPN. From this characterization, we will deduce that given a TA, the problem of deciding whether there is a TPN bisimilar to it, is $PSPACE$-complete. Furthermore, we will provide two effective constructions for such a TPN: the first one with rational constants has a size linear w.r.t. the TA, while the other, which uses only integer constants has an exponential size.

6.1 Regions of a timed automaton

Since our proofs are based on the regions of a timed automaton, we detail their definition. Recall that a *region* is a pair composed by a location and an *elementary time zone* of the grid defined by the clocks and the granularity g. In the sequel, the topology of the regions is implicitely derived from the one of its associated zone. We now formally define the particular case of regions for a maximal constant $K = \infty$. Obviously it may lead to an infinite region automaton but will be a helpful tool for proving our characterization. Note also that the following definition is equivalent to the original one but is more appropriate for our theoretical developments.

Definition 12 (Regions of an automaton w.r.t. the g-grid and constant $K = \infty$) *A time-closed region r is given by:*

- *ℓ_r the location of r,*
- *$min_r \in \mathbb{N}_g^X$ the minimal vector of the topological closure of r,*
- *The number $size_r$ of different fractional parts of clock values in the grid $\mathbb{N}_g^X$, with $1 \leq size_r \leq |X|$ and the onto mapping $ord_r : X \mapsto \{1, \ldots, size_r\}$ which gives the relative positions of these fractional parts,*

The region is then $r = \{(\ell_r, min_r + \boldsymbol{\delta}) \,|\, \boldsymbol{\delta} \in \mathbb{R}^X_{\geq 0} \ \wedge \ \forall x, y \in X [ord_r(x) = 1 \Leftrightarrow \boldsymbol{\delta}(x) = 0] \wedge \boldsymbol{\delta}(x) < 1/g \ \wedge \ [ord_r(x) < ord_r(y) \Leftrightarrow \boldsymbol{\delta}(x) < \boldsymbol{\delta}(y)]\}$.

A time-open region r is defined with the same attributes as the time-closed region by: $r = \{(\ell_r, min_r + \boldsymbol{\delta} + d) \,|\, d \in \mathbb{R}_{>0} \ \wedge \ \forall x \in X, \, \boldsymbol{\delta}(x) + d < 1/g\}$.

The set $[X]_r$ is the set of equivalence classes of clocks w.r.t. their fractional parts, i.e. x and y are equivalent iff $ord_r(x) = ord_r(y)$.

This definition needs to be slightly modified when dealing with a constant $K < \infty$, by introducing a subset of *relevant* clocks, for which the value is less than K (recall that $K > m$ where m is the maximal constant in the constraints of the timed automaton).

Definition 13 (Regions of an automaton w.r.t. the g-grid and finite constant K) *A time-closed region r is given by:*

- *ℓ_r the location of r,*
- *$min_r \in \mathbb{N}_g^X$ with $\forall x, \ min_r(x) \leq K$ the minimal vector of the topological closure of r,*

- $ActX_r = \{x \in X \mid min_r(x) < K\}$ *the subset of relevant clocks,*
- *the number* $size_r$ *of different fractionnal parts for the values of relevant clocks in the* $\mathbb{N}_g^{ActX_r}$ *grid, with* $1 \leq size_r \leq Max(|ActX_r|, 1)$ *and the onto mapping* $ord_r : X \mapsto \{1, \ldots, size_r\}$ *giving the ordering of the fractionnal parts. By convention,* $\forall x \in X \setminus ActX_r, ord_r(x) = 1$.

Then $r = \{(\ell_r, min_r + \boldsymbol{\delta}) \mid \boldsymbol{\delta} \in \mathbb{R}^X_{\geq 0} \wedge \forall x, y \in ActX_r[ord_r(x) = 1 \Leftrightarrow \boldsymbol{\delta}(x) = 0] \wedge \boldsymbol{\delta}(x) < 1/g \wedge [ord_r(x) < ord_r(y) \Leftrightarrow \boldsymbol{\delta}(x) < \boldsymbol{\delta}(y)]\}$

A time-open (description of) a region r *is given by the same attributes (and conditions) as a time-closed region with:*
$r = \{(\ell_r, min_r + \boldsymbol{\delta} + d) \mid d \in \mathbb{R}_{>0} \wedge \forall x \in ActX_r, \boldsymbol{\delta}(x) + d < 1/g\}$.

Note that letting time elapse leads to an alternation of time-open regions (where time can elapse) and time-closed ones (where no time can elapse). We also remark that $min_r \notin r$ except if there is a single class of clocks relative to r (for instance if r is a singleton). More generally, whatever be the grid and the maximal constant, we note $\overline{r}$, the topological closure of r: it is a finite union of regions and from the definition, min_r is the minimum vector of $\overline{r}$.

Reachability. Recall that a region is *reachable* if it belongs to the region automaton. However it does not mean that all the configurations of the region are reachable. Nevertheless, by induction on the reachability relation inside the region automaton it can be shown that every configuration is "quasi-reachable" in the following sense:
For each reachable region r, there is a region $reach(r)$ w.r.t. the 1-grid and the constant ∞ such that:

- $reach(r) \subset r$;
- each configuration of $reach(r)$ is reachable;
- if $reach(r)$ is a time-open region then r admits a time-open description else r admits a time-closed description.

Note that consequently $\forall x \in ActX_r, min_{reach(r)}(x) = min_r(x)$ and $\forall x \in X \setminus ActX_r, min_{reach(r)}(x) \geq K$ and that ord_r restricted to $ActX_r$ is identical to $ord_{reach(r)}$.

Let us define $\mathcal{R}$ by $(l, v)\mathcal{R}(l, v')$ iff $\forall x \in X, v'(x) = v(x) \vee (v(x) \geq K \wedge v'(x) \geq K)$. Then $\mathcal{R}$ is a strong time bisimulation relation. From the previous observations, we note that each configuration of a reachable region is strongly time bisimilar to a reachable configuration of this region. Thus speaking about reachability of regions is a slight abuse of notations.

6.2 From bisimulation to uniform bisimulation

As a first step toward our characterization, we prove that when a TPN and a TA are bisimilar, the condition can in fact be strengthened in what we call uniform bisimulation.

We first prove a lemma which is also a strengthened version of lemma 1. It points out the effect of time granularity on the behaviour of TPN when strict constraints are excluded.

Lemma 5 *Let (M, ν) and $(M, \nu + \boldsymbol{\delta})$ be two admissible configurations of a g-TPN with $\nu, \boldsymbol{\delta} \in \mathbb{R}_{\geq 0}^{En(M)}$. Let w be an instantaneous firing sequence, then:*
(a) $(M, \nu) \xrightarrow{w} \Rightarrow (M, \nu + \boldsymbol{\delta}) \xrightarrow{w}$
(b) If $\nu \in \mathbb{N}_g^{En(M)}$ *and* $\boldsymbol{\delta} \in [0, 1/g[^{En(M)}$ *then* $(M, \nu + \boldsymbol{\delta}) \xrightarrow{w} \Rightarrow (M, \nu) \xrightarrow{w}$

Proof. There are two kinds of transitions firing in w: those corresponding to a firing of a transition (say t) still enabled from the beginning of the firing sequence and those corresponding to a newly enabled transition (say t').
Proof of (a) Since t is firable from (M, ν), $\nu(t) \in I(t) \subset I(t)^{\uparrow}$, so $\nu(t) + \boldsymbol{\delta}(t) \geq \nu(t)$ also belongs to $I(t)^{\uparrow}$. Since $t \in En(M)$ and $(M, \nu + \boldsymbol{\delta})$ is reachable, $\nu(t) + \boldsymbol{\delta}(t) \in I(t)^{\downarrow}$. Thus $\nu(t) + \boldsymbol{\delta}(t) \in I(t)$ and t is also firable from $(M, \nu + \boldsymbol{\delta})$. Since t' is newly enabled, $0 \in I(t')$ and t' is also firable when it occurs starting from $(M, \nu + \boldsymbol{\delta})$.
Proof of (b) The case of newly enabled transitions in w is handled as before. Now let t be firable in $(M, \nu + \boldsymbol{\delta})$. Since $t \in En(M)$ and (M, ν) is reachable, $\nu(t) \in I(t)^{\downarrow}$. Since $\nu(t) + \boldsymbol{\delta}(t) \in I(t)^{\uparrow}$, (denoting by $eft(t)$ the minimum of $I(t)^{\uparrow}$), we have $eft(t) \leq \nu(t) + \boldsymbol{\delta}(t)$ but $eft(t)$ belongs to the g-grid, thus $eft(t) \leq \nu(t) \Leftrightarrow \nu(t) \in I(t)^{\uparrow}$. So t is firable from (M, ν). ∎

Lemma 6 (From bisimulation to uniform bisimulation) *Consider a timed automaton $\mathcal{A}$ bisimilar to some g-TPN $\mathcal{N}$ via some relation $\mathcal{R}$. The semantics considered for $\mathcal{N}$ is PA as it gives the maximal expressivity. We consider the region automaton of $\mathcal{A}$ w.r.t. the grid $\mathbb{N}_g^X$ and the constant $K = \infty$. Then:*

- *if a region r belongs to $R(\mathcal{A})$ then $\overline{r}$ also belongs to $R(\mathcal{A})$;*
- *with each reachable region r is associated a configuration of the net (M_r, ν_r) with $\nu_r \in \mathbb{N}_g^{En(M_r)}$ and a mapping $\phi_r : En(M_r) \to [X]_r$ which fulfill:*
  - *If r is time-closed, then $\forall (\ell_r, min_r + \boldsymbol{\delta}) \in \overline{r}$,*
    $(\ell_r, min_r + \boldsymbol{\delta}) \mathcal{R} (M_r, \nu_r + proj_r(\boldsymbol{\delta}))$
    where $proj_r(\boldsymbol{\delta})(t) = \boldsymbol{\delta}(\phi_r(t))$.
  - *If r is time-open, then $\forall (\ell_r, min_r + \boldsymbol{\delta} + d) \in \overline{r}$,*
    $(\ell_r, min_r + \boldsymbol{\delta} + d) \mathcal{R} (M_r, \nu_r + proj_r(\boldsymbol{\delta}) + d)$

Proof. We prove this uniform version of the bisimulation by induction on the reachability relation between regions. First note that the choice of a particular clock x in the class $\phi_r(t)$ is irrelevant when considering the value $\boldsymbol{\delta}(x)$. Thus the definition of $proj_r$ is sound.

We prove our assertion by induction on the transition relation in the region automaton. The basis case is straightforward with $\{(l_0, \mathbf{0})\}$ and $\{(M_0, \mathbf{0})\}$. For the induction part, we consider 4 cases, according to the incoming or target region and to the nature of the step.

1. A time step from a time-closed region Let r be a time-closed region which is not maximal and let us denote $r' = succ(r)$ the immediate time successor of r. Let $(\ell_r, min_r + \boldsymbol{\delta_0})$ be some item of r. $(\ell_r, min_r + \boldsymbol{\delta_0}) \xrightarrow{d}$ for some $d > 0$. Thus (by induction hypothesis) in $\mathcal{N}$ there is a step sequence of $(M_r, \nu_r + proj_r(\boldsymbol{\delta_0})) \xrightarrow{d_0 t_1 \dots t_n d_n}$ with all transitions labelled by ϵ and $\sum d_k = d$. Let d_k be the first non zero elapsing of time. By application of lemma 5-b, the firing sequence $t_1 \dots t_k$ is firable from (M_r, ν_r).

Let us choose $(M_{r'}, \nu_{r'})$ the configuration reached by this sequence. By application of lemma 5-a, this firing sequence is also fireable from any $(M_r, \nu_r + proj_r(\boldsymbol{\delta}))$ bisimilar to $(\ell_r, min_r + \boldsymbol{\delta}) \in \overline{r}$ and it leads to $(M_{r'}, \nu_{r'} + proj_{r'}(\boldsymbol{\delta}))$ (still bisimilar to $(\ell_r, min_r + \boldsymbol{\delta})$) where $\phi_{r'}$ (resp. $\nu_{r'}$) is equal to ϕ_r (resp. ν_r) for transitions always enabled during the firing sequence and $\phi_{r'}$ (resp. $\nu_{r'}$) is obtained by associating the class of index 1 (resp. by associating the value 0) to the transitions newly enabled. Since $(M_{r'}, \nu_{r'})$ let the time elapse and since $\mathcal{N}$ is a g-TPN, we note that $\forall t \in En(M_{r'}),\ \nu_{r'}(t) + 1/g \in I(t)^{\downarrow}$. Now let $(\ell_r, min_r + \boldsymbol{\delta} + d) \in \overline{r'}$, one has $\forall x \in X,\ \boldsymbol{\delta}(x) + d \leq 1/g$. Thus $\forall t \in En(M_{r'}), proj_{r'}(\boldsymbol{\delta}(x)) + d \leq 1/g$, which implies $(M_{r'}, \nu_{r'} + proj_{r'}(\boldsymbol{\delta})) \xrightarrow{d} (M_{r'}, \nu_{r'} + proj_{r'}(\boldsymbol{\delta}) + d)$, this last configuration being necessarily bisimilar to $(\ell_r, min_r + \boldsymbol{\delta} + d)$.

2. A time step from a time-open region. Let r be an time-open region and let us denote $r' = succ(r)$. Let us define X_r^{max} the class $[x]_r$ with maximal index. We remark that $min_{r'} = min_r + \boldsymbol{\delta_0}$ where if $x \in X_r^{max}$ then $\boldsymbol{\delta_0}(x) = 1/g$ else $\boldsymbol{\delta_0}(x) = 0$. We choose $(M_{r'}, \nu_{r'}) = (M_r, \nu_r + proj_r(\boldsymbol{\delta_0}))$. Let $t \in En(M_r)$ and $x \in \phi_r(t)$ then $\phi_{r'}(t) = [x]_{r'}$ (letting time elapse does not split the classes). So $proj_r$ and $proj_{r'}$ are identical.

Now let $(l_{r'}, min_{r'} + \boldsymbol{\delta}) \in \overline{r'}$. $(l_{r'}, min_{r'} + \boldsymbol{\delta}) = (\ell_r, min_r + \boldsymbol{\delta_0} + \boldsymbol{\delta})$.

Now let $d = \boldsymbol{\delta}(x)$ for x belonging the class of index 1 in $[X_r]$. Then $(\ell_r, min_r + \boldsymbol{\delta_0} + \boldsymbol{\delta}) = (\ell_r, min_r + \boldsymbol{\delta'} + d)$ where if $x \in X_r^{max}$ then $\boldsymbol{\delta'}(x) = 1/g - d$ else $\boldsymbol{\delta'}(x) = \boldsymbol{\delta}(x) - d$. $(\ell_r, min_r + \boldsymbol{\delta'} + d)$ is bisimilar to $(M_r, \nu_r + proj_r(\boldsymbol{\delta'}) + d) = (M_r, \nu_r + proj_r(\boldsymbol{\delta'} + d)) = (M_r, \nu_r + proj_r(\boldsymbol{\delta_1} + \boldsymbol{\delta})) = (M_r, \nu_r + proj_r(\boldsymbol{\delta_1}) + proj_r(\boldsymbol{\delta})) = (M_{r'}, \nu_{r'} + proj_{r'}(\boldsymbol{\delta})))$.

For this step, we have not used the characteristics of time Petri nets.

3. A discrete step into a time-closed region.

Case a. We first consider the case where r is a time-closed region.

Let $(\ell_r, \min_r + \boldsymbol{\delta_0})$ be some element of r. Suppose that $(\ell_r, \min_r + \boldsymbol{\delta_0}) \xrightarrow{e} (l', v' +$

$\boldsymbol{\delta'_0}$) with $\forall x \in R(e), v'(x) = \boldsymbol{\delta'_0}(x) = 0$, $\forall x \notin R(e), v'(x) = \min_r(x) \wedge \boldsymbol{\delta'_0}(x) = \boldsymbol{\delta_0}(x)$. Then in $\mathcal{N}$ there is a firing sequence $(M_r, \nu_r + proj_r(\boldsymbol{\delta_0})) \xrightarrow{w}$ labelled by e. Due to lemma 5, this firing sequence is also fireable from any $(M_r, \nu_r + proj_r(\boldsymbol{\delta}))$ bisimilar to $(\ell_r, min_r + \boldsymbol{\delta}) \in \overline{r}$. By bisimilarity, $(\ell_r, \min_r + \boldsymbol{\delta}) \xrightarrow{e}$ for any $(\ell_r, min_r + \boldsymbol{\delta}) \in \overline{r}$. Let r' be the region including $(\ell', v' + \boldsymbol{\delta'_0})$, then any configuration of $\overline{r'}$ is reachable by this discrete step. Note that $\ell_{r'} = l'$ and $min_{r'} = v'$.

From $(M_r, \nu_r + proj_r(\boldsymbol{\delta}))$, the sequence w leads to some (M', ν') bisimilar to $(\ell_{r'}, min_{r'} + \boldsymbol{\delta'})$. We now show how to define $M_{r'}$, $\nu_{r'}$ and $\phi_{r'}$. First $M_{r'} = M'$. Second, $\nu_{r'}(t) = \nu_r(t)$ for transitions t always enabled during the firing sequence and $\nu_{r'} = 0$ otherwise. At last, $\phi_{r'}$ is obtained from ϕ_r as follows. Let t be a transition newly enabled during the firing sequence, then $\phi_{r'}(t)$ is associated to the class of index 1. Let t be a transition always enabled during the firing sequence. There are three cases to consider for $\phi_{r'}(t)$: either there is a $x \in \phi_r(t)$ not reset, then $\phi_{r'}(t) = |x]_{r'}$ otherwise $\phi_{r'}(t)$ is the class of maximal index which preceedes $\phi_r(t)$ and contains a clock not reset or else the class of index 1. The two last affectations are sound since it means that whatever the value of $\boldsymbol{\delta}(t)$ fulfilling the order between classes, the firing sequence w leads to bisimilar configurations (as being bisimilar to the same configuration of the automaton).

Case b. The case where r is a time-open region is handled in a similar way. Let $(\ell_r, \min_r + \boldsymbol{\delta_0} + d_0)$ be some element of r. Suppose that $(\ell_r, \min_r + \boldsymbol{\delta_0} + d_0) \xrightarrow{e} (\ell', v' + \boldsymbol{\delta'_0})$ with $\forall x \in R(e), v'(x) = \boldsymbol{\delta'_0}(x) = 0$, $\forall x \notin R(e), v'(x) = \min_r(x) \wedge \boldsymbol{\delta'_0}(x) = \boldsymbol{\delta_0}(x) + d_0$. Then in $\mathcal{N}$ there is a firing sequence $(M_r, \nu_r + proj_r(\boldsymbol{\delta_0}) + d_0) \xrightarrow{w}$ labelled by e. Due to lemma 5, this firing sequence is also fireable from any $(M_r, \nu_r + proj_r(\boldsymbol{\delta}) + d)$ bisimilar to $(\ell_r, min_r + \boldsymbol{\delta} + d) \in \overline{r}$. By bisimilarity, $(\ell_r, \min_r + \boldsymbol{\delta} + d) \xrightarrow{e}$ for any $(\ell_r, min_r + \boldsymbol{\delta} + d) \in \overline{r}$. Let r' be the region including $(l', v' + \boldsymbol{\delta'_0})$, then any configuration of $\overline{r'}$ is reachable by this discrete step. Note that $l_{r'} = l'$ and $min_{r'} = v'$.

From $(M_r, \nu_r + proj_r(\boldsymbol{\delta}) + d)$, the sequence w leads to some (M', ν') bisimilar to $(l_{r'}, min_{r'} + \boldsymbol{\delta'})$. We now show how to define $M_{r'}$, $\nu_{r'}$ and $\phi_{r'}$. First $M_{r'} = M'$. Second, $\nu_{r'}(t) = \nu_r(t)$ for transitions t always enabled during the firing sequence and $\nu_{r'} = 0$ otherwise. At last, $\phi_{r'}$ is obtained from ϕ_r as follows. Let t be a transition newly enabled during the firing sequence, then $\phi_{r'}(t)$ is associated to the class of index 1. There are three cases to consider for $\phi_{r'}(t)$: either there is a $x \in \phi_r(t)$ not reset, then $\phi_{r'}(t) = |x]_{r'}$ otherwise $\phi_{r'}(t)$ is the class of maximal index which preceedes $\phi_r(t)$ and contains a clock not reset or else the class of index 1. The two last affectations are sound since it means that whatever the value of $\boldsymbol{\delta}(t)$ fulfilling the order between classes, the firing sequence w leads to bisimilar configurations (as being bisimilar to the same configuration of the automaton).

4. A discrete step into a time-open region. In order to reach a time-open region by a

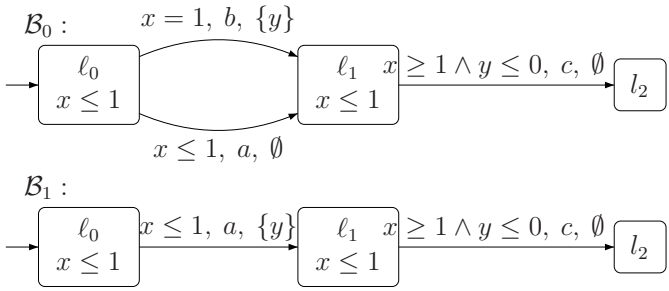


Figure 15: Two automata with different behavior w.r.t bisimulation with a TPN

discrete step, the corresponding transition must start from a time-open region and must not reset any clock. Let $(\ell_r, min_r + \boldsymbol{\delta} + d) \in r$ and $(\ell_r, min_r + \boldsymbol{\delta} + d) \xrightarrow{e} (l', min_r + \boldsymbol{\delta} + d)$. Here we have used the hypothesis that no clock is reset. Then there is a firing sequence $(M_r, \nu_r + proj_r(\boldsymbol{\delta}) + d) \xrightarrow{w}$ labelled by e. Due to the lemma 5, $(M_r, \nu_r + proj_r(\boldsymbol{\delta})) \xrightarrow{w}$. $(\ell_r, v_r + \boldsymbol{\delta})$ is bisimilar to $(M_r, \nu_r + proj_r(\boldsymbol{\delta}))$. Thus $(\ell_r, min_r + \boldsymbol{\delta}) \xrightarrow{e} (l', min_r + \boldsymbol{\delta}) \xrightarrow{d} (l', min_r + \boldsymbol{\delta} + d)$. Then this region can be reached via a discrete step into a time-closed region followed by a time step. So we do not need to examine this case. ∎

6.3 A characterization of bisimilarity

The characterization of TA bisimilar to some TPN is closely related to the topological closure of reachable regions: it states that any region intersecting the topological closure of a reachable region is also reachable and that a discrete step either from a region or from the minimal vector of its topological closure is possible in the whole topological closure. The two automata $\mathcal{B}_0$ and $\mathcal{B}_1$ in Figure 6.3 will illustrate our results: the automaton $\mathcal{B}_0$ admits a bisimilar TPN whereas $\mathcal{B}_1$ does not.In the sequel, we suppose that any atomic constraint related to a clock x occurring in the invariant of a location is added to the guard of each incoming transition which does not reset x.

Theorem 5 (Characterization of TA bisimilar to some TPN) *Let $\mathcal{A}$ be a (label-free) timed automaton, let $R(\mathcal{A})$ its region automaton w.r.t. the 1-grid and a constant K strictly greater than any constant occurring in the automaton, then $\mathcal{A}$ is weakly timed bisimilar to a time Petri net iff:*
$\forall r \in R(\mathcal{A}), \forall e$ *an edge of* $\mathcal{A}$,
(a) Every region r' s.t. $r' \cap \overline{r} \neq \emptyset$ is reachable
(b) $\forall (\ell_r, v) \in r$, $(\ell_r, v) \xrightarrow{e}\Rightarrow (\ell_r, min_r) \xrightarrow{e}$
(c) $\forall (\ell_r, v) \in \overline{r}$, $(\ell_r, min_r) \xrightarrow{e}\Rightarrow (\ell_r, v) \xrightarrow{e}$

Furthermore, if these conditions are satisfied then we can build a 1-bounded 2-TPN bisimilar to $\mathcal{A}$ whose size is linear w.r.t. the size of $\mathcal{A}$ and a 1-bounded 1-TPN bisimilar to $\mathcal{A}$ whose size is exponential w.r.t. the size of $\mathcal{A}$.

We note $\mathcal{TA}^-$ this class of automata. Using the theorem, we justify why the automaton $\mathcal{B}_1$ does not admit a bisimilar TPN. The region $r = \{(\ell_1, x = 1 \wedge 0 < y < 1\}$ is reachable. The guard of edge c is true in $min_r = (\ell_1, (1, 0))$ whereas it is false in r.

We prove Theorem 5 in three steps in the next paragraphs.

6.4 Proof of Necessity

Proof.[of Necessity] The fact that conditions *(a)*, *(b)*, and *(c)* are satisfied with respect to the g-grid and the constant $K = \infty$ is straightforward:

- *(a)* This assertion is included in the inductive assertions.
- *(b)* Let r be a reachable region, let $(\ell_r, min_r + \boldsymbol{\delta}) \in r$ be a configuration with $\boldsymbol{\delta} \in [0, 1/g[^X$, then $\exists (M, \nu)\ \nu \in \mathbb{N}_g^{En(M)}$ bisimilar to (ℓ_r, min_r) and $(M, \nu + \boldsymbol{\delta'})$ with $\boldsymbol{\delta'} \in [0, 1/g[^{En(M)}$ bisimilar to $(\ell_r, v + \boldsymbol{\delta})$. Suppose that $(\ell_r, min_r + \boldsymbol{\delta}) \xrightarrow{e}$, then $(M, \nu + \boldsymbol{\delta'}) \xrightarrow{w}$ with w an instantaneous firing sequence and $label(w) = e$. Now by lemma 5-b, $(M, \nu) \xrightarrow{w}$, thus $(\ell_r, min_r) \xrightarrow{e}$.
- *(c)* Let r be a region, and $(\ell_r, min_r + \boldsymbol{\delta}) \in \overline{r}$ with $\boldsymbol{\delta} \in [0, 1/g]^X$ thus $\exists (M, \nu)$ bisimilar to (ℓ_r, min_r) and $(M, \nu + \boldsymbol{\delta'})$ with $\boldsymbol{\delta'} \in [0, 1/g]^{En(M)}$ bisimilar to $(\ell_r, min_r + \boldsymbol{\delta})$. Suppose that $(\ell_r, min_r) \xrightarrow{e}$, then $(M, \nu) \xrightarrow{w}$ with w an instantaneous firing sequence and $label(w) = e$. Now by lemma 5-a, $(M, \nu + \boldsymbol{\delta'}) \xrightarrow{w}$, thus $(\ell_r, min_r + \boldsymbol{\delta}) \xrightarrow{e}$.

In order to complete the proof, we successively show that if the conditions are satisfied w.r.t. the g-grid and infinite constant, they are satisfied w.r.t. the 1-grid and infinite constant and when satisfied w.r.t the 1-grid and infinite constant, they are satisfied w.r.t the 1-grid and the usual finite constant. This is done by the next two lemmas.

Lemma 7 (about the conditions and the grid) *Let $\mathcal{A}$ be a timed automaton, and $g \in \mathbb{N}_{>0}$. If the conditions (a),(b),(c) are satisfied by the region automaton associated with the g-grid, then they are satisfied by the region automaton associated with the 1-grid (where in both cases the constant $K = \infty$).*

Proof. Let us denote $R(\mathcal{A})^g$ the region automaton of A w.r.t. the g-grid. By definition of regions, we remark that r a region of $R(\mathcal{A})$ is a finite union of regions of $R(\mathcal{A})^g$ (say $r = \bigcup_{i=1..k} r_i$). Thus $\overline{r} = \bigcup_{i=1..k} \overline{r_i}$ which proves the implication for (a).

Assume that (b) is satisfied by $R(\mathcal{A})^g$. Let $(\ell_r, min_r + \boldsymbol{\delta} + d) \in r$ be a region of $R(\mathcal{A})$ and assume $(\ell_r, min_r + \boldsymbol{\delta} + d) \xrightarrow{e}$. We define $\boldsymbol{\delta'}$ by $\boldsymbol{\delta'}(x) = \boldsymbol{\delta}(x)/g$. Then since $\mathcal{A}$ has integer constraints $(\ell_r, min_r + \boldsymbol{\delta'} + d/g) \xrightarrow{e}$. Moreover this configuration belongs to r and then to a region $r' \in R(\mathcal{A})^g$ whose minimal vector is min_r. Then applying ***(b)***, we obtain $(\ell_r, min_r) \xrightarrow{e}$.

Assume that ***(c)*** is satisfied by $R(\mathcal{A})^g$. Let $(\ell_r, v) \in \overline{r}$ where r is a region of $R(\mathcal{A})$ and assume $(\ell_r, min_r) \xrightarrow{e}$. Then there is an increasing path among the minimum vectors of regions of $R(\mathcal{A})^g$ all included in $\overline{r}$. This path is such that any two consecutive elements belong to the closure of some region; it starts at (ℓ_r, min_r) and finishes at (ℓ_r, min_{r_*}) such that $(\ell_r, v) \in \overline{r_*}$ (with r_* a region of $R(\mathcal{A})^g$). Thus applying iteratively ***(c)*** yields $(\ell_r, v) \xrightarrow{e}$. ∎

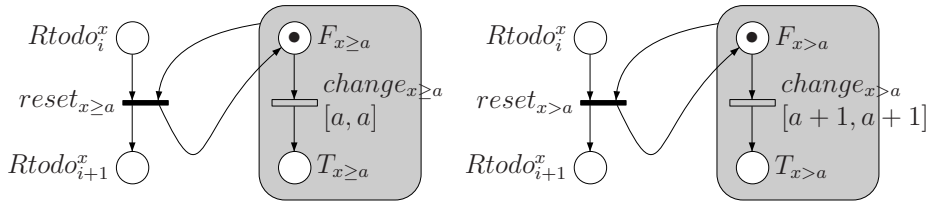
Lemma 8 (about the conditions and the constant K) *Let $\mathcal{A}$ be a timed automaton. If the conditions (a),(b),(c) are satisfied by the region automaton associated with the 1-grid and constant $K = \infty$, then they are satisfied by the region automaton associated to the 1-grid and a finite constant.*

Proof. Let us denote $R(\mathcal{A})^\infty$ the region automaton of A w.r.t. $K = \infty$. Let r be a reachable region in $R(\mathcal{A})$ and $reach(r)$ the associated region of $R(\mathcal{A})^\infty$. Note that $\ell_{reach(r)} = \ell_r$ and that $\forall x \in ActX_r, min_{reach(r)} = min_r$ and $\forall x \in X, min_{reach(r)} \geq min_r$. Suppose that $reach(r)$ is time-closed (resp. time-open) then r admits a time-closed (resp. time-open) description where the ord_r and $ord_{reach(r)}$ mappings are identical for clocks in $ActX_r$. Thus $\forall (\ell_r, v) \in r, \exists (\ell_r, v') \in reach(r)$ such that $\forall x \in ActX_r, v'(x) = v(x)$.

Now take a convergent sequence $lim_{i\to\infty}(\ell_r, v_i) = (\ell_r, v)$ with $(\ell_r, v_i) \in r$ so that $(\ell_r, v) \in \overline{r}$. Then the corresponding sequence $\{(\ell_r, v'_i)\}$ being bounded admits an accumulation point $(\ell_r, v') \in \overline{r}$. It is routine to show that (ℓ_r, v) and (ℓ_r, v') belong to the same region in $R(\mathcal{A})$. This proves that condition ***(a)*** for $R(\mathcal{A})^\infty$ implies condition ***(a)*** for $R(\mathcal{A})$.

Assume that ***(b)*** is satisfied by $R(\mathcal{A})^\infty$. Let $(\ell_r, v) \in r$ be a reachable region of $R(\mathcal{A})$ and $(\ell_r, v) \xrightarrow{e}$. Let $reach(r)$ be the associated reachable region of $R(\mathcal{A})^\infty$ then $\exists (\ell_r, v') \in reach(r)$ strongly time bisimilar to (ℓ_r, v), thus $(\ell_r, v') \xrightarrow{e}$. Using condition ***(b)***, $(\ell_r, min_{reach(r)}) \xrightarrow{e}$. Since $(\ell_r, min_{reach(r)})$ is strongly time bisimilar to (ℓ_r, min_r), we have $(\ell_r, min_r) \xrightarrow{e}$.

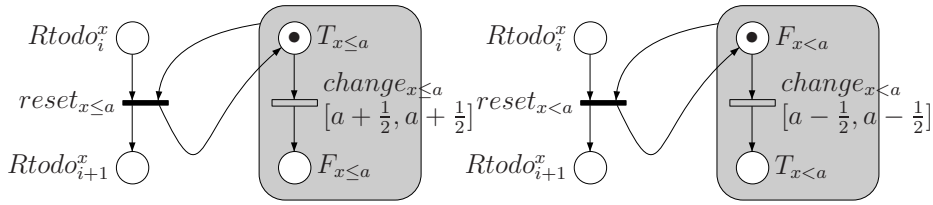
Assume that ***(c)*** is satisfied by $R(\mathcal{A})^\infty$ and consider $(\ell_r, v) \in \overline{r}$ where r is a region of $R(\mathcal{A})$ and $(\ell_r, min_r) \xrightarrow{e}$. Let $reach(r)$ be the associated reachable region of $R(\mathcal{A})^\infty$, then $\exists (\ell_r, v') \in \overline{reach(r)}$ strongly time bisimilar to (ℓ_r, v). Since $(\ell_r, min_{reach(r)})$ is strongly time bisimilar to (ℓ_r, min_r), $(\ell_r, min_{reach(r)}) \xrightarrow{e}$. Thus using condition ***(c)***, $(\ell_r, v') \xrightarrow{e}$. By bisimilarity, we obtain $(\ell_r, v) \xrightarrow{e}$. ∎



(a) Widget for condition $x \geq a$ (assume $a > 0$)

(b) Widget for condition $x > a$

Figure 16: Widgets for conditions $x \geq a$ and $x > a$



(a) Widget for condition $x \leq a$

(b) Widget for condition $x < a$ (assume $a > 0$)

Figure 17: Widgets for conditions $x \leq a$ and $x < a$

We now give the proof that the condition is sufficient. The proof is split into two parts, corresponding respectively to the construction of a 2-TPN and the construction of a 1-TPN.

6.5 First construction

Proof.[for the first construction of sufficiency] We first describe the construction of a 2-TPN $\mathcal{N}$ bisimilar to $\mathcal{A}$. The principles of this construction are similar to those used for the language equivalence. We build a subnet per elementary condition (including the part associated with the clock resetting). However except for the conditions $x \geq c$ and the resetting part, all the constructions are different. We first remark that $x < a$ occurring in an invariant may be safely omitted. Indeed (see the assumptions on timed automata), it never forbids to enter the state. If it would forbid the progress of time in some configuration, then the associated region would be a maximal time-open region r. Due to condition (a), $\overline{r}$ is reachable but since r is time-open, $\overline{r} \cap succ(r) \neq \emptyset$, so that $succ(r)$ is reachable which contradicts the maximality of r.

All edges of $\mathcal{N}$ are weighted by 1. Unless explicitely stated, the transitions are labelled by ϵ.

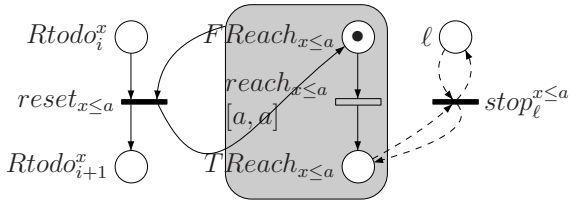


Figure 18: Widget for invariant $x \leq a$

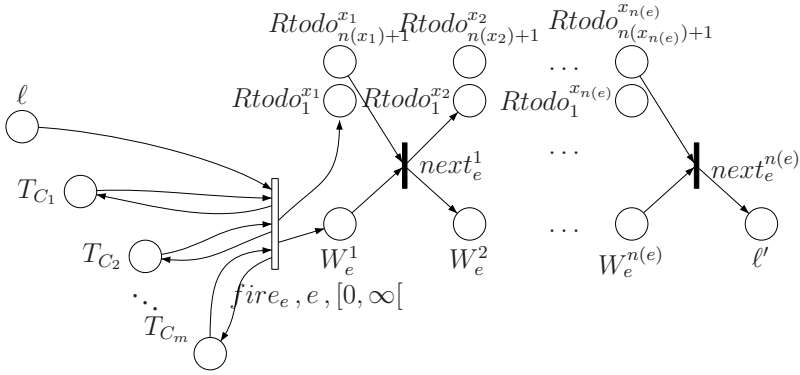


Figure 19: Widget of an edge $(l, \gamma = \{c_1, \ldots, c_{m(e)}\}, e, R = \{x_1, \ldots, x_{n(e)}\}, l')$

- With each location ℓ of the automaton, we associate an eponymous place ℓ. The place ℓ is initially marked iff the location ℓ is the initial one.

- The conditions associated with a clock x are arbitrarily numbered from 1 to $n(x)$ where $n(x)$ is the number of such conditions. We consider that when $x \leq a$ ($a \neq 0$) occurs in at least one transition and in at least one invariant it is associated to two different conditions. Then we add places $\{Rtodo_i^x\}_{i \leq n(x)+1}$ for the management of the resets.

- With each condition $x \geq a$ ($a \neq 0$) occurring in a transition of the automaton, we associate a widget (see Figure 16 (a)) composed by two places $T_{x \geq a}, F_{x \geq a}$ and three transitions $change_{x \geq a}, reset^1_{x \geq a}, reset^2_{x \geq a}$. The place $F_{x \geq a}$ is initially marked while $T_{x \geq a}$ is unmarked. The interval associated to $change_{x \geq a}$ is $[a, a]$; ${}^\bullet change_{x \geq a} = \{F_{x \geq a}\}$ and $change_{x \geq a}{}^\bullet = \{T_{x \geq a}\}$. The interval associated to $reset^1_{x \geq a}$ and $reset^2_{x \geq a}$ is $[0, 0]$. Let i be the number of the condition $x \geq a$. ${}^\bullet reset^1_{x \geq a} = \{F_{x \geq a}, Rtodo_i^x\}$ and $reset^1_{x \geq a}{}^\bullet = \{F_{x \geq a}, Rtodo_{i+1}^x\}$. ${}^\bullet reset^2_{x \geq a} = \{T_{x \geq a}, Rtodo_i^x\}$ and $reset^2_{x \geq a}{}^\bullet = \{F_{x \geq a}, Rtodo_{i+1}^x\}$.

- With each condition $x > a$ occurring in a transition of the automaton, we associate a widget (Figure 16 (b)) composed by two places $T_{x > a}, F_{x > a}$ and three transitions $change_{x > a}, reset^1_{x > a}, reset^2_{x > a}$. The place $F_{x > a}$ is initially marked while $T_{x > a}$ is unmarked. The interval associated to $change_{x > a}$ is $[a + 1, a + 1]$; ${}^\bullet change_{x > a} = \{F_{x > a}\}$ and $change_{x > a}{}^\bullet = \{T_{x > a}\}$. The interval associated to $reset^1_{x > a}$ and $reset^2_{x > a}$ is $[0, 0]$. Let i be the number of the condition $x > a$. ${}^\bullet reset^1_{x > a} = \{F_{x > a}, Rtodo_i^x\}$ and $reset^1_{x > a}{}^\bullet = \{F_{x > a}, Rtodo_{i+1}^x\}$. ${}^\bullet reset^2_{x > a} = \{T_{x > a}, Rtodo_i^x\}$ and $reset^2_{x > a}{}^\bullet = \{F_{x > a}, Rtodo_{i+1}^x\}$.

- With each condition $x \leq a$ occurring in a transition of the automaton, we associate a widget (Figure 17 (a)) composed by two places $T_{x \leq a}, F_{x \leq a}$ and three transitions $change_{x \leq a}, reset^1_{x \leq a}, reset^2_{x \leq a}$. The place $T_{x \leq a}$ is initially marked while $F_{x \leq a}$ is unmarked. The interval associated to $change_{x \leq a}$ is $[a + 1/2, a + 1/2]$; ${}^\bullet change_{x \leq a} = \{T_{x \leq a}\}$ and $change_{x \leq a}{}^\bullet = \{F_{x \leq a}\}$. The interval associated to $reset^1_{x \leq a}$ and $reset^2_{x \leq a}$ is $[0, 0]$. Let i be the number of the condition $x \leq a$. ${}^\bullet reset^1_{x \leq a} = \{T_{x \leq a}, Rtodo_i^x\}$ and $reset^1_{x \leq a}{}^\bullet = \{T_{x \leq a}, Rtodo_{i+1}^x\}$. ${}^\bullet reset^2_{x \leq a} = \{F_{x \leq a}, Rtodo_i^x\}$ and $reset^2_{x \leq a}{}^\bullet = \{T_{x \leq a}, Rtodo_{i+1}^x\}$.

- With each condition $x < a$ ($a \neq 0$) occurring in a transition of the automaton, we associate a widget (Figure 17 (b)) composed by two places $T_{x < a}$, $F_{x < a}$ and three transitions $change_{x < a}, reset^1_{x < a}, reset^2_{x < a}$. The place $T_{x < a}$ is initially marked while $F_{x < a}$ is unmarked. The interval associated to $change_{x < a}$ is $[a - 1/2, a - 1/2]$; ${}^\bullet change_{x < a} = \{T_{x < a}\}$ and $change_{x < a}{}^\bullet = \{F_{x < a}\}$. The interval associated to $reset^1_{x < a}$ and $reset^2_{x < a}$ is $[0, 0]$. Let i be the number of the condition $x < a$.

${}^\bullet reset^1_{x<a} = \{T_{x<a}, Rtodo^x_i\}$ and $reset^1_{x<a}{}^\bullet = \{T_{x<a}, Rtodo^x_{i+1}\}$. ${}^\bullet reset^2_{x<a} = \{F_{x<a}, Rtodo^x_i\}$ and $reset^2_{x<a}{}^\bullet = \{T_{x<a}, Rtodo^x_{i+1}\}$.

- With each condition $x \leq a$ ($a \neq 0$) in a invariant, we associate a widget (Figure 18) composed by two places $TReach_{x\leq a}, FReach_{x\leq a}$ and three transitions $reach_{x\leq a}, resetR^1_{x\leq a}, resetR^2_{x\leq a}$. The place $FReach_{x\leq a}$ is initially marked while $TReach_{x\leq a}$ is unmarked. The interval associated to $reach_{x\leq a}$ is $[a, a]$; ${}^\bullet change_{x\leq a} = \{FReach_{x\leq a}\}$ and $change_{x\leq a}{}^\bullet = \{TReach_{x\leq a}\}$. The intervals associated to $resetR^1_{x\leq a}$ and $resetR^2_{x\leq a}$ is $[0, 0]$. Let i be the number of the condition $x \leq a$. ${}^\bullet resetR^1_{x\leq a} = \{FReach_{x\leq a}, Rtodo^x_i\}$ and $resetR^1_{x\leq a}{}^\bullet = \{FReach_{x\leq a}, Rtodo^x_{i+1}\}$. ${}^\bullet resetR^2_{x\leq a} = \{TReach_{x\leq a}, Rtodo^x_i\}$ and $resetR^2_{x\leq a}{}^\bullet = \{FReach_{x\leq a}, Rtodo^x_{i+1}\}$.

- With each edge $(l, \gamma = \{c_1, \ldots, c_{m(e)}\}, e, R = \{x_1, \ldots, x_{n(e)}\}, l')$, we associate a widget (Figure 19) composed by places $\{W^i_e\}_{i\leq n(e)}$ and transitions $fire_e, \{next^i_e\}_{i\leq n(e)}$. The transition $fire_e$ has label e; its interval is $[0, \infty[$; ${}^\bullet fire_e = \{l, T_{c_1}, \ldots, T_{c_{m(e)}}\}$ and $fire_e{}^\bullet = \{W^1_e, Rtodo^{x_1}_1, T_{c_1}, \ldots, T_{c_{m(e)}}\}$. The interval associated to transitions $next^i_e$ is $[0, 0]$. $\forall i < n(e), {}^\bullet next^i_e = \{W^i_e, Rtodo^{x_i}_{n(x_i)+1}\}$ and $next^i_e{}^\bullet = \{W^{i+1}_e, Rtodo^{x_{i+1}}_1\}$. ${}^\bullet next^{n(e)}_e = \{W^{n(e)}_e, Rtodo^{x_{n(e)}}_{n(x_{n(e)})+1}\}$ and $next^{n(e)}_e{}^\bullet = \{\ell'\}$. When $R = \emptyset$, the widget reduces to the transition $fire_e$ with ${}^\bullet fire_e = \{\ell, T_{c_1}, \ldots, T_{c_{m(e)}}\}$ and $fire_e{}^\bullet = \{\ell', T_{c_1}, \ldots, T_{c_{m(e)}}\}$.

- If a condition $x \leq 0$ occurs in the invariant of l, then one adds a transition $stop_\ell$ with interval $[0, 0]$, ${}^\bullet stop_\ell = stop_\ell{}^\bullet = \{\ell\}$. If a condition $x \leq a$ ($a \neq 0$) occurs in the invariant of l, then one adds a transition $stop^{x\leq a}_\ell$ with interval $[0, 0]$, ${}^\bullet stop^{x\leq a}_\ell = stop^{x\leq a}_\ell{}^\bullet = \{\ell, TReach_{x\leq a}\}$.

We decompose the reachable configurations (and markings) into *intermediate* ones (some W^i_e is marked) and *permanent* ones (some ℓ is marked). An easy induction shows that in permanent configurations (M, ν) the enabled timed transitions relative to a clock are "synchronized": $\nu(change_c) = \nu(change_{c'}) = \nu(reach_{c''})$ as soon as c, c', c'' relates to the same clock x. We define $\nu(x)$ as this common value if at least one such transition is enabled and otherwise $\nu(x) = K(x)$ where $K(x)$ is the maximal value relative to clock x occuring in the net $\mathcal{N}$. Furthermore from any intermediate configuration (M, ν), the behaviour of the net is quasi-deterministic until it reaches a permanent configuration: there are only firing sequences (i.e. no time step) and some of them lead to permanent configurations. Furthermore these permanent configurations (say (M_{next}, ν_{next})) have the same marked place ℓ and the same values $\nu_{next}(x)$.

It is also obvious that once some $fire_e$ is fired, the construction ensures the existence of a "resetting" sequence which reinitializes the widgets associated to the clocks to be reset.

Bisimulation relation. We now define the relation $\mathcal{R}$ between reachable configurations of the automaton $\mathcal{A}$ and the net $\mathcal{N}$. Let us define $(\ell, v)\mathcal{R}(M, \nu)$ iff:

- either M is a permanent marking and $M(\ell)$ is marked and if $\nu(x) < K(x)$ then $v(x) = \nu(x)$ else $v(x) \geq K(x)$.
- or M is an intermediate marking leading to some permanent (M_{next}, ν_{next}) and $(\ell, v)\mathcal{R}(M_{next}, \nu_{next})$. This definition is sound due to the common features of the different (M_{next}, ν_{next}).

It remains to prove that $\mathcal{R}$ is a bisimulation, which is done in the next lemma. ∎

Lemma 9 *The relation $\mathcal{R}$ defined above is a weak timed bisimulation.*

Proof. We first consider moves from $\mathcal{A}$.
Case 1: $(\ell, v) \xrightarrow{e} (\ell', v')$ First, let us prove that $(M, \nu) \xrightarrow{\sigma}$ with σ labelled by e. At first, σ begins by σ' which consists to fire all the $change_c$ fireable leading to some (M', ν') (with $(\ell, v)\mathcal{R}(M', \nu')$). Now we prove that $(M', \nu') \xrightarrow{fire_e}$. By definition of $\mathcal{R}$, $M(\ell)$ is marked. Let c be a condition occuring in the guard of e.
If $c = [x \geq a]$ then $v(x) \geq a$ which implies

$\nu(x) \geq a$ and that $T_{x \geq a}$ is marked (eventually with the help of σ').
If $c = [x > a]$ then let r be the region to which (ℓ, v) belongs. $min_r(x) = \lfloor v(x) \rfloor$. Using condition (b), $(l, min_r) \xrightarrow{e}$. Thus $v(x) \geq min_r(x) \geq a + 1$ which implies $\nu(x) \geq a + 1$ and that $T_{x>a}$ is marked (eventually with the help of σ').
If $c = [x \leq a]$ then $v(x) \leq a$ which implies $\nu(x) \leq a$ and that $T_{x \leq a}$ is marked (remember that $change_{x \leq a}$ fires when $\nu(x) = a + 1/2$).
If $c = [x < a]$ then let r be the region to which (ℓ, v) belongs. Then there exists $(\ell, v_1) \in \overline{r}$ with $v_1(x) = \lceil v(x) \rceil$. Using condition (b) and then (c), $(l, v_1) \xrightarrow{e}$. Thus $v(x) \leq v_1(x) \leq a - 1$ which implies $\nu(x) \leq a - 1$ and that $T_{x<a}$ is marked (remember that $change_{x<a}$ fires when $\nu(x) = a - 1/2$).
Thus $fire_e$ is fireable from (M', ν'). We complete σ by the "resetting" sequence leading to a configuration bisimilar to (ℓ', v')

If M is an intermediate marking, one fires a sequence leading to some (M_{next}, ν_{next}) and performs the previous simulation.

Case 2: $(\ell, v) \xrightarrow{d} (\ell, v + d)$

Suppose that $x \leq a$ belongs to the invariant of ℓ. This means that $v(x) + d \leq a$. Thus from (M, ν), we let a time d elapse interleaved with possible firings of $change$ transitions. The $stop$ transitions associated to ℓ will be possibly firable but only at the end of this step sequence.

If M is an intermediate marking, one fires a sequence leading to some (M_{next}, ν_{next}) and performs the previous simulation.

Conversely, we consider moves from $\mathcal{N}$.
Case 3: $(M, \nu) \xrightarrow{t} (M', \nu')$

If t is labelled by ϵ, then by construction $(\ell, v)\mathcal{R}(M', \nu')$.

Thus we only to need to examine the case of $fire_e$ (M is then a permanent marking). Let r be the region to which (ℓ, v) belongs. We will show that $(\ell, min_r) \xrightarrow{e}$. Then by condition (c), we will obtain that $(\ell, v) \xrightarrow{e}$.

Let c be a condition occuring in the guard of e.
If $c = [x \geq a]$ then $T_{x \geq a}$ is marked which implies that $\nu(x) \geq a$ and then $v(x) \geq a$, thus $min_r(x) = \lfloor v(x) \rfloor \geq a$.
If $c = [x > a]$ then then $T_{x>a}$ is marked which implies that $\nu(x) \geq a + 1$ and then $v(x) \geq a + 1$ thus $min_r(x) = \lfloor v(x) \rfloor \geq a + 1 > a$
If $c = [x \leq a]$ then $T_{x \leq a}$ is marked which implies that $\nu(x) \leq a + 1/2$ and then $v(x) \leq a + 1/2$ thus $min_r(x) = \lfloor v(x) \rfloor \leq a$
If $c = [x < a]$ then $T_{x<a}$ is marked which implies that $\nu(x) \leq a - 1/2$ and then $v(x) \leq a - 1/2$ thus $min_r(x) = \lfloor v(x) \rfloor \leq a - 1 < a$

So $(\ell, v) \xrightarrow{e} (\ell', v')$ for some (ℓ', v'). By construction of $\mathcal{N}$ and definition of $\mathcal{R}$, $(\ell', v')\mathcal{R}(M', \nu')$.

Case 4: $(M, \nu) \xrightarrow{d} (M, \nu + d)$

An intermediate marking cannot let elapse time. Thus M is a permanent marking. Let $x \leq a$ belonging to the invariant of

l. $a \neq 0$ otherwise from (M, ν), $stop_l$ must be fired and time may not elapse. Similarly since $stop_l^{x \leq a}$ is only possibly fireable from $(M, \nu + d)$, it follows that $\nu(x) + d \leq a$, thus $v(x) + d \leq a$.

Consequently $(\ell, v) \xrightarrow{d} (\ell, v + d)$ and obviously $(\ell, v + d)\mathcal{R}(M, \nu + d)$. ∎ We finally illustrate this construction on the timed automaton $\mathcal{B}_0$ from Figure 6.3 above and its translation given below (with some simplifications related to this particular TA). For readability, immediate transitions (where interval $[0, 0]$ is represented in black and ε labels are not shown).

First, note that the subnet associated to the constraint $y \leq 0$ switches the condition to false (firing of $toF_{y \leq 0}$) when the implicit value of y maintained in the net reaches $1/2$. Seemingly, this translation appears to be less constrained than the original condition. We explain how we prove that this translation is nevertheless sound. Let r be the region corresponding to the current configuration (ℓ, v) of the automaton simulated by the net, if the net is able to simulate a discrete step of the automaton, we prove that in the configuration

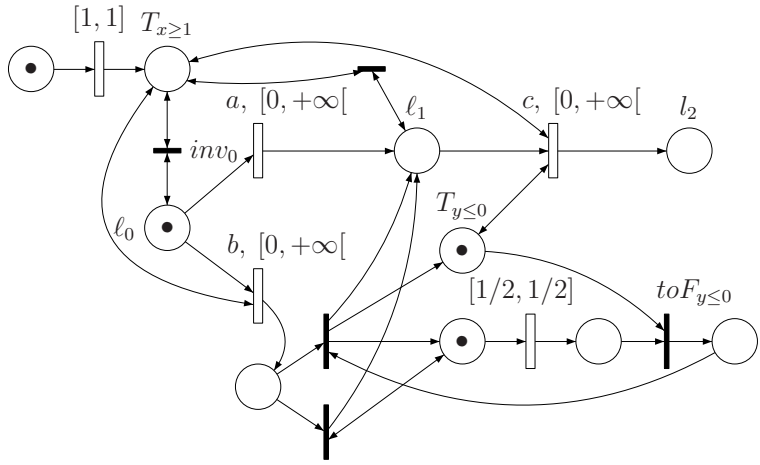
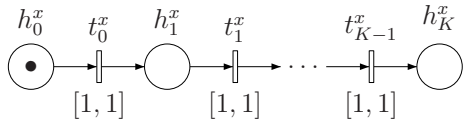


Figure 20: The TPN bisimilar to b_0

(ℓ, min_r) of the automaton this step is also possible. Thus by condition *(c)*, the step is also possible from (ℓ, v). On the other hand, if a discrete step is possible for (ℓ, v) in the automaton, we show that this step is also simulatable in the net using both conditions *(b)* and *(c)* and the following fact: $\forall x \in X, \exists(\ell_r, v'), (\ell_r, v'') \in \overline{r}$ such that $v'(x) = \lfloor v(x) \rfloor$ and $v''(x) = \lceil v(x) \rceil$. We also need to handle the invariants. First it is straightforward to observe that due to condition *(a)*, an atomic constraint $x < c$ occuring in an invariant may be safely deleted since its effectiveness leads to the existence of a region r whose time-successor (which intersects $\overline{r}$) would not be reachable. The subnet associated to the atomic constraint $x \leq 1$ occuring in the invariant of ℓ_0 leads to transition inv_0 (not modifying the marking) which is fireable as soon as the simulated value of x reaches 1 and the place ℓ_0 is marked. Thus time cannot progress except if the location is left.

6.6 Second construction

Proof.[for the second construction of sufficiency] When the conditions on the unlabeled timed automaton $\mathcal{A}$ are satisfied, we build a 1-TPN $\mathcal{N}$ with atomic semantics which is weakly timed bisimilar to $\mathcal{A}$. We suppose that all invariant conditions of a location are added to the guard of each ingoing transition. Recall that $K = m + 1$, where m is the maximal constant for $\mathcal{A}$. The construction of the TPN is a partial replication of both the region automaton of $\mathcal{A}$ and the class automaton, as explained later. There is first a subnet for each clock x, in which only the integral parts of x appear in the places (but with a fractional part that can reach 1).



Then we add one place C for each class $C = (\ell, Z)$ of the class automaton, with the initial class marked. Now let $e = (\ell, g, a, R, \ell')$ be a transition of $\mathcal{A}$. For each pair (v, v') of clock valuations in $\mathbb{N}^X$, with $v, v' \leq \overrightarrow{K}$, we build a subnet which simulates the transition $(\ell, v) \xrightarrow{e} (\ell', v')$, where we have $v'(x) = 0$ if $x \in R$ and $v'(x) = v(x)$ otherwise. Let $C_1 = (\ell, Z_1), \ldots, C_k = (\ell, Z_k)$ be the subset of classes such that $\exists v'' \in Z_i \wedge \forall x \in X, v''(x) = v(x) \vee (v''(x) \geq K \wedge v(x) = K))$ for $1 \leq i \leq k$, and $C'_1, \ldots, C'_k$ the classes obtained by applying transition e to $C_1, \ldots, C_k$ respectively. We have a transition with label e for each C_i (with $k = 2$ in the figure below), all with interval $[0, +\infty[$. Note that all reset operations for clocks in R are executed successively with instantaneous transitions. Moreover, the upper part of the net ensures that the invariant conditions of location l are satisfied (this part has been omitted for ℓ').

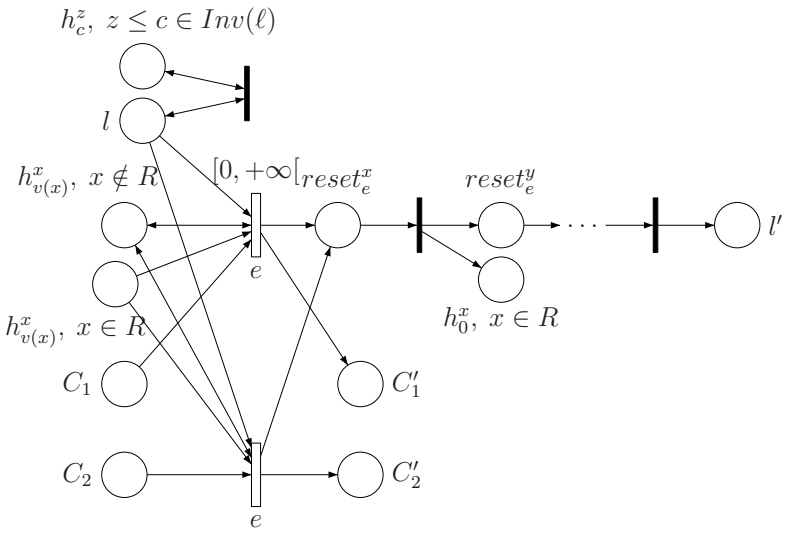


Figure 21: Simulation of a transition

Like in the previous proof, we say that a configuration (and the corresponding marking) (M, ν) of the TPN is permanent if $M(\ell) = 1$ for some l. Otherwise, it is an intermediate configuration (and marking), where $M(reset_e^x) = 1$ for some (exactly one of each)

x and e, meaning that some reset operations are in progress. Here again, a permanent configuration is reached instantaneously from such an intermediate configuration, with only firing sequences completing the reset operations for transition e (interleaved with possibly transitions firings of some t_c^x).

Furthermore, for a configuration (M, ν), there is exactly one non empty place h_c^x for each clock x. Writing c_x for the constant such that $M(h_{c_x}^x) = 1$, we have either $c_x = K$ or $0 \leq \nu(t_{c_x}^x) \leq 1$, where $\nu(t_c^x)$ is the time elapsed since arrival of the token in the place $h_{c_x}^x$. This means that the value of clock x is either $v(x) \geq K$ or $v(x) = c_x + \nu(t_{c_x}^x)$ with $\lfloor v(x) \rfloor$ equal to either c_x or $c_x + 1$. In the latter case, transition $t_{c_x}^x$ can be fired instantaneously, leading to the configuration (M', ν') with one token in place $h_{c_x+1}^x$ and either $c_x + 1 = K$ or $\nu'(t_{c_x+1}^x) = 0$. We can thus reach a configuration where $c = (c_x)_{x \in X}$ is maximal.

Bisimulation relation. The relation $\mathcal{R}$ is defined as the set of pairs $((M, \nu), (\ell, v))$ such that:

- either (M, ν) is a permanent configuration with $M(\ell) = 1$, the relation between v and ν is the one described above, and there exists exactly one class $C = (\ell, Z)$ such that $M(C) = 1$ and $v \in Z$;
- or (M, ν) is an intermediate configuration leading to some permanent configuration (M', ν') such that $((M', \nu'), (\ell, v)) \in \mathcal{R}$.

We end the proof with an auxiliary lemma and the fact that $\mathcal{R}$ is a weak time bisimulation. ∎

The following lemma which relates regions and classes, shows how the class automaton will be used to control the firing of a transition when the minimal point c is in not in the same region than v.

Lemma 10 *Let $\mathcal{A}$ be an automaton satisfying the conditions of theorem 5, let $C = (\ell, Z)$ be a class of the class automaton and $(\ell, v) \in C$. Let $(\ell, v) \in r$ where r is a region w.r.t. to the choice $K = \infty$ (which means that there is a infinite number of regions). Then $\forall (\ell, v') \in \overline{r}, (\ell, v') \in C$. In particular, $(\ell, \lfloor v \rfloor) \in C$.*

Proof. The proof is by induction on the reachability relation between regions. The case of a discrete step follows from conditions *(b)* and *(c)* of theorem 5. The case of a time step follows from the choice of $K = \infty$ which implies that given a region r, every item of $succ(r)$ is reached by a time step from an item of $\overline{r}$.

Lemma 11 *The relation $\mathcal{R}$ defined above is a weak time bisimulation.*

Proof. Assume that $(M,\nu)\mathcal{R}(\ell,v)$ and consider a move in $\mathcal{A}$.
Case 1: $(\ell,v) \xrightarrow{d} (\ell,v+d)$ (with $d \neq 0$). In this case, we must consider different subcases, according to the regions that can be reached by elapsing time. We consider only moves in which at most one different region is reached, the general case would be a combination of those elementary moves. First note that since $v' = v + d$ can be reached, no invariant condition needs to be activated in $\mathcal{N}$. Moreover, if (M,ν) is an intermediate configuration, we first apply the sequence described above and reach the equivalent configuration (M_1,ν_1). Also in this case, since classes are unchanged by elapsing time, if we prove that a delay move is possible from (M_1,ν_1), we immediately obtain that the class is the same in the resulting configuration. Thus, the resulting configuration will be equivalent to $(\ell,v+d)$.

- If v belongs to a time-open region, the case where v' belongs to the same time-open region is easy, it simply corresponds to a delay transition from (M_1,ν_1) in $\mathcal{N}$, each clock being in some h_c^x and staying inside (no token move), with (M_1,ν_1+d) equivalent to $(\ell,v+d)$.
  If v' has reached an integer value, we consider a clock x with greatest integral part, so that $v'(x) = \lfloor v(x)\rfloor + 1 = v(x) + d$ with $v(y) + d \leq \lfloor v(y)\rfloor + 1$ for all other clocks. In this case also, we obtain a delay move in $\mathcal{N}$ from (M_1,ν_1).

- If there are some clocks x for which $v(x)$ has an integer value, then elapsing time leads to the successor region, which is time-open. From (M_1,ν_1), it is possible to reach with instantaneous transitions a configuration (M_2,ν_2) where for all clocks with integer values, $M_2(h_c^x) = 1$ with c maximal, and (M_2,ν_2) still equivalent to (ℓ,v). Now from (M_2,ν_2), a delay move can be applied so that $(M,\nu) \xrightarrow{*} (M_1,\nu_1) \xrightarrow{*} (M_2,\nu_2) \xrightarrow{d} (M_2,\nu_2+d)$, with $(M_2,\nu_2+d)\mathcal{R}(\ell,v+d)$.

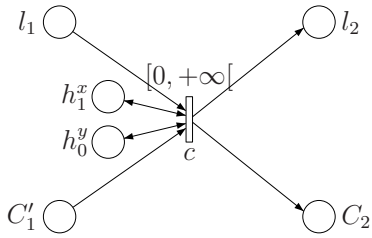
Case 2: If $(\ell,v) \xrightarrow{e} (\ell',v')$ for some $e = (\ell,g,a,R,l')$ then condition (b) implies that a transition $(\ell,\lfloor v\rfloor) \xrightarrow{e} (\ell',\lfloor v'\rfloor)$ is also possible in $\mathcal{A}$. Here again we may have to apply from (M,ν) a sequence of instantaneous transitions, leading to (M_1,ν_1) where place l is marked, and from there we can reach an equivalent configuration (M_2,ν_2) with $c = (c_x)_{x\in X}$ maximal. Let $C = (\ell,Z)$ be the class for which $M(C) = 1$, with $v \in Z$. From lemma 10, $(\ell,\lfloor v\rfloor)$ also belongs to C, and $\forall x \in X, \lfloor v\rfloor(x) = c_x \vee (\lfloor v\rfloor(x) \geq K \wedge c_x = K)$ so that the transition e (corresponding to this vector and this class) can be fired in $\mathcal{N}$, immediately followed by the corresponding reset sequence, leading to (M',ν'). Since exactly one class C' is marked after e, we have $(M',\nu')\mathcal{R}(\ell',v')$ by the definition of $\mathcal{R}$.

For the converse, we consider a move in $\mathcal{N}$.
Case 3: $(M,\nu) \xrightarrow{d} (M,\nu+d)$ (with $d \neq 0$). Then, neither reset transitions nor transitions of the form t_c^x can be fired in $\mathcal{N}$. Thus, the places h_c^x which contain a token are such that $\nu(t_c^x) < 1$ and $\nu(t_c^x)+d \leq 1$. For the state (ℓ,v), we have $M(\ell) = 1$ and $v(x) = c+\nu(t_c^x)$.

The move $(\ell, v) \xrightarrow{d} (\ell, v+d)$ is possible in $\mathcal{A}$ since $(\ell, v+d)$ belongs either to the region of (ℓ, v) or to its time successor which is reachable by condition *(a)*. Therefore $(\ell, v) \xrightarrow{d} (\ell, v+d)$ in $\mathcal{A}$ with $(M, \nu+d)\mathcal{R}(\ell, v+d)$.

Case 4: $(M, \nu) \xrightarrow{t} (M', \nu')$. For any transition of $\mathcal{N}$ which is not associated with some transition $e = (\ell, g, a, R, l')$ in $\mathcal{A}$, no time can elapse so there is no need for a move in $\mathcal{A}$ because (M', ν') is still equivalent to (ℓ, v). Suppose now that t is associated to an edge e, we have $M(\ell) = 1$, $M(C) = 1$ for some class $C = (\ell, Z)$ with $v \in Z$. Since t is fireable, considering the valuation $c = (c_x)_{x \in X}$ the construction implies that $\exists v'' \in Z$ s.t. $\forall x \in X, v''(x) = c_x \vee (v''(x) \geq K \wedge c_x = K)$, which implies that the segment $[v'', v] \subseteq Z$, from the convexity of Z, with $0 \leq v(x) - v''(x) = v(x) - c_x \leq 1$ for each x s.t. $c_x < K$. Thus, $[(\ell, v''), (\ell, v)]$ is contained in the topological closure $\overline{r}$ of some reachable region such that $min_r = c$ and $l = l_r$. Since $(\ell, c) \xrightarrow{e} (\ell', c')$ is possible in $\mathcal{A}$, and (ℓ, v'') is strongly time bisimilar to (ℓ, c), one has $(\ell, v'') \xrightarrow{e} (\ell', v''')$. Now condition *(c)* implies that a move $(\ell, v) \xrightarrow{e} (\ell', v')$ is also possible in $\mathcal{A}$. From the definition, $(M', \nu')\mathcal{R}(\ell', v')$. ∎

For instance, for the automaton $\mathcal{B}_0$ from Figure 6.3, we have four classes: $C_0 = \{l_0, 0 \leq x = y \leq 1\}$, $C_1 = \{l_1, 0 \leq x = y \leq 1\}$, $C_1' = \{l_1, x = 1 \wedge y = 0\}$ and $C_2 = \{l_2, 0 \leq y = x - 1\}$. We show below the subnet corresponding to the transition c at point $(l_1, (1,0))$ and class C_1'.



Consider the following run in $\mathcal{B}_0$: $(l_0, (0,0)) \xrightarrow{a} (l_1, (0,0)) \xrightarrow{1} (l_1, (1,1))$. The simulation of this run by $\mathcal{N}$ may lead to the following configuration: l_1, h_0^x, h_0^y and C_1 are marked and t_0^x and t_0^y have been enabled for 1 t.u. Suppose that the sequence $t_0^x t_0'^x$ is fired, marking the place t_1^x, then without the input place C_1' the transition labelled c could be erroneously fired. Since C_1' is unmarked this firing is disabled.

6.7 Complexity results

This characterization leads to the the following complexity results.

Proposition 5 (Complexity results) *Given a (label-free) timed automaton $\mathcal{A}$, deciding whether there is a TPN weakly timed bisimilar to $\mathcal{A}$ is $PSPACE$-complete. The reachability problem for the class $\mathcal{TA}^-$ is $PSPACE$-complete.*

Proof. The reachability problem for regions is in $PSPACE$. In order to check whether the condition (a) is false we *non deterministically* pick a region r and a region r' which intersects $\overline{r}$ and check whether r is reachable and r' is not reachable. In order to check whether the condition (b) is false we *non deterministically* pick a region r and a edge e and check whether r is reachable and e is firable from r and not fireable from (l_r, min_r). In order to check whether the condition (c) is false we *non deterministically* pick a region r, a region r' which intersects $\overline{r}$ and a edge e and check whether r is reachable and e is not firable from r or r' and fireable from (l_r, min_r). By Savitch construction, we obtain a deterministic algorithm in $PSPACE$.

In order to show the $PSPACE$-hardness, we use the construction given in [2] (in appendix D) which reduces the acceptation problem for linear bounded Turing machine (LBTM) to the reachability problem for TA with restricted guards. The computed TA (called $A_{\mathcal{M},w_0}$) satisfies the conditions (a) and (b) but does not satisfy the condition (c). However it can be safely transformed in order to satisfy this condition by adding the invariant $t \leq 1$ to any state (q, i) and the invariant $t \leq 0$ to any state (i, θ, j). This intermediate automaton is now bisimilar to a TPN.

Then we transform the edges entering the end state by resetting t and at last we add an edge $(end, t = 0, e, \emptyset, end)$.

If the LBTM $\mathcal{M}$ does not accept the word w_0, then the state end is not reachable and $A_{\mathcal{M},w_0}$ satisfies the conditions (a),(b),(c).

If the LBTM $\mathcal{M}$ accepts the word w_0, then the state end is reachable and $A_{\mathcal{M},w_0}$ does not satisfy the condition (c) (the additional edge is fireable when entering end but not after letting the time elapse). The fact that the reachability problem for the class $\mathcal{TA}^-$ is $PSPACE$-complete was proved implicitely within the proof above.

At last, we complete these results by adapting them to other models of TA. The previous characterization holds for TA with diagonal constraints and when satisfied a bisimilar 1-bounded 1-TPN whose size is exponential w.r.t. the TA may be built. A simpler characterization holds for TA without strict (and diagonal) constraints. Nevertheless, for these two models, the complexity of the membership and reachability problems is still $PSPACE$-complete.

Proposition 6 (TAs with diagonal constraints) *Let $\mathcal{A}$ be an unlabelled timed automaton with diagonal constraints, let $R(\mathcal{A})$ its region w.r.t. the 1-grid, then $\mathcal{A}$ is weakly timed bisimilar to a time Petri net iff:*
$\forall r \in R(\mathcal{A})$, $\forall e$ an edge of $\mathcal{A}$,

(a) Every region r' s.t. $r' \cap \overline{r} \neq \emptyset$ is reachable

(b) $\forall (\ell_r, v) \in r$, $(\ell_r, v) \xrightarrow{e} \Rightarrow (\ell_r, min_r) \xrightarrow{e}$

(c) $\forall(\ell_r, v) \in \overline{r}$, $(\ell_r, min_r) \xrightarrow{e}\Rightarrow (\ell_r, v) \xrightarrow{e}$

Furthermore, if these conditions are satisfied then one can build a 1*-bounded* 1*-TPN bisimilar to* $\mathcal{A}$ *whose size is exponential w.r.t. the size of* $\mathcal{A}$*.*

At last, deciding whether there is a TPN weakly time bisimilar to $\mathcal{A}$ *is* $PSPACE$*-complete.*

Proof. The proof of necessity and the second construction of the TPN bisimilar to $\mathcal{A}$ need to be slightly adaptated to take into account the nature of the regions of an automaton with diagonal constraints since they are based on properties of the region automaton whereas the construction of the class automaton is still valid for automata with diagonal constraints.

The $PSPACE$-hardness is obviously true while the membership to $PSPACE$ deduced from implicit explorations of the region automaton is still valid. ∎

Despite the fact that excluding strict constraints simplifies the characterization, the complexity of the membership problem remains the same.

Proposition 7 (TA without strict constraints) *Let* $\mathcal{A}$ *be an unlabelled timed automaton without strict constraints, let* $R(\mathcal{A})$ *its region w.r.t. the* 1*-grid, then* $\mathcal{A}$ *is weakly timed bisimilar to a time Petri net iff:*
$\forall r \in R(\mathcal{A}), \forall e$ *an edge of* $\mathcal{A}$*,* $\forall(\ell_r, v) \in r$*,* $(\ell_r, min_r) \xrightarrow{e}\Rightarrow (\ell_r, v) \xrightarrow{e}$

Furthermore, deciding whether there is a TPN weakly time bisimilar to $\mathcal{A}$ *is* $PSPACE$*-complete.*

Proof. It is straightforward to show that conditions (a) and (b) are satisfied by an automaton without strict constraints. Similarly the condition (c) is easily deduced from the current condition when the automaton does not include strict constraints.

The $PSPACE$ membership is obviously true. We remark that although the net of [2] (in appendix D) contains contraints $x_i > 1$, they can be safely changed to $x_i \geq 2$. Thus the $PSPACE$ hardness follows. ∎

7 Conclusion

In this paper, we have investigated different questions relative to the expressiveness of TPNs. At first, we have shown that TAs and bounded TPNs (enlarged with strict constraints) are equivalent w.r.t. the timed language equivalence. We have also provided a more general and efficient construction of a TPN equivalent to a TA than the previous

ones. Then we have focused on the weak time bisimilarity equivalence and we have developed our main contribution: a characterization of TAs time bisimulateable by a TPN. From this characterization, we have proved that deciding whether a TA admits a time bisimilar TPN is a $PSPACE$-complete problem. Furthermore the reachability problem is still $PSPACE$-complete for this subclass of TAs. Finally we have proved that for bounded TPNs the different semantics lead to equivalent models w.r.t. the time bisimilarity but that this is no more true with strict constraints.

We are now looking for similar (multiple) characterizations for TPNs enlarged with strict constraints since in this context the choice of the semantics is relevant. We will also try to apply the same techniques to compare the different models of Petri nets with time.

References

[1] P. A. Abdulla and A. Nylén. Timed Petri nets and BQOs. In *ICATPN'01*, volume 2075 of *LNCS*, pages 53–72. Springer-Verlag, june 2001.

[2] L. Aceto and F. Laroussinie. Is Your Model Checker on Time? On the Complexity of Model Checking for Timed Modal Logics. *Journal of Logic and Algebraic Programming*, volume 52-53, pages 7-51. Elsevier Science Publishers, august 2002.

[3] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science B*, 126:183–235, 1994.

[4] R. Alur and L. Fix and T. A. Henzinger. Event-Clock Automata: A Determinizable Class of Timed Automata. *Theoretical Computer Science*, 211:253–273, 1999.

[5] T. Aura and J. Lilius. A causal semantics for time Petri nets. *Theoretical Computer Science*, 243(1–2):409–447, 2000.

[6] B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Transactions on Software Engineering*, 17(3):259–273, March 1991.

[7] M.Boyer and M.Diaz. Non equivalence between time Petri nets and time stream Petri nets. In *Proceedings of 8th International Workshop on Petri Nets and Performance Modeling (PNPM'99)*, Zaragoza, Spain, pages 198–207.

[8] Franck Cassez and Olivier H. Roux. Structural Translation of Time Petri Nets into Timed Automata. In Michael Huth, editor, *Workshop on Automated Verification of Critical Systems (AVoCS'04)*, Electronic Notes in Computer Science. Elsevier, August 2004.

[9] D. de Frutos Escrig, V. Valero Ruiz, and O. Marroquín Alonso. Decidability of properties of timed-arc Petri nets. In *ICATPN'00*, Aarhus, Denmark, volume 1825 of *LNCS*, pages 187–206, june 2000.

[10] M. Diaz and P. Senac. Time stream Petri nets: a model for timed multimedia information. In *ATPN'94*, volume 815 of *LNCS*, pages 219–238, 1994.

[11] D. L. Dill. Timing assumptions and verification of finite-state concurrent systems. In Proc. Workshop on Automatic Verification Methods for Finite State Systems, Grenoble, volume 407 of *LNCS*, 1989.

[12] S. Haar, F. Simonot-Lion, L. Kaiser, and J. Toussaint. Equivalence of Timed State Machines and safe Time Petri Nets. In *Proceedings of WODES 2002*, Zaragoza, Spain, pages 119–126.

[13] W. Khansa, J.P. Denat, and S. Collart-Dutilleul. P-Time Petri Nets for manufacturing systems. In *WODES'96*, Scotland, pages 94–102, 1996.

[14] D. Lime and O. H. Roux. State class timed automaton of a time Petri net. In *PNPM'03*. IEEE Computer Society, September 2003.

[15] P. M. Merlin. *A study of the recoverability of computing systems*. PhD thesis, University of California, Irvine, CA, 1974.

[16] M. Pezzé and M. Young. Time Petri Nets: A Primer Introduction. Tutorial presented at the Multi-Workshop on Formal Methods in Performance Evaluation and Applications, Zaragoza, Spain, september 1999.

[17] C. Ramchandani. *Analysis of asynchronous concurrent systems by timed Petri nets*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1974.

[18] J. Sifakis. Performance Evaluation of Systems using Nets. In *Net Theory and Applications, Advanced Course on General Net Theory of Processes and Systems, Hamburg*, volume 84 of *LNCS*, pages 307–319, 1980.