Patricia Bouyer, Serge Haddad and Pierre-Alain Reynier

Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences

Research Report LSV-06-06

February 2006

Laboratoire Spécification et Vérification

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences

Patricia Bouyer[1], Serge Haddad[2], Pierre-Alain Reynier[1]

[1] LSV, CNRS & ENS Cachan, France
[2] LAMSADE, CNRS & Université Paris-Dauphine, France
{bouyer,reynier}@lsv.ens-cachan.fr,haddad@lamsade.dauphine.fr

Abstract. Timed Petri nets and timed automata are two standard models for the analysis of real-time systems. In this paper, we prove that they are incomparable for the timed language equivalence. Thus we propose an extension of timed Petri nets with read-arcs (RA-TdPN), whose coverability problem is decidable. We also show that this model unifies timed Petri nets and timed automata. Then, we establish numerous expressiveness results and prove that *zeno* behaviours discriminate between several sub-classes of RA-TdPNs. This has surprising consequences on timed automata, for instance on the power of non-deterministic clock resets.

1 Introduction

Timed automata (TA) [3] are a well-accepted model for representing and analyzing real-time systems: they extend finite automata with clock variables which give timing constraints on the behaviour of the system. Another prominent formalism for the design and analysis of discrete-event systems is the model of *Petri nets* (PN) [6]. Thus, in order to model concurrent systems with constraints on time, several timed extensions of PNs have been proposed as a possible alternative to TA.

Time Petri nets (TPN), introduced in the 70's, associate with each transition a time interval [4]. A transition can be fired if its enabling duration lies in its interval and time can elapse only if it does not disable some transition: firing of an enabled transition may depend on other enabled transitions even if they do not share any input or output place, which restricts a lot applicability of partial order methods in this model. Moreover, with this "urgency" requirement, all significant problems become undecidable for unbounded TPNs.

Timed Petri nets (TdPN), also called *timed-arc Petri nets*, associate with each arc an interval (or a bag of intervals) [14]. In TdPNs, each token has an age. This age is initially set to a value belonging to the interval of the arc which has produced it or set to zero if it belongs to the initial marking. Afterwards, ages of tokens evolve synchronously with time. A transition may be fired if tokens with age belonging to the intervals of its input arcs may be found in the current configuration. Note that "old" tokens may die (*i.e.* they cannot be used anymore for firing a transition but they remain in the place), and that conditions for firing transitions are thus local and do not depend on the global configuration of the system, like in PNs. This "lazy" behaviour has important consequences. Whereas the reachability problem is undecidable for TdPNs [14], the coverability problem [2] and some significant other ones are decidable [1]. Furthermore, TdPNs cannot be transformed into equivalent TA (for the language equivalence), since the untimed languages of the latter model are regular. However the question whether (bounded) TdPNs are more expressive than TA w.r.t. language equivalence was not known.

Our contributions. In this paper, we answer negatively this question, and propose an extension of TdPNs with *read-arcs*, yielding the model of *read-arc timed Petri nets* (RA-TdPN). This feature has already been introduced in the untimed framework [11] in order to define a more refined concurrent semantics for nets. However, in the untimed framework, for the interleaving semantics, they do not add any expressive power as they can be replaced by two arcs which check that a token is in the place and replace it immediately. First, we investigate the decidability of the coverability problem for the RA-TdPN model, and we prove that it remains decidable.

We then focus on the expressiveness of read-arcs, and prove quite surprising results. Indeed, we show that read-arcs add expressiveness to the model of TdPNs when considering languages of (possibly *zeno*) infinite timed words. On the contrary, we also prove that when considering languages of finite or non-*zeno* infinite timed words, read-arcs can be simulated and thus don't add any expressiveness to TdPNs.

Furthermore we investigate the relative expressiveness of several subclasses of RA-TdPNs, depending on the following restrictions: boundedness of the nets, integrality of constants appearing on the arcs, resets labelling post-arcs. We give a complete picture of their relative expressive power, and distinguish between three timed language equivalences (equivalence over finite words, or infinite words, or non-*zeno* infinite words) which, as before, lead to different results.

We finally establish that timed automata and bounded RA-TdPNs are language equivalent. From this result and former ones, we deduce several worthwhile expressiveness results, for instance we prove that non-determinism in clock resets adds expressive power to timed automata with integral constants over (possibly *zeno*) infinite timed words, which contrasts with the finite or non-*zeno* infinite timed words case [5]. If rational constants are allowed, this is no more the case: it should be emphasized that this latter result implies that the granularity of the automaton has to be refined if we want to remove non-deterministic updates while preserving expressiveness.

2 Read-Arc Timed Petri Nets

Preliminaries. If A is a set, A^* denotes the set of all finite words over A whereas A^ω denotes the set of infinite words over A. An interval I of $\mathbb{R}_{\geq 0}$ is a $\mathbb{Q}_{\geq 0}$- (resp. $\mathbb{N}$-) *interval* if its left endpoint belongs to $\mathbb{Q}_{\geq 0}$ (resp. $\mathbb{N}$) and its right endpoint belongs to $\mathbb{Q}_{\geq 0} \cup \{\infty\}$ (resp. $\mathbb{N} \cup \{\infty\}$). We denote by $\mathcal{I}$ (resp. $\mathcal{I}_\mathbb{N}$) the set of $\mathbb{Q}_{\geq 0}$- (resp. $\mathbb{N}$-) intervals of $\mathbb{R}_{\geq 0}$.

Bags. Given a set $\mathcal{E}$, $\mathsf{Bag}(\mathcal{E})$ denotes the set of mappings f from $\mathcal{E}$ to $\mathbb{N}$ s.t. the set $\mathsf{dom}(f) = \{x \in \mathcal{E} \mid f(x) \neq 0\}$ is finite. We note $\mathsf{size}(f) = \sum_{x \in \mathcal{E}} f(x)$. Let $x, y \in \mathsf{Bag}(\mathcal{E})$, then $y \leq x$ iff $\forall e \in \mathcal{E}, y(e) \leq x(e)$. If $y \leq x$, then $x - y \in \mathsf{Bag}(\mathcal{E})$ is defined by: $\forall e \in \mathcal{E}, (x-y)(e) = x(e) - y(e)$. For $d \in \mathbb{R}_{\geq 0}$ and $x \in \mathsf{Bag}(\mathbb{R}_{\geq 0})$ $x + d \in \mathsf{Bag}(\mathbb{R}_{\geq 0})$ is defined by $\forall \tau < d, (x+d)(\tau) = 0$ and $\forall \tau \geq d, (x+d)(\tau) = x(\tau - d)$. Let $x \in \mathsf{Bag}(\mathcal{E}_1 \times \mathcal{E}_2)$. The bags $\pi_i(x) \in \mathsf{Bag}(\mathcal{E}_i)$ for $i = 1, 2$ are defined by: for all $e_1 \in \mathcal{E}_1$, $\pi_1(x)(e_1) = \sum_{e_2 \in \mathcal{E}_2} x(e_1, e_2)$, and similarly for π_2.

Timed words and timed languages. Let Σ be a fixed finite alphabet s.t. $\varepsilon \notin \Sigma$ (ε is the silent action), we note $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$. A *timed word* w over Σ_ε (resp. Σ) is a finite or infinite sequence $w = (a_0, \tau_0)(a_1, \tau_1) \ldots (a_n, \tau_n) \ldots$ s.t. for every $i \geq 0$, $a_i \in \Sigma_\varepsilon$ (resp. $a_i \in \Sigma$), $\tau_i \in \mathbb{R}_{\geq 0}$ and $\tau_{i+1} \geq \tau_i$. The value τ_k gives the date at which action a_k occurs. We write *Duration*$(w) = \sup_k \tau_k$ for the duration of the timed word w. Since ε is a silent action, it can be removed in timed words over Σ_ε, and it naturally gives timed words over Σ. An infinite timed word w over Σ is said *zeno* whenever *Duration*(w) is finite. We denote by $\mathcal{TW}^*(\Sigma)$ (resp. $\mathcal{TW}^\omega(\Sigma)$, $\mathcal{TW}^{\omega_{nz}}$) the set of finite (resp. infinite, non-*zeno* infinite) timed words over Σ. A *timed language over finite (resp. infinite, non-zeno infinite) words* is a subset of $\mathcal{TW}^*(\Sigma)$ (resp. $\mathcal{TW}^\omega(\Sigma)$, $\mathcal{TW}^{\omega_{nz}}(\Sigma)$).

2.1 The Model of RA-TdPNs.

The *qualitative* component of a RA-TdPN is a Petri net extended with read-arcs. A read-arc checks for the presence of tokens in a place without consuming them. The *quantitative* part of a RA-TdPN is described by timing constraints on arcs. Roughly speaking, when firing a transition, tokens are consumed whose ages satisfy the timing constraints specified on the input arcs, and it is checked whether the constraints specified by the read-arcs are satisfied. Tokens are then produced according to the constraints specified on the output arcs.

Definition 1. *A* timed Petri net with read-arcs *(RA-TdPN for short) $\mathcal{N}$ is a tuple* $(P, m_0, T, \mathsf{Pre}, \mathsf{Post}, \mathsf{Read}, \lambda, \mathsf{Acc})$ *where:*

- *P is a finite set of places;*
- *$m_0 \in \mathsf{Bag}(P)$ denotes the initial marking of places;*
- *T is a finite set of transitions with $P \cap T = \emptyset$;*
- *Pre, the backward incidence mapping, is a mapping from T to $\mathsf{Bag}(\mathcal{I})^P$;*
- *Post, the forward incidence mapping, is a mapping from T to $\mathsf{Bag}(\mathcal{I})^P$;*
- *Read, the read incidence mapping, is a mapping from T to $\mathsf{Bag}(\mathcal{I})^P$;*
- *$\lambda : P \to \Sigma_\varepsilon$ is a labelling function;*
- *Acc is an accepting condition defined as a finite set of formulas generated by the grammar "$\mathsf{Acc} ::= \sum_{i=1}^n p_i \bowtie k \mid \mathsf{Acc} \wedge \mathsf{Acc}$" where $p_i \in P$, $k \in \mathbb{N}$ and $\bowtie \in \{\leq, \geq\}$.*

Since $\mathsf{Bag}(\mathcal{I})^P$ is isomorphic to $\mathsf{Bag}(P \times \mathcal{I})$, $\mathsf{Pre}(t)$, $\mathsf{Post}(t)$ and $\mathsf{Read}(t)$ may be also considered as bags. Given a place p and a transition t, if the bag $\mathsf{Pre}(t)(p)$ (resp. $\mathsf{Post}(t)(p)$, $\mathsf{Read}(t)(p)$) is non null then it defines a *pre-arc* (resp. *post-arc*, *read-arc*) of t connected to p.

A *configuration* ν of a RA-TdPN is an item of $\mathsf{Bag}(\mathbb{R}_{\geq 0})^P$ (or equivalently $\mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$). Intuitively, a configuration is a marking extended with age information for the tokens. We will write (p, x) for a token which is in place p and whose age is x. A configuration is then a finite sum of such pairs. Then a token (p, x) belongs to configuration

ν whenever $(p,x) \leq \nu$ (in terms of bags). The *initial configuration* $\nu_0 \in \mathsf{Bag}(\mathbb{R}_{\geq 0}^P)$ is defined as $\forall p \in P$, $\nu_0(p) = m_0(p) \cdot 0$ (there are $m_0(p)$ tokens of age 0 in place p).

We now describe the semantics of a $\mathsf{RA\text{-}TdPN}$ in terms of a transition system.

Definition 2 (Semantics of a $\mathsf{RA\text{-}TdPN}$). *Let $\mathcal{N} = (P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathit{Acc})$ be an RA-TdPN. Its semantics is the transition system $(Q, \Sigma_\varepsilon, \rightarrow)$ where $Q = \mathit{Bag}(\mathbb{R}_{\geq 0})^P$, and $\rightarrow$ is defined by:*

- *For $d \in \mathbb{R}_{\geq 0}$, $\nu \xrightarrow{d} \nu + d$ where the configuration $\nu + d$ is defined by $(\nu + d)(p) = \nu(p) + d$ for every $p \in P$.*
- *A transition t is firable from ν if for all $p \in P$, there exist $x(p), y(p) \in \mathit{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ such that* $\begin{cases} \pi_1(x(p)) + \pi_1(y(p)) \leq \nu(p), \\ \pi_2(x(p)) = \mathit{Pre}(t)(p) \text{ and } \pi_2(y(p)) = \mathit{Read}(t)(p), \\ \forall (\tau, I) \in \mathit{dom}(x(p)) \cup \mathit{dom}(y(p)),\ \tau \in I. \end{cases}$
  Let $z(p) \in \mathit{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ be such that $\begin{cases} \pi_2(z(p)) = \mathit{Post}(t)(p), \\ \forall (\tau, I) \in \mathit{dom}(z(p)),\ \tau \in I. \end{cases}$
  Define for every $p \in P$, $\nu'(p) = \nu(p) - x(p) + z(p)$. Then $\nu \xrightarrow{\lambda(t)} \nu'$.

A *path* in the $\mathsf{RA\text{-}TdPN}$ $\mathcal{N}$ is a sequence $\nu_0 \xrightarrow{d_1} \nu_1' \xrightarrow{t_1} \nu_1 \xrightarrow{d_2} \nu_2' \xrightarrow{t_2} \nu_2 \ldots$ in the above transition system. A *timed transition sequence* is a (finite or infinite) timed word over alphabet T, the set of transitions of $\mathcal{N}$. A *firing sequence* is a timed transition sequence $(t_1, \tau_1)(t_2, \tau_2)\ldots$ such that $\nu_0 \xrightarrow{\tau_1} \nu_1' \xrightarrow{t_1} \nu_1 \xrightarrow{\tau_2 - \tau_1} \nu_2' \xrightarrow{t_2} \nu_2 \ldots$ is a path. If $(p,x) \leq \nu$ is a token of a configuration ν, it is a *dead token* whenever for every interval I labelling a pre- or a read-arc of p, x is above I.

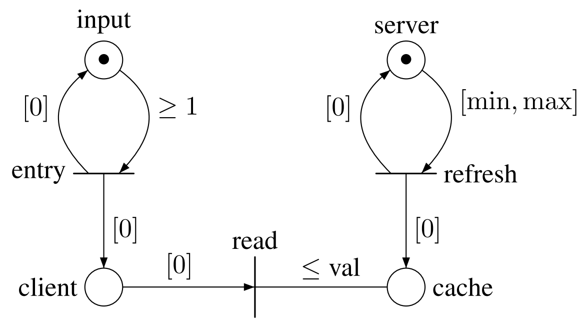
Petri nets can be considered as language acceptors. The timed word which is read along a path $\nu_0 \xrightarrow{d_1} \nu_1' \xrightarrow{t_1} \nu_1 \xrightarrow{d_2} \nu_2' \xrightarrow{t_2} \nu_2 \ldots$ is the projection over Σ of the timed word $(\lambda(t_1), d_1)(\lambda(t_2), d_1 + d_2)\ldots$

If ν is a configuration of $\mathcal{N}$, we say that ν satisfies the accepting condition $\sum_{i=1}^n p_i \bowtie k$ whenever $\sum_{i=1}^n \mathsf{size}(\nu(p_i)) \bowtie k$, and the satisfaction relation for conjunctions of accepting conditions is defined in a natural way. A finite path in $\mathcal{N}$ is accepting if it ends in a configuration satisfying one of the formulas of Acc. An infinite path is accepting if every formula of Acc is satisfied infinitely often along the path (Acc is then viewed as a generalized Büchi condition). We note $\mathcal{L}^*(\mathcal{N})$ (resp. $\mathcal{L}^\omega(\mathcal{N})$, $\mathcal{L}^{\omega_{nz}}(\mathcal{N})$ the set of finite (resp. infinite, non-*zeno* infinite) timed words accepted by $\mathcal{N}$.

Two $\mathsf{RA\text{-}TdPN}$s $\mathcal{N}$ and $\mathcal{N}'$ are $*$*-equivalent* (resp. ω*-equivalent*, ω_{nz}*-equivalent*) whenever $\mathcal{L}^*(\mathcal{N}) = \mathcal{L}^*(\mathcal{N}')$ (resp. $\mathcal{L}^\omega(\mathcal{N}) = \mathcal{L}^\omega(\mathcal{N}')$, $\mathcal{L}^{\omega_{nz}}(\mathcal{N}) = \mathcal{L}^{\omega_{nz}}(\mathcal{N}')$). These equivalences naturally extend to subclasses of $\mathsf{RA\text{-}TdPN}$s. In the following, we will use notations like "$\{*, \omega, \omega_{nz}\}$-equivalence" to mean the three equivalences altogether. *Idem* for "$\{*, \omega_{nz}\}$-equivalence" and other combinations.

Notations. Read-arcs are represented by undirected arcs. We use shortcuts to represent bags: for all $I \in \mathcal{I}$, I holds for the bag $1 \cdot I$, $[a]$ is for the interval $[a,a]$. We may write intervals as constraints, *eg* "$\leq a$" is for the interval $[0,a]$. A bag n represents the bag $n \cdot \mathbb{R}_{\geq 0}$, and no bag on an arc means that this arc is labelled by the bag $1 \cdot \mathbb{R}_{\geq 0}$.

Example 1. An example of $\mathsf{RA\text{-}TdPN}$ is depicted on the next figure. This net models an information provided by a server and asynchronously consulted by clients (transition "read"). Since the information may be obsolete with validity duration "val", the server periodically refreshes the value, but the frequency of this refresh may vary depending on the workload of the server (transition "refresh"). The admission control ensures that at least one time unit elapses between two client arrivals (transition "entry"). Note the interest of the read-arc between "cache" and "read": when transition "read" is fired the age of the token of place "cache" is not reinitialized.



Subclasses of RA-TdPNs. We define several natural subclasses of RA-TdPNs.

Definition 3. *Let $\mathcal{N} = (P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathit{Acc})$ be an RA-TdPN. It is*

- *a* timed Petri net *(TdPN for short)*[3] *if for all $t \in T$, $\mathit{size}(\mathit{Read}(t)) = 0$,*
- integral *if all intervals appearing in bags of $\mathcal{N}$ are in $\mathcal{I}_{\mathbb{N}}$,*
- 0-reset *if for all $t \in T$, for all $p \in P$, $I \neq [0, 0] \Rightarrow I \notin \mathit{dom}(\mathit{Post}(t)(p))$,*
- k-bounded *if all configurations ν appearing along a firing sequence of $\mathcal{N}$ are such that for every place $p \in P$, $\mathit{size}(\nu(p)) \leq k$,*
- bounded *if there exists $k \in \mathbb{N}$ such that $\mathcal{N}$ is k-bounded,*
- safe *if it is 1-bounded.*

2.2 The Coverability Problem.

Let $\mathcal{N}$ be an RA-TdPN with initial configuration ν_0. Let N be a finite set of configurations of $\mathcal{N}$ where all ages of tokens are rational. We note $N^{\uparrow}$ the upward closure of N, *i.e.* the set $\{\nu \mid \exists \nu' \in N,\ \nu' \leq \nu\}$.

The *coverability problem* for $\mathcal{N}$ and set of configurations N asks whether there exists a path in $\mathcal{N}$ from ν_0 to some $\nu \in N^{\uparrow}$. We obtain the following result.

Theorem 1. *The coverability problem is decidable for RA-TdPNs.*

In order to solve the coverability problem, we introduce the notion of region for a net. A *region* is a classical object used in the framework of timed automata for representing an infinite set of configurations [3], that we can extend to RA-TdPNs. Such a construction has been done for example in [10] for TdPNs, and has been used recently in several other contexts [12, 13, 9].

[3] This is the standard model, as defined in [14].

Regions of RA-TdPNs. Let $\mathcal{N} = (P, m_0, T, \mathsf{Pre}, \mathsf{Post}, \mathsf{Read}, \lambda, \mathsf{Acc})$ be a net where the bounds of intervals are in $\mathbb{N} \cup \{\infty\}$. Let N be a finite set of markings with integral ages. There is no loss of generality in assuming that bounds of the net and that values of ages are integers (otherwise we will refine the granularity of the regions). Note $\max$ the maximal integer appearing in the bounds of intervals of the net and in the ages of the tokens in the configurations of N.

Definition 4. *A region $\mathcal{R}$ for $\mathcal{N}$ is a sequence $a_0a_1 \dots a_na_\infty$ where $n \in \mathbb{N}$, for all $0 \leq i \leq n$, $a_i \in \mathit{Bag}(P \times \{0, 1, \dots, \max\})$ with $\mathit{size}(a_i) \neq 0$ if $i \neq 0$, and $a_\infty \in \mathit{Bag}(P \times \{\infty\})$.*

We first informally explain the semantics of a region. Given the bag of tokens defining a configuration, we partition it as follows. We put in a_∞ all the tokens whose ages are strictly greater than $\max$ and forget their ages. We then put in a_0 the tokens with integral ages and indicate the age. Finally, we order the remaining tokens depending on the fractional part of their ages in $a_1, \dots, a_n$ and we forget their fractional part. Hence n is the number of different fractional values.

We now define more formally the semantics of the regions. Let ϕ be the mapping from $\mathbb{R}_{\geq 0}$ to $\{0, 1, \dots, \max, \infty\}$ defined by: if $x > \max$ then $\phi(x) = \infty$ else $\phi(x) = \lfloor x \rfloor$. We extend ϕ to $P \times \mathbb{R}_{\geq 0}$ by $\phi((p, x)) = (p, \phi(x))$ and to $\mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$ by linearity.

Let $\mathcal{R} = a_0a_1 \dots a_na_\infty$ be a region then $[\mathcal{R}]$ is a set of configurations ν such that there exist $\nu_1, \nu_2, \dots, \nu_n, \nu_\infty$ belonging to $\mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$ with:

- $\nu = a_0 + \nu_1 + \nu_2 + \dots + \nu_n + \nu_\infty$,
- $\forall 1 \leq i \leq n$, $\phi(\nu_i) = a_i$, and $\phi(\nu_\infty) = a_\infty$,
- $\forall 1 \leq i \leq n$, $\forall (p, x) + (q, y) \leq \nu_i$, $0 < x - \lfloor x \rfloor = y - \lfloor y \rfloor$,
- $\forall 1 \leq i < j \leq n$, $\forall (p, x) \leq \nu_i$, $(q, y) \leq \nu_j$, $x - \lfloor x \rfloor < y - \lfloor y \rfloor$.

Note that every configuration ν belongs to a single region, we note it $\mathcal{R}(\nu)$, and that if $\nu \in N$, then $[\mathcal{R}_\nu] = \{\nu\}$. The original coverability problem thus reduces to the coverability problem for finitely many regions, which reduces to solving the coverability problem for a single region $\mathcal{R}_f$.

Lemma 1. *The region partitioning is a time-abstract bisimulation.*

Decidability of the coverability problem. We can now prove Theorem 1

Proof. We first notice that, given two regions $\mathcal{R} = a_0a_1 \dots a_na_\infty$ and $\mathcal{R}' = a'_0a'_1 \dots a'_ma'_\infty$, one can check whether $[\mathcal{R}]^\uparrow \subseteq [\mathcal{R}']^\uparrow$: the necessary and sufficient conditions are $a_0 \geq a'_0$, $a_\infty \geq a'_\infty$ and the existence of a strictly increasing mapping ψ from $\{1, \dots, n'\}$ into $\{1, \dots, n\}$ such that for every $1 \leq i \leq n'$, $a_{\psi(i)} \geq a'_i$.

We define a preorder between regions by $\mathcal{R} \leq \mathcal{R}'$ iff $[\mathcal{R}']^\uparrow \subseteq [\mathcal{R}]^\uparrow$. Then, using Higman's lemma [7], we can show that this is a well quasi-order, *i.e.* for every infinite sequence of regions $\{\mathcal{R}_i\}_{i \in \mathbb{N}}$ there exist $i < j$ such that $\mathcal{R}_i \leq \mathcal{R}_j$.

The algorithm for solving the coverability problem then consists in computing iteratively the predecessors (by time elapsing steps and by discrete steps) of $[\mathcal{R}_f]^\uparrow$, in proving that it is a finite union of upward closures of regions, and in stopping the exploration when a region is computed which is larger (for the preorder $\leq$) than an already computed region. The termination criterium is correct as all configurations reachable from $[\mathcal{R}']^{\uparrow}$ is also reachable from $[\mathcal{R}]^{\uparrow}$ whenever $\mathcal{R} \leq \mathcal{R}'$. The above computation always terminate because "$\leq$" is a well quasi-order.

It remains to explain how we compute time and discrete predecessors of the upward closure of a region $\mathcal{R} = a_0 a_1 \ldots a_n a_\infty$.

Time predecessors. If a_0 contains a token $(p, 0)$, there is no time predecessor of $[\mathcal{R}]^{\uparrow}$. Otherwise we choose which tokens will first reach (in the past) their integral part. It could be the tokens of a_1, a bag of tokens $b_\infty \leq a_\infty$ or both. We only illustrate this last case.

The above-mentioned time predecessor is $[\mathcal{R}']^{\uparrow}$ where $\mathcal{R}' = a'_0 a'_1 \ldots a'_{n'} a'_\infty$ is obtained as follows (we assume that $n \geq 1$, the other case is similar).

- $a'_\infty = a_\infty - b_\infty$,
- $a'_0 = a_1 + c_\infty$ where c_∞ is obtained from b_∞ by setting the age of each token to max,
- $\forall 1 \leq i \leq n-1, a'_i = a_{i+1}$,
- if $\mathsf{size}(a_0) = 0$, then $n' = n - 1$, otherwise $n' = n$ and $a'_n = b_0$, where b_0 is obtained from a_0 by decrementing by 1 the (integral) age of each token.

Discrete predecessors. We pick a transition t. Note that given an interval I of the net and a token (p, x) belonging to some a_i for $i \in \{0, 1, \ldots, n, \infty\}$, we can compute whether, given a configuration belonging to that region, the corresponding token belongs to I. By property of the regions, this is independent of the choice of the configuration. We then write $(i, x) \vDash I$.

We choose bags of tokens $\mathsf{post}_i, \mathsf{read}^+_i \in \mathsf{Bag}(P \times \{0, 1, \ldots, \max\} \times \mathcal{I})$ for every $i \in \{0, 1, \ldots, n\}$ and $\mathsf{post}_\infty, \mathsf{read}^+_\infty \in \mathsf{Bag}(P \times \mathcal{I})$ s.t.

- for all $(p, x, I) \leq \mathsf{post}_i + \mathsf{read}^+_i$, $(i, x) \vDash I$,
- for all $i \in \{0, 1, \ldots, n, \infty\}$, $\pi_{1,2}(\mathsf{post}_i) + \pi_{1,2}(\mathsf{read}^+_i) \leq a_i$, where $\pi_{1,2}$ projects bags onto their two first components.
- $\sum_i \pi_{1,3}(\mathsf{post}_i) \leq \mathsf{Post}(t)$,
- $\sum_i \pi_{1,3}(\mathsf{read}^+_i) \leq \mathsf{Read}(t)$.

We build an intermediate region $\mathcal{R}' = a'_0 a'_1 \ldots a'_{n'} a'_\infty$ by substracting $\pi_{1,2}(\mathsf{post}_i)$ from a_i for every i and deleting the item in resulting sequence if its size is null (for $1 \leq i \leq n$). Finally, a region $\mathcal{R}'' = a''_0 a''_1 \ldots a''_{n''} a''_\infty$ is a predecessor if there exist bags of tokens $\mathsf{pre}_i, \mathsf{read}^-_i \in \mathsf{Bag}(P \times \{0, 1, \ldots, \max\} \times \mathcal{I})$ for every $i \in \{0, 1, \ldots, n''\}$, $\mathsf{pre}_\infty, \mathsf{read}^-_\infty \in \mathsf{Bag}(P \times \{\infty\} \times \mathcal{I})$ and a strictly increasing mapping ψ from $\{1, \ldots, n'\}$ into $\{1, \ldots, n''\}$ s.t.

- for all $(p, x, I) \leq \mathsf{pre}_i + \mathsf{read}^-_i$, $(i, x) \vDash I$,
- $a''_0 = a'_0 + \pi_{1,2}(\mathsf{pre}_0) + \pi_{1,2}(\mathsf{read}^-_0)$,
  $a''_\infty = a'_\infty + \pi_{1,2}(\mathsf{pre}_\infty) + \pi_{1,2}(\mathsf{read}^-_\infty)$,
  for every $i \in \{1, \ldots, n''\}$, if there exists j s.t. $\psi(j) = i$ then $a''_i = a'_j + \pi_{1,2}(\mathsf{pre}_i) + \pi_{1,2}(\mathsf{read}^-_i)$, otherwise $a''_i = \pi_{1,2}(\mathsf{pre}_i) + \pi_{1,2}(\mathsf{read}^-_i)$,

– $\sum_i \pi_{1,3}(\mathsf{pre}_i) = \mathsf{Pre}(t)$,
– $\sum_i \pi_{1,3}(\mathsf{read}_i^-) + \sum_i \pi_{1,3}(\mathsf{read}_i^+) = \mathsf{Read}(t)$.

We have thus described an algorithm for deciding the coverability problem of RA-TdPNs. □

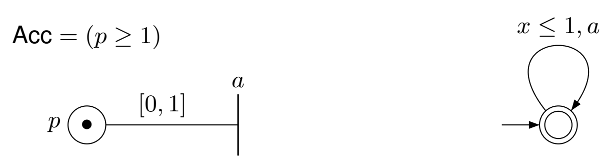
3 Relative Expressiveness of Subclasses of RA-TdPNs

In this section, we thoroughly study the relative expressiveness of subclasses of RA-TdPNs, by distinguishing whether they are bounded, integral, 0-reset, or whether they can be expressed without read-arcs. Surprisingly the results depend on the language equivalence we consider, and whereas finite timed words and non-*zeno* infinite timed words do not distinguish between (integral, bounded) 0-reset TdPNs and (integral, bounded) RA-TdPNs, *zeno* infinite timed words lead to a lattice of strict inclusions that will be summarized in Subsection 3.5.

3.1 Two Discriminating Timed Languages

We design two timed languages which distinguish between several subclasses of RA-TdPNs. Notice that these two languages are *zeno*. This remark will be important later on in this section.

The timed language L_1**.** The RA-TdPN $\mathcal{N}_1$ of Figure 1(a) (with a single accepting Büchi condition $p \geq 1$) is a 0-reset, integral and bounded RA-TdPN which recognizes the timed language $L_1 = \{(a, \tau_1) \dots (a, \tau_n) \dots \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots \leq 1\}$. Note that this timed language is also recognized by the TA $\mathcal{A}_1$ of Figure 1(b) (see Section 4 for a formal definition of TA).



(a) A RA-TdPN $\mathcal{N}_1$ recognizing L_1 (b) A TA $\mathcal{A}_1$ recognizing L_1

Fig. 1. A language L_1 not recognized by any TdPN

Lemma 2. *The timed language L_1 is recognized by no TdPN.*

Proof. Assume that there is a TdPN $\mathcal{N}$ which recognizes $L_1 = \{(a, \tau_1) \dots (a, \tau_n) \dots \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots \leq 1\}$. Let us denote d the lcm of denominators constants appearing in the intervals of $\mathcal{N}$. Pick an infinite word $w = (a, \tau_1)(a, \tau_2) \dots (a, \tau_n) \dots$ such that every $i \geq 1$, $1 - 1/2d < \tau_i < \tau_{i+1} < 1$. The word w is accepted by $\mathcal{N}_1$, there is thus an infinite firing sequence $\sigma = \sigma_1(t_1, \tau_1)\sigma_2(t_2, \tau_2) \dots \sigma_n(t_n, \tau_n) \dots$ over Σ_ε

which is an accepting run of $\mathcal{N}$ and where all transitions of σ_i are labelled by ε whereas the transitions t_i are labelled by a.

The set *Tok* of tokens part of the initial marking or produced along the sequence σ_1 is finite. Hence, there is an integer n such that tokens in *Tok* are not used for firing transitions in the sequence $(t_{n-1}, \tau_{n-1})\sigma_n(t_n, \tau_n)\ldots$ Since $\tau_{n-1} < \tau_n$, there is a suffix $(t'_0, \tau')(t'_1, \tau_n)\ldots(t'_k, \tau_n)(t_n, \tau_n)$ of the timed transition sequence $(t_{n-1}, \tau_{n-1})\sigma_n(t_n, \tau_n)$ with $\tau' < \tau_n$ (k may be equal to 0). We note σ' the finite prefix of σ up to (t'_0, τ'). We will prove that the infinite sequence $\sigma'' = \sigma'(t'_1, \tau_n+1/2d)\ldots(t'_k, \tau_n+1/2d)(t_n, \tau_n+1/2d)(\sigma_{n+1}+1/2d)(t_{n+1}, \tau_{n+1}+1/2d)\ldots$ is a firing sequence of $\mathcal{N}$ ($\sigma_{n+1}+1/2d$ is the timed transition sequence obtained from σ_{n+1} by delaying firings of transitions by $1/2d$ time units). To that aim, we will analyse the age of tokens used for firing a transition of $(t'_1, \tau_n)\ldots(t'_k, \tau_n)(t_n, \tau_n)\sigma_{n+1}(t_{n+1}, \tau_{n+1})\ldots$ in the original timed transition sequence σ, and we will show that (when necessary) we can modify the initial age of these tokens in order for the timed transition sequence σ'' to be firable.

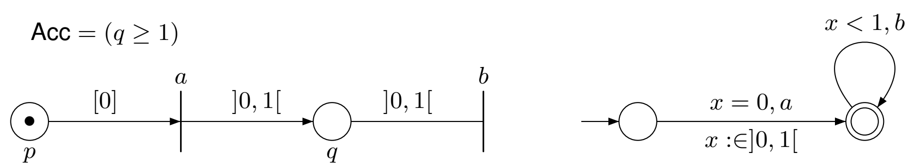
We pick a token in place p which, along σ, is produced by transition t and used for firing transition t' (which is some of the t_i's (for $i \geq 1$) or a transition appearing in some of the σ_i's (for $i \geq 2$)). Because of our choice of n, t occurs at date τ with $\tau \geq \tau_1$. If t is a transition of $(t'_1, \tau_n)\ldots(t'_k, \tau_n)(t_n, \tau_n)\sigma_{n+1}(t_{n+1}, \tau_{n+1})\ldots$, then we do not modify its initial age along σ'' since t and t' will be separated by the same delay along σ and along σ'', and the token p can be used similarly in σ and in σ''.

Otherwise $\tau \leq \tau' < \tau_n$. We set $v = \tau_n - \tau$: then $0 < v < 1/2d$. Let us call I^- the interval of $\mathsf{Post}(t)(p)$ associated with the production of the token and I^+ the interval of $\mathsf{Pre}(t')(p)$ associated with the consumption of the token. We first notice that I^- and I^+ cannot be both singletons: assume $I^- = [h/d, h/d]$ and $I^+ = [k/d, k/d]$ with $h, k \in \mathbb{N}$, then $k/d = h/d + v$, which is impossible since $0 < v < 1/2d$.

- We assume $I^- = [h/d, h/d]$ and $I^+ = (k/d, k'/d)$ with $k < k'$ (the brackets defining I^+ are either "strict" or "non-strict"). The age of the token when it is consumed by transition t' is $h/d + v \in I^+$, thus $h < k'$ and $h/d + v + 1/2d \in I^+$ (since $0 < v < 1/2d$). In this case, we do not change the initial age of the token for firing the timed transition sequence σ''.
- We assume $I^- = (h/d, h'/d)$ and $I^+ = [k/d, k/d]$ with $h < h'$. The age of the token when transition t' is fired along σ is $k/d - v \in I^-$. Thus, $h < k$ and $k/d - v - 1/2d \in I^-$ since $0 < v < 1/2d$. For firing the sequence σ'', we thus change the initial age of the token down to $k/d - v - 1/2d$.
- We assume $I^- = (h/d, h'/d)$ and $I^+ = (k/d, k'/d)$ with $h < h'$ and $k < k'$. We note α the initial age of the token when transition t is fired: $\alpha + v(\leq k'/d)$ is its age when the token is consumed for firing transition t'. If $\alpha + v < k'/d - 1/2d$, we do not modify its initial age. Assume that $\alpha \geq k'/d - 1/2d - v$: then $(k'-1)/d < \alpha < k'/d$, and thus $h \leq k' - 1 < h'$. Choose as new initial age $(k'-1)/d + 1/4d$ for the token: then $(k'-1)/d + 1/4d \in I^-$ and $(k'-1)/d + 1/4d + v \in I^+$.

With these new initial ages for the tokens, the timed transition sequence σ'' is firable, and accepts the timed word $(a, \tau_1)\ldots(a, \tau_{n-1})(a, \tau_n + 1/2d)(a, \tau_{n+1} + 1/2d)\ldots$. Moreover, the markings along the run accepting the initial word and the above word are the same, they are thus both accepted by $\mathcal{N}$. However this timed word should not be accepted by $\mathcal{N}$ as it is not accepted by $\mathcal{N}_1$. Thus, as $\tau_n + 1/2d > 1$, there is no classical TdPN which recognizes L_1. $\square$

The timed language L_2**.** The RA-TdPN $\mathcal{N}_2$ of Figure 2(a) is an integral bounded RA-TdPN which recognizes the timed language $L_2 = \{(a,0)(b,\tau_1)\ldots(b,\tau_n)\ldots \mid \exists\tau < 1 \text{ s.t. } 0 \leq \tau_1 \leq \ldots \leq \tau_n \leq \ldots < \tau\}$. Note, and that will be used in Section 4, that the timed language L_2 is also recognized by the TA of Figure 2(b) (which uses a non-deterministic reset of clock x in the intervals $]0,1[$).



(a) A RA-TdPN $\mathcal{N}_2$ recognizing L_2 (b) A TA $\mathcal{A}_2$ recognizing L_2

Fig. 2. A language L_2 not recognized by any 0-reset integral RA-TdPN

Lemma 3. *The timed language L_2 is recognized by no 0-reset integral RA-TdPN.*

Proof. Assume that the timed language L_2 is recognized by the 0-reset integral RA-TdPN $\mathcal{N}$. Pick a word $w = (a,0)(b,\tau_1)\cdots(b,\tau_i)\ldots$ of L_2, with $0 < \tau_1 \leq \tau_2 \leq \ldots \leq \tau_i \leq \ldots < \tau$ and $\lim_{i\to\infty}\tau_i = \tau$. We note σ an accepting firing sequence in $\mathcal{N}$ for w.

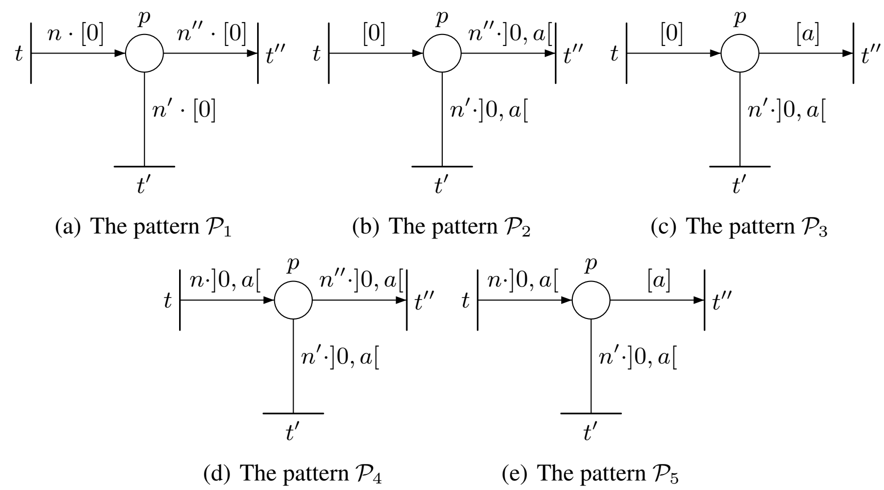
We write $\sigma = \sigma_1 d \sigma_2$ where σ_1 is an instantaneous firing sequence and $0 < d$. We claim that $\sigma' = \sigma_1(d+1-\tau)\sigma_2'$ where σ_2' is obtained from σ_2 by delaying $1-\tau$, is a firing sequence of $\mathcal{N}$. Let us select an occurrence of a transition t fired in σ_2 and a token read or consumed by t corresponding to an interval I. If the token has been produced by a transition fired in σ_2, then it has the same age in σ_2'. If the token is an initial token or has been produced by σ_1, then its age x is such that $0 < d \leq x < \tau < 1$, thus $]0,1[\subseteq I$ (because the net $\mathcal{N}$ is integral and 0-reset). The age of this token when it is checked for firing t in σ_2' is $x+1-\tau$ and satisfies $0 < x+1-\tau < 1$. Thus the same occurrence of t is firable in σ_2'.

Since the untimed firing sequences of σ and σ' are equal, σ' is an accepting sequence. The timed word which is read on σ' is $w' = (a,0)(b,\tau_1+1-\tau)\ldots(b,\tau_i+1-\tau)\ldots$ with $\lim_{i\to\infty}\tau_i+1-\tau = 1$. Thus, $w' \notin L_2$, which contradicts the assumption that it is accepted by $\mathcal{N}$, and thus by $\mathcal{N}_2$. $\square$

3.2 Normalization of RA-TdPNs

We present a transformation of RA-TdPNs which preserves both languages over finite and (*zeno* or non-*zeno*) infinite words, as well as boundedness and integrality of the nets. This construction transforms the net by imposing strong syntactical conditions on places, which will simplify further studies of RA-TdPNs. This transformation is somehow close to one-dimensional regions of [8], and records ages of tokens and how time elapses.

Proposition 1. *For any* RA-TdPN $\mathcal{N}$*, we can effectively construct a* RA-TdPN $\mathcal{N}'$ *which is* $\{*, \omega_{nz}, \omega\}$*-equivalent to* $\mathcal{N}$*, and in which all places are configured as one of the five patterns depicted in Figure 3, which reads as: "there is an* a *such that the place is connected to at most one post-arc, at most one pre-arc and possibly several read-arcs, with bags as specified on the figure". Moreover the construction preserves boundedness and integrality properties.*



(a) The pattern $\mathcal{P}_1$ (b) The pattern $\mathcal{P}_2$ (c) The pattern $\mathcal{P}_3$

(d) The pattern $\mathcal{P}_4$ (e) The pattern $\mathcal{P}_5$

Fig. 3. The five normalized patterns for an RA-TdPN.

In order to avoid difficulties with the initial marking, we first perform a straightforward transformation of the net. We add a place p_{init} with initially one token in it and a transition t_{init} labelled by ε, whose single pre-arc labelled by $[0]$ is connected to p_{init} and whose post-arcs correspond to the initial marking, *i.e.* for all $p \in P$, $\mathsf{Post}(t)(p) = m_0(p) \cdot [0]$. All other places are initially unmarked. Finally we add $p_{init} = 0$ to the acceptance conditions. It is trivial that this transformation does not modify all accepted languages. In the sequel, we assume that the net has been transformed in such a way and we apply the next transformations on every place except p_{init}.

For proving Proposition 1 we proceed in three steps, and successively construct a net which satisfies syntactical restrictions (1), (2) and (3) below:

(1) For every place, there exists a finite set of pairwise disjoint intervals $\{I_k\}_{1 \le k \le K}$ such that every arc connected to this place has a bag of the form $\sum_{1 \le k \le K} n_k \cdot I_k$. Moreover, any I_k is either $[a]$ or $]a, b[$ with $a \in \mathbb{Q}_{\ge 0}$ and $b \in \mathbb{Q}_{>0} \cup \{\infty\}$.

(2) For every place,
   - either every arc connected to the place has a bag of the form $n \cdot [0]$,
   - or there exists $a \in \mathbb{Q}_{>0} \cup \{\infty\}$ such that read- (resp. post-, pre-)arcs have bags of the form $n\cdot]0, a[$ (resp. or $[0]$, or $[a]$).

(3) Every place is configured as one of the five patterns depicted on Figure 3.

In all following lemmas, the equivalence mentioned is the $\{*, \omega, \omega_{nz}\}$-equivalence, which means that the constructions are correct for finite and infinite timed words. Let us fix an $\mathsf{RA\text{-}TdPN}$ $\mathcal{N}$.

Lemma 4. *We can build a* $\mathsf{RA\text{-}TdPN}$ $\mathcal{N}_1$*, equivalent to* $\mathcal{N}$*, and satisfying restriction* (1).

Proof. Let p be a place of $\mathcal{N}$. We first define the set $SI_p = \{I_k\}_{1 \leq k \leq K}$. We consider the finite bounds of intervals which occur in the bag of some arc connected to p, say $\{a_1, \ldots, a_m\}$ with $i < j \Rightarrow a_i < a_j$. The set SI_p is then defined by $\{[a_1, a_1],]a_1, a_2[, \ldots,]a_{m-1}, a_m[, [a_m, a_m],]a_m, \infty[\}$. W.l.o.g. we assume that $a_1 = 0$. Moreover, to ease the presentation, we define $a_{m+1} = \infty$ and set $a_{m+1} - a_m = \infty$. Note that for every interval $I_k \in SI_p$ and for every interval I which occurs in the bag of some arc connected to p, we have either $I \cap I_k = \emptyset$ or $I \cap I_k = I_k$.

We will iteratively apply the following transformation to the transitions connected to p. Let us pick a transition t connected to p by an arc whose associated bag is $x = \sum_{1 \leq k' \leq K'} n_{k'} \cdot J_{k'}$. We will duplicate the transition t with the same arcs and the same bags except the one which is concerned by the transformation. We note such a transition t_ϕ, where ϕ is a mapping from $\{1, \ldots, K\} \times \{1, \ldots, K'\}$ to $\mathbb{N}$ such that $I_k \cap J_{k'} = \emptyset \Rightarrow \phi(k, k') = 0$ and $\sum_{1 \leq k \leq K} \phi(k, k') = n_{k'}$. The modified bag is defined by:

$$\begin{aligned} x_\phi &= \textstyle\sum_{1 \leq k' \leq K'} \sum_{1 \leq k \leq K} \phi(k, k') \cdot I_k \cap J_{k'} \\ &= \textstyle\sum_{1 \leq k' \leq K'} \sum_{1 \leq k \leq K} \phi(k, k') \cdot I_k \\ &= \textstyle\sum_{1 \leq k \leq K} (\sum_{1 \leq k' \leq K'} \phi(k, k')) \cdot I_k. \end{aligned}$$

This transformation is valid. Indeed given any choice of an item $b \in \mathsf{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ with $\pi_2(b) = x$ there exists a mapping ϕ and an item $b' \in \mathsf{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ such that $\pi_1(b') = \pi_1(b)$ and $\pi_2(b') = x_\phi$. More precisely, we associate with a token $(d, J_{k'}) \leq b$ a token (d, I_k) such that $d \in I_k$. Conversely, given an item $b' \in \mathsf{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ with $\pi_2(b') = x_\phi$, we pick $\phi(k, k')$ tokens $\{(d_i, I_k)\}_{1 \leq i \leq \phi(k,k')}$ and substitute to them the tokens $\{(d_i, J_{k'})\}_{1 \leq i \leq \phi(k,k')}$. In this way, we obtain a bag $b \in \mathsf{Bag}(\mathbb{R}_{\geq 0} \times \mathcal{I})$ with $\pi_2(b) = x$ and $\pi_1(b) = \pi_1(b')$.

The resulting $\mathsf{RA\text{-}TdPN}$ is denoted $\mathcal{N}_1$. □

Lemma 5. *We can build a* $\mathsf{RA\text{-}TdPN}$ $\mathcal{N}_2$*, equivalent to* $\mathcal{N}_1$*, and satisfying restrictions* (1) *and* (2).

Proof. We iteratively do the following transformation for every place of $\mathcal{N}_1$. Let p be a place of $\mathcal{N}_1$ and assume that $\{[a_1, a_1],]a_1, a_2[, \ldots,]a_{m-1}, a_m[, [a_m, a_m],]a_m, a_{m+1}[\}$ is the set of pairwise disjoint intervals required by restriction (1).

We substitute to p a set of places $\{p_{a_1}, p_{a_1,a_2}, \ldots, p_{a_{m-1},a_m}, p_{a_m}, p_{a_m,a_{m+1}}\}$. We thus need to modify the accepting condition Acc_1 of $\mathcal{N}_1$: the accepting condition Acc_2 of $\mathcal{N}_2$ is obtained by replacing any occurrence of p in Acc_1 by the term $\sum_{i=1}^{m} (p_{a_i} + p_{a_i,a_{i+1}})$. Besides, in the transformed net, a token with age d in place p_{a_i} or $p_{a_i,a_{i+1}}$ will correspond to a token with age $d + a_i$ in place p.

In order to pick (*i.e.* produce, consume or read) a token with age a_i in place p, one must pick a token with age 0 in the new place p_{a_i}. In order to pick a token with age $d \in]a_i, a_{i+1}[$ in place p, one must pick a token with age $d \in]0, a_{i+1} - a_i[$ in the new place $p_{a_i,a_{i+1}}$.

Thus we transform an arc connected to p with bag $x = n_1 \cdot [a_1, a_1] + n_{1,2} \cdot]a_1, a_2[+ \cdots + n_{m-1,m} \cdot]a_{m-1}, a_m[+ n_m \cdot [a_m, a_m] + n_{m,m+1} \cdot]a_m, a_{m+1}[$ to arcs connected to the new places such that the bag corresponding to p_{a_i} is $n_i \cdot [0, 0]$, and the bag corresponding to $p_{a_i,a_{i+1}}$ is $n_{i,i+1} \cdot]0, a_{i+1} - a_i[$.

Finally, we add transitions to "transfer" tokens from one of the new places to another one when their age increases: $t_{a_1,a_2}, t_{a_2}, \ldots, t_{a_m}, t_{a_m,a_{m+1}}$. A transition t_{a_i} consumes a token with age $a_i - a_{i-1}$ in p_{a_{i-1},a_i} and produces a token with age 0 in place p_{a_i}. A transition $t_{a_i,a_{i+1}}$ consumes a token with age 0 in p_{a_i} and produces a token with age 0 in place $p_{a_i,a_{i+1}}$. All these transitions are labelled by ε. The initial configuration of the new net is the same as the original one except that $\nu'_0(p_{a_1}) = \nu_0(p)$ and $\nu'_0(p') = 0$ if p' is a new place different from p_{a_1}.

Let $\mathcal{N}_2$ be the transformed net and ν' be a configuration of $\mathcal{N}_2$. We associate to ν' a configuration $\nu = f(\nu')$ of $\mathcal{N}_1$ defined by:

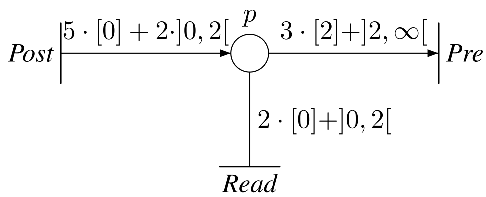
$$\begin{cases} f(p', d) & = (p', d) \quad \text{if } p' \neq p \text{ place of } \mathcal{N}_1 \\ f(p_{a_i}, d) & = (p, a_i + d) \text{ for every } p_{a_i} \\ f(p_{a_i,a_{i+1}}, d) & = (p, a_i + d) \text{ for every } p_{a_i,a_{i+1}} \end{cases}$$

which we extend on bags by linearity. Note that $f(\nu'_0) = \nu_0$. Straightforwardly, time elapsing commutes with this mapping. Moreover, firing any new transition does not modify the image of a configuration and finally the transformation of the arcs ensures that firing an existing transition is also possible in the original net and that this firing commutes with the mapping. Finally, we verify easily that the image by this mapping of a configuration satisfying Acc_1[4] is a configuration satisfying Acc_2. An accepting firing sequence of $\mathcal{N}_2$ leads thus by this mapping to an accepting firing sequence of $\mathcal{N}_1$.

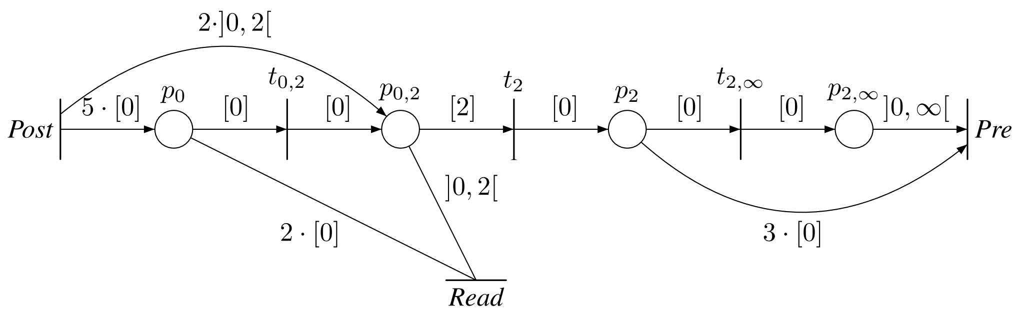
Conversely, assume that σ is an accepting firing sequence of $\mathcal{N}_1$. First, we split time elapsing steps in such a way that if at some instant a token corresponding to the sequence reaches the age a_i, this instant is associated with an intermediate configuration. In order to build the corresponding sequence σ' of $\mathcal{N}_2$, we will add firings of the new transitions at this instant some them just after the last time elapsing and some others just before the next time elapsing. The first set of firings will correspond to transitions $t_{a_{i+1}}$ and will transfer *all* tokens in place $p_{a_i,a_{i+1}}$ with age $a_{i+1} - a_i$ to place $p_{a_{i+1}}$. The second set of firings will correspond to transitions $t_{a_i,a_{i+1}}$ and will transfer *all* tokens in place p_{a_i} with age 0 in place $p_{a_i,a_{i+1}}$. With these enforced transition firings, tokens are always in the appropriate place for simulating a transition firing in σ. $\square$

Example 2. We illustrate the above construction on the following net:

[4] A configuration ν satisfies an acceptance condition Acc whenever the number of tokens in the places satisfies the constraint of Acc.



The new (part of) net which is constructed is the following:



We consider an execution in the initial net, and will give the corresponding execution in the constructed net. We consider the following execution in the initial net:

$\xrightarrow{\textit{Post}}$	$\xrightarrow{\text{wait } 0.5}$	$\xrightarrow{\textit{Post}}$	$\xrightarrow{\textit{Post}}$	$\xrightarrow{\textit{Read}}$	$\xrightarrow{\text{wait } 1}$	$\xrightarrow{\textit{Pre}}$
$0_{(\times 5)}$	$0.5_{(\times 5)}$	$0_{(\times 5)}$	$0_{(\times 10)}$	$0_{(\times 10)}$	$1_{(\times 10)}$	$1_{(\times 10)}$
1	1.5	$0.5_{(\times 5)}$	$0.5_{(\times 5)}$	$0.5_{(\times 5)}$	$1.5_{(\times 5)}$	$1.5_{(\times 5)}$
1.2	1.7	$1_{(\times 2)}$	$1_{(\times 4)}$	$1_{(\times 4)}$	$2_{(\times 4)}$	$2_{(\times 2)}$
		1.5	1.5	1.5	2.5	2.7
		1.7	1.7	1.7	2.7	

In the above sequence, a bag is represented vertically, for example the first bag means that there are 6 tokens in place p, five of age 0, one of age 1 and one of age 1.2. The corresponding sequence of transitions in the constructed net is:

Post, $t_{0,2}$, $(\text{wait } 0.5)$, *Post*, *Post*, *Read*, $t_{0,2}$, $t_{0,2}$, $(\text{wait } 0.5)$, t_2, $t_{2,\infty}$, $(\text{wait } 0.5)$, t_2, t_2, *Pre*

Lemma 6. *We can build an RA-TdPN $\mathcal{N}_3$, equivalent to $\mathcal{N}_2$, and satisfying restrictions* (1), (2)*, and* (3).

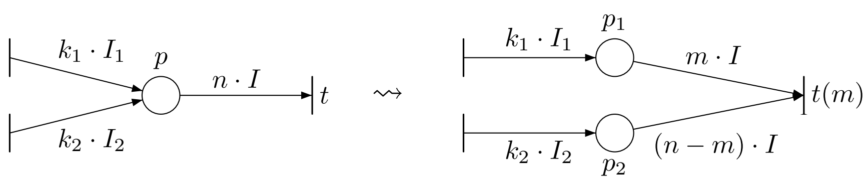
Proof. To prove this lemma, we need first to explain how we can transform the widgets built in the proof of the previous lemma into equivalent other widgets where all places will have the shape of one of the five patterns of Figure 3. In RA-TdPN $\mathcal{N}_2$,

– places p_{a_i} are connected to (possibly) several post-arcs labelled by bags $n \cdot [0]$, (possibly) several read-arcs labelled by bags $n' \cdot [0]$ and (possibly) several pre-arcs labelled by bags $n'' \cdot [0]$.

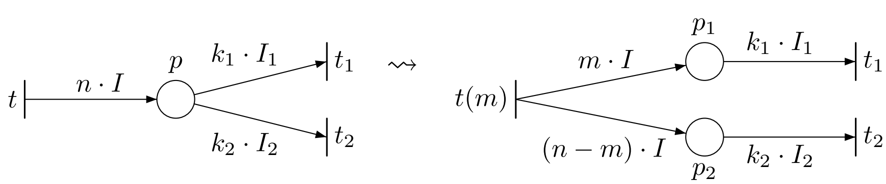
– places $p_{a_i,a_{i+1}}$ (with $a_{i+1} < \infty$) are connected to one post-arc whose bag is $[0]$, (possibly) several post-arc labelled by bags $n\cdot]0, a_{i+1}-a_i[$, (possibly) several read-arcs labelled by bags $n'\cdot]0, a_{i+1}-a_i[$, one pre-arc labelled by a bag $[a_{i+1}-a_i]$, and (possibly) several pre-arcs labelled by bags $n''\cdot]0, a_{i+1}-a_i[$.
– place $p_{a_m,\infty}$ is connected to one post-arc whose bag is $[0]$, (possibly) several post-arc labelled by bags $n\cdot]0,+\infty[$, (possibly) several read-arcs labelled by bags $n'\cdot]0,+\infty[$, and (possibly) several pre-arcs labelled by bags $n''\cdot]0,+\infty[$.

We will apply successively the following transformations to the different places:

– duplicate the place for each connected post-arc, and duplicate all transitions connected with read- and pre-arcs as depicted on the next picture (transition t can be a pre- or a read-arc):



Thus, each post- and read-transition is duplicated, one copy for every $m \le n$ if $n \cdot I$ is the bag labelling the arc between p and t.
– duplicate the place for each connected pre-arc, and duplicate all transitions connected with read- and post-arcs as depicted on the next picture (transition t can be a post- or a read-arc):



Thus, each pre- and read-transition is duplicated, one copy for every $m \le n$ if $n \cdot I$ is the bag labelling the arc between p and t.

We modify accordingly the accepting conditions by replacing occurrences of p by the sum $p_1 + p_2$ if we have duplicated the place p into the two places p_1 and p_2. It is straightforward to prove that these constructions do not change the accepted languages. There is only one point that needs to be detailed. In the last transformation, given an occurrence of t in a sequence σ of $\mathcal{N}$, we obtain the corresponding σ' of $\mathcal{N}'$ by choosing the appropriate $t(m)$ which depends on σ. Indeed, we count m_1 the number of tokens produced by t that will be consumed by t_1 and m_2 the number of tokens produced by t that will be consumed by t_2. Note that $m_1 + m_2 \le n$, so we can choose any m such that $m_1 \le m \le n - m_2$.

The places of the resulting net satisfy the property that they are connected to at most one post-arc and one pre-arc. Moreover, because of the form of the intervals in the former construction, this means that every place is of the form of one of the five patterns of Figure 3 (with possibly several read-arcs connected to the place). $\square$

Note that all transformations we have presented in the subsection preserve both boundedness and integrality properties of the nets. This concludes the proof of Proposition 1.

3.3 Removing the Read-Arcs

In this subsection, we study the role of read-arcs in RA-TdPNs. Thanks to Lemma 2 (language L_1), we already know that read-arcs add expressive power to TdPNs for the ω-equivalence. We then prove that read-arcs do not add expressiveness to the model of TdPNs when considering finite or infinite non-*zeno* timed words. We present two different constructions: the first one is correct only for finite timed words, whereas the second one, which extends the first one, is correct for non-*zeno* infinite timed words. In both correction proofs, we need to assume that places connected to read-arcs do not occur in the acceptance condition. This can be done without loss of generality, as stated by the following lemma.

Lemma 7. *Given a RA-TdPN $\mathcal{N}$, we can build a RA-TdPN $\mathcal{N}'$ $\{*, \omega, \omega_{nz}\}$-equivalent to $\mathcal{N}$ such that no place connected to a read-arc does occur in the acceptance condition.*

Proof. We iteratively apply the following transformation to every place of $\mathcal{N}$ connected to a read-arc and occurring in the acceptance condition. Let p be such a place. The net $\mathcal{N}'$ is obtained by adding to $\mathcal{N}$ a new place p' such that for every $t \in T$, $\mathsf{Post}(t)(p') = \mathsf{Post}(t)(p)$, $\mathsf{Pre}(t)(p') = \mathsf{Pre}(t)(p)$, $\mathsf{Read}(t)(p') = 0$. We assume in addition that $\nu_0(p') = \nu_0(p)$, and we set the acceptance condition of $\mathcal{N}'$ to the one of $\mathcal{N}$ where place p is replaced by place p'.

We claim that $\mathcal{N}'$ is equivalent to $\mathcal{N}$. First note that given any reachable configuration of $\mathcal{N}'$, p and p' contain the same number of tokens, but not necessarily the same (i.e. with the same age) tokens (because pre-arcs may choose different tokens).

Let σ' be a firing sequence of $\mathcal{N}'$ yielding an accepting configuration. Then σ, obtained from σ' by deleting the tokens of p' in the bags x, y, z associated with the firing of a transition, is also a sequence of $\mathcal{N}$. Indeed as $\mathcal{N}$ is a subnet of $\mathcal{N}'$ obtained by deleting places, all behaviours of the latter net are behaviours of the former one. Furthermore, due to the previous observation about markings of p and p', the configuration reached after the firing sequence σ satisfies the acceptance condition of $\mathcal{N}$.

Let σ be a firing sequence of $\mathcal{N}$ yielding an accepting configuration. Then we build σ' a firing sequence of $\mathcal{N}'$ from σ by consuming and producing in place p', the same tokens consumed and produced in p by the sequence σ. The final configuration of σ' has the same tokens in p and p' and thus satisfies the acceptance condition of $\mathcal{N}'$. $\square$

Case of finite words. We state the following result.

Theorem 2. *Let $\mathcal{N}$ be an RA-TdPN, then we can effectively build a TdPN $\mathcal{N}'$, which is $*$-equivalent to $\mathcal{N}$. Note that the construction preserves the boundedness and integrality properties of the nets.*

Proof. To prove this result, we first normalize the net. We then distinguish the five possible patterns of Figure 3 for a place p, and show that in every case, we can remove the read-arcs connected to place p.

Pattern $\mathcal{P}_1$. The construction is presented on Figure 4. This is the simplest case. Indeed, the simulation is the same as in the untimed case. It is easy to verify that the firing sequences of the two nets are exactly the same, and thus the two nets are equivalent.

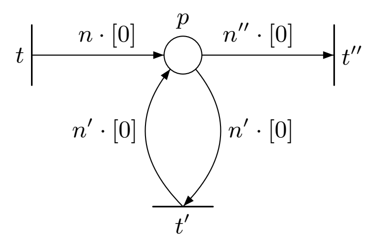


Fig. 4. Removing read-arcs in pattern $\mathcal{P}_1$

Pattern $\mathcal{P}_2$. The construction is presented on Figure 5. We also modify the accepting

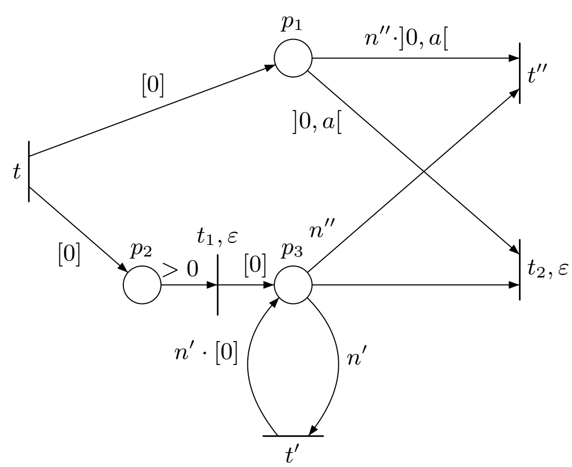


Fig. 5. Removing read-arcs in pattern $\mathcal{P}_2$

condition of $\mathcal{N}$ by adding the following constraint: $p_1 + p_2 + p_3 \leq 0$. Before proving the equivalence between the two nets, we do preliminary remarks on several invariants of the net $\mathcal{N}'$. Every configuration ν appearing on an accepting firing sequence of $\mathcal{N}'$ satisfies the following properties:

(i) $\mathsf{size}(\nu(p_1)) = \mathsf{size}(\nu(p2)) + \mathsf{size}(\nu(p_3))$

(ii) $\mathsf{size}(\nu(p_2)) \geq \mathsf{size}(\nu(p_1)_{|=0})$
where $\nu(p_1)_{|=0}$ is the bag of tokens in place p_1 whose age is equal to 0

(iii) $\mathsf{size}(\nu(p_1)) = \mathsf{size}(\nu(p_1)_{|<a})$
where $\nu(p_1)_{|<a}$ is the bag of tokens in place p_1 whose age is strictly less than a

The two first properties are simple invariants obtained by comparing producing and consuming arcs connected to places p_1, p_2 and p_3.

The last property relies on the accepting property of the sequence. Indeed, this implies that every token produced in place p_1 has to be consumed by one of the two transitions t'' and t_2. The timing requirements ($]0, a[$) of arcs connected to place p_1 of transitions t'' and t_2 then implies that these tokens cannot get older than age a.

We first consider an accepting firing sequence σ of $\mathcal{N}$, and build a corresponding accepting firing sequence σ' of $\mathcal{N}'$. We do two kinds of modifications to this sequence. First, we move tokens from place p_2 to place p_3 with the silent transition t_1 as soon as we need them for transition t' or t'' (if a token is never used, we move it when its age is equal to $\frac{a}{2}$ if $a < \infty$ or to 1 otherwise). Secondly, we empty places p_1 and p_3 using the silent transition t_2 as soon as the tokens are no more used until the end of the sequence. In this way, we consume every dead token of place p of net $\mathcal{N}$. It is possible to decide whether a token will still be used or not because we consider finite firing sequences. The silent transitions we have inserted allow to verify that we can fire the corresponding discrete transitions in the net $\mathcal{N}'$.

Conversely, we consider an accepting firing sequence σ' of $\mathcal{N}'$. We build a firing sequence σ of $\mathcal{N}$ obtained from σ' by erasing silent transitions t_1 and t_2. We now verify that transitions t' and t'' are still firable in σ. First note that the producing arcs imply the following inequality between two configurations ν and ν' obtained respectively after the same prefix of σ and σ':

$$\nu(p) \geq \nu'(p_1)$$

This implies that every firable occurrence of the transition t'' in σ' is still firable in σ. To prove the same property for t', we will use the preliminary remarks. Suppose that t' is firable in ν'. Then, there are at least n' tokens in place p_3. Properties (i), (ii) and (iii) together imply that there are at least n' tokens of age belonging to $]0, a[$ in place p_1. The previous inequality between $\nu(p)$ and $\nu'(p_1)$ finally implies that the transition t'' is also firable in $\mathcal{N}$. This concludes the proof for pattern $\mathcal{P}_2$.

Pattern $\mathcal{P}_3$. The construction is presented on Figure 6. We also modify the accepting condition of $\mathcal{N}$ by adding the following constraint: $\sum_{i=1}^{6} p_i \leq 0$. Before proving the equivalence between the two nets, we do preliminary remarks on several invariants of the net $\mathcal{N}'$. Every configuration ν appearing on an accepting firing sequence of $\mathcal{N}'$ satisfies the following properties:

(i) $\mathsf{size}(\nu(p_1)_{|=0}) + \mathsf{size}(\nu(p_2)_{|=0}) + \mathsf{size}(\nu(p_4)_{|=0}) = \mathsf{size}(\nu(p_3)_{|=0})$

(ii) $\mathsf{size}(\nu(p_2)_{|=a}) \leq \mathsf{size}(\nu(p_6)_{|>0})$

(iii) $\mathsf{size}(\nu(p_2)_{|>0}) + \mathsf{size}(\nu(p_4)_{|>0}) = \mathsf{size}(\nu(p_3)_{|>0}) + \mathsf{size}(\nu(p_5)) + \mathsf{size}(\nu(p_6))$

(iv) $\mathsf{size}(\nu(p_2)_{|]0,a[}) + \mathsf{size}(\nu(p_4)_{|>0}) \geq \mathsf{size}(\nu(p_3)_{|>0}) + \mathsf{size}(\nu(p_5))$

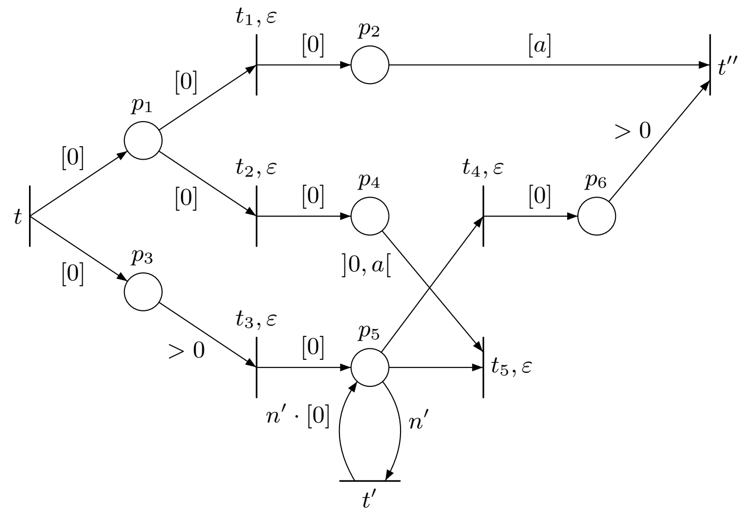


Fig. 6. Removing read-arcs in pattern $\mathcal{P}_3$

The first property is an invariant obtained by comparing producing and consuming arcs connected to the different places.

The second property relies on the accepting condition. Since a token with age a in place p_2 has to be consumed in zero time by transition t'', this transition has to be enabled, and thus we obtain the inequality (ii).

The third property is obtained from the first one by letting time elapse, using the fact that the acceptance condition implies that $\mathsf{size}(\nu(p_1)_{>0}) = 0$.

Finally, the fourth property can be obtained from properties (ii) and (iii) by subtraction.

We first consider an accepting firing sequence σ of $\mathcal{N}$, and build a corresponding accepting firing sequence σ' of $\mathcal{N}'$.

At each time a token is produced by the transition t, we move the corresponding token of place p_1. If this token will be consumed by the transition t'', then we use the silent transition t_1 to move it to the place p_2. Otherwise, we move it with t_2 to the place p_4.

Moreover, we also move the copy of the token of place p_3 to place p_5 with the silent transition t_3 as soon as we need it for transition t' (if a token is never checked by t', we move it when its age is equal to $\frac{a}{2}$). This instant must appear after a strictly positive delay of time since the interval of t' is $]0, a[$, which ensures that the transition t_3 is firable.

Finally, as soon as a token of place p_5 is no more used until the end of the sequence by the transition t', we have to consume it using t_4 or t_5. Two cases are possible:

– either the corresponding token of σ is consumed by t'', and then we move it to p_6 using t_4. Note that since the last read appears strictly before its age equals a, the age of the produced token in p_6 will be strictly positive when the age of the

corresponding token of place p_2 will reach a, and thus the transition t'' will be firable.
– or the token is never consumed by t'', and then we consume it immediately by t_5, which is possible since the last occurrence of t' appears strictly before a.

Note that the previous modifications are possible if we have done the same choices for the copies of the token placed in p_1 and p_3. In this way, we consume every dead token of place p of the net $\mathcal{N}$. This implies that the corresponding firing sequence will be accepting.

Note that once again, it is possible to decide whether a token will still be used or not because we consider finite firing sequences.

Finally, it can be checked that the silent transitions we have inserted lead to a firable sequence of the net $\mathcal{N}'$.

Conversely, we consider an accepting firing sequence σ' of $\mathcal{N}'$. We build a firing sequence σ of $\mathcal{N}$ obtained from σ' by erasing silent transitions t_1 and t_2. We now verify that transitions t' and t'' are still firable in σ. First note that the producing arcs imply the following inequality between two configurations ν and ν' obtained respectively after the same prefix of σ and σ':

$$\nu(p) \geq \nu'(p_1) + \nu'(p_2) + \nu'(p_4)$$

In particular, we have $\nu(p) \geq \nu'(p_2)$. This implies that every firable occurrence of the transition t'' in σ' is still firable in σ. To prove the same property for t', we will use the preliminary remarks. Suppose that t' is fireable in ν'. Then there are at least n' tokens in place p_5. Using inequality (iv), and the fact that the age of every token in place p_4 is strictly less than a (since we consider an accepting sequence), we get:

$$\mathsf{size}(\nu(p_2)_{|]0,a[}) + \mathsf{size}(\nu(p_4)_{|]0,a[}) \geq n'$$

This implies, using the previous inequality on ν, that there at least n' tokens in place p of age belonging to the interval $]0, a[$ in the configuration ν. This proves that t' is firable in ν and concludes the proof for pattern $\mathcal{P}_3$.

Pattern $\mathcal{P}_4$. The construction is presented on Figure 7. We also modify the accepting condition of $\mathcal{N}$ by adding the following constraint: $p_1 + p_2 \leq 0$. This pattern is treated similarly as the pattern $\mathcal{P}_2$. Indeed, the pre- and read-arcs are the same. The only modification then comes from the post-arc. In this pattern, tokens are produced with initial age belonging to the interval $]0, a[$, whereas they were produced with initial age 0 in pattern $\mathcal{P}_2$. The construction is simpler here since we do not need to let some time elapse before allowing the transition t' (corresponding to the read-arcs) to use produced tokens.

The correctness proof for this pattern can thus easily be derived from the one for pattern $\mathcal{P}_2$.

Pattern $\mathcal{P}_5$. The construction is presented on Figure 8. We also modify the accepting condition of $\mathcal{N}$ by adding the following constraint: $\sum_{i=1}^{5} p_i \leq 0$. Similarly as above, pattern $\mathcal{P}_5$ is treated in the same way than the pattern $\mathcal{P}_3$ since pre- and read-arcs are

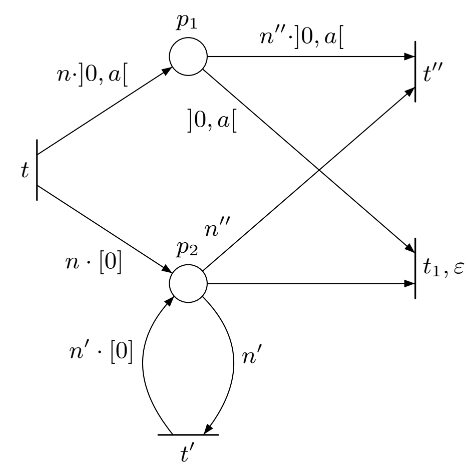


Fig. 7. Removing read-arcs in pattern $\mathcal{P}_4$

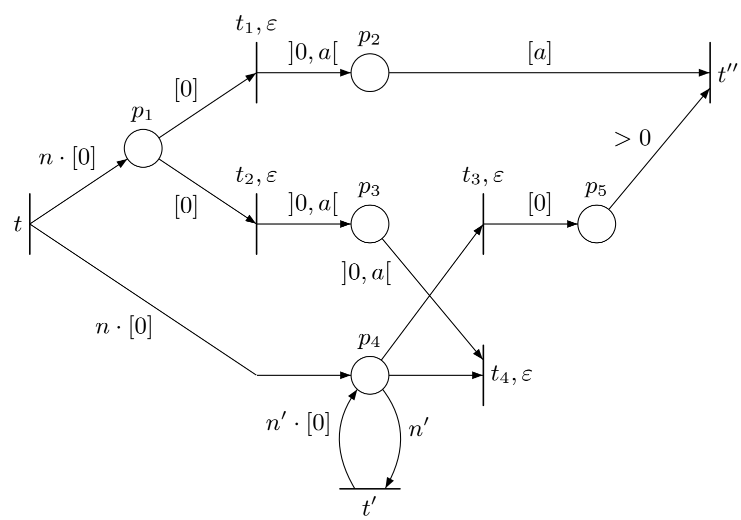


Fig. 8. Removing read-arcs in pattern $\mathcal{P}_5$

the same and the only modification comes from the post-arc: production in the interval $[0, 0]$ has been replaced by a production in the interval $]0, a[$.

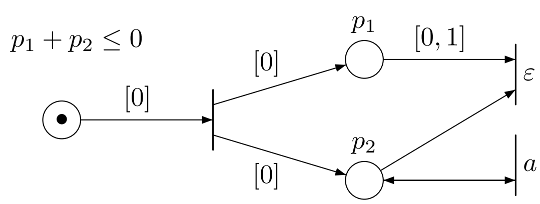
We do two main modifications to the case of pattern $\mathcal{P}_3$.

First, we let the choice of the initial age of the produced tokens to the transitions t_1 and t_2. Since there is no timed copy of the token, the choice of an initial age arises no difficulties. Recall that the choice of firing t_1 or t_2 corresponds as previously to the distinction between tokens that will be eventually consumed by the transition t'' before the end of the firing sequence, and the tokens that will not.

Second, since produced tokens have initial age belonging to the interval $]0, a[$, these tokens can immediately be used by the transition t', and thus, as in the previous case, we do not need to let some time elapse before moving tokens in the place p_4.

Finally, we claim that the correctness proof for this pattern can thus easily be derived from this of pattern $\mathcal{P}_3$. □

Example 3. We illustrate the construction on the RA-TdPN $\mathcal{N}_1$ of Figure 1(a). It is correct for finite timed words only. Indeed, to simulate a finite word, we can do as many a's as required, and then fire the silent transition, which empties both places p_1 and p_2, thus satisfying the acceptance condition. The date of this last firing ensures the satisfaction of the timing requirements. This simulation does not hold for infinite words since we can not add the silent transition after an infinite number of a's.



Case of infinite non-zeno words. The previous construction cannot be applied to languages of infinite words. Indeed, it relies on the following idea. The acceptance condition requires that one empties the places at the end of the sequence in the simulating net in order to check whether the tokens has been appropriately checked.

In the case of infinite timed words, a similar Büchi condition would "eliminate" words accepted by a sequence of the original net in which a place always contains tokens that will be checked in the future. However in the divergent case, we will first apply a transformation of the net that will not change the language, in such a way that in the new net, every infinite non-*zeno* timed word will be accepted by an appropriate generalized Büchi condition.

Theorem 3. *Let $\mathcal{N}$ be an RA-TdPN, then we can effectively build a TdPN $\mathcal{N}'$, which is ω_{nz}-equivalent to $\mathcal{N}$. Note that the construction preserves the boundedness and the integrality of the nets.*

Proof. We assume that $\mathcal{N}$ is normalized and that no place connected to a read-arc occurs in the acceptance conditions.

First we transform $\mathcal{N}$ yielding another RA-TdPN $\mathcal{N}^*$ as follows. We duplicate every place p connected to a read-arc by an arc labelled with $]0, a[$ and a finite, into two places p_{odd} and p_{even}. Then we iterate the following transformation for every place p and every arc connected to p. Let t be the transition connected by this arc to p and $n \cdot I$ be the bag labelling it. Then we substitute to t a set of transitions $\{t(k)\}_{0 \leq k \leq n}$ such that the arcs of these transitions are identical to those of t except the one under examination. We add to transition $t(k)$ two arcs (of the same kind as the original one), one labelled by $k \cdot I$ connected to p_{odd} and one labelled by $(n-k) \cdot I$ connected to p_{even}. Note that an original transition may be duplicated several times. The label of the duplicated transitions is the one of the original one.

It is clear that $\mathcal{N}$ and $\mathcal{N}^*$ are equivalent for all the language equivalences and in particular for the ω_{nz}-equivalence. However $\mathcal{N}^*$ satisfies an additional property we will now explain. Select any integer strictly greater than every finite interval bound occurring in $\mathcal{N}^*$ and call it $\max$. Given a sequence σ and a token initially present or produced by the sequence, we say that a token is useless in some configuration reached by σ, if it will be no more "used" in the sequence by a read-arc or a pre-arc connected to the place which contains it.

Let w be an infinite non-*zeno* timed word accepted by a firing sequence σ of $\mathcal{N}$ then we build a firing sequence σ^* of $\mathcal{N}^*$ whose label is w and such that:

- at any time $(2k) \cdot \max$ with $k \in \mathbb{N}$, there is a configuration such that every place p_{even} contains only useless tokens,
- at any time $(2k+1) \cdot \max$ with $k \in \mathbb{N}$, there is a configuration such that every place p_{odd} contains only useless tokens.

Note that, due to the divergence of σ, a token produced in place p (defined as before) will either become useless or be consumed in some configuration. If this configuration occurs in some interval $[(2k+1) \cdot \max, (2k+2) \cdot \max[$ we say that this token is *even* otherwise we say that it is *odd*. We build σ^* by appropriately replacing a transition by one of its duplicate depending on where to check, to consume or to produce the tokens in an odd or even place. In the case of a token production, we put an odd token into the corresponding odd place and *vice versa*.

Now take the last configuration of σ^* reached at time $(2k+1) \cdot \max$ and suppose that place p_{odd} contains a token which it is not useless then it will become useless during the interval $](2k+1) \cdot \max, (2k+2) \cdot \max[$. So it is an even token and should have been produced in p_{even}. The proof for the last configuration of σ^* reached at time $(2k) \cdot \max$ is similar.

We now apply the transformation of theorem 2 to $\mathcal{N}^*$ yielding $\mathcal{N}'$. In the transformation of patterns 2, 3, 4, 5 when a is finite we memorize the character of the new places. For instance, in the pattern $\mathcal{P}_4$, a place p_{odd} is replaced by two places $p_{odd,1}$ and $p_{odd,2}$. Then we add to the generalized Büchi condition of $\mathcal{N}'$ two new conditions: the sum of tokens in odd (resp. even) places must be infinitely often 0.

Let w be a non *zeno* infinite timed word of $\mathcal{N}$ (and of $\mathcal{N}^*$). Now take a sequence σ^* of $\mathcal{N}^*$ accepting w with the additional property. Simulate the sequence in $\mathcal{N}'$ as for theorem 2 except that tokens not consumed by σ^* are consumed by the "emptying" transitions of $\mathcal{N}'$ as soon as they become useless. Due to the property of σ^*, this simulating sequence fulfills the new conditions added to the generalized Büchi condition.

Conversely let σ' be an infinite non *zeno* sequence of $\mathcal{N}'$ and suppose that it "cheats". Then some tokens in odd or even places will never be consumed in σ' and σ' is not accepting. Thus for an accepting sequence σ' of $\mathcal{N}'$, we apply exactly the same transformations as those performed in theorem 2 in order to obtain an accepting sequence of $\mathcal{N}^*$. $\square$

Remark 1. Note the importance of the generalized Büchi condition since nothing ensures that odd and even places will be infinitely often *simultaneously* empty.

3.4 Removing General Resets

In this subsection, we study the role of general resets in RA-TdPNs. Thanks to Lemma 3 (language L_2), we know that the class of integral RA-TdPNs is strictly more expressive than the class of 0-reset integral RA-TdPNs for the ω-equivalence. We then prove two results, which show that this is the combination of the presence of read-arcs together with the integrality property which explains the expressiveness gap between 0-reset nets and nets with general resets. Indeed, we design a first construction which holds if there is no read-arc, and which preserves integrality of the net. Then we design a second construction, which holds even for nets with read-arcs, but which does not preserve the integrality of the nets.

Theorem 4. *For every TdPN $\mathcal{N}$, we can effectively build a 0-reset TdPN $\mathcal{N}'$ which is $\{*, \omega, \omega_{nz}\}$-equivalent to $\mathcal{N}$. Moreover, this construction preserves the boundedness and integrality properties of the net.*

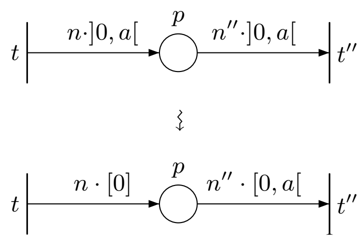
This result is not difficult and consists in shifting intervals of pre-arcs connected to a place, depending on the intervals which label post-arcs connected to this place.

Proof. Let $\mathcal{N}$ be a TdPN. Thanks to Proposition 1, we can assume that every place p of $\mathcal{N}$ satisfies one of the five patterns of Figure 3, in which there is no read-arc.

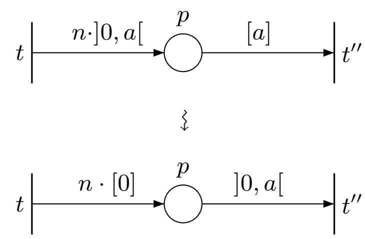
Only patterns $\mathcal{P}_4$ and $\mathcal{P}_5$ have general resets, we thus only describe a construction for these two cases. The constructions are depicted on Figure 9, and it is straightforward to prove their correctness. Indeed, in the case of pattern $\mathcal{P}_4$, if, in the initial net, a token enters place p with age $x \in]0, a[$ and leaves place p with age $y \in]0, a[$, then in the second net, it will enter place p with age 0, and leave place p with age $y - x \in [0, a[$. Conversely, if a token arrives in place p (with age 0) in the second net, and leaves the place with age $x \in [0, a[$, then it will arrive in place p (in the first net) with age $\frac{a-x}{2} \in]0, a[$ if $a < \infty$ (with age 1 otherwise) and it will leave place p at age $a - \frac{a-x}{2} \in]0, a[$ if $a < \infty$ (at age $1 + x$ otherwise). Dead tokens in the first net correspond to dead tokens in the second net. The case of pattern $\mathcal{P}_5$ is similar. $\square$

The second result is much more involved, and requires to refine the granularity of the net we build. However, it is correct for the whole class of RA-TdPNs.

Theorem 5. *For every RA-TdPN $\mathcal{N}$, we can build a 0-reset RA-TdPN $\mathcal{N}'$ which is $\{*, \omega_{nz}, \omega\}$-equivalent to $\mathcal{N}$. The construction preserves the boundedness of the net, but **not** its integrality.*



(a) Case of pattern $\mathcal{P}_4$



(b) Case of pattern $\mathcal{P}_5$

Fig. 9. Removing general resets in TdPNs.

Proof. First, it it worth noticing that in the case of finite words, and non-*zeno* infinite words, this result is a corollary of previous results (Theorems 2, 3 and 4). This proof, though correct for all finite and infinite timed words, is thus only necessary to deal with *zeno* infinite timed words.

Let $\mathcal{N}$ be a RA-TdPN which we assume satisfies Proposition 1. The only places of $\mathcal{N}$ which are connected to non 0-reset post-arcs are those which satisfy pattern $\mathcal{P}_4$ or pattern $\mathcal{P}_5$ (Figures 3(d) and 3(e)).

Case of pattern $\mathcal{P}_4$. The construction for this case is depicted on Figure 10. We denote $\mathcal{N}'$ the resulting net. We prove now the equivalence of the two nets.

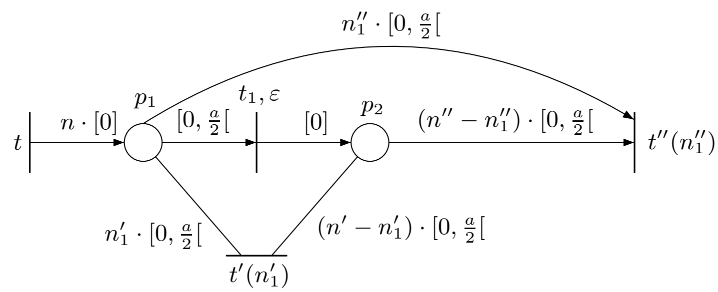


Fig. 10. 0-reset equivalent for pattern $\mathcal{P}_4$

First, let σ be an (infinite) accepting firing sequence of $\mathcal{N}$. We obtain a sequence σ' of $\mathcal{N}'$ with same timed word as follows.

Let us pick a token of p with initial age δ. Two cases are possible:

- First case: this token will not be consumed by t''. If $\delta \geq \frac{a}{2}$ then we let it definitively in p_1. Otherwise ($\delta < \frac{a}{2}$), after passing $\frac{a}{2} - \delta$ time, we transfer it to p_2 using the silent transition t_1. Note that the token in $\mathcal{N}'$ is at least as long available in p_1 or in p_2 as it is in $\mathcal{N}$.
- Second case: this token will be consumed by t'' when its age is δ'. If $0 < \delta' - \delta(< a)$, then we transfer it to p_2 after passing time $\frac{\delta' - \delta}{2}$. Otherwise, the token is

immediately consumed and no time elapses: we thus do not transfer the token. Note again that the token in $\mathcal{N}'$ is at least as long available in p_1 or in p_2 as it is in $\mathcal{N}$.

Now the sequence σ' is obtained from σ by inserting the occurrences of the transfer transition and by substituting to t' (resp. t'') the appropriate $t'(n'_1)$ (resp. $t''(n''_1)$) depending on the locations of the tokens of p in $\mathcal{N}'$ used by the firing of t' (resp. in t'') $\mathcal{N}$.

Conversely, let σ' be an (infinite) accepting firing sequence of $\mathcal{N}'$. We obtain a sequence σ of $\mathcal{N}$ with same timed word as follows.

We simply delete the occurrences of the transfer transition and we substitute to $t'(k'_1)$ (resp. $t''(k''_1)$) the transition t' (resp. t''). It remains to define the initial age of a token produced in p. If this token corresponds to a token in $\mathcal{N}'$ which is not transfered to p_2, its initial age is $\frac{a}{2}$. If the token is transfered to p_2 when its age is δ, then in $\mathcal{N}$, its initial age is $\frac{a}{2} - \delta$. Due to this choice, the token is at least as long available as it is in in p_1 or in p_2 of $\mathcal{N}'$, and thus every firable transition of σ' will be firable in σ.

This concludes the case of pattern $\mathcal{P}_4$.

Case of pattern $\mathcal{P}_5$. The construction for this second case is depicted on Figure 11. We again denote by $\mathcal{N}'$ the resulting RA-TdPN. Before showing the validity of the construction, we give some explanations about $\mathcal{N}'$. First, place $ready$ is connected to any transition of $\mathcal{N}$ by a Read-Arc whose bag is $[0, 0]$. Second, we denote by K the largest constant n' appearing on a bag $n' \cdot]0, a[$ of a Read-Arc and, for any integer k such that $0 \leq k \leq K$, we define a place $q(k)$ and two silent transitions $in(k)$ and $out(k)$. The lower part of the net is used to control the behaviours of the upper part of the net. Any behaviour of $\mathcal{N}'$ must then be an iteration of the following sequence:

- First, exactly one of the transitions $in(k)$ is fired, thus putting in zero time a token in some place $q(k)$ and in the place $ready$.
- Then the net fires the transitions of $\mathcal{N}$, including t, t', t'', (or more precisely their versions in $\mathcal{N}'$) in 0 time. Simultaneously, the token of the lower part of the net has moved to place $wait$.
- After that some time elapses, enabling the firing of the silent transition $out(k)$, which picks the token of the place $q(k)$ and puts a token in place tr.
- The upper part of the net can then transfer in 0 time some tokens from p_1 to p_2 using the silent transition t_1.
- Finally, the silent transition t_{sel} is fired in 0 time and puts back the token of the lower part in place sel.

We can now prove that the two nets are equivalent.

First, let σ be an (infinite) accepting firing sequence of $\mathcal{N}$. We obtain a sequence σ' of $\mathcal{N}'$ with same timed word as follows.

First, we describe how we transfer tokens from p_1 to p_2. Note that this is done in the same way than for the case of $\mathcal{P}_4$. Let us pick a token of p with initial age δ. Two cases are possible:

- First case: this token will not be consumed by t''. If $\delta \geq \frac{a}{2}$, then we let it definitively in p_1. Otherwise ($\delta < \frac{a}{2}$), after passing $\frac{a}{2} - \delta$ time we transfer it to p_2. Note that the token in $\mathcal{N}'$ is at least as long available in p_1 or in p_2 as it is in $\mathcal{N}$.

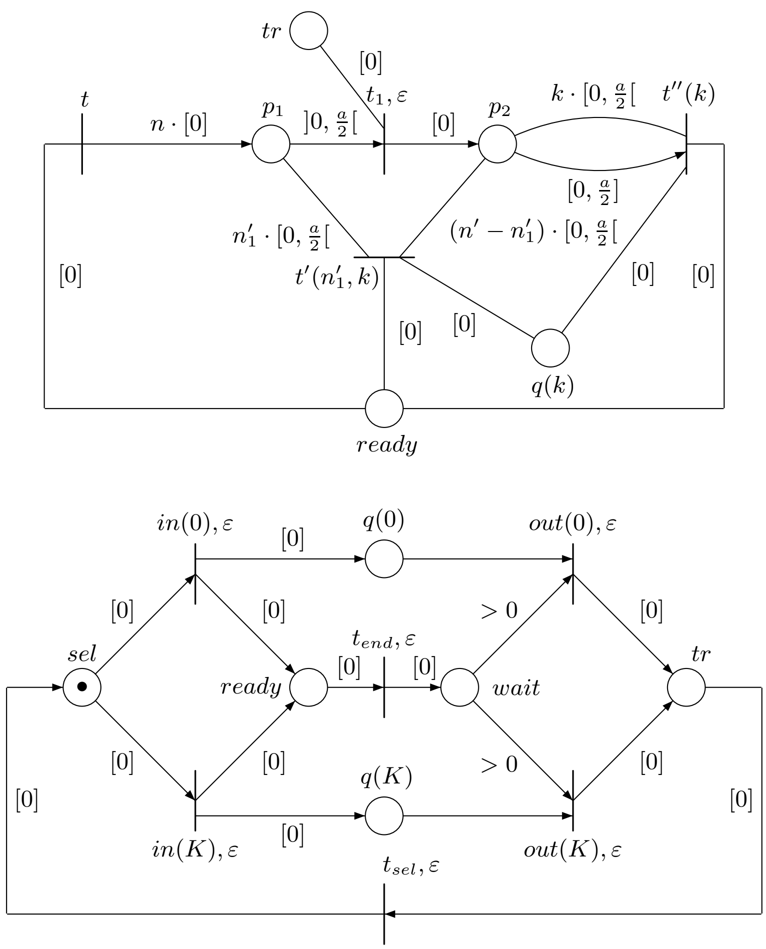


Fig. 11. 0-reset equivalent for pattern $\mathcal{P}_5$.

– Second case: this token will be consumed by t'' (necessarily when its age is a). We then transfer it to p_2 after passing time $\frac{a-\delta}{2}$. Note again that the token in $\mathcal{N}'$ is at least as long available in p_1 or in p_2 as it is in $\mathcal{N}$.

We then insert the occurrences of the transfer transition t_1 described above just before any transition which occurs at the same time.

Let us now consider a maximal instantaneous firing sequence ρ, *i.e.* a maximal subsequence of σ of time length equal to 0. We can consider the set of the constants $n' - n'_1$ such that there exists a Read Arc used in ρ whose bag connected to the place p_2 is equal to $(n' - n'_1) \cdot [0, \frac{a}{2}[$. We denote by k the maximum value of this set (0 if this set is empty). In such a subsequence ρ, we substitute to t' the appropriate $t'(n'_1, k)$ depending on the locations of the tokens of p in $\mathcal{N}'$ used by the firing of t', and we substitute to t'' the transition $t''(k)$.

Finally we decompose the sequence between time elapsing and instantaneous firing sequences. For each instantaneous firing sequence we apply the substitutions described above.

We claim that we obtain in this way a firing sequence of $\mathcal{N}'$ with same timed word. The only point to be detailed is the validity of a $t''(k)$ firing in $\mathcal{N}'$ since there is an additional read-arc. However, this firing takes place in a maximal instantaneous firing subsequence where k tokens have been read in p_2 with an age belonging to $[0, a/2[$. These tokens correspond in $\mathcal{N}$ to tokens in p whose age was strictly less than a during this subsequence. So they cannot be consumed by this subsequence and thus are present when firing $t''(k)$.

Conversely, let σ' be an (infinite) accepting firing sequence of $\mathcal{N}'$. We obtain a sequence σ of $\mathcal{N}$ with same timed word as follows. First we remark that each time a transition $t''(k)$ is fired in σ', we can consume the oldest token in p_2 with age less or equal than $\frac{a}{2}$ without modifying the firability of the sequence (since tokens in p_2 are checked for downwards closed intervals). Thus we assume this behaviour.

We simply delete the occurrences of the transfer transition and the cycle transitions (*i.e.* those occurring in the lower net) and we substitute to $t'(n'_1, k)$ (resp. $t''(k)$) the transition t' (resp. t''). It remains to define the initial age of a token produced in p. If this token corresponds to a token in $\mathcal{N}'$ which is not transfered to p_2, its initial age is $\frac{a}{2}$. If the token is transfered to p_2 when its age is δ and not consumed by some $t''(k)$, then in $\mathcal{N}$, its initial age is $\frac{a}{2} - \delta$. At last, if the token is transfered to p_2 when its age is δ and consumed by some transition $t''(k)$ when its age is δ', then its initial age is $a - \delta - \delta'$ (note that this last choice implies that the transition t'' will also be firable in $\mathcal{N}$).

Finally, we need to verify that these definitions of the initial ages of the tokens in $\mathcal{N}$ are compatible with the firing of the transitions t'. Let us consider an occurrence in σ of a Read-Arc with bag $n' \cdot]0, a[$. To be firable, this Read-Arc requires the presence of n' tokens in p with age less than a. This checking corresponds in $\mathcal{N}'$ to the firing of a transition $t'(n'_1, k)$ with $n' - n'_1 \leq k$ in some instantaneous firing sequence ρ. The n'_1 tokens in p_1 used by this firing have, by construction, an age less than a (note that these tokens will be possibly transfered to p_2 after a time elapsing). Now take the $n' - n'_1$ youngest tokens in p_2 at the beginning of ρ. We will prove that they all have an age in $\mathcal{N}$ strictly less than a. First, note that none of them can be consumed by a transition t'' during ρ since a firing of t'' requires at least $k \geq n' - n'_1$ tokens in addition to the one to be consumed, and since we have assumed above that transitions $t''(k)$ consume the oldest tokens. Now, let us consider one of these tokens. Two cases are possible: either it is consumed later (*i.e.* in another instantaneous firing sequence) by a transition $t''(k)$, and then its age in $\mathcal{N}$ is necessarily less than a. Or this token is never consumed, and then if its age in $\mathcal{N}'$ is equal to some $\delta' < a/2$, we have defined above its age in $\mathcal{N}$ as $a/2 + \delta'$, which satisfies $a/2 + \delta' < a$.

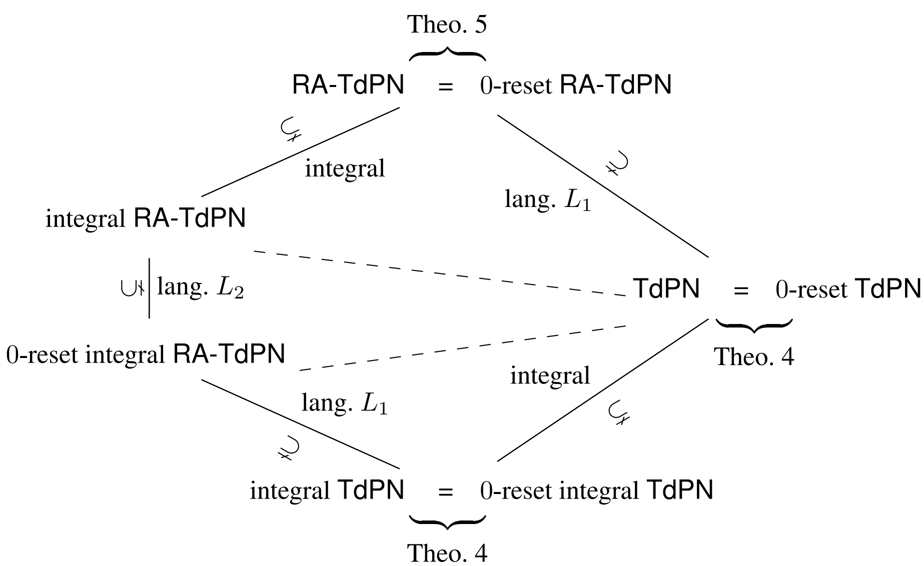
This concludes the proof of the second case. □

3.5 Summary of Our Expressiveness Results

Case of finite and infinite non-zeno words. Applying the results of the two previous subsections, we get equality of all subclasses of RA-TdPNs mentioned on the following picture, for the $\{*, \omega_{nz}\}$-equivalence. Note that this picture is correct for the general classes, for the restriction to integral nets, and also for the restriction to bounded nets.

$$\mathsf{RA\text{-}TdPN} \underbrace{\ =\ }_{\text{Theo. 2,3}} \mathsf{TdPN} \underbrace{\ =\ }_{\text{Theo. 4}} \text{0-reset } \mathsf{TdPN}$$

Case of infinite words. The picture in the case of infinite words is much different. Indeed the hierarchy in the previous case collapses, whereas we get here the lattice below. Plain arcs represent strict inclusion, and dashed arcs indicate that the classes are incomparable. We also write on the arc the reason for this inclusion ("integral" is to indicate that we restrict to the integral nets which immediately restrict the class; "lang. L_i" means that we use the language L_i of Subsection 3.1 for proving the strict inclusion). Finally note that this picture holds for both bounded and general nets.



4 Application to Timed Automata

First defined in [3], the model of timed automata (TA) associates with a finite automaton a finite set of non negative real-valued variables called *clocks*.

4.1 Introduction of Timed Automata

Let X be a finite set of variables, which we call *clocks*. We write $\mathcal{C}(X)$ for the set of *constraints* over X, which consist of conjunctions of atomic formulas of the form

$x \bowtie h$ for $x \in X$, $h \in \mathbb{Q}_{\geq 0}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$. The model we will define here is a slight extension of the classical model of [3] and a subclass of *updatable timed automata* [5].

Definition 5 (Timed Automaton (TA)). *A* timed automaton $\mathcal{A}$ *over* Σ_ε *is a tuple* $(L, \ell_0, X, \Sigma_\varepsilon, E, A)$ *where* L *is a finite set of* locations*,* $\ell_0 \in L$ *is the* initial location*,* X *is a finite set of* clocks*,* $E \subseteq L \times \mathcal{C}(X) \times \Sigma_\varepsilon \times (X \hookrightarrow \mathcal{I}) \times L$ *is a finite set of* edges*, and* A *is an accepting condition given as a finite set of subsets of* L *An edge* $e = \langle \ell, \gamma, a, \mu, \ell' \rangle \in E$ *represents a transition from location* ℓ *to location* ℓ' *labelled by* a *with constraint* γ *and update partially defined function* μ *called a* reset*.*

A *valuation* v is a mapping in $\mathbb{R}_{\geq 0}^X$. If $\mu : X \hookrightarrow \mathcal{I}$ is a partially defined function, if v is a valuation, $\mu(v)$ is the set of valuations v' such that $v'(x) \in \mu(x)$ if μ is defined in x, and $v'(x) = v(x)$ otherwise. Constraints of $\mathcal{C}(X)$ are interpreted over valuations, and the relation $v \models \gamma$ is defined inductively by $v \models (x \bowtie h)$ when $v(x) \bowtie h$, and $v \models (\gamma_1 \wedge \gamma_2)$ whenever $v \models \gamma_1$ and $v \models \gamma_2$.

The semantics of timed automata is defined as a timed transition system.

Definition 6 (Semantics of a TA). *The semantics of a TA* $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E)$ *is a TTS* $S_\mathcal{A} = (Q, q_0, \rightarrow)$ *where* $Q = L \times (\mathbb{R}_{\leq 0})^X$*,* $q_0 = (\ell_0, \mathbf{0})$ *and* $\rightarrow$ *is defined by:*

- *either a delay move* $(\ell, v) \xrightarrow{d} (\ell, v + d)$*,*
- *or a discrete move* $(\ell, v) \xrightarrow{e} (\ell', v')$ *iff there exists some* $e = (\ell, \gamma, a, \mu, \ell') \in E$ *s.t.* $v \models \gamma$ *and* $v' \in \mu(v)$*.*

We recover classical timed automata by restricting the resets to partial functions μ assigning only the interval $[0]$, but we will call them here 0-*reset timed automata*. If all constants appearing in guards and updates are integers, we say that the timed automaton is *integral*.

As for RA-TdPNs, we define the various timed languages accepted by a TA $\mathcal{A}$: $\mathcal{L}^*(\mathcal{A})$, $\mathcal{L}^\omega(\mathcal{A})$, and $\mathcal{L}^{\omega_{nz}}(\mathcal{A})$. We extend the $*$-(resp. ω-, ω_{nz}-)equivalences to TA and to comparisons between subclasses of RA-TdPNs and subclasses of TA.

Examples of TA have already been given in this paper: see Figures 1(b) and 2(b).

4.2 TA and Bounded RA-TdPNs.

The following theorem relates TA and bounded RA-TdPNs.

Theorem 6. *Bounded RA-TdPNs and TA are* $\{*, \omega_{nz}, \omega\}$*-equivalent.*

Proof. ***From bounded RA-TdPNs to TA.*** Let $\mathcal{N}$ be a bounded RA-TdPN, and assume that the net is bounded by k. We will build a TA $\mathcal{A}$ equivalent to $\mathcal{N}$. The construction is made in two steps. We first construct an equivalent (structurally) safe RA-TdPN $\mathcal{N}'$, and we then build an equivalent timed automaton $\mathcal{A}$.

Every place p of $\mathcal{N}$ is replaced by $2k$ places $\{p_i^0, p_i^1 \mid 1 \leq i \leq k\}$ in $\mathcal{N}'$. The two places p_i^0 and p_i^1 will be mutually exclusive, and the (at most) k tokens in place p in $\mathcal{N}$ will be spread in the places p_i^1's. The intuition of the construction is to use the places p_i^1 to simulate one of the at most k tokens of place p. To ensure that these places are safe, we use the complementary places p_i^0. We make these two places (p_i^0 and p_i^1) mutually exclusive by imposing, when producing (resp. consuming) a token in p_i^1, to consume (resp. produce) a token in place p_i^0. We now describe formally the construction.

Let t be a transition of $\mathcal{N}$. Transition t will be simulated by several sequences of transitions. $\mathsf{Pre}(t)(p)$ (resp. $\mathsf{Post}(t)(p)$, $\mathsf{Read}(t)(p)$) is a bag in $\mathsf{Bag}(\mathcal{I})$, whose size is denoted by $s(t)(p)$ (resp. $s'(t)(p)$, $s''(t)(p)$). We order the tokens in these bags and assume that $\mathsf{Pre}(t)(p) = I_1 + \ldots + I_{s(t)(p)}$, $\mathsf{Post}(t)(p) = I'_1 + \ldots + I'_{s'(t)(p)}$ and $\mathsf{Read}(t)(p) = I''_1 + \ldots + I''_{s''(t)(p)}$. We add four new places in $\mathcal{N}'$ which will be used as intermediate places for simulating t: q_t^0, q_t^1, ${q'_t}^0$ and ${q'_t}^1$. The simulation of t proceeds in three steps. First, we consume the tokens as required by the Pre Arcs. Second, we proceed the Read Arcs. Third, we produce the tokens as required by the Post Arcs.

Simulation of a pre-arc. Let p be a place such that $s(t)(p) > 0$. We fix an injective function $\iota_t(p)$ defined from $\{1, \ldots, s(t)(p)\}$ onto $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the Pre Arc between t and p will consume the $s(t)(p)$ tokens. We then add a transition t_{ι_t} such that $\forall 1 \leq i \leq s(t)(p)$,

$$\begin{cases} \mathsf{Pre}(t_{\iota_t})(p^1_{\iota_t(p)(i)}) = I_i, & \text{(consumes the tokens chosen by } \iota_t(p)\text{)} \\ \mathsf{Post}(t_{\iota_t})(p^0_{\iota_t(p)(i)}) = [0], & \text{(produces the corresponding tokens)} \\ \mathsf{size}(\mathsf{Read}(t_{\iota_t})) = 0, & \text{(reads nothing)} \\ \mathsf{Post}(t_{\iota_t})(q_t^1) = [0], & \text{(goes to the read step of the simulation of } t\text{)} \\ \mathsf{Pre}(t_{\iota_t})(q_t^0) = \mathbb{R}_{\geq 0} & \text{(consumes the corresponding token)} \end{cases}$$

Simulation of a read-arc. Let p be a place such that $s'(t)(p) > 0$. We fix an injective function $\iota'_t(p)$ defined from $\{1, \ldots, s'(t)(p)\}$ onto $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the Read Arc between t and p will read the $s'(t)(p)$ tokens. We then add a transition $t_{\iota'_t}$ such that $\forall 1 \leq j \leq s'(t)(p)$,

$$\begin{cases} \mathsf{Pre}(t_{\iota'_t})(q_t^1) = [0] & \text{(starts the simulation of the read-arcs)} \\ \mathsf{Post}(t_{\iota'_t})(q_t^0) = [0] & \text{(puts the corresponding token back)} \\ \mathsf{Read}(t_{\iota'_t})(p^1_{\iota'_t(p)(j)}) = I'_j, & \text{(reads the tokens chosen by } \iota'_t(p)\text{)} \\ \mathsf{Post}(t_{\iota'_t})({q'_t}^1) = [0], & \text{(goes to the simulation of the post-arcs)} \\ \mathsf{Pre}(t_{\iota'_t})({q'_t}^0) = \mathbb{R}_{\geq 0} & \text{(consumes the corresponding token)} \end{cases}$$

Simulation of a post-arc. Let p be a place such that $s''(t)(p) > 0$. We fix an injective function $\iota''_t(p)$ defined from $\{1, \ldots, s''(t)(p)\}$ onto $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the Post Arc between t and p will produce the $s''(t)(p)$ tokens. We then add a transition $t_{\iota''_t}$ such that $\forall 1 \leq k \leq s''(t)(p)$,

$$\begin{cases} \mathsf{Pre}(t_{\iota''_t})({q'_t}^1) = [0], & \text{(starts the simulation of the post-arcs)} \\ \mathsf{Post}(t_{\iota''_t})({q'_t}^0) = \mathbb{R}_{\geq 0} & \text{(puts the corresponding token back)} \\ \mathsf{size}(\mathsf{Read}(t_{\iota''_t})) = 0, & \text{(reads nothing)} \\ \mathsf{Post}(t_{\iota''_t})(p^1_{\iota''_t(p)(k)}) = I''_k, & \text{(produces the tokens chosen by } \iota''_t(p)\text{)} \\ \mathsf{Pre}(t_{\iota''_t})(p^0_{\iota_t(p)(k)}) = \mathbb{R}_{\geq 0}, & \text{(consumes the corresponding tokens)} \end{cases}$$

Label of transitions. The two first transitions t_{ι_t} and $t_{\iota'_t}$ are labelled by ε. The third one, $t_{\iota''_t}$, is labelled by the label of t.

Initial marking The initial marking is extended by the following affectations. $m_0(q_t^0) = 1$, $m_0({q'_t}^0) = 1$, $m_0(q_t^1) = 0$ and $m_0({q'_t}^1 = 0)$.

Finally, the acceptance condition is transformed in a natural way: every occurrence of a place p in the acceptance condition is replaced by the term $\sum_{i=1}^{k}(p_i^0 + p_i^1)$, and we add to the acceptance condition, for any transition t, the constraints $q_t^1 = 0$ and ${q'_t}^1 = 0$.

The construction is illustrated on Figure 12.

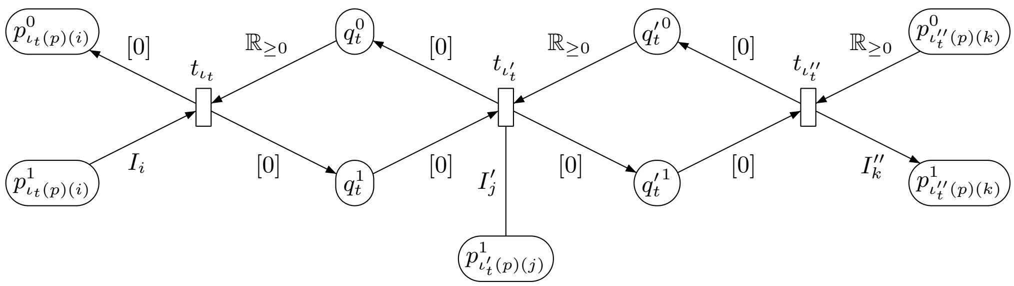


Fig. 12. Simulating a bounded RA-TdPN using a safe RA-TdPN

The correctness of this construction relies on the fact that a configuration with n tokens in place p is encoded by a configuration where n places p_i^1 contains 1 token whereas for the $k-n$ other i's, there is 1 token in place p_i^0. Then transition t is simulated by selecting correctly the tokens needed by the pre-arc in the places p_i^1, removing them, then reading correctly the read-arc (by selecting as many tokens as necessary in the corresponding places) and then creating tokens by the post-arc in (and only in) the places p_i^1 which do not have a token. The other side of the proof is similar, we just remove the intermediary steps. This construction is correct for the $\{*, \omega, \omega_{nz}\}$-equivalence.

We now present the construction which transforms a safe RA-TdPN into a TA. Let $\mathcal{N} = (P, m_0, T, \mathsf{Pre}, \mathsf{Post}, \mathsf{Read}, \lambda, \mathsf{Acc})$ be a safe RA-TdPN. We will define a TA $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, A)$ equivalent to $\mathcal{N}$. By abuse of notation, given a transition t of $\mathcal{N}$, we simply write in this construction $\mathsf{Pre}(t)$ for the set of places $p \in P$ such that $\mathsf{size}(\mathsf{Pre}(t)(p)) > 0$ (and similarly for Post and Read). Note that since $\mathcal{N}$ is safe, we can assume that for any transition $t \in T$, we have $\mathsf{Pre}(t) \cap \mathsf{Read}(t) = \emptyset$ and $\mathsf{Read}(t) \cap \mathsf{Post}(t) = \emptyset$ (otherwise the transition will never be firable).

We define $\mathcal{A}$ as follows:

- $L = 2^P$,
- $\ell_0 = \mathsf{dom}(m_0)$, (there is exactly one token per initially marked place)
- $X = P$, (x_p denotes the clock corresponding to the place p)
- there is a transition $\ell \xrightarrow{\gamma, a, \mu} \ell'$ whenever there exists a transition t in $\mathcal{N}$ such that:

  - $\mathsf{Pre}(t) \cup \mathsf{Read}(t) \subseteq \ell$,
  - $\mathsf{Post}(t) \cap (\ell \setminus \mathsf{Pre}(t)) = \emptyset$,
  - $\ell' = (\ell \setminus \mathsf{Pre}(t)) \cup \mathsf{Post}(t)$,
  - γ is the conjunction of all $x_p \in I_p$ such that $(p, I_p) \in \mathsf{Pre}(t) \cup \mathsf{Read}(t)$,
  - a is the label of transition t in $\mathcal{N}$,
  - μ resets clock x_p in interval I_p if $(p, I_p) \in \mathsf{Post}(t)$.
- if $\mathsf{Acc} = \{f_1, \dots, f_k\}$, A is defined as the set of formulas $\{A_1, \dots, A_k\}$ where for every $1 \le i \le k$, $A_k = \{Q \subseteq 2^P \mid (\bigwedge_{q \in Q} q = 1 \wedge \bigwedge_{q \notin Q} q = 0) \Rightarrow f_i\}$.

Note that since a place contains at most one token, one clock is enough to encode the behaviour of a place. It is then routine to verify that this construction is correct.

From TA to bounded RA-TdPNs. Let $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, F)$ be a TA. We construct the RA-TdPN $\mathcal{N} = (P, m_0, T, \mathsf{Pre}, \mathsf{Post}, \mathsf{Read}, \lambda, \mathsf{Acc})$ as follows.

- $P = L \cup X$,
- $m_0 = \ell_0 + \sum_{x \in X} x$
- $T = E$,
- for all $e = \ell \xrightarrow{g,a,\mu} \ell'$ in E,
  - if x is such that $\mu(x)$ is defined, $\mathsf{Post}(e)(x) = \mu(x)$, $\mathsf{Pre}(e)(x) = g_{|x}$, where $g_{|x}$ is the interval of x imposed by constraint g,
  - if x is such that $\mu(x)$ is not defined, $\mathsf{Read}(e)(x) = g_{|x}$,
  - $\mathsf{Pre}(e)(\ell) = \mathbb{R}_{\ge 0}$, $\mathsf{Post}(e)(\ell') = [0]$,
  - $\lambda(e) = a$,
- $\mathsf{Acc} = \{(f = 1), f \in F\}$,

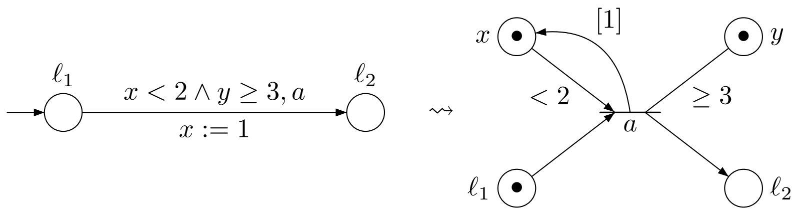
The net $\mathcal{N}$ that we have constructed is strongly bisimilar to the original timed automaton. Indeed, we consider the relation $\mathcal{R}$ defined by

$$(\ell, v)\mathcal{R}\nu \text{ iff } \begin{cases} \mathsf{size}(\nu(\ell)) > 0 & \\ \mathsf{size}(\nu(\ell')) = 0 & \forall \ell' \neq \ell \\ \nu(x) = v(x) & \forall x \in X, \end{cases}$$

where $(\ell, v) \in L \times \mathbb{R}_{\ge 0}^X$ is a configuration of $\mathcal{A}$, and $\nu \in \mathsf{Bag}(\mathbb{R}_{\ge 0})^P$ is a configuration of $\mathcal{N}$. It is straightforward to verify that $\mathcal{R}$ is a bisimulation relation which respects accepting configurations.

Finally, just notice that there is always exactly one token in one of the places l for $l \in L$. This justifies the definition of Acc. Moreover, it is easy to verify that the net we have constructed is safe, thus bounded. $\square$

Example 4. We illustrate the transformation of a TA into a bounded RA-TdPN on an example. It is worth noticing that for the clock x, which is both checked and reset, we can use a pre- and a post-arc, whereas for the clock y, which is checked but not reset, we use a read-arc.



4.3 Expressiveness Results for TA

Combining this former result with the results of the previous section on Petri nets, we get interesting side results on timed automata, and in particular quite surprising results for languages of infinite timed words.

Corollary 1. *For the* $\{*, \omega_{nz}\}$*-equivalence,*

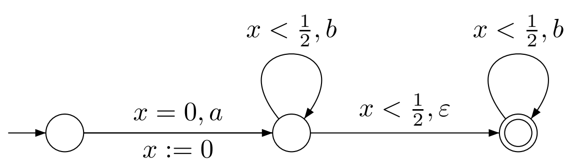
1. *bounded* TdPN*s and* TA *are equally expressive;*
2. *(integral)* TA *and* 0*-reset (integral)* TA *are equally expressive.*

Corollary 2. *For the* ω*-equivalence,*

3. TdPN*s and* TA *are incomparable;*
4. TA *are strictly more expressive than bounded* TdPN*s;*
5. *integral* TA *are strictly more expressive than integral* 0*-reset* TA*;*
6. TA *and* 0*-reset* TA *are equally expressive.*

As a "folk" result, it was thought that TA and bounded TdPNs are equally expressive. We have proved that this is indeed the case for finite and infinite non-*zeno* behaviours (item *1.*), but that it is wrong when considering also *zeno* behaviours (item *4.*). Indeed, the result is even stronger: even though TdPNs can be somehow seen as timed systems with infinitely many clocks, we have proved that TA and TdPNs are in general incomparable (item *3.*).

The three other results complete the picture of known results about general resets in TA [5]. Item *2.* was already partially proved in the above-mentioned paper, and we provide here a new proof of this result. Items *5.* and *6.* are quite surprising, since they show that refining the granularity of the guards is necessary for removing general resets in TA (and for preserving the languages of infinite timed words). It is one of the first such results in the framework of timed systems (up to our knowledge). Finally, the construction provided in the proof of Theorem 5 applied to TA provides an extension to infinite words of the construction presented in [5] for removing general resets in TA (which is indeed only correct for finite and infinite non-*zeno* timed words). We illustrate this construction by giving a 0-reset TA ω-equivalent to the timed automaton of Figure 2(b).



5 Conclusion

In this paper , we have thoroughly studied the relative expressiveness of TdPNs and TA, and we have proved in particular that they are incomparable in general. This has motivated the introduction of read-arcs in TdPNs, yielding the model of RA-TdPNs. This model unifies TA and TdPNs, has a decidable coverability problem, and enjoys pretty surprising expressiveness results.

We have studied the expressive power of read-arcs in RA-TdPNs, and we have proved that, when restricting to finite or infinite non-*zeno* behaviours, read-arcs do not add expressiveness. On the other hand, we show that *zeno* behaviours discriminate between several subclasses of RA-TdPNs. For instance, RA-TdPNs are strictly more expressive than TdPNs. Since we also prove that bounded RA-TdPNs and TA are equally expressive, we get the surprising result that TA are strictly more expressive than bounded TdPNs, which is quite counter-intuitive.

Classically, TdPNs use quite general resets, whereas TA use only resets to 0. We have thus studied the expressive power of these general resets, compared with resets to 0. We have shown that they don't add any expressiveness to the above-mentioned models, but that the granularity has to be refined for removing general resets in RA-TdPN when considering *zeno* behaviours. Up to our knowledge, this is one of the first expressiveness results (at least in the domain of timed systems), which requires to refine the granularity of the model. As side results, we complete the work in [5], and get that it is necessary to refine the granularity of guards in TA for removing general resets, when considering languages of infinite possibly *zeno* timed words.

Our main further work will be to develop partial-order techniques for RA-TdPNs, taking advantage of the locality of the firing rules.

References

1. P. A. Abdulla, P. Mahata, and R. Mayr. Decidability of Zenoness, syntactic boundedness and token-liveness for dense-timed petri nets. In *Proc. 24th Conf. Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *LNCS*, pages 58–70. Springer, 2004.
2. P. A. Abdulla and A. Nylén. Timed Petri nets and bqos. In *Proc. 22nd Int. Conf. Application and Theory of Petri Nets (ICATPN'01)*, volume 2075 of *LNCS*, pages 53–70. Springer, 2001.
3. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
4. B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Transactions in Software Engineering*, 17(3):259–273, 1991.
5. P. Bouyer, C. Dufourd, E. Fleury, and A. Petit. Updatable timed automata. *Theoretical Computer Science*, 321(2–3):291–345, 2004.
6. C. Girault and R. Valk, editors. *Petri Nets for Systems Engineering*. Springer, 2002.
7. G. Higman. Ordering by divisibility in abstract algebras. In *Proc. London Math. Soc.*, volume 2, pages 326–336, 1952.
8. F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. 15th Int. Conf. Concurrency Theory (CONCUR'04)*, volume 3170 of *LNCS*, pages 387–401. Springer, 2004.

9. S. Lasota and I. Walukiewicz. Alternating timed automata. In *Proc. 8th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05)*, volume 3441 of *Lecture Notes in Computer Science*, pages 250–265. Springer, 2005.
10. P. Mahata. *Model Checking Parameterized Timed Systems*. PhD thesis, Dept. Information Technology, Uppsala University, Sweden, 2005.
11. U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6):545–596, 1995.
12. J. Ouaknine and J. B. Worrell. On the language inclusion problem for timed automata: Closing a decidability gap. In *Proc. 19th Annual Symposium on Logic in Computer Science (LICS'04)*, pages 54–63. IEEE Computer Society Press, 2004.
13. J. Ouaknine and J. B. Worrell. On the decidability of metric temporal logic. In *Proc. 19th Annual Symposium on Logic in Computer Science (LICS'05)*, pages 188–197. IEEE Computer Society Press, 2005.
14. V. Valero Ruiz, F. Cuartero Gomez, and D. de Frutos-Escrig. On non-decidability of reachability for timed-arc Petri nets. In *Proc. 8th Int. Work. Petri Nets and Performance Models (PNPM'03)*, pages 188–196. IEEE Computer Society Press, 1999.