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

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

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

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

1 Introduction

Timed automata (TA) [3] are a well-accepted model for representing and analyzing real-time systems: they extend finite automata with clock variables which give timing constraints on the behaviour of the system. Another prominent formalism for the design and analysis of discrete-event systems is the model of *Petri nets* (PN) [8]. Thus, in order to model concurrent systems with constraints on time, several timed extensions of PNs have been proposed as a possible alternative to TA.

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

Timed Petri nets (TdPN), also called *timed-arc Petri nets*, associate with each arc an interval (or bag of intervals) [12]. In TdPNs, each token has an age. This age is initially set to a value belonging to the interval of the arc which has produced it or set to zero if it belongs to the initial marking. Afterwards, ages of tokens evolve synchronously with time. A transition may be fired if tokens with age belonging to the intervals of its input arcs may be found in the current configuration. Note that "old" tokens may die (*i.e.* they cannot be used anymore for firing a transition but they remain in the place), and that conditions for firing transitions are thus local and do not depend on the global configuration of the system, like in PNs. This "lazy" behaviour has important consequences.

* Work partially supported by ACI SI Cortos, a program of the French Ministry of Research.

Whereas the reachability problem is undecidable for TdPNs [12], the coverability problem [2] and some significant other ones are decidable [1]. Furthermore, TdPNs cannot be transformed into equivalent TA (for the language equivalence), since the untimed languages of the latter model are regular. However the question whether (bounded) TdPNs are more expressive than TA w.r.t. language equivalence was not known.

Our contributions. In this paper, we answer negatively this question, and propose an extension of TdPNs with *read-arcs*[3], yielding the model of *read-arc timed Petri nets* (RA-TdPN). This feature has already been introduced in the untimed framework [10] in order to define a more refined concurrent semantics for nets. However, in the untimed framework, for the interleaving semantics, they do not add any expressive power as they can be replaced by two arcs which check that a token is in the place and replace it immediately. First, we investigate the decidability of the coverability problem for the RA-TdPN model, and we prove that it remains decidable.

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

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

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

Due to lack of space, proofs are omitted, but can be found in [6].

2 Read-Arc Timed Petri Nets

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

Bags. Given a set $\mathcal{E}$, $\mathsf{Bag}(\mathcal{E})$ denotes the set of mappings f from $\mathcal{E}$ to $\mathbb{N}$ s.t. the set $\mathsf{dom}(f) = \{x \in \mathcal{E} \mid f(x) \neq 0\}$ is finite. We note $\mathsf{size}(f) = \sum_{x \in \mathcal{E}} f(x)$. Let $x, y \in$

[3] A similar extension has been proposed independently by Srba in [11].

$\mathsf{Bag}(\mathcal{E})$, then $y \leq x$ iff $\forall e \in \mathcal{E}, y(e) \leq x(e)$. If $y \leq x$, then $x - y \in \mathsf{Bag}(\mathcal{E})$ is defined by: $\forall e \in \mathcal{E}, (x-y)(e) = x(e) - y(e)$. For $d \in \mathbb{R}_{\geq 0}$ and $x \in \mathsf{Bag}(\mathbb{R}_{\geq 0})$ $x + d \in \mathsf{Bag}(\mathbb{R}_{\geq 0})$ is defined by $\forall \tau < d, (x+d)(\tau) = 0$ and $\forall \tau \geq d, (x+d)(\tau) = x(\tau - d)$. Let $x \in \mathsf{Bag}(\mathcal{E}_1 \times \mathcal{E}_2)$. The bags $\pi_i(x) \in \mathsf{Bag}(\mathcal{E}_i)$ for $i = 1, 2$ are defined by: for all $e_1 \in \mathcal{E}_1$, $\pi_1(x)(e_1) = \sum_{e_2 \in \mathcal{E}_2} x(e_1, e_2)$, and similarly for π_2.

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

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

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

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

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

A *configuration* ν of a RA-TdPN is an item of $\mathsf{Bag}(\mathbb{R}_{\geq 0})^P$ (or equivalently $\mathsf{Bag}(P \times \mathbb{R}_{\geq 0})$). Intuitively, a configuration is a marking extended with age information for the tokens. We will write (p, x) for a token which is in place p and whose age is x. A configuration is then a finite sum of such pairs. Then a token (p, x) belongs to configuration ν whenever $(p, x) \leq \nu$ (in terms of bags). The *initial configuration* $\nu_0 \in \mathsf{Bag}(\mathbb{R}^P_{\geq 0})$ is defined as $\forall p \in P$, $\nu_0(p) = m_0(p) \cdot 0$ (there are $m_0(p)$ tokens of age 0 in place p).

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

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

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

A *path* in the 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. A *timed transition sequence* is a (finite or infinite) timed word over alphabet T, the set of transitions of $\mathcal{N}$. A *firing sequence* is a timed transition sequence $(t_1, \tau_1)(t_2, \tau_2)\ldots$ such that $\nu_0 \xrightarrow{\tau_1} \nu'_1 \xrightarrow{t_1} \nu_1 \xrightarrow{\tau_2 - \tau_1} \nu'_2 \xrightarrow{t_2} \nu_2 \ldots$ is a path. If $(p, x) \leq \nu$ is a token of a configuration ν, it is a *dead token* whenever for every interval I labelling a pre- or a read-arc of p, x is above I.

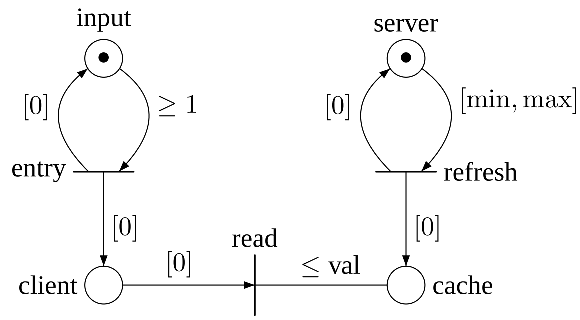
Petri nets can be considered as language acceptors. The timed word which is read along a path $\nu_0 \xrightarrow{d_1} \nu'_1 \xrightarrow{t_1} \nu_1 \xrightarrow{d_2} \nu'_2 \xrightarrow{t_2} \nu_2 \ldots$ is the projection over Σ of the timed word $(\lambda(t_1), d_1)(\lambda(t_2), d_1 + d_2)\ldots$

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

Two 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 three equivalences altogether. *Idem* for "$\{*, \omega_{nz}\}$-equivalence" and other combinations.

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

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



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

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

– *a* timed Petri net *(TdPN for short)*[4] *if for all $t \in T$, $\mathit{size}(\mathit{Read}(t)) = 0$,*
– integral *if all intervals appearing in bags of $\mathcal{N}$ are in $\mathcal{I}_{\mathbb{N}}$,*
– 0-reset *if for all $t \in T$, for all $p \in P$, $I \neq [0, 0] \Rightarrow I \notin \mathit{dom}(\mathit{Post}(t)(p))$,*
– k-bounded *if all configurations ν appearing along a firing sequence of $\mathcal{N}$ are such that for every place $p \in P$, $\mathit{size}(\nu(p)) \leq k$,*
– bounded *if there exists $k \in \mathbb{N}$ such that $\mathcal{N}$ is k-bounded,*
– safe *if it is 1-bounded.*

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

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

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

The proof of this theorem is an extension of the proof done in [9] for TdPNs, based on an extension of classical regions in timed automata [3].

3 Relative Expressiveness of Subclasses of RA-TdPNs

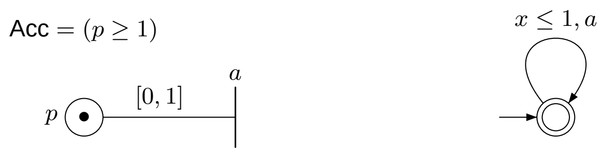
In this section, we thoroughly study the relative expressiveness of subclasses of RA-TdPNs, by distinguishing whether they are bounded, integral, 0-reset, or whether they can be expressed without read-arcs. Surprisingly the results depend on the language equivalence we consider, and whereas finite timed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

[4] This is the standard model, as defined in [12].

3.1 Two Discriminating Timed Languages

We design two timed languages which distinguish between several subclasses of RA-TdPNs. Notice that these two languages are *Zeno*.

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

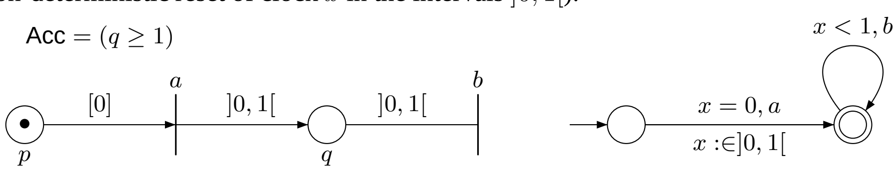


(a) A RA-TdPN $\mathcal{N}_1$ recognizing L_1 (b) A TA $\mathcal{A}_1$ recognizing L_1

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

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

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



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

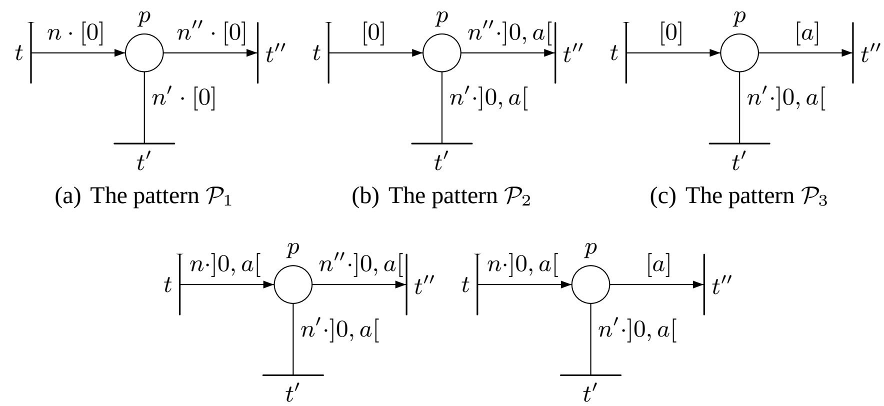
Fig. 2. A language L_2 not recognized by any 0-reset integral RA-TdPN

Lemma 2. *The timed language L_2 is recognized by no 0-reset integral RA-TdPN.*

3.2 Normalization of RA-TdPNs

We present a transformation of RA-TdPNs which preserves both languages over finite and (*Zeno* or non-*Zeno*) infinite words, as well as boundedness and integrality of the nets. This construction transforms the net by imposing strong syntactical conditions on places, which will simplify further studies of RA-TdPNs.

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



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

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

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

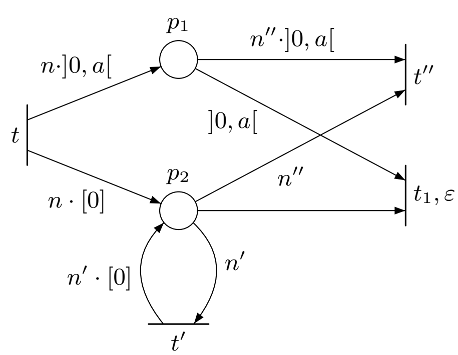
3.3 Removing the Read-Arcs

In thus subsection, 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.

Case of finite words. We state the following result.

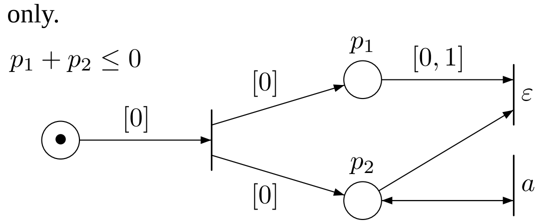
Theorem 2. *Let $\mathcal{N}$ be an RA-TdPN, then we can effectively build a TdPN $\mathcal{N}'$, which is $*$-equivalent to $\mathcal{N}$. Note that the construction preserves the boundedness and integrality properties of the nets.*

Proof (Sketch). To prove this result, we first normalize the net. We then distinguish the five possible patterns of Fig. 3 for a place p, and show that in every case, we can remove the read-arcs connected to place p. The construction for pattern $\mathcal{P}_4$ is given on the next picture.



The accepting condition is reinforced by the constraint $p_1 + p_2 \leq 0$, thus imposing to consume (by t'' or t_1) every token produced by t. The idea of this construction is to check pre-arcs with tokens which are in place p_1 and to check read-arcs with tokens in place p_2, but with no timing constraints (there is no sense to check the age of the tokens in p_2 since it is reset each time a read-arc checks the presence of a token in the place). *A posteriori*, before tokens are dead (thus before their age reaches a), they will be consumed by transition t'' or t_1, together with one token in place p_1. □

We illustrate the construction on the RA-TdPN $\mathcal{N}_1$ of Fig. 1(a). It is correct for finite timed words only.



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

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

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

3.4 Removing General Resets

In this subsection, 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 then prove two results, which show that this is the combination of the presence of read-arcs together with the integrality property which explains the expressiveness gap between 0-reset nets and nets with general resets. Indeed, we design a first construction which holds if there is no read-arc, and which preserves integrality of the net. Then we design a second construction, which holds even for nets with read-arcs, but which does not preserve the integrality of the nets.

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

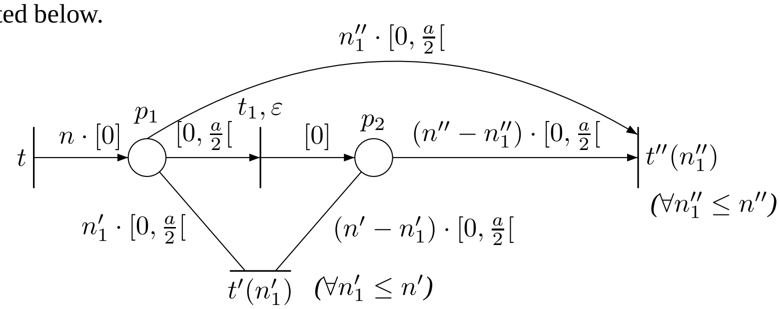
This result is not difficult and consists in shifting intervals of pre-arcs connected to a place, depending on the intervals which label post-arcs connected to this place.

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

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

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

Let $\mathcal{N}$ be a RA-TdPN which we assume satisfies Proposition 1. The only places of $\mathcal{N}$ which are connected to non 0-reset post-arcs are those which satisfy pattern $\mathcal{P}_4$ or pattern $\mathcal{P}_5$ (Fig. 3(d) and 3(e)). Here, we only present the construction for pattern $\mathcal{P}_4$, it is depicted below.



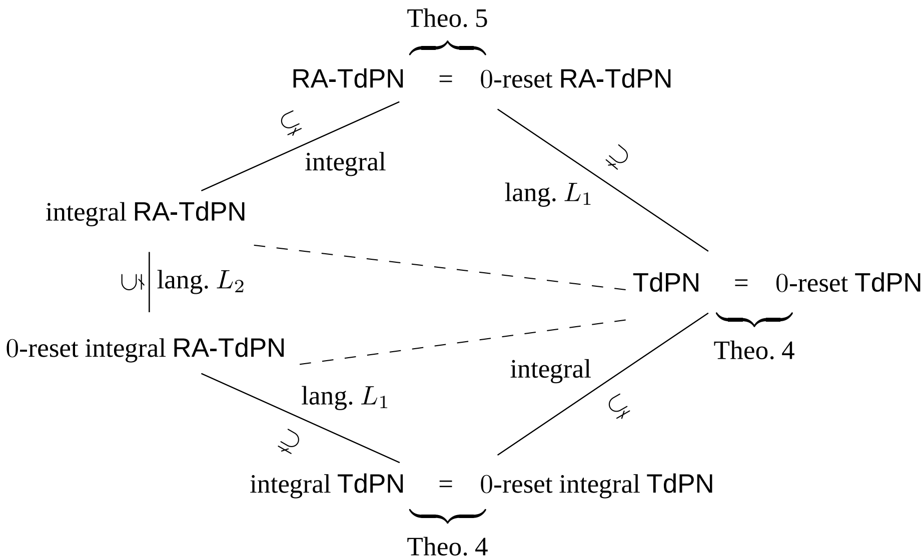
A token which enters place p in the original net (and which will not die) will either be consumed by transition t'' before $\frac{a}{2}$ units of time has elapsed, or after a delay which is greater than $\frac{a}{2}$ but strictly less than 1. In the first case, the token can stay in place p_1 (place in which it can be used by a read-arc) and leave when it is consumed by transition t''. In the second case, the token will stay in place p_1 for some amount of time, and then go to place p_2 where it can also be consumed by transition t''. The read-arc can read tokens in place p_1 or in place p_2 with the constraint that ages of the token are in the interval $[0, \frac{a}{2}[$. □

3.5 Summary of Our Expressiveness Results

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

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

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

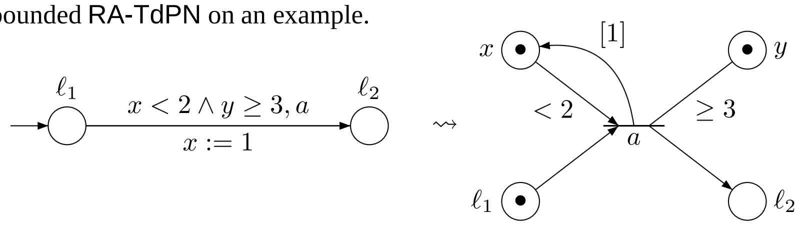


4 Application to Timed Automata

First defined in [3], the model of timed automata (TA) associates with a finite automaton a finite set of non negative real-valued variables called *clocks*. We assume the reader is familiar with TA, and refer to [5] for a formal definition (we allow, in addition to classical resets to 0 of clocks, general resets of the form $x :\in I$ if $I \in \mathcal{I}$ which sets a clock to a value non-deterministically chosen in I). Two examples of TA are given on Fig. 1(b) and 2(b). The following theorem, close to a result by Srba [11], relates TA and bounded RA-TdPNs.

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

Proof (Sketch). For transforming a bounded RA-TdPNs into an equivalent TA, we first build a safe RA-TdPN, and then a TA, in which a clock is associated with a place and records the age of the token in the place. We illustrate the transformation of a TA into a bounded RA-TdPN on an example.



□

Expressiveness Results for TA. Combining this 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,*

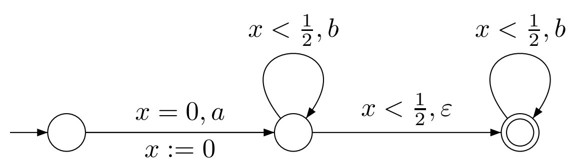
1. *bounded TdPNs and TA are equally expressive;*
2. *(integral) TA and 0-reset (integral) TA are equally expressive.*

Corollary 2. *For the ω-equivalence,*

3. *TdPNs and TA are incomparable;*
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 "folk" result, it was thought that TA and bounded TdPNs are equally expressive. We have proved that this is indeed the case for finite and infinite non-*Zeno* 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 by giving a 0-reset TA ω-equivalent to the timed automaton of Fig. 2(b).



5 Conclusion

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

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

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

Our main further work will be to develop partial-order techniques for RA-TdPNs, taking advantage of the locality of the firing rules (see [7]). Another research direction is to study arcs which do not reset age of tokens.

References

1. P. A. Abdulla, P. Mahata, and R. Mayr. Decidability of Zenoness, syntactic boundedness and token-liveness for dense-timed petri nets. In *Proc. 24th Conf. Foundations of Software Technology and Theoretical Computer Science (FST&TCS'04)*, volume 3328 of *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 Int. Conf. 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. Research Report LSV-06-06, ENS de Cachan, France, 2006.
7. P. Bouyer, S. Haddad, and P.-A. Reynier. Timed unfoldings of networks of timed automata. Research Report LSV-06-09, ENS de Cachan, France, 2006.
8. C. Girault and R. Valk, editors. *Petri Nets for Systems Engineering*. Springer, 2002.
9. P. Mahata. *Model Checking Parameterized Timed Systems*. PhD thesis, Dept. Information Technology, Uppsala University, Sweden, 2005.
10. U. Montanari and F. Rossi. Contextual nets. *Acta Informatica*, 32(6):545–596, 1995.
11. 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.
12. V. Valero Ruiz, F. Cuartero Gomez, and D. de Frutos-Escrig. On non-decidability of reachability for timed-arc Petri nets. In *Proc. 8th Int. Work. Petri Nets and Performance Models (PNPM'03)*, pages 188–196. IEEE Computer Society Press, 1999.