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

Patricia Bouyer, Serge Haddad, Pierre-Alain Reynier

PII: S0890-5401(07)00108-3
DOI: 10.1016/j.ic.2007.10.004
Reference: YINCO 3521

To appear in: *Information and Computation*

Received Date: 26 July 2006
Revised Date: 15 May 2007
Accepted Date: 14 October 2007

Please cite this article as: P. Bouyer, S. Haddad, P-A. Reynier, Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences, *Information and Computation* (2007), doi: 10.1016/j.ic.2007.10.004

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

Patricia Bouyer [a] Serge Haddad [b] Pierre-Alain Reynier [a]

[a] *LSV, CNRS & ENS Cachan, France*

[b] *LAMSADE, CNRS & Université Paris-Dauphine, France*

Abstract

Timed Petri nets and timed automata are two standard models for the analysis of real-time systems. We study in this paper their relationship, and prove in particular that they are incomparable w.r.t. language equivalence. In fact, we study the more general model of timed Petri nets with read-arcs (RA-TdPN), already introduced in [17], which unifies both models of timed Petri nets and of timed automata, and prove that the coverability problem remains decidable for this model. 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.

Key words: Timed Automata, Timed Petri Nets, Expressiveness

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) [8]. An important interest of PNs lies in their applicability to the verification of infinite-state systems because some standard problems are decidable for this model (boundedness, coverability, reachability, action-based linear-time formula checking, etc.).

⋆ A preliminary version of this work has been published in [6].

Email addresses: bouyer@lsv.ens-cachan.fr (Patricia Bouyer), haddad@lamsade.dauphine.fr (Serge Haddad), reynier@lsv.ens-cachan.fr (Pierre-Alain Reynier).

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 it is enabled (every input place contains the required number of tokens) and if the time since it has been enabled lies in the specified interval. Time can elapse only if it does not disable some transition: thus, the decision to wait some amount of time and then fire a transition cannot be done locally, but requires to check globally that no other transition is disabled during the delay, even though this transition does not share any input or output place with the transition we plan to fire. This restricts a lot applicability of partial order methods for this model. Moreover, because of this "urgency" requirement, all significant problems are undecidable for unbounded TPNs.

Timed Petri nets (TdPN), also called *timed-arc Petri nets*, associate with each arc an interval [18]. 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 hence local and do not depend on the global configuration of the system, unlike in TPNs. This "lazy" behaviour has important consequences. Whereas the reachability problem is undecidable for TdPNs [18], 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.

Read-arc timed Petri nets (RA-TdPN) extend TdPNs with read-arcs, *i.e.* arcs that check the presence of a token (with an age as specified on the arc), without consuming it. This model has been first introduced by Jiří Srba in [17] in order to compare TA with 1-bounded TdPNs (and its extension with read-arcs). Moreover, this feature has already been introduced in the untimed framework [14] in order to define a more refined concurrent semantics for the nets. For semantics taking into account fairness, it has been shown in [19] that read arcs add expressive power. For the interleaving semantics, they however do not add any expressive power in the untimed framework as they can be replaced by two arcs which check that a token is in the place and replace it immediately.

Our contributions. We first investigate the decidability of the coverability problem for the RA-TdPN model, and we prove that, as for TdPNs, it is 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.

Organisation of the paper. In Section 2, we define the RA-TdPN model and its different subclasses. We show in Section 3 that the coverability problem is decidable for that model. In Sections 4, 5, 6 and 7 we establish our numerous expressiveness results on RA-TdPNs and their subclasses. We present an overview of these results in Section 8. In Section 9, we give expressiveness results for timed automata.

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. Given a function f over some set X, we may extend wordlessly f to the set of subsets of X, by $f(Y) = \{f(y) \mid y \in Y\}$, for every subset Y of X. An interval I of $\mathbb{R}_{\geq 0}$ is a $\mathbb{Q}_{\geq 0}$-(resp. $\mathbb{N}_{\geq 0}$-)*interval* if its left endpoint belongs to $\mathbb{Q}_{\geq 0}$ (resp. $\mathbb{N}_{\geq 0}$) and its right endpoint belongs to $\mathbb{Q}_{\geq 0} \cup \{\infty\}$ (resp. $\mathbb{N}_{\geq 0} \cup \{\infty\}$). We denote by $\mathcal{I}$ (resp. $\mathcal{I}_{\mathbb{N}_{\geq 0}}$) the set of $\mathbb{Q}_{\geq 0}$-(resp. $\mathbb{N}_{\geq 0}$-)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 $\mathbb{N}_{\geq 0}$ such that

the set $\mathsf{dom}(f) = \{x \in \mathcal{E} \mid f(x) \neq 0\}$ is finite. Given such an element $f \in \mathsf{Bag}(\mathcal{E})$, we use the notation $f = \sum_{x \in \mathsf{dom}(f)} f(x) \cdot x$ (omitting $f(x)$ when $f(x) = 1$). 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 $\forall \tau \geq d, (x+d)(\tau) = x(\tau - d)$. We finally define the operation of projection. Let $x \in \mathsf{Bag}(\mathcal{E}_1 \times \ldots \times \mathcal{E}_n)$, and let $I = \{i_1, \ldots, i_k\}$ be a set of indices such that $1 \leq i_i < \ldots < i_k \leq n$. The bag $\pi_{i_1,\ldots,i_k}(x) \in \mathsf{Bag}(\mathcal{E}_{i_1} \times \ldots \times \mathcal{E}_{i_k})$ is defined by: for all $(e_{i_1}, \ldots, e_{i_k}) \in \mathcal{E}_{i_1} \times \ldots \times \mathcal{E}_{i_k}$, $\pi_{i_1,\ldots,i_k}(x)(e_{i_1}, \ldots, e_{i_k}) = \sum_{e_{j_1},\ldots,e_{j_{n-k}} \in \mathcal{E}_{j_1} \times \ldots \times \mathcal{E}_{j_{n-k}}} x(e_1, \ldots, e_n)$, where $\{j_1, \ldots, j_{n-k}\}$ is the unique set of indices such that $1 \leq j_1 < \ldots < j_{n-k} \leq n$ satisfying $\{i_1, \ldots, i_k\} \cap \{j_1, \ldots, j_{n-k}\} = \emptyset$. Finally, note that if A is a finite set and B a set, then $\mathsf{Bag}(B)^A$, the set of applications from A to $\mathsf{Bag}(B)$, is isomorphic to $\mathsf{Bag}(A \times B)$.

Timed words and timed languages. Let Σ be a fixed finite alphabet such that $\varepsilon \notin \Sigma$ (ε is the silent action), we denote $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$. A *timed word* w over Σ_ε (resp. Σ) is a finite or infinite sequence $w = (a_0, \tau_0)(a_1, \tau_1) \ldots (a_n, \tau_n) \ldots$ such that 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 time point at which action a_k occurs. We write $\mathit{Duration}(w) = \sup_k \tau_k$ 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 to be *Zeno* whenever $\mathit{Duration}(w)$ is finite. We denote by $\mathcal{TW}^*(\Sigma)$ (resp. $\mathcal{TW}^\omega(\Sigma)$, $\mathcal{TW}^{\omega_{nz}}(\Sigma)$) the set of finite (resp. infinite, non-*Zeno* infinite) timed words over Σ. A *timed language of finite (resp. infinite, non-Zeno infinite) words* is a subset of $\mathcal{TW}^*(\Sigma)$ (resp. $\mathcal{TW}^\omega(\Sigma)$, $\mathcal{TW}^{\omega_{nz}}(\Sigma)$).

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 composed of timing constraints on arcs. Informally, when firing a transition, tokens are consumed whose ages satisfy the timing constraints specified on the input-arcs (they are specified using bags), 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* $(P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathsf{Acc})$ *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 $\mathsf{Bag}(\mathcal{I})^P$;*
- *$\lambda : T \to \Sigma_\varepsilon$ is a labelling function;*

- Acc *is an accepting condition defined as a finite set of formulas, each of which is generated by the grammar*

$$\mathsf{acc} \ ::= \ \sum_{i=1}^{n} p_i \bowtie k \ \mid \ \mathsf{acc} \wedge \mathsf{acc}$$

where $p_i \in P$, $k \in \mathbb{N}_{\geq 0}$ *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 also be 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, τ) for a token which is in place p and whose age is τ. A configuration is then a finite sum of such pairs. A token (p, τ) then belongs to the configuration ν whenever $(p, \tau) \leq \nu$ (in terms of bags). The *initial configuration* $\nu_0 \in \mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$ is defined as $\nu_0 = \sum_{p \in P} m_0(p) \cdot (p, 0)$, where it means that for each p, there are $m_0(p)$ tokens of age 0 in place p. Given a configuration $\nu \in \mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$ and a bag $f \in \mathsf{Bag}(P \times \mathcal{I})$, we say that ν satisfies f, and write $\nu \models f$, if and only if there exists a bag $x \in \mathsf{Bag}(P \times \mathbb{R}_{\geq 0} \times \mathcal{I})$ verifying the following conditions.

$$\begin{cases} \pi_{1,2}(x) = \nu, \\ \pi_{1,3}(x) = f, \\ \forall (p, \tau, I) \in \mathsf{dom}(x), \ \tau \in I. \end{cases}$$

We now describe the semantics of a RA-TdPN as a transition system.

Definition 2 (Semantics of a RA-TdPN) *Let* $\mathcal{N} = (P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathsf{Acc})$ *be an RA-TdPN. Its semantics is the transition system* $(Q, \nu_0, \Sigma_\varepsilon, \rightarrow)$ *where* $Q = \mathit{Bag}(\mathbb{R}_{\geq 0})^P$, $\nu_0 = \sum_{p \in P} m_0(p) \cdot (p, 0)$, *and the transition relation* $\rightarrow$ *is composed of delay and discrete transitions as follows:*

- *For each* $d \in \mathbb{R}_{\geq 0}$, *there is a delay transition* $\nu \xrightarrow{d} \nu + d$ *where the configuration* $\nu + d$ *is defined by* $(\nu + d)(p) = \nu(p) + d$ *for every* $p \in P$.
- *Given a transition* $t \in T$ *and two configurations* $\nu, \nu' \in \mathit{Bag}(P \times \mathbb{R}_{\geq 0})$, *there exists a discrete transition from* ν *to* ν' *labelled by* $\lambda(t)$, *denoted by* $\nu \xrightarrow{\lambda(t)} \nu'$, *if and only*

if there exist three bags ${}^\bullet\nu, {}^\circ\nu, \nu^\bullet \in \mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$ *such that:*

$$\begin{cases} {}^\bullet\nu \models \mathit{Pre}(t), \\ {}^\circ\nu \models \mathit{Read}(t), \\ \nu^\bullet \models \mathit{Post}(t), \\ {}^\bullet\nu + {}^\circ\nu \leq \nu, \\ \nu' = \nu - {}^\bullet\nu + \nu^\bullet. \end{cases}$$

The intuition of the previous definition is as follows: ${}^\bullet\nu$ is the set [1] of tokens which is removed from the configuration ν when firing transition t, whereas ${}^\circ\nu$ is the set of tokens that needs to be in p for transition t to be fired (note that these two sets of tokens need to be disjoint, hence the fourth condition ${}^\bullet\nu + {}^\circ\nu \leq \nu$); finally $\nu^\bullet$ is the set of tokens that are created by the transition firing. Moreover, the ages of all these tokens need to satisfy the constraints specified by the various arcs (conditions written using the $\models$ operator defined above). Finally, all tokens used by a read-arc are not removed, that's why the new configuration is given by ν' computed as $\nu' = \nu - {}^\bullet\nu + \nu^\bullet$.

To reason about the behaviour of the net, we also consider the transition system obtained when λ is the identity mapping. We then write $\nu \xrightarrow{t} \nu'$ when transition t is fired, according to the previous definition.

A *path* in the RA-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, which alternates between delay and discrete transitions. 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 $(p, \tau) \leq \nu$ 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, τ is strictly greater than I. It means that this token cannot be used anymore (either by a pre- or a read-arc) to fire a transition.

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 $(\lambda(t_1), d_1)(\lambda(t_2), d_1 + d_2)\ldots$ Petri nets can be considered as language acceptors, as formally defined by the next definition.

[1] This is a language misuse, the right term should be "bag", as there can be several tokens with the same age.

We first define a satisfaction relation for the accepting conditions. It is defined over configurations of the nets, inductively as follows:

$$\begin{cases} \nu \text{ satisfies } \sum_{i=1}^{n} p_i \bowtie k \text{ iff } \sum_{i=1}^{n} \mathsf{size}(\nu(p_i)) \bowtie k \\ \nu \text{ satisfies } \mathsf{acc}_1 \wedge \mathsf{acc}_2 \text{ iff } \nu \text{ satisfies } \mathsf{acc}_1 \text{ and } \nu \text{ satisfies } \mathsf{acc}_2 \end{cases}$$

where $\bowtie \in \{\leq, \geq\}$.

Definition 3 (Language accepted by a RA-TdPN) *Let* $\mathcal{N} = (P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathsf{Acc})$ *be an RA-TdPN. 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*[2] *). 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* $\mathcal{N}$ *along finite (resp. infinite) paths.*

It is worth noticing that the accepting conditions only depend on the untimed markings associated with configurations. Note also that infinite paths leading to finite timed words are not considered in this work.

Two RA-TdPNs $\mathcal{N}$ 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 RA-TdPNs. In the following, we will use notations like "$\{*, \omega, \omega_{nz}\}$-equivalence" to mean the intersection of all three equivalences. *Idem* for "$\{*, \omega_{nz}\}$-equivalence" and other combinations. We will also use notations like $\equiv_\omega$ or $\equiv_{*,\omega_{nz}}$ to denote the ω- (resp. $\{*, \omega_{nz}\}$) equivalence between classes of nets.

Notations. Read-arcs are represented by undirected arcs. On pictures, we may use shorthands to represent bags: for all $I \in \mathcal{I}$, I stands for the bag $1 \cdot I$, $[a]$ is for the interval $[a, a]$. We may write intervals as constraints, *e.g.* "$\leq a$" stands 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 RA-TdPN is depicted on Figure 1. 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 between* min *and* max *depending on the workload of the server (transition "*start*"). Note that, due to the "lazy" semantics of RA-TdPNs, nothing prevents the*

[2] We do not know whether generalized Büchi conditions could be reduced to Büchi conditions in the context of timed Petri nets. Nevertheless, the standard construction for finite automata does not extend to Petri nets.

*token in place "*busy*" (resp. "*ready*") to die (i.e., to reach an age strictly greater than* max*, resp. than* 0*), hence blocking the system. A suitable accepting condition like "*Acc = {busy = 0, ready = 0}*" prevents such a blocking behaviour by enforcing infinitely often the server to refresh the cache. Note the importance of using a generalized Büchi condition to enforce the firing of both transitions "*start*" and "*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 the places "*cache*" and "*read*": when transition "*read*" is fired, a token in place "*client*" with age* 0 *is consumed, and it is checked whether at least one token in place "*cache*" has age less than or equal to "*val*". However, this token is not consumed (and can hence be used later on again) and its age is unchanged.*

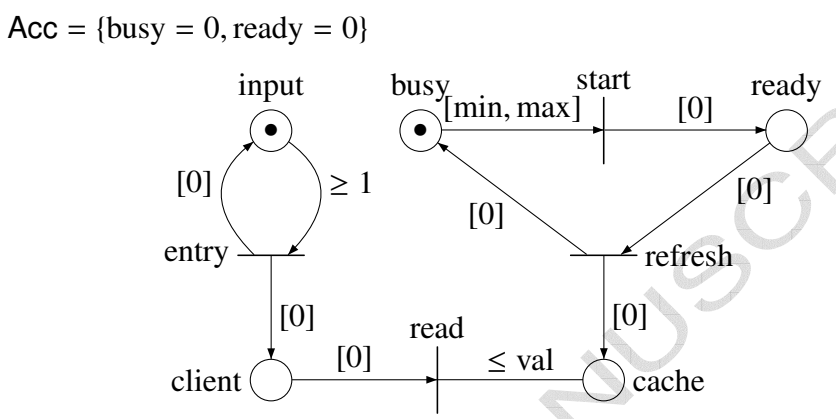


Fig. 1. An example of RA-TdPN.

We give an example of a path in this RA-TdPN, assuming that min = 2*,* max = 4*, and* val = 3*.*

$$
\begin{aligned}
(\text{input}, 0) + (\text{busy}, 0) &\xrightarrow{(2)} (\text{input}, 2) + (\text{busy}, 2) \\
&\xrightarrow{\text{start}} (\text{input}, 2) + (\text{ready}, 0) \\
&\xrightarrow{\text{refresh}} (\text{input}, 2) + (\text{busy}, 0) + (\text{cache}, 0) \\
&\xrightarrow{(3)} (\text{input}, 5) + (\text{busy}, 3) + (\text{cache}, 3) \\
&\xrightarrow{\text{entry}} (\text{input}, 0) + (\text{client}, 0) + (\text{busy}, 3) + (\text{cache}, 3) \\
&\xrightarrow{\text{read}} (\text{input}, 0) + (\text{busy}, 3) + (\text{cache}, 3)
\end{aligned}
$$

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

Definition 4 *Let* $\mathcal{N} = (P, m_0, T, \mathit{Pre}, \mathit{Post}, \mathit{Read}, \lambda, \mathsf{Acc})$ *be an RA-TdPN. It is*

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

All above notions are quite standard, except the 0-reset property which implies that all tokens which are produced are produced with initial age 0.

Note that the RA-TdPN of Example 1 is integral, 0-reset, but not bounded as there can be an unbounded number of tokens in place "cache" or "client".

3 The Coverability Problem.

Let $\mathcal{N}$ be an RA-TdPN. Let N be a set of configurations of $\mathcal{N}$. By $N^{\uparrow}$, we denote the upward closure of N, *i.e.*, the set $\{\nu \mid \exists \nu' \in N,\ \nu' \leq \nu\}$.

Let N be a finite set of configurations of $\mathcal{N}$ where all ages of tokens are rational. The *coverability problem* for $\mathcal{N}$ and set of configurations N asks whether there exists a path in $\mathcal{N}$ from ν_0, the initial configuration of $\mathcal{N}$, to some $\nu \in N^{\uparrow}$. We prove the following result.

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

In order to prove this theorem, 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 [12] for TdPNs, and has been used recently in several other contexts [15,16,11]. An alternative proof based on zones rather than regions could be used as well, like in [2].

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 $\mathbb{N}_{\geq 0} \cup \{\infty\}$. Let N be a finite set of markings with integral ages. There is no loss of generality in assuming that finite bounds of the net and that values of ages are integers or $+\infty$ (otherwise we refine the granularity of the regions). By max we denote the maximal integer appearing in the bounds of intervals of the net and in the ages of the tokens in the configurations of N.

[3] This is the standard model, as defined in [18].

Definition 5 *A region $\mathcal{R}$ for $\mathcal{N}$ is a sequence $a_0 a_1 \dots a_n a_\infty$ where $n \in \mathbb{N}_{\geq 0}$, for all $0 \leq i \leq n$, $a_i \in \mathsf{Bag}(P \times \{0, 1, \dots, \max\})$ with $\mathsf{size}(a_i) \neq 0$ if $i \neq 0$, and $a_\infty \in \mathsf{Bag}(P \times \{\infty\})$.*

We first informally explain the semantics of a region. Given the bag of tokens defining a configuration, we obtain its associated region 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 add the information about their ages. Finally, we order the remaining tokens depending on the fractional part of their ages in $a_1, \dots, a_n$, forget their fractional part, and only store the integral part of their ages. Hence n is the number of different positive fractional values for ages of the remaining tokens. For instance, consider the bag of tokens $(p, 1) + (p, 2.8) + (q, 0.8) + (q, 5.1) + (r, 1.5)$. Then, if the maximal constant is 4, its region encoding will be $a_0 a_1 a_2 a_\infty$ where $a_0 = (p, 1)$ (because there is a single token with integral age), $a_\infty = (q, \infty)$ (because the age of token $(q, 5.1)$ is 5.1, hence above the maximal constant), $a_1 = (r, 1)$ (among all fractional parts, 0.5 is the smallest one), and $a_2 = (p, 2) + (q, 0)$ (all tokens with fractional part 0.8).

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 $\mathcal{R} = a_0 a_1 \dots a_n a_\infty$ 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:

- $\nu = a_0 + \nu_1 + \nu_2 + \ldots + \nu_n + \nu_\infty$,
- $\forall 1 \leq i \leq n$, $\phi(\nu_i) = a_i$, 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, that we write $\mathcal{R}(\nu)$, 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 itself reduces to solving the coverability problem for a single region $\mathcal{R}$.

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

Proof. We first notice that, given two regions $\mathcal{R} = a_0 a_1 \dots a_n a_\infty$ and $\mathcal{R}' = a'_0 a'_1 \dots a'_{n'} a'_\infty$, 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 partial order between regions by $\mathcal{R} \leq \mathcal{R}'$ iff $[\mathcal{R}']^\uparrow \subseteq [\mathcal{R}]^\uparrow$. Then, using Higman's lemma [9], we can show that this is a well quasi-order, *i.e.*, for every

infinite sequence of regions $\{\mathcal{R}_i\}_{i\in\mathbb{N}_{\geq 0}}$ there exist $i < j$ such that $\mathcal{R}_i \leq \mathcal{R}_j$. Indeed, each region $\mathcal{R}$ is a finite sequence of bags over a finite set, hence applying [2, Theorem 1], the above-mentioned partial order is a well quasi-order.

The algorithm for solving the coverability problem for the upward closure of a single region $\mathcal{R}$ then consists in computing iteratively the predecessors (by time elapsing and by discrete steps) of $[\mathcal{R}]^{\uparrow}$. As we will see, each such predecessor is a finite union of upward closures of regions. We stop exploring the predecessors of an upward closure of a region when it is larger (for partial order $\leq$) than an already computed region. Note that all configurations reachable from $[\mathcal{R}_2]^{\uparrow}$ are also reachable from $[\mathcal{R}_1]^{\uparrow}$ whenever $\mathcal{R}_1 \leq \mathcal{R}_2$. The computation can then be seen as a finitely branching tree. To prove that it terminates, it is sufficient to prove that this tree is finite. Suppose it is not. By applying König lemma, this tree has an infinite branch. However, as $\leq$ is a well quasi-order, we will eventually obtain a region which is larger than a previous one. This leads to a contradiction. Hence, the computation tree is finite, and the computation terminates. The set of configurations N is covered by the RA-TdPN $\mathcal{N}$ if and only if its initial configuration ν_0 occurs in the upward closure of some region of the tree.

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

Time predecessors. If a_0 contains a token $(p, 0)$, there is no strict time predecessor of $[\mathcal{R}]^{\uparrow}$. Otherwise if $\mathsf{size}(a_0) \neq 0$, then the time predecessor is $[\mathcal{R}']^{\uparrow}$ with $\mathcal{R}' = a'_0 a_1 \dots a_n a'_{n+1} a_\infty$ where a'_0 is the empty bag and a'_{n+1} is obtained from a_0 by decrementing by 1 the (integral) age of each token. Informally, this operation represents a (reverse) small time elapse such that no token of a_1 reaches an integral value and no token of a_∞ reaches back max.

Otherwise (*i.e.*, $\mathsf{size}(a_0) = 0$) we need to choose if tokens of a_1 will first reach an integral value or some tokens of a_∞ will first reach max. 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 (which assumes $n \geq 1$). The above-mentioned time predecessor is $[\mathcal{R}']^{\uparrow}$ where $\mathcal{R}' = a'_0 a'_1 \dots a'_{n-1} a'_\infty$ is obtained as follows.

- $a'_\infty = a_\infty - b_\infty$,
- $a'_0 = a_1 + c_\infty$ where c_∞ is obtained from b_∞ by setting the age of each token to max,
- $\forall 1 \leq i \leq n-1, a'_i = a_{i+1}$.

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 $i \in \{0, 1, \dots, n, \infty\}$, 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 consider the upward closure of the region $a_0 a_1 \dots a_n a_\infty$, and want to compute its preimage by transition t. Transition t produces the bag of tokens $\mathsf{Post}(t)$. These tokens may appear in one of the a_i's, but this is not required, they may only be in the upward closure. Similarly, some tokens of $\mathsf{Read}(t)$ may appear in some of the a_i's, but this is also not required. Hence, we choose bags of tokens $\mathsf{post}_i, \mathsf{read}_i^+ \in \mathsf{Bag}(P \times \{0, 1, \dots, \max\} \times \mathcal{I})$ for every $i \in \{0, 1, \dots, n\}$ and $\mathsf{post}_\infty, \mathsf{read}_\infty^+ \in \mathsf{Bag}(P \times \{\infty\} \times \mathcal{I})$ such that

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

The bag post_i represents the tokens produced by t which "belong" to a_i, whereas the bag read_i^+ represents the tokens read by t which also "belong" to a_i. However, there might be additional tokens (either that are read or that are produced) which do not appear in one of the a_i's (this is possible as we consider the upward closure of the region), that's why the two last conditions are inequalities and not equalities. Figure 2 illustrates the decomposition.

Applying this first decomposition, we build an intermediate region $\mathcal{R}' = a'_0 a'_1 \dots a'_{n'} a'_\infty$ by substracting $\pi_{1,2}(\mathsf{post}_i)$ from a_i for every i and deleting the item in the resulting sequence if its size is null (for $1 \leq i \leq n$).

Then, to really simulate the discrete transition t, we need to initially have all tokens required by the read-arcs and all tokens that are consumed by the pre-arcs. We set bags of tokens $\mathsf{pre}_i, \mathsf{read}_i^- \in \mathsf{Bag}(P \times \{0, 1, \dots, \max\} \times \mathcal{I})$ for every $i \in \{0, 1, \dots, n''\}$ for some integer n'', $\mathsf{pre}_\infty, \mathsf{read}_\infty^- \in \mathsf{Bag}(P \times \{\infty\} \times \mathcal{I})$ and a strictly increasing mapping ψ from $\{1, \dots, n'\}$ into $\{1, \dots, n''\}$ such that

- 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 $i \in \{1, \dots, n''\}$, if there exists j such that $\psi(j) = i$ then
  $a''_i = a'_j + \pi_{1,2}(\mathsf{pre}_i) + \pi_{1,2}(\mathsf{read}_i^-)$, otherwise $a''_i = \pi_{1,2}(\mathsf{pre}_i) + \pi_{1,2}(\mathsf{read}_i^-)$,
- $\sum_i \pi_{1,3}(\mathsf{pre}_i) = \mathsf{Pre}(t)$,
- $\sum_i \pi_{1,3}(\mathsf{read}_i^-) + \sum_i \pi_{1,3}(\mathsf{read}_i^+) = \mathsf{Read}(t)$.

The bags read_i^- complement the already defined bags read_i^+'s to satisfy the $\mathsf{Read}(t)$ constraint, whereas pre_i are the tokens required by the pre-arcs of the transition. See Figure 2 for an illustration of the construction.

Under those conditions, the region $\mathcal{R}'' = a''_0 a''_1 \dots a''_{n''} a''_\infty$ is a predecessor by t of $[\mathcal{R}]^\uparrow$. Note that the constructed region $\mathcal{R}''$ depends on the various choices we have made (all bags read^+, read^-, pre, *etc.* and also the indices n', n'', the mapping

ψ, *etc.*). For each of these (finitely many) choices, it gives a region which is in the preimage of $\mathcal{R}$ by t (indeed, take any configuration $\nu'' \in [\mathcal{R}'']^{\uparrow}$, then quite straightforwardly, any configuration image of ν by t is in $[\mathcal{R}]^{\uparrow}$), and all regions in the preimage by t can of course be obtained in that way.

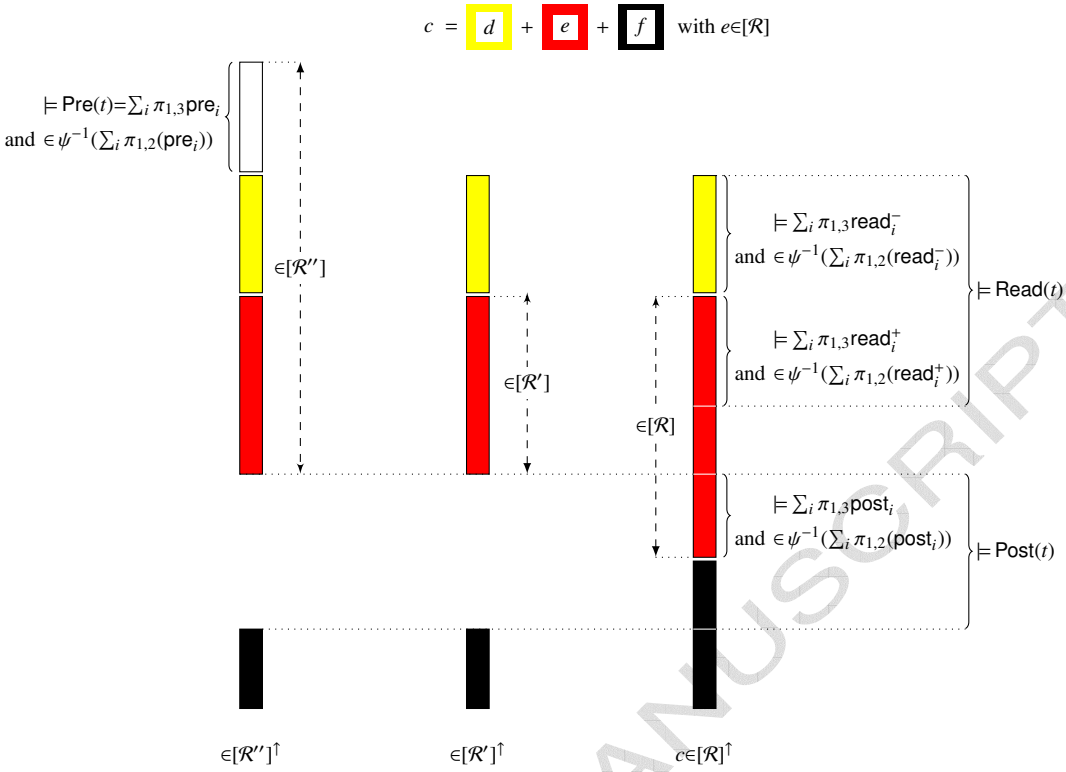


Fig. 2. Decomposition of the set of tokens for the discrete predecessor computation

Hence, time predecessors and discrete predecessors of regions are finite unions of regions, and can be effectively computed, which concludes the proof of the theorem. □

4 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 3 (with a single Büchi accepting condition $p \geq 1$) is a 0-reset, integral and bounded RA-TdPN which recognizes the timed language (of infinite timed words)

$$L_1 = \{(a, \tau_1) \dots (a, \tau_n) \dots \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots \leq 1\}.$$

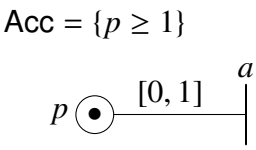


Fig. 3. A RA-TdPN $\mathcal{N}_1$ recognizing L_1

Lemma 1 *The timed language L_1 is recognized by no TdPN.*

Proof. Assume that there is a TdPN $\mathcal{N}$ which recognizes the timed language L_1. We denote by d the least common multiple of the denominators of the constants appearing in the intervals of $\mathcal{N}$. We pick an infinite word $w = (a, \tau_1)(a, \tau_2)\dots(a, \tau_n)\dots$ such that for every $i \geq 1$, $1 - 1/(2d) < \tau_i < \tau_{i+1} < 1$.

The word w is accepted by $\mathcal{N}_1$, and thus by $\mathcal{N}$: there is 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 $(t_{n-1}, \tau_{n-1})\sigma_n(t_n, \tau_n)\dots$ Since $\tau_{n-1} < \tau_n$, there is a suffix $(t'_0, \overline{\tau})(t'_1, \tau_n)\dots(t'_k, \tau_n)(t_n, \tau_n)$ of the timed transition sequence $(t_{n-1}, \tau_{n-1})\sigma_n(t_n, \tau_n)$ with $\overline{\tau} < \tau_n$ (k may be equal to 0). We note σ' the finite prefix of σ up to $(t'_0, \overline{\tau})$, and σ'' the suffix starting right after $(t'_0, \overline{\tau})$ (hence $\sigma = \sigma'\sigma''$). We will prove that the infinite sequence $\widetilde{\sigma} = \sigma'(\sigma'' + 1/(2d))$ is a firing sequence of $\mathcal{N}$ ($\sigma'' + 1/(2d)$ is the timed transition sequence obtained from σ'' 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 $\sigma'' = (t'_1, \tau_n)\dots(t'_k, \tau_n)(t_n, \tau_n)\sigma_{n+1}(t_{n+1}, \tau_{n+1})\dots$ 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 $\widetilde{\sigma}$ to be firable.

We pick a token in place p which, along σ, is produced by some transition t and used for firing a transition t' along σ''. This means in particular that this token is not in *Tok*, and thus that transition t occurs along σ at some date τ with $\tau_1 \leq \tau$. If t is a transition of σ'', then we do not need to modify the initial age of p along $\widetilde{\sigma}$, since t and t' will be separated by the same delay along σ and along $\widetilde{\sigma}$, hence the token p can be used similarly in σ and in $\widetilde{\sigma}$. Otherwise, t occurs along σ' in σ, hence $1 - 1/(2d) < \tau_1 \leq \tau \leq \overline{\tau} < \tau_n \leq \tau' < 1$ where τ' is the date at which t' is fired along σ. We set $\delta = \tau' - \tau$: obviously, $0 < \delta < 1/(2d)$. 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 $h, k \in \mathbb{N}_{\geq 0}$, then $k/d = h/d + \delta$, which is impossible since $0 < \delta < 1/(2d)$. We distinguish between several cases for I^- and I^+:

- 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' along σ is $h/d + \delta \in I^+$. Thus $h < k'$, and we get that $h/d + \delta + 1/(2d) \in I^+$ (since $0 < \delta < 1/(2d)$). In this case, we do not change the initial age of the token for firing the timed transition sequence $\widetilde{\sigma}$, and the firing of t' can be delayed by $1/(2d)$ time units.
- We assume $I^- = (h/d, h'/d)$ and $I^+ = [k/d, k/d]$ with $h < h'$. The age of the token when it is produced (*i.e.*, when transition t is fired) along σ is $k/d - \delta \in I^-$. Thus, $h < k$ and $k/d - \delta - 1/(2d) \in I^-$ since $0 < \delta < 1/(2d)$. For firing the sequence $\widetilde{\sigma}$, we thus change the initial age of the token down to $k/d - \delta - 1/(2d)$, and the firing of t' can then be delayed by $1/(2d)$ time units.
- 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 along σ: $\alpha + \delta\,(\leq k'/d)$ is its age when the token is consumed for firing transition t' along σ. If $\alpha + \delta < k'/d - 1/(2d)$, we do not modify its initial age in $\widetilde{\sigma}$, and the firing of t' can be delayed safely by $1/(2d)$ time units.

  Assume conversely that $\alpha \geq k'/d - 1/(2d) - \delta$. Then, $(k'-1)/d < \alpha < k'/d$, and thus $h \leq k' - 1 < h'$. Along $\widetilde{\sigma}$, choose as new initial age $\alpha' = (k'-1)/d + \beta$ with $0 < \beta < 1/(2d) - \delta$ for the token (when transition t is fired), then we can check that $\alpha' \in I^-$ and $\alpha' + \delta + 1/(2d) \in I^+$, hence the firing of t' can also be delayed by $1/(2d)$ time units.

With these new initial ages for the tokens, the timed transition sequence $\widetilde{\sigma}$ is firable, and accepts the timed word $(a, \tau_1) \dots (a, \tau_{n-1})(a, \tau_n + 1/(2d))(a, \tau_{n+1} + 1/(2d)) \dots$. Moreover, the discrete markings along the run accepting the initial word and the above word are the same, both timed words are thus accepted by $\mathcal{N}$. However this timed word should not be accepted by $\mathcal{N}$ as it is not accepted by $\mathcal{N}_1$ (because $\tau_n + 1/(2d) > 1$), which contradicts the existence of a TdPN $\mathcal{N}$ equivalent to $\mathcal{N}_1$. Thus, there is no classical TdPN which recognizes L_1. □

The timed language L_2. The RA-TdPN $\mathcal{N}_2$ of Figure 4 is an integral bounded RA-TdPN which recognizes the timed language (of infinite timed words)

$$L_2 = \{(a,0)(b,\tau_1)\dots(b,\tau_n)\dots \mid \exists \tau < 1 \text{ s.t. } 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots < \tau\}.$$

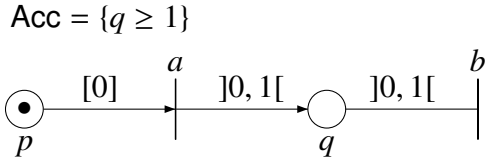


Fig. 4. A RA-TdPN $\mathcal{N}_2$ recognizing L_2

Lemma 2 *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 $w = (a,0)(b,\tau_1)\cdots(b,\tau_i)\ldots$ in 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.

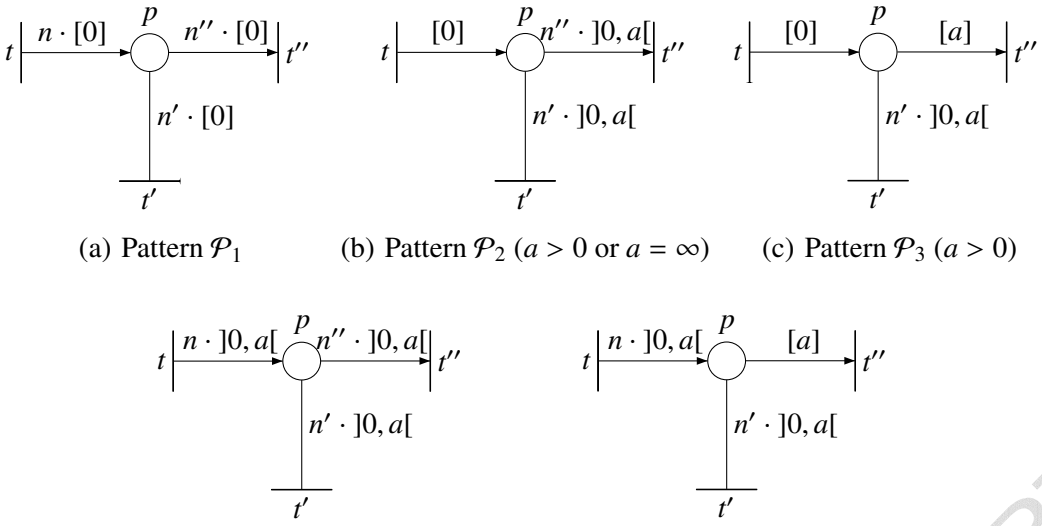
We write $\sigma = \sigma_1\sigma_2$ where σ_1 is an instantaneous firing sequence, and $\sigma_2 = (t_0,d)\overline{\sigma_2}$ for some delay $d > 0$ (hence, t_0 is the first transition along σ which does not occur at date 0). We claim that $\sigma' = \sigma_1\sigma'_2$ where σ'_2 is obtained from σ_2 by delaying all dates by $1-\tau$ time units, 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 when firing t in σ_2 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 $\mathcal{N}_2$. Finally, there is no 0-reset integral RA-TdPN which recognizes the language L_2. □

5 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 construction is decomposed into three steps. The first step consists in splitting intervals so that two intervals are either disjoint or equal. The second step is somehow close to one-dimensional regions of [10], and records ages of tokens and how time elapses. The third step duplicates places so that all pre- (resp post-) arcs connected to a place are labelled by the same interval.

Proposition 1 *For every 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 5, which reads as: "there is an $\boldsymbol{a}$ (which is a positive rational, or is possibly equal to $+\infty$ for patterns $\mathcal{P}_2$ and $\mathcal{P}_4$) such that the place is connected to possibly several post-arcs, pre-arcs and read-arcs, with bags as specified on the figure". Note that parameters n, n′ and n" are not necessarily shared by arcs (whereas a is). Moreover the construction preserves boundedness and integrality.*



(a) Pattern $\mathcal{P}_1$ (b) Pattern $\mathcal{P}_2$ ($a > 0$ or $a = \infty$) (c) Pattern $\mathcal{P}_3$ ($a > 0$)

(d) Pattern $\mathcal{P}_4$ ($a > 0$ or $a = \infty$) (e) Pattern $\mathcal{P}_5$ ($a > 0$)

Fig. 5. The five normalized patterns for an RA-TdPN.

To avoid difficulties due to the initial marking, we first apply a straightforward transformation to the net. We add a place p_{init} containing initially one token 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 any accepted language. In the sequel, we assume that we have already applied this transformation to the net, and we apply the next transformations on each place, except p_{init}.

As announced above, 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, every I_k is either of the form $[a]$ or $]a, b[$ with $a \in \mathbb{Q}_{\ge 0}$ and $b \in \mathbb{Q}_{>0} \cup \{\infty\}$.

(2) For every place,
- either it is 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 $n'' \cdot [0]$.
- or there exists $a \in \mathbb{Q}_{>0}$ such that it is connected to one post-arc whose bag is [0], (possibly) several post-arc labelled by bags $n \cdot]0, a[$, (possibly) several read-arcs labelled by bags $n' \cdot]0, a[$, one pre-arc labelled by a bag $[a]$, and (possibly) several pre-arcs labelled by bags $n'' \cdot]0, a[$.
- or it 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 $n'' \cdot]0, +\infty[$.
(3) Every place is configured as one of the five patterns depicted on Figure 5.

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.

The transformation proceeds as follows: it starts with an RA-TdPN $\mathcal{N}$ and successively builds the three RA-TdPNs $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ obtained respectively by Lemma 3, 4 and 5.

Lemma 3 *We can build a RA-TdPN $\mathcal{N}_1$, equivalent to $\mathcal{N}$, and satisfying restriction* (1).

Proof. Let p be a place of $\mathcal{N}$. 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 $i < j \Rightarrow a_i < a_j$. We then define the set $SI_p = \{[a_1, a_1],]a_1, a_2[, \ldots,]a_{m-1}, a_m[, [a_m, a_m],]a_m, \infty[\}$. W.l.o.g. we assume that $a_1 = 0$. Moreover, to ease the presentation, we define $a_{m+1} = \infty$ and set $a_{m+1} - a_m = \infty$, and write the set SI_p as $SI_p = \{I_k\}_{1 \le k \le K}$. 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 $x = \sum_{1 \le k' \le K'} n_{k'} \cdot J_{k'}$. We will replace the transition t by copies with the same arcs and the same bags except the one which is concerned by the transformation. We denote such copies by t_ϕ, where ϕ is a mapping from $\{1, \ldots, K\} \times \{1, \ldots, K'\}$ to $\mathbb{N}_{\ge 0}$ such that $I_k \cap J_{k'} = \emptyset \Rightarrow \phi(k, k') = 0$ and $\sum_{1 \le k \le K} \phi(k, k') = n_{k'}$. The modified bag is defined by:

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

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

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

Lemma 4 *We can build a RA-TdPN $\mathcal{N}_2$, equivalent to $\mathcal{N}_1$, and satisfying restrictions* (1) *and* (2).

Proof. We iteratively apply the following transformation to each 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 all occurrences 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 $d - a_i \in]0, a_{i+1} - a_i[$ in the new place $p_{a_i,a_{i+1}}$.

Thus we transform an arc connected to p with bag

$$x = n_1 \cdot [a_1, a_1] + n_{1,2} \cdot]a_1, a_2[+ \cdots + n_m \cdot [a_m, a_m] + n_{m,m+1} \cdot]a_m, a_{m+1}[$$

into 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.

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

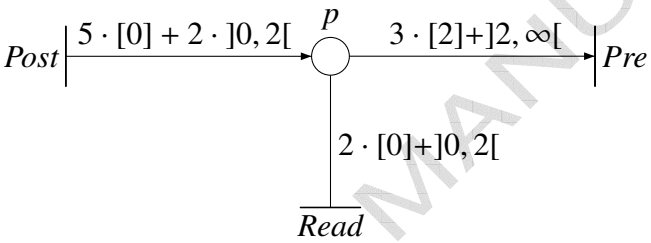
$$\begin{cases} f(p', d) = (p', d) & \text{if } p' \neq p \text{ place of } \mathcal{N}_1 \\ f(p_{a_i}, d) = (p, a_i + d) & \text{for every } p_{a_i} \\ f(p_{a_i,a_{i+1}}, d) = (p, a_i + d) & \text{for every } p_{a_i,a_{i+1}} \end{cases}$$

which we extend on bags by linearity. Note that $f(\nu'_0) = \nu_0$. Straightforwardly, time elapsing commutes with this mapping. Moreover, firing a 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 easily check 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 $\mathcal{N}_1$.

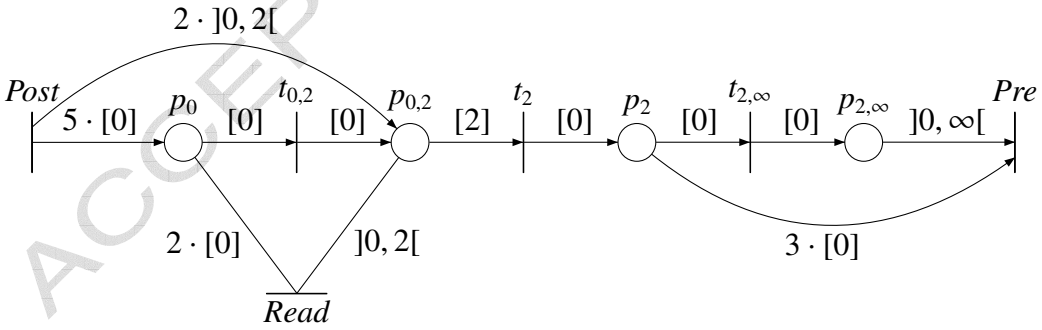
[4] Recall that a configuration ν satisfies an acceptance condition Acc whenever the number of tokens in the places satisfies the constraint of Acc.

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 time 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. □

Example 2 *We illustrate the above construction on the net below:*



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:

$$\xrightarrow{Post} 5 \cdot (p,0) + (p,1) + (p,1.2)$$
$$\xrightarrow{(0.5)} 5 \cdot (p,0.5) + (p,1.5) + (p,1.7)$$
$$\xrightarrow{Post} 5 \cdot (p,0) + 5 \cdot (p,0.5) + 2 \cdot (p,1) + (p,1.5) + (p,1.7)$$
$$\xrightarrow{Post} 10 \cdot (p,0) + 5 \cdot (p,0.5) + 4 \cdot (p,1) + (p,1.5) + (p,1.7)$$
$$\xrightarrow{Read} 10 \cdot (p,0) + 5 \cdot (p,0.5) + 4 \cdot (p,1) + (p,1.5) + (p,1.7)$$
$$\xrightarrow{(1)} 10 \cdot (p,1) + 5 \cdot (p,1.5) + 4 \cdot (p,2) + (p,2.5) + (p,2.7)$$
$$\xrightarrow{Pre} 10 \cdot (p,1) + 5 \cdot (p,1.5) + (p,2) + (p,2.7)$$

In the above sequence, tokens are gathered by age, for example the first bag means that there are seven 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:*

$Post, (t_{0,2})^5, (0.5), Post, Post, Read, (t_{0,2})^{10}, (0.3), t_2, t_{2,\infty}, (0.2), t_2, t_{2,\infty}, (0.5), (t_2)^4, Pre$

Lemma 5 *We can build an* RA-TdPN $\mathcal{N}_3$*, equivalent to* $\mathcal{N}_2$*, and satisfying restrictions* (1)*,* (2)*, and* (3)*.*

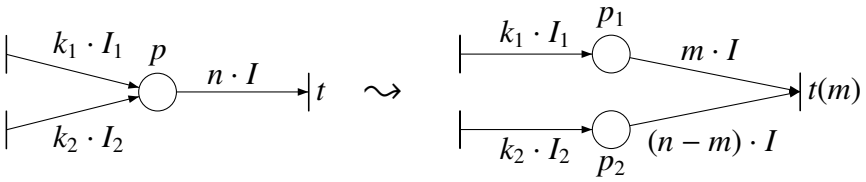
Proof. To prove this lemma, we need to explain how we can transform the snippets built in the proof of the previous lemma into equivalent other snippets where all places have the shape of one of the five patterns of Figure 5. In RA-TdPN $\mathcal{N}_2$ we have [5] ,

- 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 apply successively the following transformations to the different places:

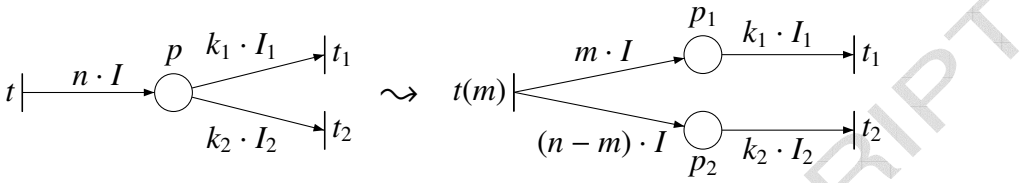
- duplicate the place for each incident post-arc, and duplicate all transitions connected with read- and pre-arcs as depicted on the next picture (transition t can be connected by a pre- or a read-arc):

[5] Parameters n, n' and n'' are not necessarily shared by arcs.



Thus, each transitions connected by a pre- or read-arc is replaced by copies, one for every $m \leq n$ if $n \cdot I$ is the bag labelling the arc between p and t.

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



Thus, each transition connected by a post- or a read-arc is replaced by copies, one for every $m \leq n$ 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 $m_1 + m_2 \leq n$, so we can choose any m such that $m_1 \leq m \leq n - m_2$.

Finally, the places of the resulting net satisfy the property that they are connected to post-arcs (resp. pre-arcs) labelled by the same interval. 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 5. □

Note that all transformations we have presented in this section preserve both boundedness and integrality of the nets. Note also that the transformation is doubly-exponential. This bound may be improved, but here we only focus on expressiveness. This concludes the proof of Proposition 1.

6 Removing the Read-Arcs

In this section, we study the role of read-arcs in RA-TdPNs. Thanks to Lemma 1 (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 6 *Given an 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 $\mathcal{N}'$ leading to 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 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 $\mathcal{N}$ leading to 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 $\mathcal{N}'$. □

6.1 Case of finite words

As announced above, we establish now a result proving that with respect to the equivalence of finite timed words, it is possible, given an RA-TdPN, to build another one which is equivalent. One of the key ideas underlying this construction is the resort to a modification of the acceptance conditions which allows us to add some vivacity to the model. Before stating our result, we illustrate this idea on an example.

Example 3 *We consider the RA-TdPN $\mathcal{N}_1$ depicted on Figure 3. We transform this net into the net illustrated on Figure 6, which recognizes the same language of finite timed words. In this net, initially, transition t_1 labelled by ε puts one token in place p_1 and another one in place p_2. Then a's are produced by firings of the transition t_3, and finally before one time unit has elapsed, the transition t_2 labelled by ε is taken, which empties places p_1 and p_2. This last firing is enforced by the accepting condition $p_1 + p_2 = 0$.*

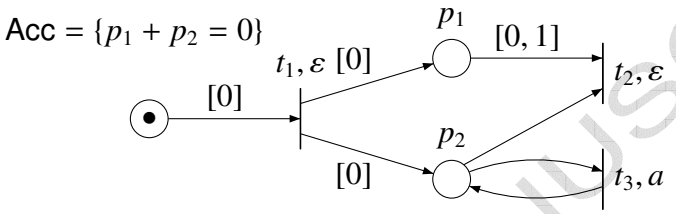


Fig. 6. An illustration of the ideas used for removing the read-arcs.

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 boundedness and integrality of the net.*

Proof. To prove this result, we first normalize the net. We consider only places incident to read-arcs and, thanks to the previous lemma, we can suppose that these places are not in the acceptance condition. We then distinguish between the five possible patterns of Figure 5 for a place p incident to a read-arc, and show that in every case, we can remove the read-arcs incident to place p.

Pattern $\mathcal{P}_1$. The construction is presented on Figure 7. 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.

Pattern $\mathcal{P}_2$. We handle separately the cases $a = +\infty$ and $a < +\infty$. The construction for the first case is presented on Figure 8. For the second case, the construction is presented on Figure 9.

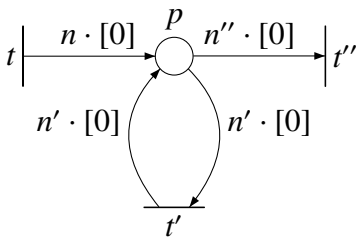


Fig. 7. Removing read-arcs in pattern $\mathcal{P}_1$

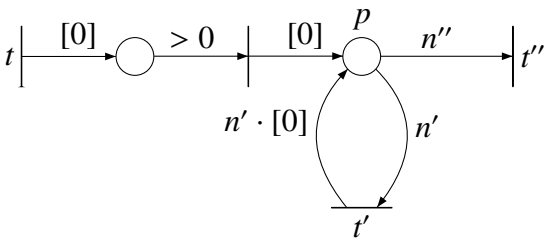


Fig. 8. Removing read-arcs in pattern $\mathcal{P}_2$, case $a = +\infty$

The case $a = +\infty$ is relatively simple. It is indeed sufficient to notice that, once a token has a positive age, it can be used forever by read-arcs and pre-arcs, since its age does not constrain their firings. In particular, we do not modify the accepting condition.

The case $a < +\infty$ is a little bit more involved since we have to take into account the ages of the tokens. Simulating the read-arcs is thus not so easy. To ensure the

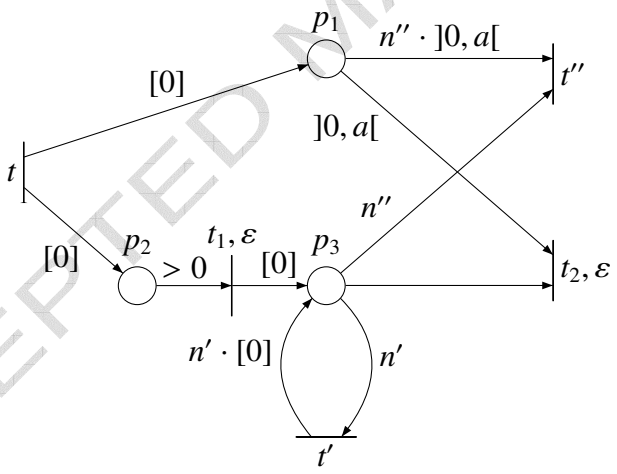


Fig. 9. Removing read-arcs in pattern $\mathcal{P}_2$, case $a < +\infty$

correctness of this construction, we also modify the accepting 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 make 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 the age of these tokens is always strictly less than a.

We first consider an accepting firing sequence σ of $\mathcal{N}$, and build a corresponding accepting firing sequence σ' of $\mathcal{N}'$. We make 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 $a/2$). 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 $\mathcal{N}$. 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 10. 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 also make preliminary remarks

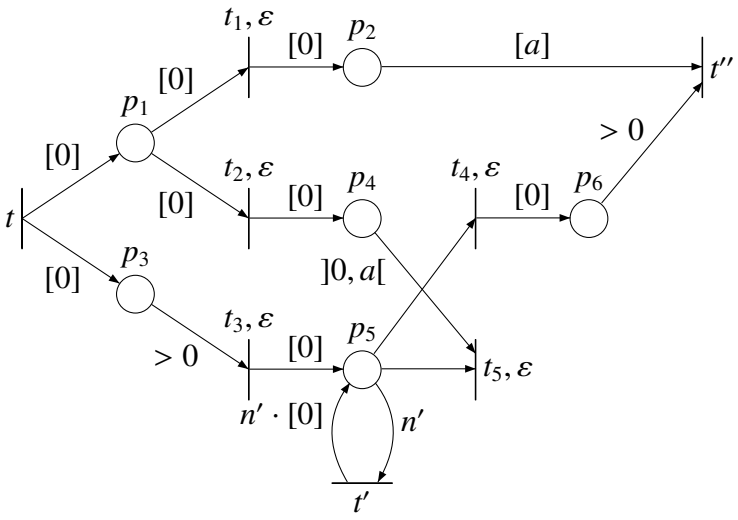


Fig. 10. Removing read-arcs in pattern $\mathcal{P}_3$

on several invariants of the net $\mathcal{N}'$. Every configuration ν appearing on an accepting firing sequence of $\mathcal{N}'$ satisfies the following properties:

$$(i)\ \ \mathsf{size}(\nu(p_1)_{|=0}) + \mathsf{size}(\nu(p_2)_{|=0}) + \mathsf{size}(\nu(p_4)_{|=0}) = \mathsf{size}(\nu(p_3)_{|=0})$$
$$(ii)\ \ \mathsf{size}(\nu(p_2)_{|=a}) \leq \mathsf{size}(\nu(p_6)_{|>0})$$
$$(iii)\ \ \mathsf{size}(\nu(p_2)_{|>0}) + \mathsf{size}(\nu(p_4)_{|>0}) = \mathsf{size}(\nu(p_3)_{|>0}) + \mathsf{size}(\nu(p_5)) + \mathsf{size}(\nu(p_6))$$
$$(iv)\ \ \mathsf{size}(\nu(p_2)_{|]0,a[}) + \mathsf{size}(\nu(p_4)_{|>0}) \geq \mathsf{size}(\nu(p_3)_{|>0}) + \mathsf{size}(\nu(p_5))$$

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 $a/2$). 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.

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 $t_1, \ldots, t_5$. 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 $\mathcal{P}_4$. We distinguish the two cases $a = +\infty$ (Figure 11) and $a < +\infty$ (Figure 12).

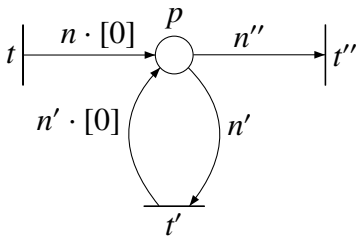


Fig. 11. Removing read-arcs in pattern $\mathcal{P}_4$, case $a = +\infty$

For the case $a = +\infty$, the construction is similar to that for pattern $\mathcal{P}_1$. Indeed, a token produced is immediately and forever available for use since its age does not constrain the firing of transitions. Note that we do not modify the accepting conditions.

As for the pattern $\mathcal{P}_2$, the case $a < +\infty$ is more involved since we have to take into account the ages of the tokens. We also modify the accepting condition of $\mathcal{N}$

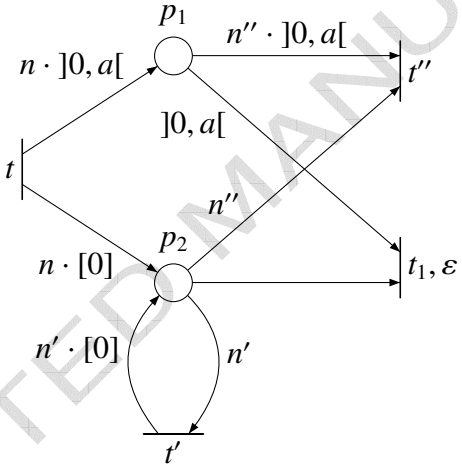


Fig. 12. Removing read-arcs in pattern $\mathcal{P}_4$, case $a < +\infty$

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 easily be derived from the proof for pattern $\mathcal{P}_2$.

Pattern $\mathcal{P}_5$. The construction is presented on Figure 13. We also modify the ac-

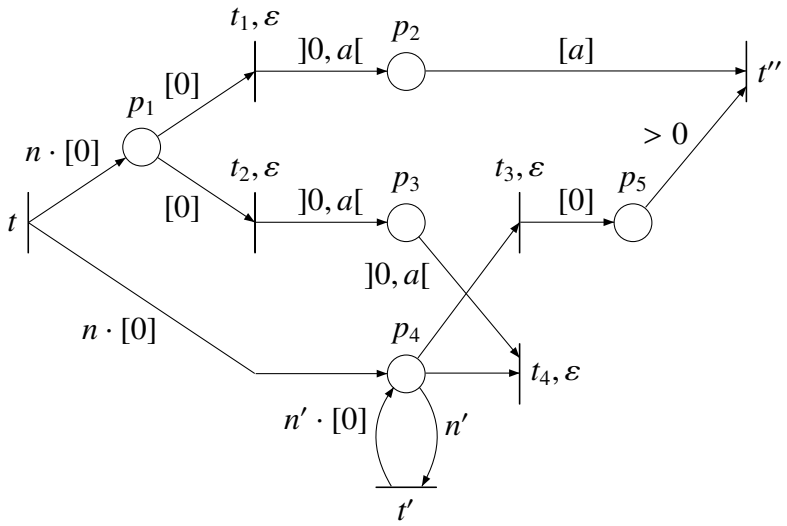


Fig. 13. Removing read-arcs in pattern $\mathcal{P}_5$

cepting condition of $\mathcal{N}$ by adding the following constraint: $\sum_{i=1}^{5} p_i \leq 0$. Pattern $\mathcal{P}_5$ is treated in a way similar to pattern $\mathcal{P}_3$ since pre- and read-arcs are 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 make 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 raises no difficulty. 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.

Then, 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.

The correctness proof for this pattern can easily be derived from the one for pattern $\mathcal{P}_3$. □

6.2 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 ages of tokens have been appropriately simulated.

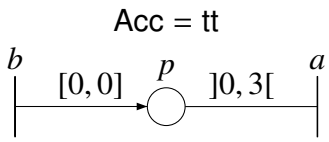


Fig. 14. A RA-TdPN $\mathcal{N}_3$

In the case of infinite timed words, a similar Büchi condition would require that the places of the simulating net are empty infinitely often, but this may not be the case. Consider for example the net $\mathcal{N}_3$ depicted on Figure 14. This net recognizes the following language of infinite timed words :

$$\mathcal{L}^{\omega}(\mathcal{N}_3) = \{w = (a_i, \tau_i)_{i\geq 0} \mid \ a_i = a \Rightarrow \exists j < i, a_j = b \text{ and } \tau_i - \tau_j \in]0, 3[\}$$

In particular the following timed word belongs to $\mathcal{L}(\mathcal{N}_3)$:

$$w = (b, 0)(b, 2)(a, 2)(b, 4)(a, 4) \ldots (b, 2i)(a, 2i) \ldots$$

Any configuration of the execution accepting w always contains a token in place p that needs to be read later on and thus a Büchi condition similar to the one used for finite words would "eliminate" the timed word w. 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. Roughly, this construction consists on this example in creating two copies of the net and producing tokens alternatively in a copy of place p or in the other one. As a consequence, each copy will be empty infinitely often.

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 note that the only cases in which bounds of intervals may be infinite are in patterns $\mathcal{P}_2$ and $\mathcal{P}_4$. Moreover, in these cases, when a is infinite, we have proposed constructions which do not rely on a modification of the acceptance conditions and which are thus also correct for equivalences on infinite timed words. In the sequel, we are thus only interested in cases of finite bounds, *i.e.*, when a is finite.

First we transform $\mathcal{N}$ into another RA-TdPN $\mathcal{N}^*$ as follows. We duplicate every place p connected to a read-arc by an arc labelled with $]0, a[$ (a finite), into two places p_{odd} and p_{even}. Then we apply the following transformation iteratively to every place p and every arc connected to p. Let t be a transition connected to p and $n \cdot I$ be the bag labelling the arc connecting them. We replace t by 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 transition.

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 that we explain now. We select an integer strictly greater than every finite interval bound occurring in $\mathcal{N}^*$ and call it max. Given an infinite sequence σ and a token initially present or produced along the sequence, we say that a token is *useless* in some configuration reached along σ, if it will not be "used" in the remaining sequence by a read-arc or a pre-arc.

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 $(2k) \cdot \text{max}$ with $k \in \mathbb{N}_{\geq 0}$, there is a configuration such that all places p_{even} contain only useless tokens,
- at any time $(2k+1) \cdot \text{max}$ with $k \in \mathbb{N}_{\geq 0}$, there is a configuration such that all places p_{odd} contain only useless tokens.

Note that, due to the (time) divergence of σ, a token produced in some place p (defined as before) will either become useless or it will be consumed in some configuration. This is true because we are concerned with intervals whose bounds are finite. If this configuration occurs in some interval $[(2k+1)\cdot\text{max}, (2k+2)\cdot\text{max}[$, 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 copies: the choice of the copy depends on whether tokens that are read, consumed or produced are even or odd. For instance, an odd (resp. even) token will be produced in the odd (resp. even) copy of the place.

Now take the last configuration of σ^* reached at time $(2k+1) \cdot \text{max}$ and suppose that place p_{odd} contains a token which is not useless yet, then it will become useless during the interval $](2k+1)\cdot\text{max}, (2k+2)\cdot\text{max}[$. 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 $(2k) \cdot \text{max}$ 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 does not respect previous conditions, *i.e.*, that it produces tokens in the wrong copies of the place p, or that it does not consume tokens that are useless. 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 $\mathcal{N}^*$. □

Example 4 (Application of the construction of Theorem 3) *Consider the net $\mathcal{N}_3$ depicted on Figure 14. It is easy to see that the net depicted on Figure 15, say $\mathcal{N}'_3$, is the net obtained by the construction presented in the proof of Theorem 3. Indeed, the only place p of $\mathcal{N}_3$ is configured as pattern $\mathcal{P}_2$. The construction thus consists in duplicating this place into two copies called "Even" and "Odd", and then applying the construction described for finite timed words to each of this copies. The accepting condition is a generalized Büchi condition requiring that the two sets of places obtained respectively for the even copy and for the odd copy are empty infinitely often. Recall that the following word is accepted by $\mathcal{N}_3$.*

$$\mathsf{Acc} = \left\{ \sum_{i=1}^{3} p_e^i = 0, \sum_{i=1}^{3} p_o^i = 0 \right\}$$

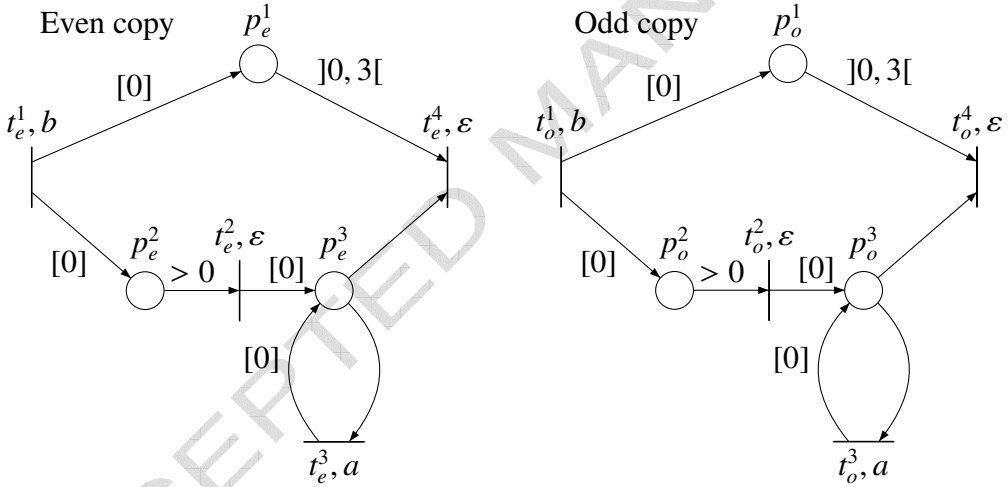


Fig. 15. Application of Theorem 3 to the net $\mathcal{N}_3$

$$w = (b,0)\ (b,2)\ (a,2)\ (b,4)\ (a,4)\ \ldots\ (b,2i)\ (a,2i)\ \ldots$$
$$\textit{odd}\quad \textit{even}\qquad\qquad \textit{even}$$

We give here the corresponding execution σ' in $\mathcal{N}'_3$, as it is defined in the proof of Theorem 3. Note that it could be possible on this example to provide a simpler execution. By definition, we consider 4 *as the constant* max *of the proof. Then a*

token is "even" if it becomes useless in an interval of the form $[(2k+1).4, (2k+2).4[$*, and "odd" otherwise. We have indicated under the occurrences of* b *in* w *whether the produced token is odd or even. Using this information, we can derive the following sequence* σ'*.*

$$\sigma' = (t_o^1, 0)(t_o^2, 2)(t_e^1, 2)(t_o^3, 2)(t_o^4, 2)(t_e^1, 4)(t_e^2, 4)(t_e^3, 4)(t_e^4, 4)(t_o^1, 6)(t_e^2, 6)(t_e^3, 6)(t_e^4, 6)\dots$$

Let us note ν_1 *(respectively* ν_2*) the configuration reached after firing the* 5 *first transitions (respectively* 13*) of* σ'*. It is routine to verify that* ν_1 *satisfies the accepting condition* $\sum_{i=1}^{3} p_o^i = 0$ *and that* ν_2 *satisfies the accepting condition* $\sum_{i=1}^{3} p_e^i = 0$*.*

7 Removing General Resets

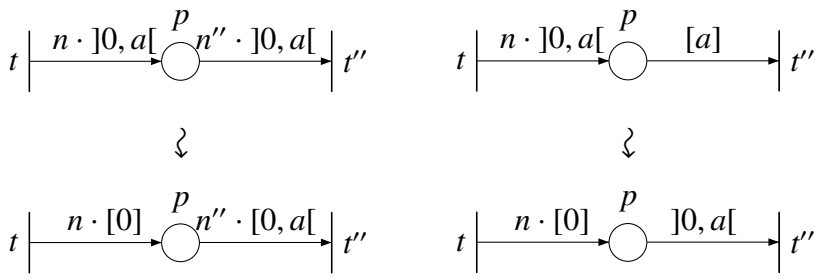
In this section, we study the role of general resets in RA-TdPNs. Thanks to Lemma 2 (language L_2), we know that the class of integral RA-TdPNs is strictly more expressive than the class of 0-reset integral RA-TdPNs for the ω-equivalence. We now prove two results, which show that this is the combination 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 first propose a construction which is correct for TdPNs (*i.e.*, without read-arcs), and which preserves integrality of the net. Then we present a second construction, which is correct 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 boundedness and integrality 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. Observing that the transformation related to Proposition 1 preserves the absence of read-arcs, we can assume that every place p of $\mathcal{N}$ satisfies one of the five patterns of Figure 5, 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 16, 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



(a) Case of pattern $\mathcal{P}_4$ (b) Case of pattern $\mathcal{P}_5$

Fig. 16. Removing general resets in TdPNs.

at age $\frac{a+x}{2} \in]0, a[$ 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. □

The second construction is much more involved, and requires to refine the granularity of the net which is built. 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 boundedness of the net, but* ***not*** *its integrality.*

Proof. First, it it worth noticing that in the case of finite timed words, and non-*Zeno* infinite timed words, this result is a corollary of previous results (Theorems 2, 3 and 4). The construction we explain now, 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, only includes the patterns of 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 5(d) and 5(e)).

Case of pattern $\mathcal{P}_4$. The construction for this case is depicted on Figure 17. We denote $\mathcal{N}'$ the resulting net. We prove now the equivalence of the two nets $\mathcal{N}$ and $\mathcal{N}'$.

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

Let us pick a token of p with initial age δ. Two cases have to be distinguished:

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

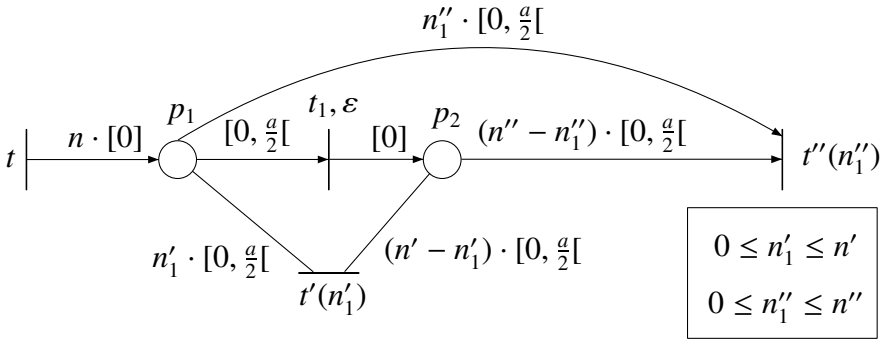


Fig. 17. 0-reset equivalent for pattern $\mathcal{P}_4$

- *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 letting $\frac{\delta' - \delta}{2}$ time units pass. 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 available in p_1 or in p_2 at least as long as it is available in $\mathcal{N}$.

Now the sequence σ' is obtained from σ by inserting the occurrences of the transfer transition and by substituting the appropriate $t'(n_1')$ (resp. $t''(n_1'')$) for t' (resp. t'') depending on the locations of the tokens of p in $\mathcal{N}'$ used by the firing of t' (resp. t'') in $\mathcal{N}$.

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

We simply delete the occurrences of the transfer transition and we substitute the transition t' (resp. t'') for $t'(k_1')$ (resp. $t''(k_1'')$). 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 available in p at least as long as it is available in p_1 or in p_2 of $\mathcal{N}'$, and every firable transition of σ' will thus be firable in σ.

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

Case of pattern $\mathcal{P}_5$. This construction is more involved since read actions and consumptions happen in different intervals $(]0, a[$ and $[a]$ respectively). In order to understand the problem raised by this new constraint, compared to pattern $\mathcal{P}_4$, we start with a wrong simulation (depicted in Figure 18) directly adapted from the previous simulation.

Using a proof similar to the one for pattern $\mathcal{P}_4$, we can show that every firing se-

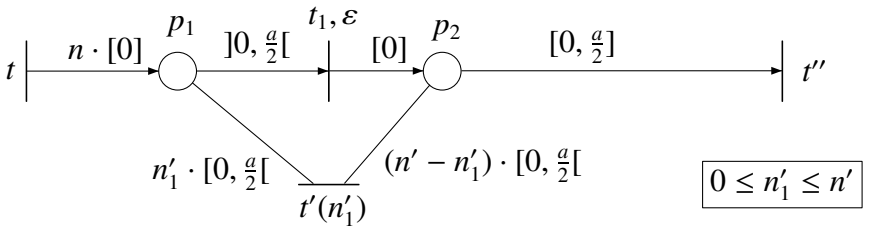


Fig. 18. A wrong 0-reset simulation for pattern $\mathcal{P}_5$

quence σ in $\mathcal{N}$ can be simulated in this net. However the converse is wrong. Indeed, assume for instance that $n = n' = 1$. Then, the firing sequence $(t, 0)(t_1, \frac{a}{4})(t'(1), \frac{a}{4})$ $(t'', \frac{a}{4})$ does not correspond to any sequence in the original net. Indeed if such a sequence did exist then the token produced by t would have an age belonging to $]0, a[$ at time $\frac{a}{4}$ in order to fire t'. But then at time $\frac{a}{4}$, the transition t'' is not firable. The problem with this simulation is that at the same point in time, a token may be used first to simulate a firing of t' and then to simulate the firing of t''.

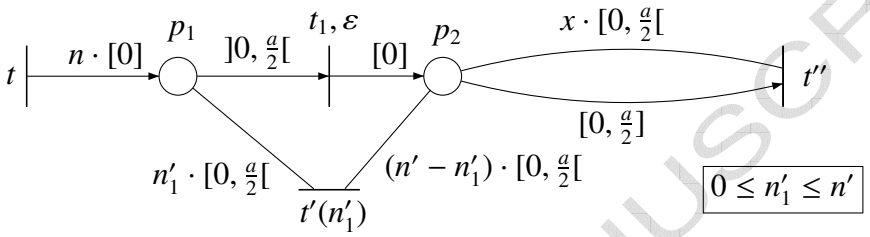


Fig. 19. A 0-reset simulation for pattern $\mathcal{P}_5$ ... with a dynamical weight

We now present a second simulation (depicted in Figure 19) which is correct but uses a "dynamical" weight x on an arc. Let us explain the semantics of x: when firing t'' at some time point τ, x is the maximum value of $n' - n'_1$ corresponding to a previous firing of some $t'(n'_1)$ at date τ. Thus, one avoids the problem faced by the previous simulation, but there are no dynamical weights in the RA-TdPN model. The next (correct) simulation, depicted on Figure 20, mainly consists in simulating such a dynamical weight. We again denote by $\mathcal{N}'$ the resulting RA-TdPN.

Before proving the correctness of the construction, we give some explanations about $\mathcal{N}'$. First, place *ready* is connected to every transition of $\mathcal{N}$ by a read-arc whose bag is [0]. Secondly, we denote by K the largest constant n' appearing on a bag $n' \cdot]0, a[$ of a read-arc and, for every 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 plays three roles. First it schedules the upper part as follows: it makes explicit the alternation between time elapsing and "simulating" instantaneous firing sequences in the upper part of the net. Then before any "simulating" instantaneous firing sequence, it selects the maximal number of tokens that will be simultaneously checked by a firing of t' in this firing sequence (*i.e.*, selects the number k which corresponds to the previous dynamical weight). Finally, after some time has elapsed, it moves tokens from p_1 to p_2 in order to avoid these transfers during the "simulating" instantaneous firing sequences. More precisely, every behaviour of $\mathcal{N}'$ must be a (possibly

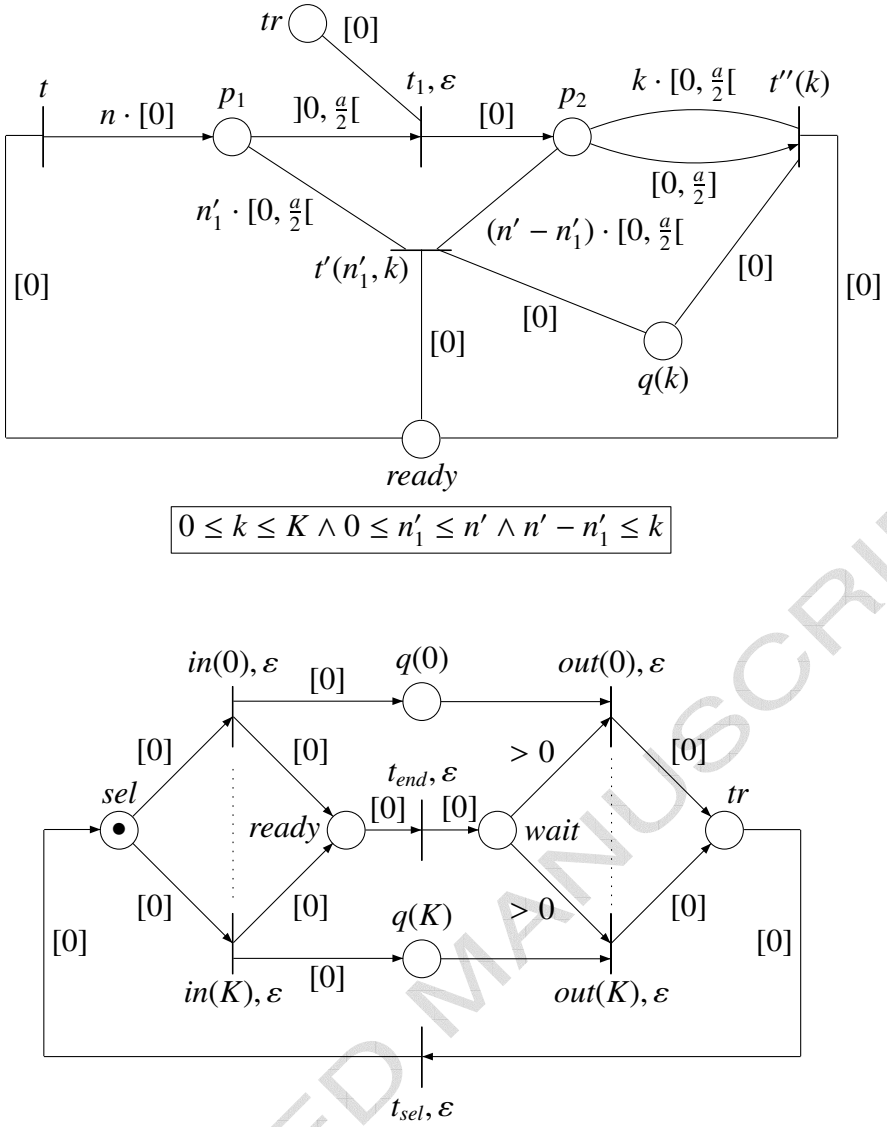


Fig. 20. 0-reset equivalent for pattern $\mathcal{P}_5$.

infinite) iteration of the following sequence:

- First, exactly one of the transitions $in(k)$ is fired, thus putting instantaneously (*i.e.*, in zero delay) 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 zero delay. Then, instantaneously, transition t_{end} is fired and the token in place *ready* is moved to the place *wait*.
- Afterwards, some time elapses, enabling the firing of the silent transition $out(k)$, which picks the token out of the place $q(k)$ and puts a token in place *tr*.
- The upper part of the net can then transfer in zero delay some tokens from p_1 to p_2 using the silent transition t_1.
- Finally, the silent transition t_{sel} is fired instantaneously and puts back the token of the lower part in place *sel*.

We can now prove that the two nets are equivalent.

Let σ be an (infinite) accepting firing sequence in $\mathcal{N}$. We add to this sequence additional information in order to build a sequence σ' in $\mathcal{N}'$ accepting the same timed word. We assume that the sequence includes the intermediate markings and that the tokens in the markings are distinguished (meaning for instance, that if two tokens have the same age, then one of them is the first, the other is the second).

First, we add a (possibly infinite) *transfer date* to all tokens produced in p. Let us pick a token of p with initial age δ produced at time τ. Two cases are possible:

- *First case:* this token will not be consumed by t''. If $\delta \geq \frac{a}{2}$, then we affect to it a transfer date equal to ∞. Otherwise ($0 < \delta < \frac{a}{2}$), its transfer date will be $\tau + \frac{a}{2} - \delta$.
- *Second case:* this token will be consumed by t'' (necessarily when its age is a). Its transfer date will be $\tau + \frac{a-\delta}{2}$.

Let us now consider a maximal instantaneous firing sequence ρ, *i.e.*, a (possibly infinite) maximal subsequence of σ of time length equal to 0. In this subsequence, we add to every occurrence of some transition t' connected to p with a read-arc $n'\cdot]0, a[$, the number of tokens checked by this read-arc which have not yet reached their transfer date, let say n'_1. We affect to the whole subsequence the (finite) maximal value among $n' - n'_1$ for such n'_1 (0 if this set is empty). Let us denote this value k: $k = \max\{n' - n'_1 \mid n'_1 \text{ is attached to } t', t' \in \rho\}$. We have $k \leq K$.

We now build σ' as follows. Let $0 = \tau_0 < \tau_1 < \tau_2 < \cdots$ be the (finite or infinite) sequence of instants corresponding to either a firing of a transition in σ or to a finite transfer date (or both).

In σ', the lower part of the net of Figure 20 "decomposes" time elapsing according to $\tau_0, \tau_1, \tau_2, \ldots$. Let us describe the iterative "behaviour" of σ'. If τ_i corresponds to a firing subsequence of σ then it selects the value k described above by firing $in(k)$, otherwise, it selects 0 by firing $in(0)$. Afterwards, the upper part simulates the maximal subsequence substituting $t'(n'_1, k)$ (resp. $t''(k)$) for t' (resp. t') with n_1 specified above. Then, after firing t_{end}, it lets $\tau_{i+1} - \tau_i$ time units elapse and (after firing $out(k)$), fires t_1 as many times as specified by the number of tokens with transfer date τ_{i+1} and finally fires t_{sel}.

We claim that we obtain in this way a firing sequence in $\mathcal{N}'$ accepting the same timed word. The validity of the firing of a transition $t'(n_1, k)$ is obtained as for pattern $\mathcal{P}_4$. Thus 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 $[0, \frac{a}{2}[$. Due to our choice of firings of the transfer transition t_1, 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 than or equal to $\frac{a}{2}$ 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 the transition t' (resp. t'') for $t'(n'_1, k)$ (resp. $t''(k)$). 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 corresponding occurrence of 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 transition t' with a read-arc labelled by bag $n' \cdot]0, a[$. To be firable, t' 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 $\delta' < \frac{a}{2}$, we have defined above its age in $\mathcal{N}$ as $\frac{a}{2} + \delta'$, which satisfies $\frac{a}{2} + \delta' < a$.

This concludes the proof of the second case. □

8 Summary of Our Expressiveness Results

Case of finite and infinite non-Zeno words. Applying the results of the two previous sections, we get equality of all subclasses of RA-TdPNs mentioned on Fig-

ure 21, 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.

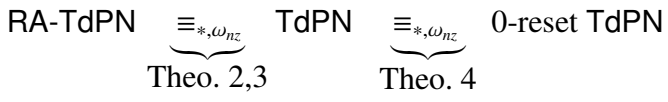


Fig. 21. Relative expressiveness of RA-TdPNs for finite and infinite non-*Zeno* words

Case of infinite words. The picture in the case of infinite timed words is much different (see Figure 22). Indeed the hierarchy in the previous case collapses, whereas we get in that case the lattice depicted on Figure 22. Plain arcs represent strict inclusion, and dashed arcs indicate that the classes are incomparable. Finally note that this picture holds for both bounded and general nets.

9 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.

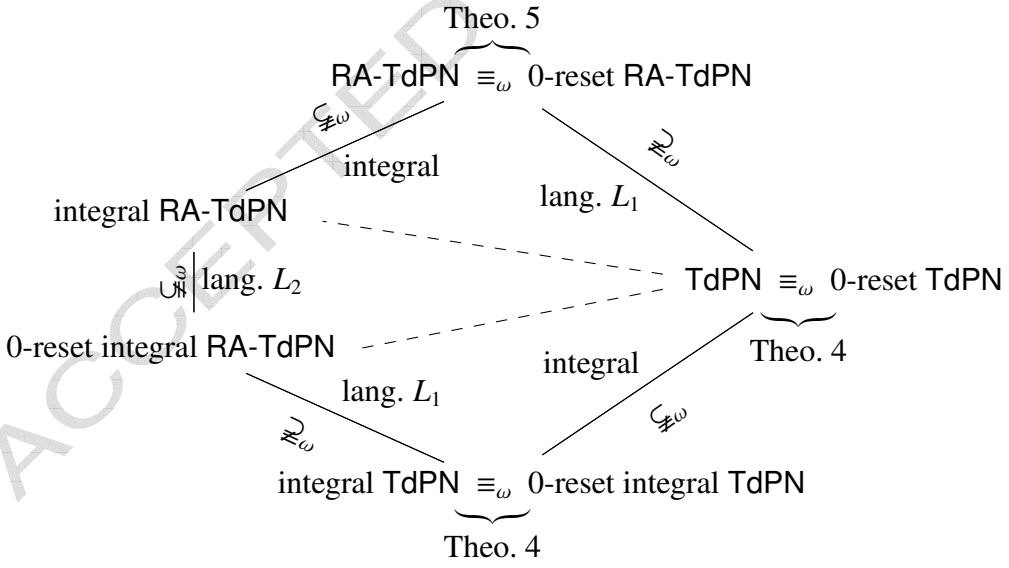


Fig. 22. Relative expressiveness of RA-TdPN for infinite words

Let X be a finite set of variables, which we call *clocks*. We write $C(X)$ 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 6 (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*,* $E \subseteq L \times C(X) \times \Sigma_\varepsilon \times (X \hookrightarrow \mathcal{I}) \times L$ *is a finite set of* edges*,* $A \subseteq 2^L$ *is the accepting condition. 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 $\mathbb{R}^X_{\geq 0}$. 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 $C(X)$ 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 7 (Semantics of a TA) *The semantics of a* TA $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E)$ *is a TTS* $S_{\mathcal{A}} = (Q, q_0, \to)$ *where* $Q = L \times (\mathbb{R}_{\leq 0})^X$*,* $q_0 = (\ell_0, \mathbf{0})$ *and* $\to$ *is defined by:*

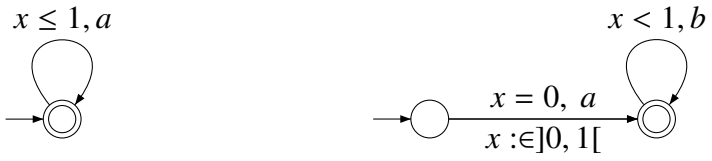
- *either a delay move* $(\ell, v) \xrightarrow{d} (\ell, v + d)$*,*
- *or a discrete move* $(\ell, v) \xrightarrow{e} (\ell', v')$ *iff there exists some* $e = (\ell, \gamma, a, \mu, \ell') \in E$ *s.t.* $v \models \gamma$ *and* $v' \in \mu(v)$*.*

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 $\mathcal{L}^{\omega_{nz}}(\mathcal{A})$, where the acceptance condition is given by the set of finite locations $\bigcup_{F \in A} F$ for finite timed words, and by the generalized Büchi condition A for infinite timed words [6]. We extend the $*$-(resp. ω-, ω_{nz}-)equivalences to TA and to comparisons between subclasses of RA-TdPNs and subclasses of TA.

[6] Here we could use standard Büchi conditions since the classical construction for finite state automata also works for TA.

Two examples of TA are given on Figure 23. Note that the TA $\mathcal{A}_1$ of Figure 23(a) recognizes the timed language L_1 introduced in section 4. Similarly, the TA $\mathcal{A}_2$ of Figure 23(b), which uses a non-deterministic reset of clock x in the interval $]0, 1[$, recognizes the timed language L_2 also introduced there.



(a) A TA $\mathcal{A}_1$ recognizing L_1 (b) A TA $\mathcal{A}_2$ recognizing L_2

Fig. 23. Two examples of timed automata

9.2 TA and Bounded RA-TdPNs.

Our aim was to compare the relative expressiveness of RA-TdPNs and TA. In this subsection, we prove the equivalence between bounded RA-TdPNs and TA. In this context, the following result has been obtained by Jiří Srba:

Theorem 6 ([17]) *Safe RA-TdPNs and TA are $\{*, \omega_{nz}, \omega\}$-equivalent.* [7]

We strengthen the above result and prove that this also holds for bounded RA-TdPNs.

Theorem 7 *Bounded RA-TdPNs and TA are $\{*, \omega_{nz}, \omega\}$-equivalent. Moreover, the translation preserves integrality and 0-reset.*

To improve readability, we however give here a self-contained proof of the complete result.

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 $\mathcal{A}$.

Copies of places. 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 in place p. To ensure that these places are safe, we use the complementary places p_i^0. We make these

[7] The result proved in [17] is even stronger because the equivalence considered is not a language equivalence, but isomorphism of timed transition systems.

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.

Copies of transitions. Let us consider a transition t of $\mathcal{N}$. Transition t will be replaced by copies. $\mathsf{Pre}(t)(p)$ (resp. $\mathsf{Read}(t)(p)$, $\mathsf{Post}(t)(p)$) is a bag in $\mathsf{Bag}(\mathcal{I})$, whose size is denoted by $s(p)$ (resp. $s'(p)$, $s''(p)$). We order the tokens in these bags and assume that $\mathsf{Pre}(t)(p) = I_1 + \ldots + I_{s(p)}$, $\mathsf{Read}(t)(p) = I'_1 + \ldots + I'_{s'(p)}$ and $\mathsf{Post}(t)(p) = I''_1 + \ldots + I''_{s''(p)}$. The copies of t are parameterized by three functions indicating for every place p in which copies of the place p the tokens should be consumed (resp. read, produced).

Pre-arcs. For every place p such that $s(p) > 0$. Consider an injective function ζ_p defined from $\{1, \ldots, s(p)\}$ into $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the pre-arc between t and p will consume the $s(p)$ tokens.

Read-arcs. For every place p such that $s'(p) > 0$. Consider an injective function ζ'_p defined from $\{1, \ldots, s'(p)\}$ into $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the read-arc between t and p will read the $s'(p)$ tokens.

Post-arcs. For every place p such that $s''(p) > 0$. Consider an injective function ζ''_p defined from $\{1, \ldots, s''(p)\}$ into $\mathbb{N}_k = \{1, \ldots, k\}$. This function defines in which places the post-arc between t and p will produce the $s''(p)$ tokens.

We now define the function ζ (resp. ζ', ζ'') as the function mapping each place $p \in P$ to the function ζ_p (resp. ζ'_p, ζ''_p).

Suppose moreover that these three functions satisfy the following conditions:

$$\forall p \in P, \begin{cases} \zeta_p \text{ and } \zeta'_p \text{ have disjoint images,} \\ \zeta'_p \text{ and } \zeta''_p \text{ have disjoint images.} \end{cases}$$

These conditions simply require that for every place p, the simulation of t does not try to consume and read a token in the same copy of the place p (resp. does not try to read and produce a token in the same copy of the place p).

For every 3-tuple of such functions (ζ, ζ', ζ'') verifying these conditions, we add to the new net the transition $t' = t_{\zeta,\zeta',\zeta''}$ defined, for every place $p \in P$, by

$$\forall i \in \{1, \dots, s(p)\}, \quad \begin{cases} \mathsf{Pre}(t')(p^1_{\zeta_p(i)}) = I_i \\ \mathsf{Post}(t')(p^0_{\zeta_p(i)}) = [0] \end{cases}$$

$$\forall j \in \{1, \dots, s'(p)\}, \ \mathsf{Read}(t')(p^1_{\zeta'_p(j)}) = I'_j$$

$$\forall l \in \{1, \dots, s''(p)\}, \begin{cases} \mathsf{Pre}(t')(p^0_{\zeta''_p(l)}) = \mathbb{R}_{\geq 0} \\ \mathsf{Post}(t')(p^1_{\zeta''_p(l)}) = I''_l \end{cases}$$

Moreover, the label of $t_{\zeta,\zeta',\zeta''}$ is the one of t.

Given a place p, the arcs connecting transition $t_{\zeta,\zeta',\zeta''}$ to copies of p are represented on Figure 24.

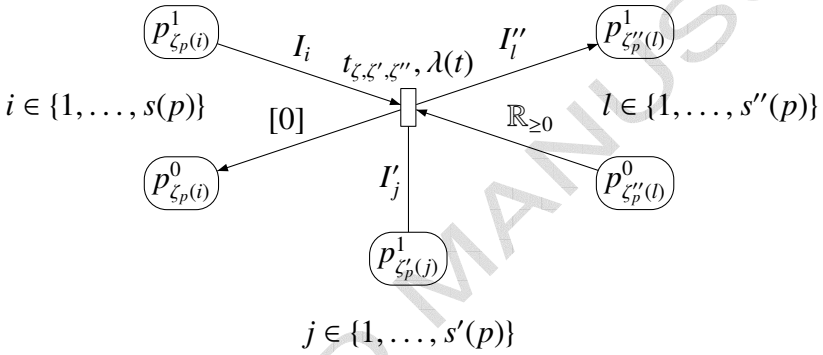


Fig. 24. Simulating a bounded RA-TdPN using a safe RA-TdPN

Initial marking. Given the original initial marking $M_0 \in \mathsf{Bag}(P)$, the new initial marking M'_0 is defined by

$$M'_0 = \sum_{p \in P} \sum_{i=1}^{M_0(p)} p^1_i.$$

Acceptance condition. 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 $\sum_{i=1}^{k} p^1_i$.

It is easy to verify that this transformation provides a structurally 1-safe RA-TdPN $\mathcal{N}'$ which is strongly bisimilar to $\mathcal{N}$. The fact that $\mathcal{N}'$ is 1-safe is obvious by construction (recall that places p^0_i and p^1_i are mutually exclusive). The existence of a bisimulation relation relies on the fact that a configuration with n tokens in place p is encoded by a configuration where n places p^1_i contains 1 token (whose ages are

the ones of tokens of p) whereas for the $k - n$ other i's, there is 1 token in place p_i^0 (with arbitrary ages). It is then easy to prove that t is firable from a configuration ν of $\mathcal{N}$ if and only if there exists a copy of t which is firable from a corresponding configuration in $\mathcal{N}'$. Since the initial markings and the acceptance conditions are preserved by the bisimulation, the strong bisimulation implies 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 define a TA $\mathcal{A} = (L, \ell_0, X, \Sigma_\varepsilon, E, F, A)$ equivalent to $\mathcal{N}$. By notation misuse, 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 every 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 $\mathsf{Acc} = \{\mathsf{acc}_1, \ldots, \mathsf{acc}_k\}$, A is defined as the set of formulas $\{A_1, \ldots, A_k\}$ where for every $1 \leq i \leq k$, $A_i = \left\{\ell \in 2^P \mid \left(\bigwedge_{q \in \ell} q = 1 \wedge \bigwedge_{q \notin \ell} q = 0\right) \Rightarrow \mathsf{acc}_i\right\}$.

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, $\mathsf{Read}(e)(x) = g_{|x}$,

- $\mathsf{Pre}(e)(\ell) = \mathbb{R}_{\geq 0}$, $\mathsf{Post}(e)(\ell') = [0]$,
- $\lambda(e) = a$,

- if $A = \{A_1, \ldots, A_k\}$, then Acc is the set $\{\mathsf{acc}_1, \ldots, \mathsf{acc}_k\}$ where for every $1 \leq i \leq k$, $\mathsf{acc}_i = \bigwedge_{\ell \in A_i} \ell = 1$

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

$$(\ell, val)\ \mathcal{R}\ \nu \text{ iff } \begin{cases} \mathsf{size}(\nu(\ell)) > 0 & \\ \mathsf{size}(\nu(\ell')) = 0 & \forall \ell' \neq \ell \\ \nu(x) = 1 \cdot val(x) & \forall x \in X, \end{cases}$$

where $(\ell, val) \in L \times \mathbb{R}_{\geq 0}^X$ is a configuration of $\mathcal{A}$, and $\nu \in \mathsf{Bag}(\mathbb{R}_{\geq 0})^P$ 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 $\ell \in L$. This justifies the definition of Acc. Moreover, it is easy to verify that the net we have constructed is safe, thus bounded. □

Example 5 *We illustrate the transformation of a TA into a bounded RA-TdPN on the automaton depicted on Figure 25.*

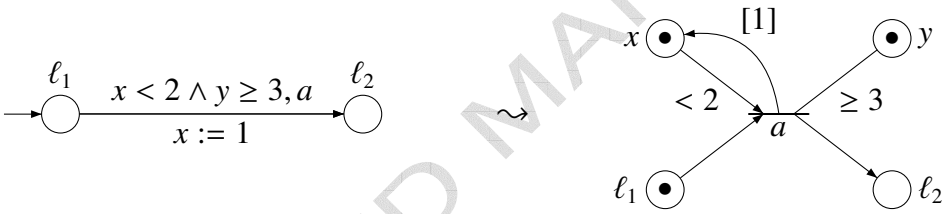


Fig. 25. An example of the construction from TA to safe RA-TdPNs.

9.3 Expressiveness Results for TA

Combining the previous 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 TdPNs and TA are equally expressive;
(2) (integral) TA and 0-reset (integral) TA are equally expressive.

Corollary 2 *For the ω-equivalence,*

(3) TdPNs and TA are incomparable; [8]
(4) TA are strictly more expressive than bounded TdPNs;
(5) integral TA are strictly more expressive than integral 0*-reset TA;*
(6) TA and 0*-reset TA are equally expressive.*

As a folklore theorem, 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* timed words (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 on Figure 26 by giving a 0-reset TA ω-equivalent to the timed automaton of Figure 23(b).

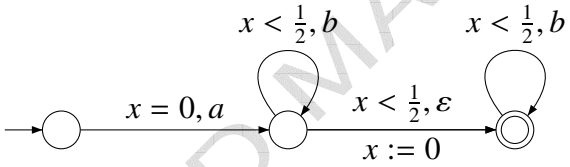


Fig. 26. An example of the construction for removing general resets in TA.

10 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 makes the model of RA-TdPNs (introduced earlier in [17]) very interesting, as it unifies TA and TdPNs while it enjoys the interesting property that coverability is still decidable. Surprisingly, this rather general model also enjoys nice expressiveness properties.

[8] Recall that (untimed) Petri nets may recognize non regular languages, unlike timed automata whose untimed languages are always regular.

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. This implies in particular that, in this context, the classical assumption which consists in forgetting *Zeno* behaviours is restrictive. 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 only use 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-TdPNs 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 is to develop unfolding techniques for RA-TdPNs, taking advantage of the locality of the firing rules. A first step in that direction is [7], where we have extended the seminal work of McMillan [13] to networks of timed automata with invariants (using some ideas presented in this paper for translating timed autamata to RA-TdPNs). Note that read-arcs increase concurrency between events, but they require some attention when building unfoldings [20,21]. Another possible research direction is to study other kinds of arcs, for instance arcs which do not reset ages of tokens when moving the tokens from one place to another one.

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 Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 58–70. Springer, 2004.

[2] P. A. Abdulla and A. Nylén. Timed Petri nets and bqos. In *Proc. 22nd International Conference on Application and Theory of Petri Nets (ICATPN'01)*, volume 2075 of *Lecture Notes in Computer Science*, 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] P. Bouyer, S. Haddad, and P.-A. Reynier. Timed Petri nets and timed automata: On the discriminating power of Zeno sequences. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 420–431. Springer, 2006.

[7] P. Bouyer, S. Haddad, and P.-A. Reynier. Timed unfoldings of networks of timed automata. In *Proc. 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06)*, volume 4218 of *Lecture Notes in Computer Science*, pages 292–306. Springer, 2006.

[8] C. Girault and R. Valk, editors. *Petri Nets for Systems Engineering*. Springer, 2002.

[9] G. Higman. Ordering by divisibility in abstract algebras. In *Proc. London Math. Soc.*, volume 2, pages 326–336, 1952.

[10] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science*, pages 387–401. Springer, 2004.

[11] 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.

[12] P. Mahata. *Model Checking Parameterized Timed Systems*. PhD thesis, Department of Information Technology, Uppsala University, Uppsala, Sweden, 2005.

[13] K. McMillan. A technique of state space search based on unfolding. *Formal Methods in Syst. Design*, 6(1):45–65, 1995.

[14] U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6):545–596, 1995.

[15] 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.

[16] 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.

[17] J. Srba. Timed-arc petri nets vs. networks of timed automata. In *Proc. 26th International Conference Application and Theory of Petri Nets (ICATPN'05)*, volume 3536 of *Lecture Notes in Computer Science*, pages 385–402. Springer, 2005.

[18] V. Valero, D. Frutos-Escrig, and F. R. F. . Cuartero. 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.

[19] W. Vogler. Efficiency of asynchronous systems, read arcs, and the MUTEX-problem. *Theor. Comput. Sci.*, 275(1–2):589–631, 2002.

[20] W. Vogler, A. L. Semenov, and A. Yakovlev. Unfolding and finite prefix for nets with read arcs. In *Proc. 9th Int. Conf. Concurrency Theory (CONCUR'98)*, volume 1466 of *LNCS*, pages 501–516. Springer, 1998.

[21] J. Winkowski. Reachability in contextual nets. *Fundam. Inform.*, 51(1-2):235–250, 2002.