



HAL
open science

When are Timed Automata Weakly Timed Bisimilar to Time Petri Nets?

Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, Olivier Henri Roux

► To cite this version:

Béatrice Bérard, Franck Cassez, Serge Haddad, Didier Lime, Olivier Henri Roux. When are Timed Automata Weakly Timed Bisimilar to Time Petri Nets?. *Theoretical Computer Science*, 2008, 403 (2-3), pp.202-220. inria-00363024

HAL Id: inria-00363024

<https://inria.hal.science/inria-00363024v1>

Submitted on 20 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

When are Timed Automata weakly timed bisimilar to Time Petri Nets?

B. Bérard¹, F. Cassez², S. Haddad³, D. Lime², O.H. Roux²

¹ LAMSADE, Paris, France

E-mail: Beatrice.Berard@lamsade.dauphine.fr

² IRCCyN, Nantes, France

E-mail: {Franck.Cassez | Didier.Lime | Olivier-h.Roux}@irccyn.ec-nantes.fr

³ LSV, Cachan, France

E-mail: Serge.Haddad@lsv.ens-cachan.fr

Abstract. In this paper⁴, we compare Timed Automata (TA) and Time Petri Nets (TPN) with respect to weak timed bisimilarity. It is already known that the class of bounded TPNs is strictly included in the class of TA. It is thus natural to try and identify the subclass \mathcal{TA}^{wtb} of TA equivalent to some TPN for the weak timed bisimulation relation. We give a characterization of this subclass and we show that the membership problem and the reachability problem for \mathcal{TA}^{wtb} are *PSPACE*-complete. Furthermore we show that for a TA in \mathcal{TA}^{wtb} with integer constants, an equivalent TPN can be built with integer bounds but with a size exponential w.r.t. the original model. Surprisingly, using rational bounds yields a TPN whose size is linear.

Keywords: Time Petri Nets, Timed Automata, Weak Timed Bisimilarity.

1 Introduction

Time in Petri nets. Adding explicit time to classical models of dynamic systems was first done in the seventies for Petri nets [24,27], with the aim to verify quantitative properties of systems. Since then, various timed models based on Petri nets were proposed. Among them, the two most prominent ones are Time Petri Nets (TPN) [24,10] and Timed Petri Nets (TdPN) [27,2]. In TPNs, a time interval is associated with each transition and a transition can fire if its enabling duration belongs to its interval. Since time elapsing must not disable transitions, TPNs naturally model urgency requirements. Efficient verification methods have been designed for bounded TPNs (*e.g.* [11]) implemented in several tools [19,11]. Roughly speaking, in TdPNs a time interval is associated with each arc, a token can be consumed along an input arc if its age belongs to the corresponding interval and the initial age of token produced along an output arc is non deterministically selected inside the corresponding interval. Contrary to

⁴ A preliminary version of this work has been published in [8].

TPNs, there is no urgency mechanism and tokens may become useless due to time elapsing. However the lazy behaviour of such nets has led to the design of verification methods for unbounded nets [1].

Time in finite automata. Timed Automata (TA), introduced in the seminal paper [4] have yielded a significant breakthrough in the theory of modelling and analysis of timed systems. The most commonly used variant of TA, called Safety TA has been defined in [21] and this is the one we study here. A TA is a finite automaton equipped with a set of clocks which evolve synchronously with time. Elementary constraints on clock values restrict the sojourn in a location and the firing of transitions. In addition, transitions may involve some clock reset. Extensions of this model have been subsequently proposed (*e.g.* diagonal or linear constraints, silent transitions, non deterministic updates, etc. [13]). In these models, verification is based on a finite partition of clock values and is supported by various tools [28,25].

Expressiveness of timed models. Due to the diversity of time mechanisms involved in these models, their relative expressiveness is a natural issue. More precisely, given two different formalisms, some questions must be considered:

- Are they equally expressive? Otherwise is one formalism more powerful than the other?
- Given a model of the first kind can we decide whether it is equivalent to a model of the second kind (*membership problem*)? In this case, can we build such an equivalent model?

Two standard equivalence criteria are the family of timed languages generated and the weak timed bisimilarity of the models. For instance, in the framework of TdPNs w.r.t. the language criterion, it has been shown that read arcs add expressive power and that TdPNs and TA are incomparable [14]. In the framework of TA w.r.t. the language criterion, it has been shown that silent transitions add expressive power [9] and (very recently) that the corresponding membership problem is undecidable [15].

Comparison of TA and TPNs. In [20], the authors compare Timed State Machines (TSM, a restricted version of TA) and TPNs, giving a translation from TSM to TPN that preserves timed languages. In [7], we have designed a more general translation between TA and TPNs with better complexity. In [6], we studied the effect of different semantics for TPNs on expressiveness w.r.t. weak timed bisimilarity. Here, we are interested in comparing the expressive power of TA and TPN for weak timed bisimilarity. Recall that there are unbounded TPNs for which no bisimilar TA exists. This is a direct consequence of the following observation: the untimed language of a TA is regular which is not necessarily the case for TPNs. It was proved in [17] that bounded TPNs form a strict subclass of the class of timed automata, in the sense that for each bounded TPN \mathcal{N} , there exists a TA which is weakly timed bisimilar to \mathcal{N} but the converse is false. A similar result can be found in [22], where it is obtained by a completely different approach. However given a TA, deciding whether there exists a bisimilar TPN remained an open question.

Our Contribution. In this work, we give a characterization of the maximal subclass \mathcal{TA}^{wtb} of timed automata which admit a weakly timed bisimilar TPN. This condition is not intuitive and relates to the topological properties of the so-called region automaton associated with a TA. To prove that the condition is necessary, we introduce the notion of *uniform bisimilarity*, which is stronger than weak timed bisimilarity. Conversely, when the condition holds for a TA, we provide two effective constructions of bisimilar TPNs: the first one with rational constants has a size linear w.r.t. the TA, while the other one, which uses only integer constants has an exponential size. From this characterization, we deduce that given a TA, the problem of deciding whether there is a TPN bisimilar to it, is *PSPACE*-complete. Thus, we obtain that the membership problem is *PSPACE*-complete. Finally we also prove that the reachability problem is *PSPACE*-complete.

Outline of the paper. In section 2, we recall the definitions for Timed Transition Systems which are used to describe the semantics of TPNs and TA, and for timed bisimilarity relations. We also describe Timed Automata (TA) and the associated notions of zones and regions. Section 3 presents Time Petri Nets (TPNs) and explains the characterization of TA bisimilar to TPNs. The following sections are devoted to the proof for this characterization: section 4 defines the notion of uniform bisimulation and gives the proof that the condition is necessary, while sections 5 and 6 show that the condition is sufficient by exhibiting two constructions which differ by their complexity. We give complexity results in section 7 and we conclude in section 8.

2 Timed Transition Systems and Timed Automata

Notations. Let Σ be a finite alphabet, Σ^* (resp. Σ^ω) the set of finite (resp. infinite) words of Σ and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. We also use $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ with ε (the empty word) not in Σ .

The sets \mathbb{N} , $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ are respectively the sets of natural, non-negative rational and non-negative real numbers. We denote by $\mathbf{0}$ the tuple $v \in \mathbb{N}^n$ such that $v(k) = 0$ for all $1 \leq k \leq n$. For a natural number K and a tuple $v \in \mathbb{R}_{\geq 0}^n$, we write $v \leq \vec{K}$ if $v(k) \leq K$ for all $1 \leq k \leq n$. For a non negative real number z , we denote by $\lfloor z \rfloor$ the integral part of z (the greatest natural number which is less than or equal to z) and, similarly, by $\lceil z \rceil$ the smallest natural number greater than or equal to z . We also denote by $fract(z)$ the fractional part of z . Let $g > 0$ in \mathbb{N} , we write $\mathbb{N}_g = \{\frac{i}{g} \mid i \in \mathbb{N}\}$. A tuple $v \in \mathbb{Q}^n$ belongs to the g -grid if $v(k) \in \mathbb{N}_g$ for all $1 \leq k \leq n$.

An interval I of $\mathbb{R}_{\geq 0}$ is a $\mathbb{Q}_{\geq 0}$ -interval iff its left endpoint belongs to $\mathbb{Q}_{\geq 0}$ and its right endpoint belongs to $\mathbb{Q}_{\geq 0} \cup \{\infty\}$. We set $I^\downarrow = \{x \mid x \leq y \text{ for some } y \in I\}$, the *downward closure* of I and $I^\uparrow = \{x \mid x \geq y \text{ for some } y \in I\}$, the *upward closure* of I . We denote by $\mathcal{I}(\mathbb{Q}_{\geq 0})$ the set of $\mathbb{Q}_{\geq 0}$ -intervals of $\mathbb{R}_{\geq 0}$.

2.1 Timed Transition Systems and Equivalence Relations

Timed transition systems describe systems which combine discrete and continuous evolutions. They are used to define and compare the semantics of time Petri nets and timed automata.

A *Timed Transition System (TTS)* is a transition system $S = (Q, q_0, \rightarrow)$, where Q is the set of configurations, $q_0 \in Q$ is the initial configuration and the relation \rightarrow consists of either delay moves $q \xrightarrow{d} q'$, with $d \in \mathbb{R}_{\geq 0}$, or discrete moves $q \xrightarrow{a} q'$, with $a \in \Sigma_\varepsilon$. Moreover, we require standard properties for the relation \rightarrow :

Time-Determinism: if $q \xrightarrow{d} q'$ and $q \xrightarrow{d} q''$ with $d \in \mathbb{R}_{\geq 0}$, then $q' = q''$

0-delay: $q \xrightarrow{0} q$

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 $q \xrightarrow{d'} q'' \xrightarrow{d''} q'$.

These properties have been formally introduced in the framework of Algebra of Timed Processes [23] but are also satisfied in TTS studied here. With these properties, a *run* of S can be defined as a finite or infinite sequence of moves $\rho = q_0 \xrightarrow{d_0} q'_0 \xrightarrow{a_0} q_1 \xrightarrow{d_1} q'_1 \xrightarrow{a_1} \dots q_n \xrightarrow{d_n} q'_n \dots$ where discrete actions alternate with durations. For a finite run, we also write $q \xrightarrow{d_0 a_0 \dots d_n \dots} q'$. The word $Untimed(\rho)$ in Σ^∞ is obtained by the concatenation $a_0 a_1 \dots$ of labels in Σ_ε (so empty labels disappear), and $Duration(\rho) = \sum_{i=0}^{|\rho|} d_i$, where $|\rho|$ is the length of ρ .

From a TTS, we define the relation $\Rightarrow \subseteq Q \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times Q$ for $a \in \Sigma$ and $d \in \mathbb{R}_{\geq 0}$ by:

- $q \xRightarrow{d} q'$ iff $\exists \rho = q \xrightarrow{w} q'$ with $Untimed(\rho) = \varepsilon$ and $Duration(\rho) = d$,
- $q \xRightarrow{a} q'$ iff $\exists \rho = q \xrightarrow{w} q'$ with $Untimed(\rho) = a$ and $Duration(\rho) = 0$.

Definition 1 (Weak Timed Bisimilarity). Let $S_1 = (Q_1, q_0^1, \rightarrow_1)$ and $S_2 = (Q_2, q_0^2, \rightarrow_2)$ be two TTS and let \approx be a binary relation over $Q_1 \times Q_2$. We write $q \approx q'$ for $(q, q') \in \approx$. The relation \approx is a weak timed bisimulation between S_1 and S_2 iff $q_0^1 \approx q_0^2$ and for all $a \in \Sigma \cup \mathbb{R}_{\geq 0}$:

- if $q_1 \xrightarrow{a}_1 q'_1$ and $q_1 \approx q_2$ then $\exists q_2 \xrightarrow{a}_2 q'_2$ such that $q'_1 \approx q'_2$;
 - conversely, if $q_2 \xrightarrow{a}_2 q'_2$ and $q_1 \approx q_2$ then $\exists q_1 \xrightarrow{a}_1 q'_1$ such that $q'_1 \approx q'_2$.
- Two TTS S_1 and S_2 are weakly timed bisimilar, written $S_1 \approx_{\mathcal{W}} S_2$, if there exists a weak timed bisimulation relation between them.

Strong timed bisimilarity would require similar properties for transitions labeled by $a \in \Sigma \cup \mathbb{R}_{\geq 0}$, but with \xrightarrow{a} instead of \xRightarrow{a} . Thus it forbids the possibility of simulating a move by a sequence. On the other hand, weak timed bisimilarity is more precise than language equivalence and it is well-known to be central among equivalence relations between timed systems. In the rest of the paper, we abbreviate weak timed bisimilarity by bisimilarity and we explicitly name other equivalences when needed.

2.2 Timed Automata

First defined in [4], the model of timed automata (TA) associates with a finite automaton a set of non negative real-valued variables called *clocks*. Let X be a finite set of *clocks*. A *constraint* over X is a conjunction of atomic formulas of the form $x \bowtie h$ for $x \in X$, $h \in \mathbb{N}$ and $\bowtie \in \{<, \leq, \geq, >\}$ and we write $\mathcal{C}(X)$ for the set of constraints.

Definition 2 (Timed Automaton). A Timed Automaton \mathcal{A} over alphabet Σ_ε is a tuple $(L, \ell_0, X, \Sigma_\varepsilon, E, Inv)$ where

- L is a finite set of locations with $\ell_0 \in L$ is the initial location,
- X is a finite set of clocks,
- $E \subseteq L \times \mathcal{C}(X) \times \Sigma_\varepsilon \times 2^X \times L$ is a finite set of edges and
- $Inv \in \mathcal{C}(X)^L$ assigns an invariant to each location.

An edge $e = \langle \ell, \gamma, a, R, \ell' \rangle \in E$, also written $\ell \xrightarrow{\gamma, a, R} \ell'$, represents a transition from location ℓ to location ℓ' with guard γ and reset set $R \subseteq X$. We restrict the invariants to conjunctions of terms of the form $x \bowtie h$ for $x \in X$, $h \in \mathbb{N}$ and $\bowtie \in \{<, \leq\}$.

We consider *injectively-labelled* automata *i.e.*, where two different edges have different labels (and no label is ε). Figure 1 shows two injectively-labelled timed automata that will be used to illustrate the constructions throughout the paper.

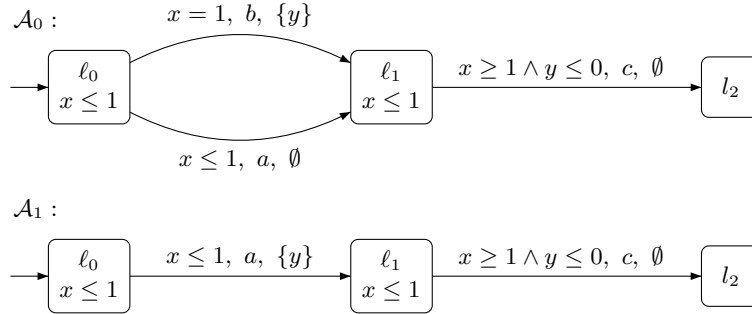


Fig. 1. Two timed automata

A *valuation* v is a mapping in $\mathbb{R}_{\geq 0}^X$. The effect of reset operations and time elapsing on valuations are described as follows. For $R \subseteq X$, the valuation $v[R \mapsto 0]$ maps each variable in R to the value 0 and agrees with v over $X \setminus R$. For valuation v and $d \in \mathbb{R}_{\geq 0}$, valuation $v+d$ is defined by: $\forall x \in X, (v+d)(x) = v(x) + d$. Constraints of $\mathcal{C}(X)$ are interpreted over valuations: we write $v \models \gamma$ when the constraint γ is satisfied by v . The set of valuations satisfying a constraint γ is denoted by $\llbracket \gamma \rrbracket$.

Definition 3 (Semantics of TA). *The semantics of a timed automaton $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, Inv)$ is a timed transition system $S_{\mathcal{A}} = (Q, q_0, \rightarrow)$ where $Q = L \times (\mathbb{R}_{\leq 0})^X$, $q_0 = (\ell_0, \mathbf{0})$ and \rightarrow is defined by:*

- either a delay move $(\ell, v) \xrightarrow{d} (\ell, v + d)$ iff $v + d \models Inv(\ell)$,
- or a discrete move $(\ell, v) \xrightarrow{e} (\ell', v')$ iff there exists some $e = (\ell, \gamma, a, R, \ell')$ in E such that $v \models \gamma$, $v' = v[R \mapsto 0]$ and $v' \models Inv(\ell')$.

In order to ensure in a syntactic way that a move associated with a discrete transition e always leads to a configuration (ℓ', v') such that $v' \models Inv(\ell')$, we may modify the initial timed automaton in the following way : any atomic constraint related to a clock x occurring in the invariant of ℓ' is added to the guard of each transition arriving in ℓ' which does not reset x . This transformation does not change the resulting transition system but makes some proofs simpler and in the sequel, we always assume that the timed automaton has this property.

Since our results are mainly based on the *region automaton* and some of its variants, we now recall the material needed.

2.3 Elementary zones of a timed automaton

Notations. A *zone* is a subset of $(\mathbb{R}_{\geq 0})^X$ defined by a conjunction of atomic clock constraints of the form $x \bowtie h$ or $x - y \bowtie h$, where x and y are clocks in X , $h \in \mathbb{N}$ and $\bowtie \in \{<, \leq, \geq, >\}$. In particular, if γ is a clock constraint in $\mathcal{C}(X)$, then the set $\llbracket \gamma \rrbracket$ is a zone. Constraints of the form $x - y \bowtie h$ are usually called *diagonal constraints*. The future of a zone Z is defined by $fut(Z) = \{v + d \mid v \in Z, d \in \mathbb{R}_{\geq 0}\}$. A zone Z satisfies a constraint $\gamma \in \mathcal{C}(X)$, written $Z \models \gamma$, if all valuations in Z satisfy γ . For $k \in \mathbb{N}$, a *k-zone* is a zone for which no atomic clock constraint in its definition involves a constant greater than k .

Elementary zones. Recall [4] that, if m is the maximal constant appearing in atomic formulas $x \bowtie h$ 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. In the original definition, an element of the partition is specified by a pair $(\{I_x\}_{x \in X}, \alpha)$ where I_x is an interval in the set $\{\{0\},]0, 1[, \{1\}, \dots, \{m\},]m, +\infty[$ and α is a relation on the subset of clocks x such that $I_x \neq]m, +\infty[$ defined by $x \propto y \equiv fract(x) \leq fract(y)$.

Note that the property related to the behaviour of the systems holds for any partition which refines the above partition. In this paper, we define a family of refining partitions $\mathcal{P}_{K,g}$ for $K \geq m + 1$ and $g \in \mathbb{N} \setminus \{0\}$. whose elements are called *elementary zones*. The constant g is called the *granularity* and $\mathcal{P}_{K,g}$ is called a *g-grid*. For such a partition the interval I_x belongs to the set $\{\{0\},]0, \frac{1}{g}[, \{\frac{1}{g}\},]\frac{1}{g}, \frac{2}{g}[, \dots, \{K - \frac{1}{g}\},]K - \frac{1}{g}, K[, [K, +\infty[$ (instead of keeping $\{K\}$ separately). As before, we also specify the ordering on the fractional parts w.r.t.

g for all clocks x (i.e. the value $\delta \in [0, \frac{1}{g}[$ such that $\exists i, x = \frac{i}{g} + \delta$) with valuation less than K . We include the case $K = +\infty$.

When K is finite, reachability can be decided by the usual method applied on these partitions; however we consider $[K, \infty[$ (rather than the usual $]K, \infty[$) as the last interval for topological reasons explained later. The partition $K = +\infty$ is used here as a tool for our expressiveness results.

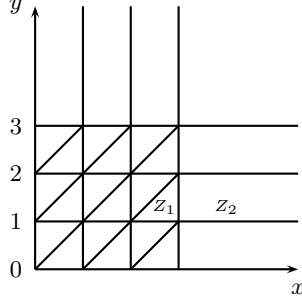


Fig. 2. Partition of $(\mathbb{R}^+)^2$ with granularity $g = 1$ and $K = 3$

Figures 2 and 3 represent different partitions for the set of two clocks $X = \{x, y\}$. The horizontal and vertical lines correspond to the possible intervals I_x and I_y whereas the diagonal lines specify the ordering of fractional parts.

Figure 2 is associated with $g = 1$ and $K = 3$. For this example, elementary zones Z_1 and Z_2 are described as follows: $Z_1 : (2 < x < 3) \wedge (1 < y < 2) \wedge (0 < \text{frac}(y) < \text{frac}(x))$ or equivalently, $(2 < x < 3) \wedge (1 < y < 2) \wedge (x - y > 1)$ and $Z_2 : (x \geq 3) \wedge (1 < y < 2)$. Figure 3 (left) is associated with $g = 2$ and $K = 2$ while figure 3 (right) is associated with $g = 2$ and $K = +\infty$. The dotted lines of this figure mean that the pattern is infinitely repeated.

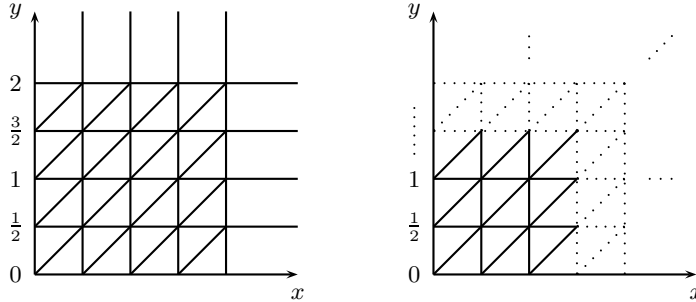


Fig. 3. Partitions of $(\mathbb{R}^+)^2$ with granularity $g = 2$, for $K = 2$ and $K = +\infty$

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'' different from Z and Z' such that $Z \leq Z'' \leq Z'$. When it exists, this elementary zone is called the immediate time successor of Z and denoted by $\text{succ}(Z)$. From the definition of elementary zones as equivalent classes consistent with reset operations, the reset can be extended to elementary zones, so that $Z[R \mapsto 0]$ is the elementary zone containing any $v[R \mapsto 0]$, for $v \in Z$.

Standard topological notions on $(\mathbb{R}_{\geq 0})^n$ apply to zones. Moreover, due to the particular form of the constraints, the topological closure of any elementary zone has a minimal element.

For a timed automaton \mathcal{A} , a constant K and a granularity g , we call *region* a pair (ℓ, Z) , where ℓ is a location of \mathcal{A} and Z an elementary zone of $(\mathbb{R}_{\geq 0})^X$. We now give a description for regions, which distinguishes *time-closed* and *time-open descriptions*. It is equivalent to the original one but more convenient for our proofs and it fits both cases, whether K is finite or infinite.

Definition 4 (Region description w.r.t. constant K and granularity g).
A time-closed description of a 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 ,
- $\text{Act}X_r = \{x \in X \mid \min_r(x) < K\}$ the subset of relevant clocks,
- the number size_r of different fractional parts for the values of relevant clocks in the $\mathbb{N}_g^{\text{Act}X_r}$ grid, with $1 \leq \text{size}_r \leq \text{Max}(|\text{Act}X_r|, 1)$ and the onto mapping $\text{ord}_r : X \mapsto \{1, \dots, \text{size}_r\}$ giving the ordering of the fractional parts.

By convention, $\forall x \in X \setminus \text{Act}X_r, \text{ord}_r(x) = 1$.

Then $r = \{(\ell_r, \min_r + \delta) \mid \delta \in \mathbb{R}_{\geq 0}^X \text{ and } \forall x, y \in \text{Act}X_r [\text{ord}_r(x) = 1 \Leftrightarrow \delta(x) = 0] \wedge \delta(x) < 1/g \wedge [\text{ord}_r(x) < \text{ord}_r(y) \Leftrightarrow \delta(x) < \delta(y)]\}$

A time-open description of a region r is defined with the same attributes (and conditions) as the time-closed one with:

$r = \{(\ell_r, \min_r + \delta + \mathbf{d}) \mid \exists d \in \mathbb{R}_{> 0} \wedge \forall x \in \text{Act}X_r, \mathbf{d}(x) = d \wedge \delta(x) + d < 1/g \wedge \forall x \notin \text{Act}X_r, \mathbf{d}(x) = 0\}$.

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 $\text{ord}_r(x) = \text{ord}_r(y)$.

Thus, a time-open region corresponds to an immediate time successor of a time-closed region and the decomposition of a valuation in a time-open region as $\min_r + \delta + \mathbf{d}$ is unique if $\min_r + \delta$ belongs to the *previous* time-closed region. This property is used in the sequel.

Also remark that $\min_r \notin r$ except if there is a single class of clocks relative to r (for instance if the corresponding zone is a singleton). Of course, when $K = +\infty$, the part about *relevant* clocks, for which the value is less than K ,

can be omitted (since $ActX_r = X$). The hypothesis $K = +\infty$ makes some proofs simpler, because the extremal case where a clock value is greater than K is avoided, and it can be lifted afterward. Furthermore when K is finite, some regions admit both time-open and time-closed descriptions (for instance a region associated with zone Z_2 in fig. 3), whereas when $K = +\infty$, a region admits a single description, so that time elapsing leads to an alternation of time-open regions (where time can elapse) and time-closed ones (where no time can elapse). Furthermore in this case, the representation of a valuation in a time-open region can be written $min_r + \delta + d$ with d a (scalar) duration.

2.4 Region automaton and class automaton of a TA.

For a timed automaton \mathcal{A} , a constant K and a granularity g , the region automaton $R(\mathcal{A})_{g,K}$ is a finite automaton, the states of which are regions. These regions are built inductively from the initial one $(\ell_0, \mathbf{0})$ by the following transitions over the set of labels $\{succ\} \cup \Sigma_\varepsilon$:

- $(\ell, Z) \xrightarrow{succ} (\ell, succ(Z))$ if $succ(Z) \models Inv(\ell)$ and
- $(\ell, Z) \xrightarrow{a} (\ell', Z')$ if there is a transition $(\ell, \gamma, a, R, \ell') \in E$ such that $Z \models \gamma$ and $Z' = Z[R \mapsto 0]$, with $Z' \models Inv(\ell')$.

A region $r = (\ell, Z)$ is said to be *maximal* in $R(\mathcal{A})_{g,K}$ with respect to ℓ if no *succ*-transition is possible from r . In the sequel, the topological properties of r are implicitly derived from those of Z . We write \bar{r} for the topological closure of r , and recall that min_r denotes the minimal vector of \bar{r} .

We also consider another automaton, called the *class automaton*, in which the states, called *classes*, are of the form (ℓ, Z) , where Z is a zone. In this case, the automaton is built 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 $(\ell, \gamma, a, R, \ell') \in E$ such that $Z_1 \cap \llbracket \gamma \rrbracket \neq \emptyset$, and $Z_2 = fut((Z_1 \cap \llbracket \gamma \rrbracket)[R \mapsto 0]) \cap Inv(\ell')$.

While this construction does not always yield a finite number of classes, it is now well known [12] that for a timed automaton without diagonal constraints (as is our case), replacing each zone Z from the previous construction by its K -approximation ($Approx_K(Z)$, the smallest K -zone containing Z), we obtain a finite automaton such that any configuration belonging to a zone is strongly timed bisimilar to a reachable configuration (see also the discussion of the next paragraph). This method is usually implemented in tools like UPPAAL [25] or KRONOS [28] for on-the-fly verification, with each zone represented by a Difference Bounded Matrix [18].

2.5 Reachability

For a region r of $R(\mathcal{A})_{g,K}$, not all configurations of r are reachable. Nevertheless, by induction on the reachability relation, the following property can be shown: for any region r of $R(\mathcal{A})_{g,K}$, there is a region $reach(r)$ with respect to the g -grid and constant $K = \infty$ such that

- (i) $reach(r) \subset r$,
- (ii) each configuration of $reach(r)$ is reachable and
- (iii) if $reach(r)$ admits a time-open description then this is also the case for r , else r admits a time-closed description.

As a consequence, we have: $\forall x \in ActX_r, min_{reach(r)}(x) = min_r(x)$ and $\forall x \in X \setminus ActX_r, min_{reach(r)}(x) \geq K$ and ord_r restricted to $ActX_r$ is identical to $ord_{reach(r)}$.

Consider now the relation \mathcal{R} defined by $(\ell, v) \mathcal{R} (\ell, v')$ iff $\forall x \in X, v'(x) = v(x) \vee (v(x) \geq K \wedge v'(x) \geq K)$. It is a strong timed bisimulation relation. From the previous observations, we note that each configuration of a region is strongly timed bisimilar to a reachable configuration of this region. Thus speaking about reachability of regions is a slight abuse of notations. Note that the same property holds for the classes of the class automaton.

3 Characterizing TA bisimilar to Time Petri Nets

3.1 Time Petri Nets

Introduced in [24], Time Petri Nets (TPNs) extend Petri nets by associating a (topological) closed time interval with each transition (see figure 4). The meaning of this interval is detailed after the definition.

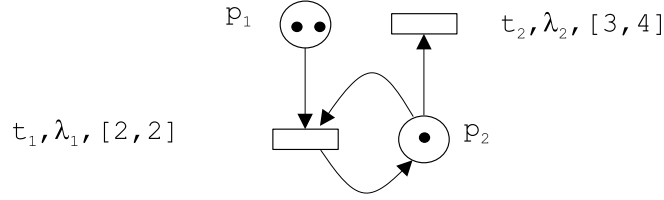


Fig. 4. A labelled time Petri net

Definition 5 (Labeled Time Petri Net). A Labeled Time Petri Net \mathcal{N} over Σ_ε is a tuple $(P, T, \Sigma_\varepsilon, \bullet(\cdot), (\cdot)^\bullet, M_0, \Lambda, I)$ where:

- P is a finite set of places,
- T is a finite set of transitions with $P \cap T = \emptyset$,
- $\bullet(\cdot) \in (\mathbb{N}^P)^T$ is the backward incidence mapping,
- $(\cdot)^\bullet \in (\mathbb{N}^P)^T$ is the forward incidence mapping,
- $M_0 \in \mathbb{N}^P$ is the initial marking,
- $\Lambda : T \rightarrow \Sigma_\varepsilon$ is the labeling function,
- $I : T \mapsto \mathcal{I}(\mathbb{Q}_{\geq 0})$ associates with each transition a closed firing interval.

A TPN \mathcal{N} is a g -TPN if for all $t \in T$, the interval $I(t)$ has its 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 is common in the literature. The intended meaning of these notations will be clear from the context.

As usual, a *marking* M is a mapping in \mathbb{N}^P , with $M(p)$ the number of tokens in place p . A transition t is *enabled* in a marking M iff $M \geq \bullet t$. We denote by $En(M)$ the set of enabled transitions in M . A *configuration* of a TPN is a pair (M, ν) where M is a marking and the *valuation* ν is a mapping in $(\mathbb{R}_{\geq 0})^{En(M)}$, which describes the values of clocks implicitly associated with transitions enabled in M . Roughly speaking, such a clock measures the time elapsed since the enabling of the corresponding transition as explained more precisely below.

An enabled transition t can be *fired* if $\nu(t)$ belongs to the interval $I(t)$. The result of this firing is as usual the new marking $M' = M - \bullet t + t^\bullet$. Moreover, some valuations are reset and we say that the corresponding transitions are newly enabled. Different semantics are possible for this operation. In this paper, we choose *persistent atomic semantics*, which is different from the classical semantics [10,5], but equivalent when the net is bounded [6]. The predicate describing when a transition t' is newly enabled by the firing of transition t is defined by:

$$\uparrow enabled(t', M, t) = t' \in En(M - \bullet t + t^\bullet) \wedge (t' \notin En(M)).$$

Thus, firing a transition is considered as an atomic step and the transition currently fired behaves like the other transitions ($\nu(t)$ need not be reset when t is fired). In classical semantics, when firing a transition, disabling of the other transitions is checked after the consumption of tokens from input places and before the production of tokens to output places. Furthermore the clock associated with the fired transition is always reset. Observe that our results also hold for the classical semantics but we have chosen this one as it leads to more elegant constructions in sections 5 and 6. More precisely, the sufficient condition is obtained by construction of a bounded TPN and it has been proven that for bounded nets, both semantics are equivalent [6]. The necessary condition is based on lemmas 1 and 2 which do not depend on the triggering of clock reset in TPNs.

The set $ADM(\mathcal{N})$ of *admissible configurations* consists of the pairs (M, ν) such that $\nu(t) \in I(t)^\downarrow$ for each transition $t \in En(M)$. Thus time can progress in a marking only until it reaches the first right endpoint of the intervals for all enabled transitions. For $d \in \mathbb{R}_{\geq 0}$, the valuation $\nu + d$ is defined by $(\nu + d)(t) = \nu(t) + d$ for each $t \in En(M)$.

Definition 6 (Semantics of TPN). *The semantics of a TPN $\mathcal{N} = (P, T, \Sigma_\varepsilon, \bullet(\cdot), (\cdot)^\bullet, M_0, \Lambda, I)$ is a TTS $S_{\mathcal{N}} = (Q, q_0, \rightarrow)$ where $Q = ADM(\mathcal{N})$, $q_0 = (M_0, \mathbf{0})$ and \rightarrow is defined by:*

$$\begin{aligned} & - \text{either a delay move } (M, \nu) \xrightarrow{d} (M, \nu + d) \\ & \text{iff } \forall t \in En(M), \nu(t) + d \in I(t)^\downarrow, \end{aligned}$$

- or a discrete move $(M, \nu) \xrightarrow{A(t)} (M - \bullet t + t \bullet, \nu')$ where $\forall t' \in \text{En}(M - \bullet t + t \bullet)$, $\nu'(t') = 0$ if $\uparrow \text{enabled}(t', M, t)$ and $\nu'(t') = \nu(t')$ otherwise, iff $t \in \text{En}(M)$ and $\nu(t) \in I(t)$.

We simply write $(M, \nu) \xrightarrow{w}$ to emphasise that a sequence of transitions w can be fired. If $\text{Duration}(w) = 0$, we say that w is an *instantaneous firing sequence*. A net is said to be k -bounded if for each reachable configuration (M, ν) and for each place p , $M(p) \leq k$.

Denoting $((M_0(p_1), M_0(p_2)), (\nu_0(t_1), \nu_0(t_2)))$ by $((2, 1), (0, 0))$, the sequence $((2, 1), (0, 0)) \xrightarrow{2} ((2, 1), (2, 2)) \xrightarrow{t_1} ((1, 1), (2, 2)) \xrightarrow{t_1} ((0, 1), (-, 2))$ is a firing sequence of the net of figure 4. Note that, due to the semantics, the first firing of t_1 does not *reset* its clock even if the token in p_2 is consumed and produced again. Note also that time cannot progress beyond time 2 as long as t_1 remains fireable. Afterwards time can progress for at most 2 t.u. at which point t_2 must fire (it could also fire before). This temporal feature is sometimes called an *urgent* behaviour in contrast to a *lazy* behaviour.

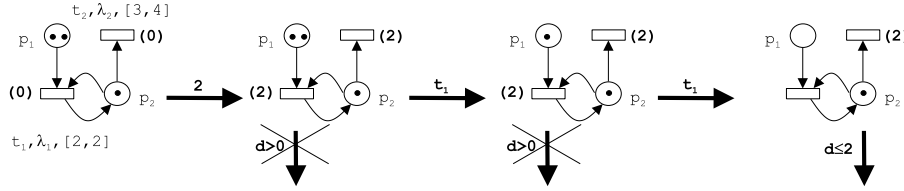


Fig. 5. An execution in a time Petri net

3.2 A characterization of TA bisimilar to TPNs

We aim at characterizing TA bisimilar to some TPN whatever the labelling of the TA. In fact to check this property is equivalent to test whether an associated injectively-labelled TA is bisimilar to some TPN. This is why we only consider injectively-labelled TA in the sequel.

TA include both lazy mechanisms since time elapsing can falsify the guard of an edge and urgent mechanisms since invariants may forbid time elapsing in some configurations. Hence a characterization of TA bisimilar to some TPN seems to be closely related to the absence of laziness features. In this paragraph, we first present two intuitive conditions for a TA to be bisimilar to a TPN which do not characterize this property before introducing the exact characterization.

The first condition is obtained by observing that a transition (and more generally an instantaneous firing sequence) of a TPN cannot be disabled by time elapsing. Consequently, our first attempt to characterize this property is: “A TA is bisimilar to some TPN if time elapsing cannot falsify the guard of an edge

starting from the current location". By virtue of the observation this condition is necessary. However it is not sufficient and the automaton \mathcal{A}_1 of figure 1 does not admit a bisimilar TPN whereas it fulfills this condition as can be checked on its region automaton represented in figure 6. The three parts of this figure (from left to right) represent respectively the reachable regions associated with locations ℓ_0, ℓ_1 and ℓ_2 . Thin arrows represent time elapsing (thus staying in the same location) while thick (labelled) arrows represent discrete transitions from some (ℓ, Z) to another region (ℓ', Z') . For instance, $(\ell_0, 0 < x = y < 1) \xrightarrow{a} (\ell_1, 0 < x < 1 \wedge y = 0)$

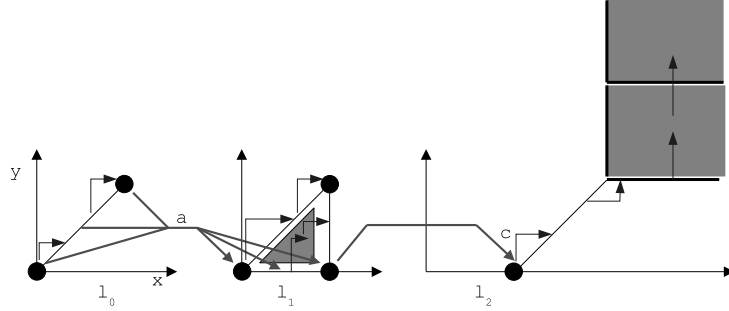


Fig. 6. The region automaton of \mathcal{A}_1 ($K = 2, g = 1$)

The previous observation about the behaviour of a TPN may be sharpened and yields the following lemma useful for subsequent developments.

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

- (i) $(M, \nu) \xrightarrow{w}$ implies $(M, \nu + \delta) \xrightarrow{w}$
- (ii) If $\nu \in \mathbb{N}_g^{En(M)}$ and $\delta \in [0, 1/g]^{En(M)}$ then $(M, \nu + \delta) \xrightarrow{w}$ implies $(M, \nu) \xrightarrow{w}$

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 (i) Since t is fireable from (M, ν) , $\nu(t) \in I(t) \subset I(t)^\uparrow$, so $\nu(t) + \delta(t) \geq \nu(t)$ also belongs to $I(t)^\uparrow$. Since $t \in En(M)$ and $(M, \nu + \delta)$ is reachable, $\nu(t) + \delta(t) \in I(t)^\downarrow$. Thus $\nu(t) + \delta(t) \in I(t)$ and t is also fireable from $(M, \nu + \delta)$. Since t' is newly enabled, $0 \in I(t')$ and t' is also fireable when it occurs starting from $(M, \nu + \delta)$.

Proof of (ii) The case of newly enabled transitions in w is handled as before. Now let t be fireable in $(M, \nu + \delta)$. Since $t \in En(M)$ and (M, ν) is reachable, $\nu(t) \in I(t)^\downarrow$. Since $\nu(t) + \delta(t) \in I(t)^\uparrow$, (denoting by $eft(t)$ the minimum of $I(t)^\uparrow$), we have $eft(t) \leq \nu(t) + \delta(t)$ but $eft(t)$ belongs to the g -grid, thus $eft(t) \leq \nu(t) \Leftrightarrow \nu(t) \in I(t)^\uparrow$. So t is fireable from (M, ν) . \square

Taking into account this lemma, one can elaborate a slightly modified version of the first condition: “A TA is bisimilar to some TPN if given any two reachable configurations (l, v) and (l, v') with $v' \geq v$ and an edge e , then $(l, v) \xrightarrow{e} \Rightarrow (l, v') \xrightarrow{e}$ ”. For instance, \mathcal{A}_1 does not fulfill this condition (take $(l_1, (1, 0)), (l_1, (1, 1))$ and the edge labelled by c in figure 6). Unfortunately this condition is not necessary. The TA \mathcal{A}_0 of figure 1 does not fulfill this condition whereas there exists a TPN bisimilar to it (see figure 13 in section 5). Indeed take $(l_1, (1, 0)), (l_1, (1, 1))$ and the edge labelled by c and see its region automaton presented in figure 7 (with the same conventions as in figure 6).

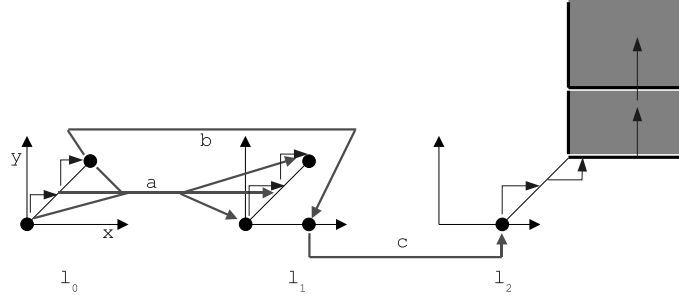


Fig. 7. The region automaton of \mathcal{A}_0 ($K = 2, g = 1$)

The difference between the two automata w.r.t. the property to be checked is the existence of the reachable region $(l_1, x = 1 \wedge 0 < y < 1)$ whose topological closure includes the configuration $(l_1, (1, 0))$. In fact, the exact characterization takes into account topological considerations as stated by the following theorem which also contains effectiveness and complexity results.

Theorem 1. *Let \mathcal{A} be an injectively-labelled timed automaton and $R(\mathcal{A})_{1,K}$ its region automaton with 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 for each region r of $R(\mathcal{A})_{1,K}$ and for each edge e from \mathcal{A} ,*

- (a) *Every region r' such that $r' \cap \bar{r} \neq \emptyset$ is reachable*
- (b) *$\forall (l_r, v) \in r$, if $(l_r, v) \xrightarrow{e}$ then $(l_r, \min_r) \xrightarrow{e}$*
- (c) *$\forall (l_r, v) \in \bar{r}$, if $(l_r, \min_r) \xrightarrow{e}$ then $(l_r, v) \xrightarrow{e}$.*

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 denote by \mathcal{TA}^{wtb} the corresponding subclass of timed automata.

Thus \mathcal{TA}^{wtb} is the maximal subclass of TA that are weakly timed bisimilar to a TPN. In [7], we have proposed a “syntactical” (proper) subclass of \mathcal{TA}^{wtb} which avoids to check this characterization.

The characterization of Theorem 1 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. Consider again the two TA \mathcal{A}_0 and \mathcal{A}_1 in Figure 1. The automaton \mathcal{A}_0 admits a bisimilar TPN whereas \mathcal{A}_1 does not. Indeed, 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 .

The next sections are devoted to the proof of Theorem 1.

4 Proof of necessary condition for Theorem 1

4.1 From bisimulation to uniform bisimulation

As a first step, we prove that when a g -TPN and a TA are bisimilar, this relation can in fact be strengthened in what we call *uniform bisimulation*. Lemma 2 is the central point for the proof of necessity. It shows that bisimulation implies uniform bisimulation for the g -grid with $K = \infty$. Roughly speaking, uniform bisimulation means that a unique mechanism is used for every configuration of the topological closure of the region to obtain a bisimilar configuration of the net.

Lemma 2 (From bisimulation to uniform bisimulation). *Let \mathcal{A} be a timed automaton bisimilar to some g -TPN \mathcal{N} via some relation \mathcal{R} and let $R(\mathcal{A})_{g,\infty}$ be a region automaton of \mathcal{A} . Then:*

- if a region r belongs to $R(\mathcal{A})_{g,\infty}$ then any region included in \bar{r} also belongs to $R(\mathcal{A})_{g,\infty}$;
- for each region r , there exist a configuration of the net (M_r, ν_r) with $\nu_r \in \mathbb{N}_g^{En(M_r)}$ and a mapping $\phi_r : En(M_r) \rightarrow [X]_r$ such that:
 - If r is time-closed, then for each $\delta \in \mathbb{R}_{\geq 0}^X$ such that $(\ell_r, \min_r + \delta) \in \bar{r}$, $(\ell_r, \min_r + \delta) \mathcal{R} (M_r, \nu_r + \text{proj}_r(\delta))$,
 - If r is time-open, then for each $\delta \in \mathbb{R}_{\geq 0}^X$, $d \in \mathbb{R}_{\geq 0}$ such that $(\ell_r, \min_r + \delta + d) \in \bar{r}$, $(\ell_r, \min_r + \delta + d) \mathcal{R} (M_r, \nu_r + \text{proj}_r(\delta) + d)$,
where $\text{proj}_r(\delta)(t) = \delta(\phi_r(t))$.

Proof. First note that the choice of a particular clock x in the class $\phi_r(t)$ is irrelevant when considering the value $\delta(x)$. Thus the definition of proj_r is sound. The proof is an induction on the transition relation in the region automaton. The basis case is straightforward with $\{(l_0, \mathbf{0})\}$ and $\{(M_0, \mathbf{0})\}$. The induction part relies on lemma 1, with 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, 2. a time step from a time-open region, 3. a discrete step into a time-closed region, and 4. a discrete step into a time-open region.

1. A time step from a time-closed region (see figure 8(1)). Let r be a time-closed region in $R(\mathcal{A})_{g,\infty}$ and let us denote $r' = \text{succ}(r)$ the immediate

time successor of r . Let $(\ell_r, \min_r + \delta_0)$ be some item of r . $(\ell_r, \min_r + \delta_0) \xrightarrow{d}$ for some $d > 0$. Thus (by induction hypothesis) in \mathcal{N} there is a step sequence of $(M_r, \nu_r + \text{proj}_r(\delta_0)) \xrightarrow{d_0 t_1 \dots t_n d_n}$ 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 1-(ii), the firing sequence $t_1 \dots t_k$ is fireable from (M_r, ν_r) .

Let us choose $(M_{r'}, \nu_{r'})$ the configuration reached by this sequence. By application of lemma 1-(i), this firing sequence is also fireable from any $(M_r, \nu_r + \text{proj}_r(\delta))$ bisimilar to $(\ell_r, \min_r + \delta) \in \bar{r}$ and it leads to $(M_{r'}, \nu_{r'} + \text{proj}_{r'}(\delta))$ (still bisimilar to $(\ell_r, \min_r + \delta)$) 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) with the transitions newly enabled. Since $(M_{r'}, \nu_{r'})$ let the time elapse and since \mathcal{N} is a g -TPN, we note that $\forall t \in \text{En}(M_{r'}), \nu_{r'}(t) + 1/g \in I(t)^\downarrow$. Now let $(\ell_r, \min_r + \delta + d) \in \bar{r}'$, one has $\forall x \in X, \delta(x) + d \leq 1/g$. Thus $\forall t \in \text{En}(M_{r'}), \text{proj}_{r'}(\delta(x)) + d \leq 1/g$, which implies $(M_{r'}, \nu_{r'} + \text{proj}_{r'}(\delta)) \xrightarrow{d} (M_{r'}, \nu_{r'} + \text{proj}_{r'}(\delta) + d)$, this last configuration being necessarily bisimilar to $(\ell_r, \min_r + \delta + d)$.

2. A time step from a time-open region (see figure 8(2)). Let r be an time-open region and let us denote $r' = \text{succ}(r)$. Note that $\bar{r}' \subset \bar{r}$. Let us define X_r^{\max} the class $[x]_r$ with maximal index. We remark that $\min_{r'} = \min_r + \delta_0$ where if $x \in X_r^{\max}$ then $\delta_0(x) = 1/g$ else $\delta_0(x) = 0$. We choose $(M_{r'}, \nu_{r'}) = (M_r, \nu_r + \text{proj}_r(\delta_0))$. Let $t \in \text{En}(M_r)$ and $x \in \phi_r(t)$ then $\phi_{r'}(t) = [x]_{r'}$ (letting time elapse does not split the classes). So proj_r and $\text{proj}_{r'}$ are identical.

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

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

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

3. A discrete step into a time-closed region (see figure 8(3)).

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

Let $(\ell_r, \min_r + \delta_0)$ be some element of r . Suppose that $(\ell_r, \min_r + \delta_0) \xrightarrow{e} (\ell', \nu' + \delta'_0)$ with $\forall x \in R(e), \nu'(x) = \delta'_0(x) = 0, \forall x \notin R(e), \nu'(x) = \min_r(x) \wedge \delta'_0(x) = \delta_0(x)$. Then in \mathcal{N} there is an instantaneous firing sequence $(M_r, \nu_r + \text{proj}_r(\delta_0)) \xrightarrow{w}$ labelled by e . Due to lemma 1, this firing sequence is also fireable from any $(M_r, \nu_r + \text{proj}_r(\delta))$ bisimilar to $(\ell_r, \min_r + \delta) \in \bar{r}$. By bisimilarity, $(\ell_r, \min_r + \delta) \xrightarrow{e}$ for any $(\ell_r, \min_r + \delta) \in \bar{r}$. Let r' be the region including $(\ell', \nu' + \delta'_0)$, then any configuration of \bar{r}' is reachable by this discrete step. Note that $\ell_{r'} = \ell'$ and $\min_{r'} = \nu'$.

From $(M_r, \nu_r + \text{proj}_r(\delta))$, the sequence w leads to some (M', ν') bisimilar to $(\ell_{r'}, \min_{r'} + \delta')$. 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

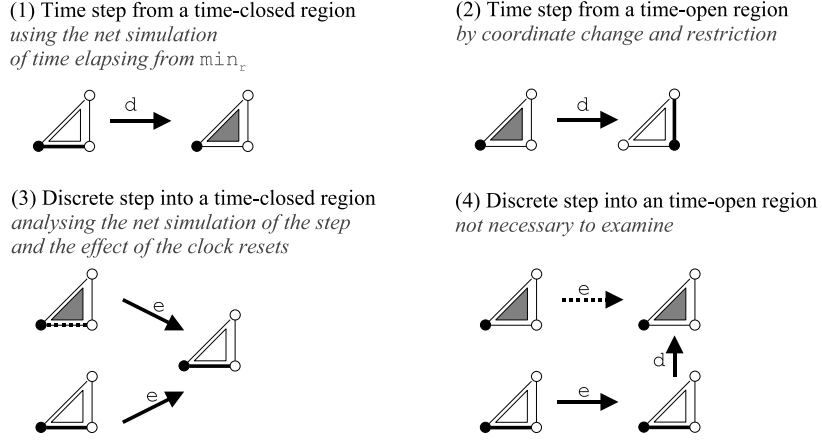


Fig. 8. The different cases of the proof

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 two cases to consider for $\phi_{r'}(t)$: if 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 precedes $\phi_r(t)$ and contains a clock not reset or else the class of index 1. The two last affectations are sound since, it means that whatever, before the firing of w , the fractional value of the implicit clock associated with t between the fractional value of the new clock corresponding to t and the fractional value of a clock of $\phi_r(t)$, 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 $(\ell_r, \min_r + \delta_0 + d_0)$ be some element of r . Suppose that $(\ell_r, \min_r + \delta_0 + d_0) \xrightarrow{e} (\ell', v' + \delta'_0)$ with $\forall x \in R(e), v'(x) = \delta'_0(x) = 0, \forall x \notin R(e), v'(x) = \min_r(x) \wedge \delta'_0(x) = \delta_0(x) + d_0$. Then in \mathcal{N} there is an instantaneous firing sequence $(M_r, \nu_r + proj_r(\delta_0) + d_0) \xrightarrow{w}$ labelled by e . Due to lemma 1, this firing sequence is also fireable from any $(M_r, \nu_r + proj_r(\delta) + d)$ bisimilar to $(\ell_r, \min_r + \delta + d) \in \bar{r}$. By bisimilarity, $(\ell_r, \min_r + \delta + d) \xrightarrow{e}$ for any $(\ell_r, \min_r + \delta + d) \in \bar{r}$. Let r' be the region including $(\ell', v' + \delta'_0)$, then any configuration of \bar{r}' is reachable by this discrete step. Note that $l_{r'} = \ell'$ and $\min_{r'} = v'$.

From $(M_r, \nu_r + proj_r(\delta) + d)$, the sequence w leads to some (M', ν') bisimilar to $(\ell_{r'}, \min_{r'} + \delta')$. 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 two cases to consider

for $\phi_{r'}(t)$: if there is a $x \in \phi_r(t)$ not reset, then $\phi_{r'}(t) = \lfloor x \rfloor_{r'}$, otherwise $\phi_{r'}(t)$ is the class of maximal index which precedes $\phi_r(t)$ and contains a clock not reset or else the class of index 1. The two last affectations are sound since, it means that whatever, before the firing of w , the fractional value of the implicit clock associated with t between the fractional value of the new clock corresponding to t and the fractional value of a clock of $\phi_r(t)$, 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 (see figure 8(4)). In order to reach a time-open region by a discrete step, the corresponding transition must start from a time-open region and must not reset any clock. Let $(\ell_r, \min_r + \delta + d) \in r$ and $(\ell_r, \min_r + \delta + d) \xrightarrow{e} (\ell', \min_r + \delta + d)$. Here we have used the hypothesis that no clock is reset. Then there is a firing sequence $(M_r, \nu_r + \text{proj}_r(\delta) + d) \xrightarrow{w}$ labelled by e . Due to the lemma 1, $(M_r, \nu_r + \text{proj}_r(\delta)) \xrightarrow{w} (\ell_r, \nu_r + \delta)$ is bisimilar to $(M_r, \nu_r + \text{proj}_r(\delta))$. Thus $(\ell_r, \min_r + \delta) \xrightarrow{e} (\ell', \min_r + \delta) \xrightarrow{d} (\ell', \min_r + \delta + d)$. 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. \square

4.2 Proof of Necessity

The fact that conditions (a), (b) and (c) of Theorem 1 hold for $R(\mathcal{A})_{g,\infty}$ is now straightforward:

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

In order to complete the proof, we successively show that if the conditions are satisfied in $R(\mathcal{A})_{g,\infty}$ for some g , they also hold for $R(\mathcal{A})_{1,\infty}$ (lemma 3), and finally that they are satisfied in the standard region automaton $R(\mathcal{A})_{1,K}$, with a finite constant K , sufficiently large (lemma 4). Recall (section 2.2) 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 .

Lemma 3 (about the conditions and the grid). *Let \mathcal{A} be a timed automaton and $g > 0$ in \mathbb{N} . If conditions (a),(b),(c) are satisfied in the region automaton $R(\mathcal{A})_{g,\infty}$, then they are satisfied in $R(\mathcal{A})_{1,\infty}$.*

Proof. From the definition of regions, a region r of $R(\mathcal{A})_{1,\infty}$ is a finite union of regions of $R(\mathcal{A})_{g,\infty}$ (say $r = \bigcup_{i=1..k} r_i$). Thus $\bar{r} = \bigcup_{i=1..k} \bar{r}_i$ which proves the implication for (a).

Assume that (b) is satisfied by $R(\mathcal{A})_{g,\infty}$. Let $(\ell_r, \min_r + \delta + d) \in r$ be a region of $R(\mathcal{A})_{1,\infty}$ and assume $(\ell_r, \min_r + \delta + d) \xrightarrow{e}$. We define δ' by $\delta'(x) = \delta(x)/g$. Then since \mathcal{A} has integer constraints $(\ell_r, \min_r + \delta' + d/g) \xrightarrow{e}$. Moreover this configuration belongs to r and then to a region $r' \in R(\mathcal{A})_{g,\infty}$ whose minimal vector is $\min_{r'}$. Then applying (b), we obtain $(\ell_r, \min_{r'}) \xrightarrow{e}$.

Assume that (c) is satisfied by $R(\mathcal{A})_{g,\infty}$. Let $(\ell_r, v) \in \bar{r}$ where r is a region of $R(\mathcal{A})_{1,\infty}$ and assume $(\ell_r, \min_r) \xrightarrow{e}$. Then there is an increasing path among the minimum vectors of regions of $R(\mathcal{A})_{g,\infty}$ all included in \bar{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 $(\ell_r, v) \in \bar{r}_*$ (with r_* a region of $R(\mathcal{A})_{g,\infty}$). Thus applying iteratively (c) yields $(\ell_r, v) \xrightarrow{e}$. \square

Lemma 4 (about the conditions and the constant K). *Let \mathcal{A} be a timed automaton. If conditions (a),(b),(c) are satisfied in $R(\mathcal{A})_{1,\infty}$, then they hold in the region automaton $R(\mathcal{A})_{1,K}$, for some finite constant K .*

Proof. Let r be a region in $R(\mathcal{A})_{1,K}$ where K is greater than the maximal constant in \mathcal{A} and $reach(r)$ the associated region of $R(\mathcal{A})_{1,\infty}$. 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 $\lim_{i \rightarrow \infty} (\ell_r, v_i) = (\ell_r, v)$ with $(\ell_r, v_i) \in r$ so that $(\ell_r, v) \in \bar{r}$. Then the corresponding sequence $\{(\ell_r, v'_i)\}$ being bounded admits an accumulation point $(\ell_r, v') \in \bar{r}$. It is routine to show that (ℓ_r, v) and (ℓ_r, v') belong to the same region in $R(\mathcal{A})_{1,K}$. This proves that condition (a) for $R(\mathcal{A})_{1,\infty}$ implies condition (a) for $R(\mathcal{A})_{1,K}$.

Assume that (b) is satisfied by $R(\mathcal{A})_{1,\infty}$. Let $(\ell_r, v) \in r$ be a reachable region of $R(\mathcal{A})_{1,K}$ and $(\ell_r, v) \xrightarrow{e}$. Let $reach(r)$ be the associated reachable region (as explained in section 2.5) of $R(\mathcal{A})_{1,\infty}$ 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 $R(\mathcal{A})_{1,\infty}$ and consider $(\ell_r, v) \in \bar{r}$ where r is a region of $R(\mathcal{A})_{1,K}$ and $(\ell_r, \min_r) \xrightarrow{e}$. Again let $reach(r)$ be the associated reachable region of $R(\mathcal{A})_{1,\infty}$, then $\exists (\ell_r, v') \in reach(r)$ 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 $(\ell_r, v) \xrightarrow{e}$. \square

5 Sufficient condition: first construction

Starting from a timed automaton \mathcal{A} satisfying the conditions of Theorem 1, we build a 2-TPN bisimilar to \mathcal{A} . We describe the construction and give the proof of correctness.

For the figures corresponding to all constructions, all edges are weighted by 1. Omitted labels for transitions stand for ε . A firing interval $[0, 0]$ is indicated by a blackened transition and intervals $[0, \infty[$ are omitted. A double arrow between a place p and a transition t indicates that p is both an input and an output place for t .

5.1 Construction

First remark that any constraint of the form $x < c$ occurring in an invariant of \mathcal{A} may be safely omitted. 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), \bar{r} is reachable but since r is time-open, $\bar{r} \cap \text{succ}(r) \neq \emptyset$, so that $\text{succ}(r)$ is reachable which contradicts the maximality of r .

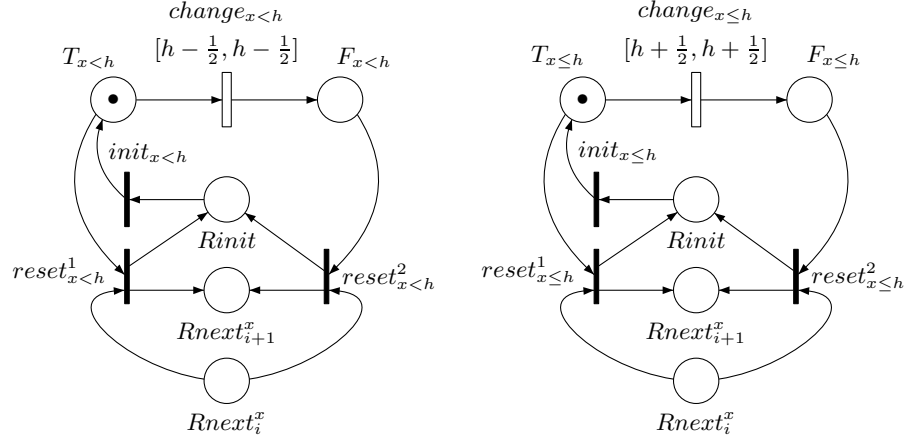


Fig. 9. Subnets for $x < h$ (with $h > 0$) and $x \leq h$ in guards

Clock constraints. The atomic constraints associated with a clock x are arbitrarily numbered from 1 to $n(x)$ where $n(x)$ is the number of such conditions. When $x \leq h$ occurs in at least one transition and in at least one invariant, we consider it as two different conditions. Then we add places $Rinit$ and $(Rnext_i^x)_{i \leq n(x)+1}$ for the reset operations. We build a subnet for each atomic constraint $x \bowtie h$ occurring in a transition of the TA, and one for each condition $x \leq h$ occurring in an invariant. Figure 9 shows the subnets corresponding to $x < h$ (with $h > 0$) on the left and $x \leq h$ on the right, while Figure 10 shows

the subnets for $x > h$ on the left and $x \geq h$ (with $h > 0$) on the right, in the case where the constraint has number i . Figure 11 shows the subnet for invariant $x \leq h$.

Since constant $\frac{1}{2}$ appears in interval bounds, the resulting TPN is a 2-TPN.

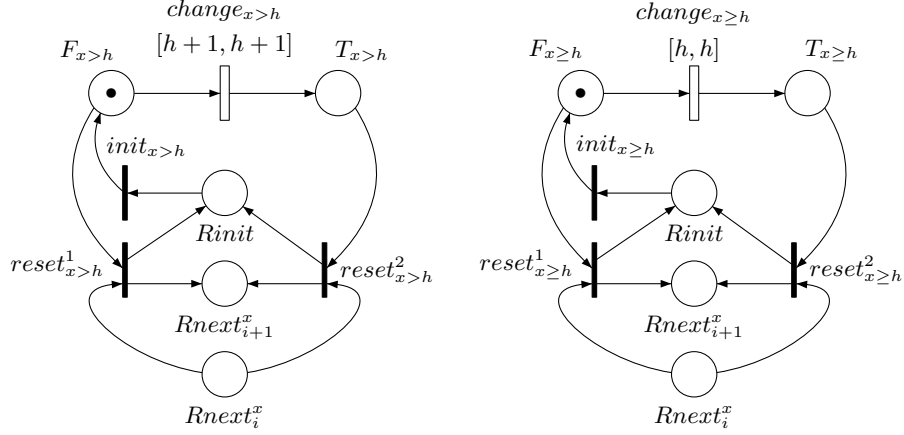


Fig. 10. Subnets for $x > h$ and $x \geq h$ (with $h > 0$) in guards

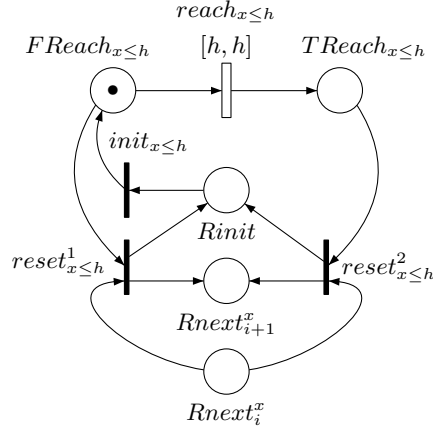


Fig. 11. Subnet for $x \leq h$ in an invariant

Locations and edges. 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 invariant $Inv(\ell)$ is tested with the subnets corresponding to its atomic constraints:

- if condition $x \leq h$ (with $h > 0$) occurs in the invariant, then we add a transition $stop_\ell^{x \leq h}$, with both ℓ and $TReach_{x \leq h}$ as input and output places, and interval $[0, 0]$,
- if condition $x \leq 0$ occurs in the invariant, we simply add a transition $stop_\ell$ with ℓ as only input and output place, and also interval $[0, 0]$.

To simulate an edge $e = (\ell, \gamma, a, R, \ell')$, we must test the atomic constraints from $\gamma = \gamma_1 \wedge \dots \wedge \gamma_{m(e)}$, using the places corresponding to true in the associated subnets, and reset successively all the clocks in $R = \{x_1, \dots, x_{k(e)}\}$ by instantaneous transitions. This is done by the subnet in Figure 12, which must be connected to some subnets like those of Figure 9, 10 or 11.

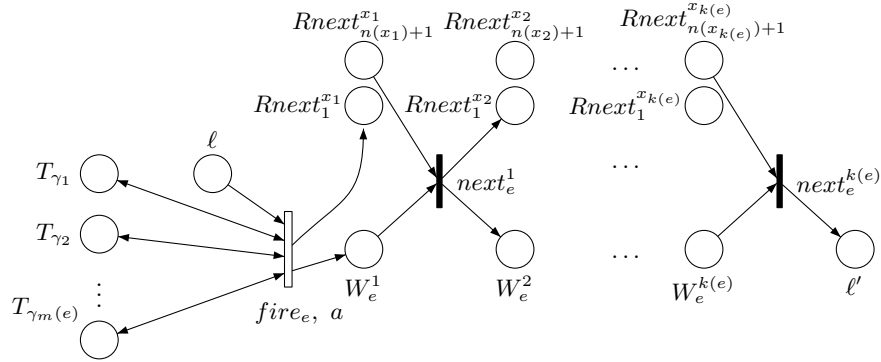


Fig. 12. The subnet for edge $e = (\ell, \gamma = \gamma_1 \wedge \dots \wedge \gamma_{m(e)}, a, R = \{x_1, \dots, x_{k(e)}\}, \ell')$

This construction is illustrated in figure 13 for the timed automaton \mathcal{A}_0 from figure 1 with some simplifications related to this particular timed automaton. More precisely, places $FReach_{x \leq 1}$ and $F_{x \geq 1}$ (resp. $TReach_{x \leq 1}$ and $T_{x \geq 1}$) have been merged. Moreover since x is never reset in \mathcal{A}_0 the corresponding parts in the TPN are omitted.

First note that the subnet associated to the constraint $y \leq 0$ switches the condition to false (marking $F_{y \leq 0}$) when the implicit value of y maintained in the net reaches $1/2$. This translation thus seems less constrained than the original condition, so we explain how we prove that it 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 (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 can also be simulated in the net using both conditions (b) and (c) and the following fact: $\forall x \in X, \exists (\ell_r, v'), (\ell_r, v'') \in \bar{r}$ such that $v'(x) = \lfloor v(x) \rfloor$ and $v''(x) = \lceil v(x) \rceil$. The subnet associated to the atomic constraint $x \leq 1$ occurring in the invariant of ℓ_0 leads to transition inv_0 (not modifying the marking)

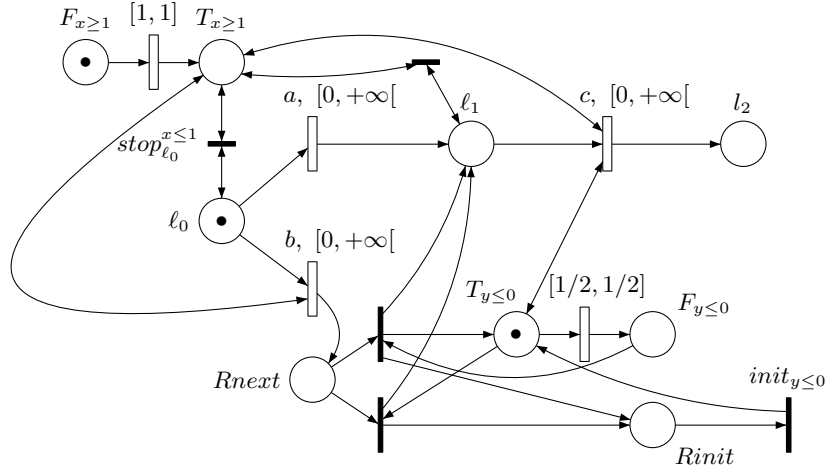


Fig. 13. A 2-TPN bisimilar to \mathcal{A}_0

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.

5.2 Correctness proof

We decompose the reachable configurations (and markings) into *intermediate* ones (some W_e^i 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”: $\nu(\text{change}_c) = \nu(\text{change}_{c'}) = \nu(\text{reach}_{c'})$ 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 occurring 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 instantaneous firing sequences (i.e. no time step) and the finite maximal ones lead to permanent configurations. These permanent configurations (say (M_{next}, ν_{next})) have the same marked place ℓ and the same values $\nu_{next}(x)$. They may only differ depending on whether some transitions related to a condition switch have been fired.

It is also obvious that once some fire_e is fired, the construction ensures the existence of a “resetting” sequence which reinitializes the subnets 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 $(\ell, \nu)\mathcal{R}(M, \nu)$ iff:

- either M is a permanent marking and $M(\ell)$ is marked and if $\nu(x) < K(x)$ then $\nu(x) = \nu(x)$ else $\nu(x) \geq K(x)$.

- 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 5. *The relation \mathcal{R} defined above is a weak timed bisimulation.*

Proof. We first consider moves from \mathcal{A} .

Case 1: $(\ell, v) \xrightarrow{e} (\ell', v')$.

We 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 occurring 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 (possibly 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), $(\ell, min_r) \xrightarrow{e}$. Thus $v(x) \geq min_r(x) \geq a + 1$ which implies $\nu(x) \geq a + 1$ and that $T_{x > a}$ is marked (possibly 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 $(\ell, v_1) \in \bar{r}$ with $v_1(x) = \lceil v(x) \rceil$. Using condition (b) and then (c), $(\ell, v_1) \xrightarrow{e}$.

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, we have to fire a sequence leading to some (M_{next}, ν_{next}) and perform the previous simulation.

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

If M is an intermediate marking, again we fire an instantaneous sequence leading to some (M_{next}, ν_{next}) still bisimilar to (ℓ, v) and perform the following simulation for the case when M is permanent.

We must determine which transitions could prevent d time units to elapse: either *change* or *stop* transitions. When after $d' < d$ time units, a *change* transition prevents time elapsing, it is fired leading to another permanent marking bisimilar to $(\ell, v + d')$. The only possible *stop* transitions that could prevent time elapsing correspond to conjuncts of the form $x \leq a$ belonging 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 with ℓ will be possibly fireable but only at the end of this step sequence.

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 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 occurring 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 $T_{x > a}$ 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 time elapse. Thus M is a permanent marking. Let $x \leq a$ belonging to the invariant of ℓ . Then $a \neq 0$, otherwise from (M, ν) , transition $stop_\ell$ must be fired and time may not elapse. Similarly since $stop_\ell^{x \leq a}$ 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 $(\ell, v + d)\mathcal{R}(M, \nu + d)$. \square

6 Sufficient condition: second construction

When the conditions on the injectively-labelled timed automaton \mathcal{A} are satisfied, we now build a 1-TPN \mathcal{N} which is weakly timed bisimilar to \mathcal{A} .

6.1 Construction

The construction of the TPN contains a partial replication of both the region automaton of \mathcal{A} and its class automaton. Recall that we consider $K = m + 1$, where m is the maximal constant for \mathcal{A} . 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).

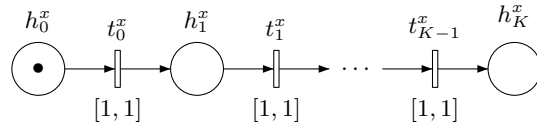


Fig. 14. Subnet for clock x

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 $v, v' \leq \overline{K}$, we build a subnet (see figure 15) 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 $C_1 = (\ell, Z_1), \dots, C_k = (\ell, Z_k)$ 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 $1 \leq i \leq k$. Otherwise stated, there is a configuration in every Z_i which is strongly time bisimilar to v . Let C'_1, \dots, C'_k the classes obtained by applying transition e to C_1, \dots, C_k respectively. We have a transition with label e for each C_i (with $k = 2$ in figure 15), 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' in the figure).

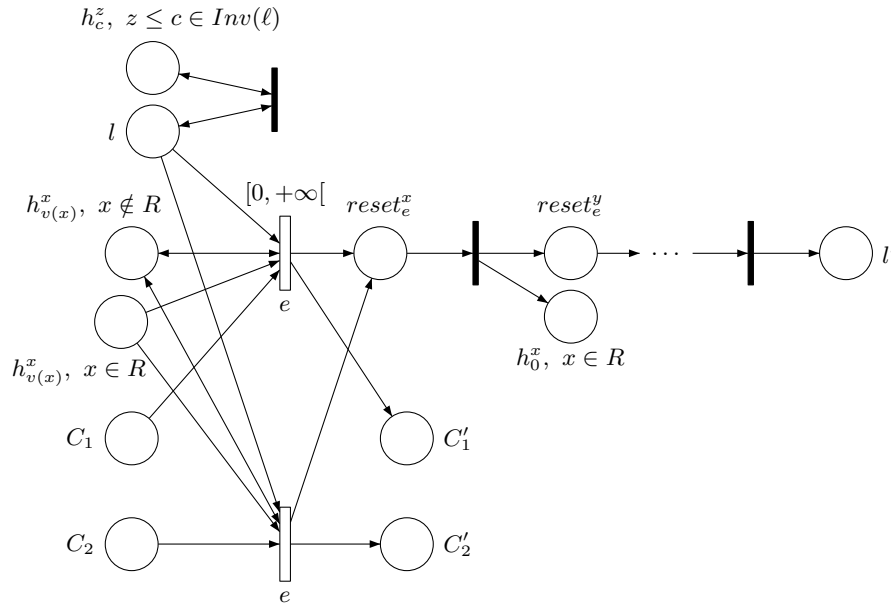


Fig. 15. Simulation of a transition

For instance, for the automaton \mathcal{A}_0 from Figure 1, we have four classes: $C_0 = \{l_0, 0 \leq x = y \leq 1\}$, $C_1 = \{l_1, 0 \leq x = y \leq 1\}$, $C_2 = \{l_1, x = 1 \wedge y = 0\}$ and $C_3 = \{l_2, 0 \leq y = x - 1\}$. The subnet in Figure 16 corresponds to transition c at point $(l_1, (1, 0))$ and class C_2 .

Consider the following run in \mathcal{A}_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_1^x, h_0^y

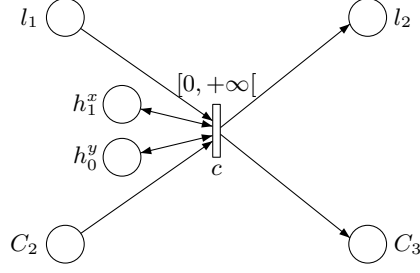


Fig. 16. Subnet of transition c

and C_1 are marked and t_0^x and t_0^y have been enabled for 1 t.u. Suppose that the transition t_0^x is fired, marking the place t_1^x , then without the input place C_2 the transition labelled c could be erroneously fired. Since C_2 is unmarked this firing is disabled.

6.2 Correctness proof

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 $\nu(t_{c_x}^x)$ 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 its associated $c = (c_x)_{x \in X}$ is *maximal*, *i.e.* for every x , either $c_x = K$ or $\nu(t_{c_x}^x) < 1$.

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 achieve the proof with an auxiliary lemma and the fact that \mathcal{R} is a weak timed 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 6. *Let \mathcal{A} be an automaton satisfying the conditions of theorem 1, 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 $\forall (\ell, v') \in \bar{r}, (\ell, v') \in C$.*

In particular, $(\ell, \lfloor v \rfloor) \in C$ (note that $\lfloor v \rfloor = \min_r$).

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 1. The case of a time step follows from the choice of $K = \infty$ which implies that given a region r , every item of $\overline{\text{succ}(r)}$ is reached by a time step from an item of \bar{r} . Recall that classes are closed by time elapsing. \square

Lemma 7. *The relation \mathcal{R} defined above is a weak timed 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 $d \neq 0$). Let us note $v' = v + d$. 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' can be reached, no transition related to an invariant condition in \mathcal{N} is enabled before d . 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 $(M_1, \nu_1 + d)$ 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 every other clock. 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 its associated 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 its associated $c = (c_x)_{x \in X}$ maximal. Let $C = (\ell, Z)$ be the class for which $M(C) = 1$, with $v \in Z$. From lemma 6, $(\ell, \lfloor v \rfloor)$ 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 t 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 with 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 $c = (c_x)_{x \in X}$ associated with (M, ν) 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 \bar{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} , condition (c) implies that a move $(\ell, v) \xrightarrow{e} (\ell', v')$ is also possible in \mathcal{A} . From the definition, $(M', \nu')\mathcal{R}(\ell', v')$. \square

7 Complexity results

This characterization leads to the the following complexity results.

Proposition 1. *Given a (injectively-labelled) 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}^{wtb} 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 \bar{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 fireable 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 \bar{r} and a edge e and check whether r is reachable and e is not fireable 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 adapt the construction given in [3] (appendix D) which reduces the acceptance problem for linear bounded Turing machines (LBTM) to the reachability problem for TA with restricted guards. In order to be self-content, we develop the complete proof.

The Turing machine $\mathcal{M} = \langle Q, \Sigma, q_0, q_F, Tr \rangle$ is defined by Q its finite set of states, $\Sigma = \{a, b\}$ its alphabet, q_0 (resp. q_F) its initial (resp. final) state and Tr the transitions of the machine. Each transition $\theta = (q, \alpha, \alpha', \delta, q')$ is defined by its current state q , the character to be read α , the character to be written α' , the move to be performed $\delta \in \{L, R\}$ and the next state q' . Let w_0 be a word of length n . We first build a TA $A_{\mathcal{M}, w_0}$ bisimilar to a TPN which reaches location *end* iff w_0 is accepted by \mathcal{M} .

The set of clocks of $A_{\mathcal{M}, w_0}$ is $\{x_i\}_{0 \leq i \leq n}$. Clock x_0 rules the behaviour of the simulation by letting exactly 1 time unit elapse before performing instantaneously the simulation of a machine transition. For $1 \leq i$, at the time of transition execution, the value of clock x_i is related to the value of the i^{th} cell: $x_i = 1$ iff the cell contains an a and $x_i \geq 2$ iff the cell contains an b .

The set of locations of $A_{\mathcal{M}, w_0}$ is:

$$\{(q, i) \mid q \in Q, 1 \leq i \leq n\} \cup \{init, end\} \cup \{(i, \theta, l) \mid 1 \leq i \leq n, \theta \in Tr, 1 \leq l \leq n\}$$

When the machine is in state q reading the i^{th} cell, the location of an automaton is (q, i) . When the machine changes its configuration to state q' reading the i^{th} cell by transition $\theta = (q, \alpha, \alpha', \delta, q')$, the automaton will first let 1 time unit elapse and then will successively visit $(i, \theta, 1), \dots, (i, \theta, n), (q', i')$ in zero time. Thus the invariant associated with every (q, i) is $x_0 \leq 1$ and the one associated with every (i, θ, l) is $x_0 \leq 0$.

Let us describe the automaton transitions related to such a transition (we do not give the labels as the TA is injectively-labelled). We define $s(i, \delta)$ by: if $\delta = L$ then $s(i, \delta) = i - 1$ else $s(i, \delta) = i + 1$.

- $(q, i) \xrightarrow{g, \{x_0\}} (i, \theta, 1)$ with $g = (x_0 \geq 1) \wedge (x_i \leq 1)$ (resp. $g = (x_0 \geq 1) \wedge (x_i \geq 2)$) if $\alpha = a$ (resp. $\alpha = b$)
- $(i, \theta, i) \xrightarrow{true, r} (i, \theta, i + 1)$ if $i < n$ and $(n, \theta, n) \xrightarrow{true, r} (q', s(n, \delta))$ with $r = \{x_i\}$ (resp. $r = \emptyset$) if $\alpha' = a$ (resp. $\alpha' = b$)
- $(i, \theta, l) \xrightarrow{x_l \leq 1, \{x_l\}} (i, \theta, l + 1)$ if $l < n \wedge i \neq l$ and $(i, \theta, n) \xrightarrow{x_n \leq 1, \{x_n\}} (q', s(i, \delta))$ if $i \neq n$ (in this case δ must be L).
- $(i, \theta, l) \xrightarrow{x_l \geq 2, \emptyset} (i, \theta, l + 1)$ if $l < n \wedge i \neq l$ and $(i, \theta, n) \xrightarrow{x_n \geq 2, \emptyset} (q', s(i, \delta))$ if $i \neq n$

The first step of the simulation consists in checking whether the i^{th} cell contains α whereas the other steps consists in resetting the clocks corresponding

to a cell containing an a . By this way, such clocks will have value 1 at the next stage of the simulation whereas the other ones will have a value at least 2. In the last step, the new location $(q', s(i, \delta))$ is reached.

It remains to “initialize” the clocks according to w_0 . This is performed through the transition $init \xrightarrow{x_0 \geq 1, \{x_0\} \cup r_{w_0}} (q_0, 1)$ with r_{w_0} being the positions of a in w_0 . Once again, the invariant associated with $init$ is $x_0 \leq 1$. At last, we add transitions in order to reach $end: (q_F, i) \xrightarrow{true, \{x_0\}} end$.

From the details of the construction, it is clear that the size of $A_{\mathcal{M}, w_0}$ is polynomial w.r.t. n . Furthermore it satisfies the conditions (a), (b) and (c). This is mainly due to the fact that in a configuration with $x_0 \in \{0, 1\}$ all the other clocks have integral values.

We build another TA $A'_{\mathcal{M}, w_0}$ by adding an edge $end \xrightarrow{x_0=0, \emptyset} end$.

If the LBTM \mathcal{M} does not accept the word w_0 , then the state end is not reachable and $A'_{\mathcal{M}, w_0}$ which behaves as $A_{\mathcal{M}, w_0}$, 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}^{wtb} is *PSPACE*-complete was proved implicitly within the proof above. \square

8 Conclusion

In this paper, we considered the subclass \mathcal{TA}^{wtb} of injectively-labelled TA such that a timed automaton \mathcal{A} is in \mathcal{TA}^{wtb} if and only if there is a TPN \mathcal{N} weakly timed bisimilar to \mathcal{A} . We obtained a characterization of this class, based on the region automaton associated with \mathcal{A} . To prove that our condition is necessary, we introduced the notion of uniform bisimulation between TA and TPNs. For the sufficiency, we proposed two constructions. From this characterization, we have proved that for the class \mathcal{TA}^{wtb} , the membership problem and the reachability problem are *PSPACE*-complete. Of course, checking this condition is therefore expensive but it should be noted that the syntactic subclass of TA proposed in [7] fulfills the condition. Hence there is a simpler (even if coarser) way to check a sufficient condition avoiding the complexity. Furthermore translation of a TA into a TPN provides an alternative method for verification with tools like TINA [11] or ROMEO [19] in case UPPAAL [25] or KRONOS [28] are inefficient. Besides the techniques introduced in this paper give some insight for use of the region automaton in order to obtain expressivity results.

References

1. P. A. Abdulla, P. Mahata and R. Mayr. Decidability of Zenoness, Syntactic Boundedness and Token-Liveness for Dense-Timed Petri Nets. *Foundations of Software Technology and Theoretical Computer Science, 24th International Conference, Chennai, India., LNCS* volume 3328 pages 58-70, december 2004.

2. P. A. Abdulla and A. Nylén. Timed Petri Nets and BQOs. *International Conference on Application and Theory of Petri nets. Newcastle. U.K., LNCS volume 2075* pages 53-70, june 2001.
3. 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.
4. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science B*, 126:183–235, 1994.
5. T. Aura and J. Lilius. A causal semantics for time Petri nets. *Theoretical Computer Science*, 243(1–2):409–447, 2000.
6. B. Bérard, F. Cassez, S. Haddad, D. Lime and O.H. Roux. Comparison of Different Semantics for Time Petri Nets. *Automated Technology for Verification and Analysis: Third International Symposium (ATVA 2005) Taipei, Taiwan, LNCS volume 3707* pages 293-307, october 2005.
7. B. Bérard, F. Cassez, S. Haddad, D. Lime and O.H. Roux. Comparison of the Expressiveness of Timed Automata and Time Petri Nets . *Third International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'05), Uppsala, Sweden, LNCS volume 3829*, pages 211-225, september 2005.
8. B. Bérard, F. Cassez, S. Haddad, D. Lime and O.H. Roux. When are timed automata weakly timed bisimilar to time Petri nets ? *25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005), Hyderabad, India LNCS volume 3821*, pages 273-284, december 2005.
9. B. Bérard, V. Diekert, P. Gastin and A. Petit. Characterization of the expressive power of silent transitions in timed automata. *Fundamenta Informaticae*, 36(2-3):145-182, 1998.
10. 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.
11. B. Berthomieu and F. Vernadat. Time Petri Nets Analysis with TINA. *Third International Conference on the Quantitative Evaluation of Systems (QEST 2006), Riverside, California, USA IEEE Computer Society Press*, pages 123–124, september 2006.
12. P. Bouyer. Forward Analysis of Updatable Timed Automata. *Formal Methods in System Design*, 24(3):281–320, May 2004.
13. P. Bouyer, C. Dufourd, E. Fleury and A. Petit. Updatable Timed Automata. *Theoretical Computer Science*, 321(2-3):291-345, Elsevier Science Publishers, August 2004.
14. P. Bouyer, S. Haddad and P.-A. Reynier. Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences. *33rd International Colloquium on Automata, Languages and Programming (ICALP'06), Venice, Italy, (LNCS) volume 4052*, pages 420–431, July 2006.
15. P. Bouyer, S. Haddad and P.-A. Reynier. Undecidability Results for Timed Automata with Silent Transitions. *Research Report LSV-07-12*, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2007-12.pdf, February 2007.
16. G. Bucci and A. Fedeli and L. Sassoli and E. Vicario. Timed State Space Analysis of Real-Time Preemptive Systems. *IEEE Trans. Software Eng.* 30(2): 97-111, 2004.
17. F. Cassez and O. H. Roux. Structural Translation of Time Petri Nets into Timed Automata. In Michael Huth, editor, *Workshop on Automated Verification of Critical Systems (AVoCS'04) London, UK*, Electronic Notes in Computer Science. Elsevier, volume 128, issue 6, 23 pages 145-160, august 2004.

18. D. L. Dill. Timing assumptions and verification of finite-state concurrent systems. In Proc. Workshop on Automatic Verification Methods for Finite State Systems, Grenoble, *LNCS* volume 407 pages 197-212, 1989.
19. G. Gardey, D. Lime, M. Magnin and O. H. Roux. ROMÉO: A Tool for Analyzing Time Petri Nets. In *Proc. of Computer Aided Verification, 17th International Conference, CAV 2005, Edinburgh, Scotland*, volume 3576 of *LNCS*, pages 418–423, 2005.
20. S. Haar, F. Simonot-Lion, L. Kaiser, and J. Toussaint. Equivalence of Timed State Machines and safe Time Petri Nets. In *Proceedings of 6th International Workshop on Discrete Event Systems (WODES 2002)*, Zaragoza, Spain, pages 119–126, 2002.
21. T.A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.
22. D. Lime and O. H. Roux. State class timed automaton of a time Petri net. In *Proceedings of the 10th International Workshop on Petri Nets and Performance models (PNPM 2003) Urbana-Champaign, Illinois, USA* IEEE Computer Society, pages 124-133 september 2003.
23. X. Nicollin and J. Sifakis. An Overview and Synthesis on Timed Process Algebras In *CAV'91, Aalborg, invited talk, LNCS 575, pages 376-398 July 1991*
24. P. M. Merlin. A study of the recoverability of computing systems. *PhD thesis, University of California, Irvine, CA*, 1974.
25. P. Pettersson and K. G. Larsen. UPPAAL2k. *Bulletin of the European Association for Theoretical Computer Science*, 70:40–44, February 2000.
26. L. Popova-Zeugmann and M. Heiner and I. Koch. Time Petri Nets for Modelling and Analysis of Biochemical Networks. *Fundamenta Informaticae (FI)*, 67(2005), pp 149-162, IOS-Press, Amsterdam.
27. C. Ramchandani. Analysis of asynchronous concurrent systems by timed Petri nets. *PhD thesis, Massachusetts Institute of Technology, Cambridge, MA*, 1974.
28. S. Yovine. Kronos: A Verification Tool for real-Time Systems. *Journal of Software Tools for Technology Transfer*, 1(1/2):123–133, October 1997.