Robustness of Time Petri Nets under Guard Enlargement

Sundararaman Akshay, Loïc Hélouët, Claude Jard, Pierre-Alain Reyniers

Fundamenta Informaticae XXI (2001) 1–27
DOI 10.3233/FI-2012-0000
IOS Press

Robustness of Time Petri Nets under Guard Enlargement

S. Akshay
Indian Institute of Technology Bombay, India
akshayss@cse.iitb.ac.in

Loïc Hélouët
INRIA/IRISA Rennes, France
loic.helouet@inria.fr

Claude Jard
Université de Nantes, Nantes, France
Claude.Jard@univ-nantes.fr

Pierre-Alain Reynier
Aix Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France
pierre-alain.reynier@lif.univ-mrs.fr

Abstract. Robustness of timed systems aims at studying whether infinitesimal perturbations in clock values can result in new discrete behaviors. A model is robust if the set of discrete behaviors is preserved under arbitrarily small (but positive) perturbations. We tackle this problem for time Petri nets (TPNs, for short) by considering the model of parametric guard enlargement which allows time-intervals constraining the firing of transitions in TPNs to be enlarged by a (positive) parameter.

We show that TPNs are not robust in general and checking if they are robust with respect to standard properties (such as boundedness, safety) is undecidable. We then extend the marking class timed automaton construction for TPNs to a parametric setting, and prove that it is compatible with guard enlargements. We apply this result to the (undecidable) class of TPNs which are robustly bounded (i.e., whose finite set of reachable markings remains finite under infinitesimal perturbations): we provide two decidable robustly bounded subclasses, and show that one can effectively build a timed automaton which is timed bisimilar even in presence of perturbations. This allows us to apply existing results for timed automata to these TPNs and show further robustness properties.

Address for correspondence: INRIA/IRISA Rennes, France

Keywords: Time Petri nets, Robustness, timed automata

1. Introduction

Formal methods can be used to specify and verify properties of complex real-life systems. For instance, safety-critical systems with several interacting components have been studied by modeling them as networks of timed automata [2] (TA), time Petri nets [22] (TPNs) and so on. However, the usual semantics of many of these classical models rely on hypotheses which may not be met at the implementation level, such as the infinite precision of clocks or instantaneous mode transitions. Obviously, the semantics of these systems is idealized : first, in implementations of timed systems, clock values are discretized, which may lead to approximations of real clock values. Second, in distributed systems, the clocks of two different processes may evolve at slightly different rates. As a result, the extreme precision of the models leads to unexpected outcomes when there is even a slight imprecision at the level of implementation. A solution to handle this problem is to introduce perturbations in the models, and then study implementability issues for these systems. This means providing tools to verify properties of models under perturbation, and also developing robust models of systems, which preserve some good properties even in the presence of small perturbations. For timed automata, a model of guard enlargement has been extensively studied in the last decade [23, 7, 8, 14, 9, 24]. In [15], it is proved that this model of perturbation covers both the issue of discretization and drift of clocks, by reducing the implementability problem to the analysis of the enlarged semantics.

In this paper, we tackle the problem of robustness under small perturbations in the distributed and timed setting of time Petri nets [22]. TPNs associate time intervals to transitions representing "guards" within which the transition must fire once it is enabled and our aim is to study the effect of small enlargement of intervals. In this work, we address mainly three problems. The first is the *robust boundedness* problem, which consists of deciding, for a given bounded TPN, whether there exists a positive enlargement for which the set of reachable markings is finite. The second problem considered in this paper is *robust untimed language preservation*, which consists in deciding whether there exists a positive enlargement for which the untimed language remains unchanged. The third problem considered is *preservation of markings for a subset of places* under enlargement. As mentioned, robustness issues have been well studied for TA. Hence, a possible way to address the robustness problem for TPNs is to translate TPNs to TA, and reuse existing techniques. However, we show in this paper that results on TA do not always extend to TPNs. For instance, robust safety, i.e., avoidance of some bad configurations under perturbation, is decidable in TA, but not for TPNs. The objectives of this paper are to consider robustness issues for TPNs, and to study to what extent results proven for TA can be applied on TPN.

We first show that the phenomenon of accumulation of perturbations, which Puri exhibited in TA in [23], also occurs in TPN, but in a slightly different way. In a TPN, firing of transitions which are not causally related may occur systematically at distinct dates in a non-perturbed model, and after accumulation of some delays, become concurrent in the perturbed model. This has two consequences: first, reachable markings of a net may change under perturbation. Second, a bounded net may become unbounded under perturbation. This is a significant difference from the TA model which is defined over a finite set of locations which does not change under perturbation. We show an example of a TPN whose unbounded perturbed semantics cannot be captured by a finite timed automaton. We then use this example to prove that the three problems we consider are undecidable. There are several translations from

TPN to TA [16, 11, 21, 13]. We study which of these translations can be used to lift robustness results on TA to the model of TPNs. In particular, we prove that the marking class timed automaton construction of [13] is compatible with guard enlargement, in the sense that the property of timed bisimulation is preserved when guards are enlarged by the same parameter in the TA and in the TPN. We use this result to exhibit subclasses of bounded TPNs for which robust boundedness, language preservation and the preservation of markings for a subset of places are decidable.

This paper is an extended version of [1], and is organized as follows: Section 2 defines the models used. Section 3 introduces our perturbation model for TPNs, and the robust boundedness and language, markings preservation problems. Section 4 shows that many robustness issues are undecidable for TPNs. Section 5 presents a robust translation from TPNs to TA, i.e. compatible with guard enlargement. Sections 6 and 7 build on this result to exhibit decidable subclasses of TPNs. Section 8 considers robustness of markings of a subset of places under enlargement. Section 9 presents a case study, namely the robustness of schedules in a train carrousel, before conclusion.

2. Preliminaries

Let Σ be a finite alphabet, Σ^* is the set of finite words over Σ. We also use $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ with ε (the empty word) not in Σ. The sets $\mathbb{N}$, $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ are respectively the sets of natural, non-negative rational and non-negative real numbers. An interval I of $\mathbb{R}_{\geq 0}$ is a $\mathbb{Q}_{\geq 0}$-*interval* iff its left endpoint belongs to $\mathbb{Q}_{\geq 0}$ and its right endpoint belongs to $\mathbb{Q}_{\geq 0} \cup \{\infty\}$. We set $I^{\downarrow} = \{x \in \mathbb{R}_{\geq 0} \mid x \leq y \text{ for some } y \in I\}$, the *downward closure* of I. We denote by $\mathcal{I}(\mathbb{Q}_{\geq 0})$ the set of $\mathbb{Q}_{\geq 0}$-intervals of $\mathbb{R}_{\geq 0}$. A *valuation* v over a finite set X is a mapping from X to $\mathbb{R}_{\geq 0}$. We denote by $\mathbf{0}$ the valuation which assigns to every $x \in X$ the value 0. For any value $d \in \mathbb{R}_{\geq 0}$, the valuation $v + d$ is defined by $(v + d)(x) = v(x) + d,\ \forall x \in X$.

Definition 2.1. (Timed Transition System (TTS))
A *timed transition system* over Σ_ε is a transition system $S = (Q, q_0, \to)$, where Q is the set of states, $q_0 \in Q$ is the initial state, and the transition relation $\to$ consists of delay moves $q \xrightarrow{d} q'$ (with $d \in \mathbb{R}_{\geq 0}$), and discrete moves $q \xrightarrow{a} q'$ (with $a \in \Sigma_\varepsilon$). Moreover, we require standard properties of time-determinism, additivity and continuity for the transition relation $\to$.

TTSs describe systems combining discrete and continuous evolutions. They are used to define and compare semantics of TPNs and TA. With these properties, a *run* of S can be defined as a finite sequence of moves $\rho = q_0 \xrightarrow{d_0} q'_0 \xrightarrow{a_0} q_1 \xrightarrow{d_1} q'_1 \xrightarrow{a_1} q_2 \dots \xrightarrow{a_n} q_{n+1}$ where discrete actions and delays alternate, and which starts in the initial state. To such a run corresponds a word $a_0 \dots a_n$ over Σ_ε; we say that this word is accepted by S. The language of S is the set of words accepted by S.

Definition 2.2. (Time Petri Nets (TPNs))
A time Petri net $\mathcal{N}$ over Σ_ε is a tuple $(P, T, {}^\bullet(.), (.)^\bullet, m_0, \Lambda, I)$ where P is a finite set of *places*, T is a finite set of *transitions* with $P \cap T = \emptyset$, ${}^\bullet(.) \in (\mathbb{N}^P)^T$ is the *backward* incidence mapping, $(.)^\bullet \in (\mathbb{N}^P)^T$ is the *forward* incidence mapping, $m_0 \in \mathbb{N}^P$ is the *initial* marking, $\Lambda : T \to \Sigma_\varepsilon$ is the *labeling function* and $I : T \mapsto \mathcal{I}(\mathbb{Q}_{\geq 0})$ associates with each transition a *firing interval*. We denote by $\alpha(t)$ (resp. $\beta(t)$) the lower bound (resp. the upper bound) of interval $I(t)$.

Semantics. Introduced in [22], TPNs associate a time interval with each transition of a Petri net. A *configuration* of a TPN is a pair (m, ν), where m is a *marking* in the usual sense, *i.e.* a mapping in

$\mathbb{N}^P$, with $m(p)$ the number of tokens in place p. A transition t is *enabled* in a marking m if $m \geq {}^\bullet t$. We denote by $\mathit{En}(m)$ the set of enabled transitions in m. The second component of the pair (m, ν) is a valuation over $\mathit{En}(m)$ which associates with each enabled transition its age, *i.e.* the amount of time that has elapsed since this transition was last enabled. We choose the classical semantics [1] (see for instance [5]) defined as follows. An enabled transition t can be fired if $\nu(t)$ belongs to the interval $I(t)$. The set of configurations from which a transition t can be fired is called the *firing domain* of t. The result of this firing is as usual the new marking $m' = m - {}^\bullet t + t^\bullet$. Moreover, some valuations are reset. We say that transition t' is *newly enabled* by firing of t from marking m, and write $\uparrow\mathit{enabled}(t', m, t)$ iff:

$$t' \in \mathit{En}(m - {}^\bullet t + t^\bullet) \wedge ((t' \notin \mathit{En}(m - {}^\bullet t)) \vee t = t')$$

Reset valuations correspond to newly enabled clocks. Thus, firing a transition is not an atomic step and the transition currently fired is always reset. The set $\mathit{ADM}(\mathcal{N})$ of *(admissible) configurations* consists of the pairs (m, ν) such that $\nu(t) \in I(t)^\downarrow$ for every transition $t \in \mathit{En}(m)$. Thus time can progress in a marking only when it does not leave the firing interval of any enabled transition. The semantics of a TPN $\mathcal{N} = (P, T, {}^\bullet(.), (.)^\bullet, m_0, \Lambda, I)$ is a TTS $[\![\mathcal{N}]\!] = (Q, q_0, \to)$ where $Q = \mathit{ADM}(\mathcal{N})$, $q_0 = (m_0, \mathbf{0})$ and $\to$ is defined by:

- **delay moves:** $(m, \nu) \xrightarrow{d} (m, \nu + d)$ iff $\forall t \in \mathit{En}(m)$, $\nu(t) + d \in I(t)^\downarrow$,
- **discrete moves:** $(m, \nu) \xrightarrow{\Lambda(t)} (m - {}^\bullet t + t^\bullet, \nu')$ iff $t \in \mathit{En}(m)$ is s.t. $\nu(t) \in I(t)$, and $\forall t' \in \mathit{En}(m - {}^\bullet t + t^\bullet)$, $\nu'(t') = 0$ if $\uparrow\mathit{enabled}(t', m, t)$ and $\nu'(t') = \nu(t)$ otherwise.

One may immediately notice that the semantics of TPN integrates a notion of urgency: as time can elapse only up to a point that does not violate any upper bound of time constraints attached to enabled transitions, some transitions have to fire before other ones. Consider for instance the TPN of Figure 1. Transition a' has to fire at a date d that lays between 1 and 2 time units after its enabling, and transition a at a date d' that is strictly greater than 2 time units after its enabling. According to the semantics, one cannot let 2 time units elapse without firing a'. Hence, within this setting, only transition a' can fire. The (untimed) language of $\mathcal{N}$ is defined as the untimed language of $[\![\mathcal{N}]\!]$ and is denoted by $\mathcal{L}(\mathcal{N})$. The reachability set of $\mathcal{N}$, denoted $\mathsf{Reach}(\mathcal{N})$, is the set of markings $m \in \mathbb{N}^P$ such that there exists a reachable configuration (m, ν). A *bounded TPN* is a TPN $\mathcal{N}$ such that $\mathsf{Reach}(\mathcal{N})$ is finite.

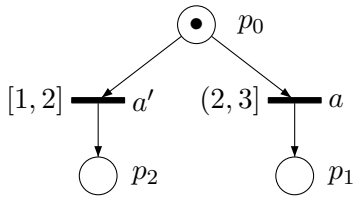


Figure 1. The TPN $\mathcal{N}_0$.

Timed automata: First defined in [2], the model of timed automata associates a set of non-negative real-valued variables called *clocks* with a finite automaton. Let X be a finite set of clocks. We write $\mathcal{C}(X)$ for the set of *constraints* over X, which consist of conjunctions of atomic formulae of the form $x \bowtie c$ for $x \in X$, $c \in \mathbb{Q}_{\geq 0}$ and $\bowtie \in \{<, \leq, \geq, >\}$. We also define the proper subset $\mathcal{C}_{ub}(X)$ of *upper bounds* constraints over X where $\bowtie \in \{<, \leq\}$.

[1] Alternative semantics exist, called atomic and persistent atomic (see [3] for a comparison). They only differ in the definition of the predicate $\uparrow\mathit{enabled}(t', m, t)$. One can prove that the undecidability results as well as the decidability results presented in this work also hold for TPN equipped with these alternative semantics.

Definition 2.3. (Timed Automata (TA))
A *timed automaton* $\mathcal{A}$ over Σ_ε is a tuple (L, ℓ_0, X, E, Inv) where L is a finite set of *locations*, $\ell_0 \in L$ is the *initial location*, X is a finite set of *clocks*, $Inv \in \mathcal{C}_{ub}(X)^L$ assigns an *invariant* to each location and $E \subseteq L \times \mathcal{C}(X) \times \Sigma_\varepsilon \times 2^X \times L$ is a finite set of *edges*. An edge $e = (\ell, \gamma, a, R, \ell') \in E$ represents a transition from location ℓ to location ℓ' labeled by a with constraint γ and reset $R \subseteq X$.

Semantics. For $R \subseteq X$, the valuation $v[R]$ is the valuation v' such that $v'(x) = v(x)$ when $x \notin R$ and $v'(x) = 0$ otherwise. Finally, constraints of $\mathcal{C}(X)$ are interpreted over valuations: we write $v \models \gamma$ when the constraint γ is satisfied by v. The semantics of a TA $\mathcal{A} = (L, \ell_0, X, E, Inv)$ is the TTS $[\![\mathcal{A}]\!] = (Q, q_0, \to)$ where $Q = \{(\ell, v) \in L \times (\mathbb{R}_{\geq 0})^X \mid v \models Inv(\ell)\}$, $q_0 = (\ell_0, \mathbf{0})$ and $\to$ is defined by:

- **delay moves:** $(\ell, v) \xrightarrow{d} (\ell, v + d)$ if $d \in \mathbb{R}_{\geq 0}$ and $v + d \models Inv(\ell)$;
- **discrete moves:** $(\ell, v) \xrightarrow{a} (\ell', v')$ if there exists some $e = (\ell, \gamma, a, R, \ell') \in E$ s.t. $v \models \gamma$ and $v' = v[R]$.

The (untimed) language of $\mathcal{A}$ is defined as that of $[\![\mathcal{A}]\!]$ and is denoted by $\mathcal{L}(\mathcal{A})$.

Timed (bi)-simulation:

Let $S = (Q, q_0, \to)$ and $S' = (Q', q'_0, \to')$ be two TTSs. A relation $\mathcal{R} \subseteq Q \times Q'$ is a *timed simulation* if and only if, $(q_0, q'_0) \in \mathcal{R}$ and for every $\sigma \in \Sigma_\epsilon \cup \mathbb{R}_{\geq 0}$, $q_1 \in Q$, $q'_1 \in Q'$ such that $(q_1, q'_1) \in \mathcal{R}$, if $q_1 \xrightarrow{\sigma} q_2$, then there exists q'_2 such that $q'_1 \xrightarrow{\sigma} q'_2$ and $(q_2, q'_2) \in \mathcal{R}$. We will say that S' *simulates* S and write $S \preceq S'$ when such a relation $\mathcal{R}$ among states of S and S' exists. If in addition $\mathcal{R}^{-1}$ is a timed simulation relation from S' to S, then we say that $\mathcal{R}$ is a timed bisimulation. We say that S and S' are *timed bisimilar* when such a relation $\mathcal{R}$ among states of S and S' exists, and write $S \approx S'$.

3. Perturbations in TPN

Perturbations in timed automata [7, 8, 14]: We start by fixing a parameter $\Delta \in \mathbb{R}_{\geq 0}$. Given a constraint $g \in \mathcal{C}(X)$, we define its Δ-enlargement as the constraint obtained by replacing any atomic formulae of the formulae $x \bowtie c$ for $x \in X$, $c \in \mathbb{N}$ and $\bowtie \in \{<, \leq, \geq, >\}$, by the formulae $x \bowtie c + \Delta$ if $\bowtie \in \{<, \leq\}$, and by the formulae $x \bowtie c - \Delta$ if $\bowtie \in \{\geq, >\}$. Now, given a timed automaton $\mathcal{A}$, we denote by $\mathcal{A}_\Delta$ the TA obtained by replacing every constraint by its Δ-enlarged version (both in guards and invariants). In the rest of the paper, we will denote by $\mathsf{Reach}(\mathcal{A}_\Delta)$ the set of locations of $\mathcal{A}$ that are reachable in $[\![\mathcal{A}_\Delta]\!]$). This model of perturbation verifies the following monotony property: for TA $\mathcal{A}$ and any $\Delta \leq \Delta' \in \mathbb{R}_{\geq 0}$, we have $[\![\mathcal{A}_\Delta]\!] \preceq [\![\mathcal{A}_{\Delta'}]\!]$. In the sequel, we will use the following result:

Proposition 3.1. ([9])
Let $\mathcal{A}$ be a timed automaton and S be a subset of locations of $\mathcal{A}$. One can decide whether there exists $\Delta \in \mathbb{Q}_{>0}$ such that $\mathsf{Reach}(\mathcal{A}_\Delta) \cap S = \emptyset$.

This property means that one can decide robustness of some simple safety properties for timed automata. If one takes a set Bad of locations that are not in $\mathsf{Reach}(\mathcal{A})$, and that the modeled system should never reach to remain safe, then finding $\Delta \in \mathbb{Q}_{>0}$ such that $\mathsf{Reach}(\mathcal{A}_\Delta) \cap S = \emptyset$ amounts to ensuring that some clock precision allows to preserve safety of the system.

Introducing perturbations in TPNs: Our goal is to consider a similar model of perturbation for time Petri nets. Given an interval $I \in \mathcal{I}(\mathbb{Q}_{\geq 0})$, we denote by I_Δ the interval obtained by replacing its lower bound α by the bound $\max(0, \alpha - \Delta)$, and its upper bound β by the bound $\beta + \Delta$. Given a TPN $\mathcal{N}$, we denote by $\mathcal{N}_\Delta$ the TPN obtained by replacing every interval I by the interval I_Δ. We can then easily prove that Petri nets enjoy a monotony property similar to that of timed automata. This entails that if the system verifies a safety property for some perturbation Δ_0, it will also verify this property for any $\Delta \leq \Delta_0$:

Lemma 3.2. Let $\mathcal{N}$ be a TPN and $\Delta \leq \Delta' \in \mathbb{R}_{\geq 0}$. We have $[\![\mathcal{N}_\Delta]\!] \preceq [\![\mathcal{N}_{\Delta'}]\!]$.

Proof:
First of all, one can notice that moves of a TPN are deterministic, that is if we have $(m, \nu) \xrightarrow{d} (m', \nu')$ or $(m, \nu) \xrightarrow{\Lambda(t)} (m', \nu')$, (m', ν') is uniquely defined. It is then sufficient to prove that every run of $[\![\mathcal{N}_\Delta]\!]$ is a run of $[\![\mathcal{N}_{\Delta'}]\!]$ to show existence of a simulation relation. We can now reason by induction on the length of runs of $[\![\mathcal{N}_\Delta]\!]$ to prove the following claim. Let $\rho = (m_0, \nu_0) \to \ldots \to (m_n, \nu_n)$ be a run of $[\![\mathcal{N}_\Delta]\!]$ and $[\![\mathcal{N}_{\Delta'}]\!]$. Then, any move $(m_n, \nu_n) \to (m_{n+1}, \nu_{n+1})$ of $[\![\mathcal{N}_\Delta]\!]$, is also a move of $[\![\mathcal{N}_{\Delta'}]\!]$.

Suppose ρ is of size 0. If there is a delay move from (m_0, ν_0) for some value d, then we have $\forall t \in \mathit{En}(m_0)$, $\nu(t) + d \in I_\Delta(t)^\downarrow$, that is $\nu(t) + d \leq \beta(t) + \Delta \leq \beta(t) + \Delta'$ and this delay move is also allowed in $[\![\mathcal{N}_{\Delta'}]\!]$. If there is a discrete move $(m_0, \nu_0) \xrightarrow{t} (m_1, \nu_1)$ in $[\![\mathcal{N}_\Delta]\!]$, then t is enabled in m_0, and we have $\alpha(t) - \Delta \leq x_t \leq \beta(t) + \Delta$. Hence, we have $\alpha(t) - \Delta' \leq x_t \leq \beta(t) + \Delta'$, and discrete transition t is allowed from (m_0, ν_0) in $[\![\mathcal{N}_{\Delta'}]\!]$. Similar reasoning holds for any configuration reached after a run of arbitrary length in $[\![\mathcal{N}_\Delta]\!]$. □

3.1. Problems considered

We now define robustness problems on TPNs in a way which is consistent with the monotony property stated above. Recall that given a TPN $\mathcal{N}$ over a set of places P and a marking $m \in \mathbb{N}^P$, $\mathcal{N}$ is said to *cover* m iff there exists $m' \in \mathsf{Reach}(\mathcal{N})$ such that $m \leq m'$ (i.e., $m(p) \leq m'(p)$ for all $p \in P$).

Robust boundedness: Given a bounded TPN $\mathcal{N}$, does there exist $\Delta \in \mathbb{Q}_{>0}$ such that $\mathcal{N}_\Delta$ is bounded?

Robust untimed language preservation: Given a bounded TPN $\mathcal{N}$, does there exist $\Delta \in \mathbb{Q}_{>0}$ such that $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$?

Subset-markings robustness (SMR): Given a bounded TPN $\mathcal{N}$ defined over a set of places P, and given a subset of places $X \subseteq P$, does there exists $\Delta \in \mathbb{Q}_{>0}$ such that the projection onto X of $\mathsf{Reach}(\mathcal{N})$ is equal to the projection onto X of $\mathsf{Reach}(\mathcal{N}_\Delta)$?

We call a TPN $\mathcal{N}$ *robustly bounded* if there exists $\Delta \in \mathbb{Q}_{>0}$ such that $\mathcal{N}_\Delta$ is bounded. This problem is strongly related to the problem of ***robust safety*** asking, given a bounded TPN $\mathcal{N}$ with set of places P, and a marking $m \in \mathbb{N}^P$, whether there exists $\Delta \in \mathbb{Q}_{>0}$ s.t., $\mathcal{N}_\Delta$ does not cover m. In fact, our undecidability and decidability results for robust boundedness will easily extend to this problem. However, the situation differs for robust untimed language preservation and so we treat this problem separately.

The above properties are considered in the setting of bounded TPNs. However, from the undecidability results presented in section 4, we can infer undecidability in the unbounded setting as well.

3.2. Robustness under a bounded horizon

Consider again the example net $\mathcal{N}_0$ in Figure 1. Due to the open interval and urgency condition, transition a' has to fire before transition a, and furthermore, it has to fire at most 2 time units after enabling. Hence, in the non-enlarged semantics of this net, transition a never fires, and no marking in which place p_1 is marked can be reached. However, any enlargement of guards results in reachability of markings with place p_1 marked. From this example, we can easily construct TPNs that are not robustly bounded nor robustly safe. Now, we observe that, in this example, the firing domain of transition a is not reachable, but is a "neighbor" of the reachable configuration $(p_0, \nu(a') = 2)$.

Formally, given a set of configurations S and a configuration $(m, \nu) \notin S$, we say that S is a *neighbor* of (m, ν) if ν is in the topological closure of the set $V = \{\nu' \mid (m, \nu') \in S\}$. As in the above example, it is immediate to see that the presence of such neighbors leads to additional behaviors under the enlarged semantics. However, this simple form of non-robustness can be easily checked in bounded TPNs.

Proposition 3.3. Let $\mathcal{N}$ be a bounded net. Then one can check whether there exists a reachable configuration (m, ν) and a transition t such that, the firing domain of t is a neighbor of (m, ν).

Proof:
We can compute a (finite) symbolic representation of the reachability set (using the state-class graph construction [5, 21] for instance). The state class graph is a transition system where states are pairs of the form (m, C) in which m is a reachable marking of the net, and C is a constraint on the values of the clocks attached to transitions that are enabled in m. Transitions of the state class graph are of the form $(m, C) \xrightarrow{t} (m', C')$, where m' is the marking reached from m after firing t. Furthermore, t is fireable iff $C \wedge I(t)$ is satisfiable. The resulting constraint C' is obtained from C by performing the following operations: $i)$ add to C the constraints that the value of clock x_t, attached to transition t, belongs to $I(t)$, and that no other urgent transition could have been fired, $ii)$ eliminate constraints attached to transitions that are disabled by the firing of t, and $iii)$ add new constraints for transitions that are newly enabled. We refer readers to [5, 21] for technical details on this construction. Now, for a bounded net, the state class graph is finite, and for every state class (m, C) and any transition t such that ${}^\bullet t \leq m$ with open interval $I(t) = (a, b]$, $I(t) = [a, b)$ or $I(t) = (a, b)$ that cannot be fired from (m, C), one can check whether t is fireable with $I'(t) = [a, b]$. □

Requiring that all intervals must be closed solves robustness problems due to neighbor configurations. In the next subsection, we will show that robustness issues may not be only due to neighbor configurations. In fact, when we consider unbounded executions, some delay accumulation mechanisms – similar to those exhibited in [23] for timed automata, can occur in TPNs and may result in new reachable markings.

But first let us examine the case of bounded executions. Instead of considering infinite or unboundedly long executions, the authors of [25] study robustness issues for timed automata under a *bounded horizon*, *i.e.* considering executions with a fixed number of discrete transitions. In the setting of TPN, assuming that we only use closed intervals one can prove that any net is robust under such a bounded horizon. Formally, this means that given an integer K, one can always pick a sufficiently small $\Delta > 0$ to ensure that no new behavior occurs during the K first discrete steps of the system.

Lemma 3.4. Let $\mathcal{N}$ be a net with closed intervals, and let $K \in \mathbb{N}$. Then, we can compute $\Delta_K > 0$ s.t.:

- the set of markings reachable by runs with at most K discrete moves is the same in $\mathcal{N}$ and in $\mathcal{N}_{\Delta_K}$,

- the restrictions of the untimed languages of $\mathcal{N}$ and $\mathcal{N}_{\Delta_K}$ to words of length at most K are the same.

Proof:
We will consider, without loss of generality, that the considered net has integral bounds. We first build a (finite) unfolding of $\mathcal{N}$, that is an acyclic automaton that memorizes paths of length at most K in the state class graph of $\mathcal{N}$. Its states are of the form (m, C, n), meaning that a marking (m, C) is reached at step n, and a transition $(m, C, n) \xrightarrow{t} (m', C', n+1)$ occurs iff $(m, C) \xrightarrow{t} (m', C')$ is a transition of the state class graph of $\mathcal{N}$, and $n < K$. Note that constraints are conjunctions of inequations of the form $a \leq x_t \leq b$ and $a \leq x_t - x'_t \leq b$. Indeed, it is known that given a TPN with closed intervals, every state class built is defined as a conjunction of non-strict inequalities. Let us denote by $\mathcal{U}_K(\mathcal{N})$ this finite unfolding.

We then build a parametric unfolding of $\mathcal{N}_\Delta$ up to depth K, that memorizes (parameterized) constraints attached to transitions, a constraint on the value of Δ, and the number of transitions already fired. States of this parametric unfolding are of the form (m, C, D, n) where m is a marking of $\mathcal{N}$, C is a conjunction of constraints on clocks and clock differences involving the parameter Δ, D is an interval describing admissible values for Δ and $0 \leq n \leq K$ indicates the depth in the unfolding. More precisely, C is a conjunction of constraints of the form $a - q.\Delta \leq x_t \leq b + q'.\Delta$ and $a - q.\Delta \leq x_t - x_{t'} \leq b + q'.\Delta$, with $q, q' \in \mathbb{N}$. Given such a constraint C, we denote by $C_{\Delta=0}$ the constraint on clocks and clock differences (*without parameter* Δ) obtained by setting Δ to 0. For every transition t, the time interval $I(t) = [a, b]$ is represented by the constraint $\phi_t = a - \Delta \leq x_t \leq b + \Delta$. We start from initial configuration $(m_0, \nu_0, [0, \infty), 0)$. Then for each configuration (m, C, D, n) we can stop if $n = K$, or add a new transition $(m, C, D, n) \xrightarrow{t} (m', C', D', n+1)$, where C' and D' are computed as follows. Observe that considering the constraint $C_{\Delta=0}$, the triple $(m, C_{\Delta=0}, n)$ is a state of $\mathcal{U}_K(\mathcal{N})$, for which we know which transitions are fireable, and which are not. Let us consider some transition t fireable in $\mathcal{U}_K(\mathcal{N})$ from the state $(m, C_{\Delta=0}, n)$. Observe that when enlarging intervals, it may happen that enabled transitions which were not fireable from $(m, C_{\Delta=0}, n)$ become fireable. To avoid this situation, we will identify an upper bound constraint on Δ. Let us denote by $T' \subseteq T$ the set of transitions enabled in m, but not fireable from $(m, C_{\Delta=0}, n)$. We consider the constraint $C \wedge \phi_t \wedge \left(\bigwedge_{t' \in T'} \neg \phi_{t'} \right)$, eliminate all variables but Δ, and end up with a constraint on Δ expressed as some interval I_Δ. First observe that $\Delta = 0$ is a solution of this set of constraints. Second, the integral bounds appearing in C match those of the corresponding state of $\mathcal{U}_K(\mathcal{N})$. Moreover, as $\mathcal{N}$ only has closed intervals, every transition t' that is not fireable has a firing interval which is at least 1-unit far away from the domain of its clock variable (in a setting with rational upper and lower bounds for intervals, this firing interval is at a computable and strictly positive rational distance). This implies that the interval I_Δ of admissible values for Δ is of the form $[0, d]$ or $[0, d)$ with $d > 0$. We thus let $D' = D \cap I_\Delta$, and compute C' from C as usually in the state class graph by elimination of variables attached to disabled transitions from $C \wedge \phi_t$, and adding new constraints $\phi_{t'}$ attached to each newly enabled transition t'.

The construction of the unfolding terminates, and the maximal value allowed for Δ_K is given by the tightest interval D appearing in the leaves of the parametric unfolding. As the two unfoldings built are in bijection, both the sets of runs and the untimed languages are preserved. □

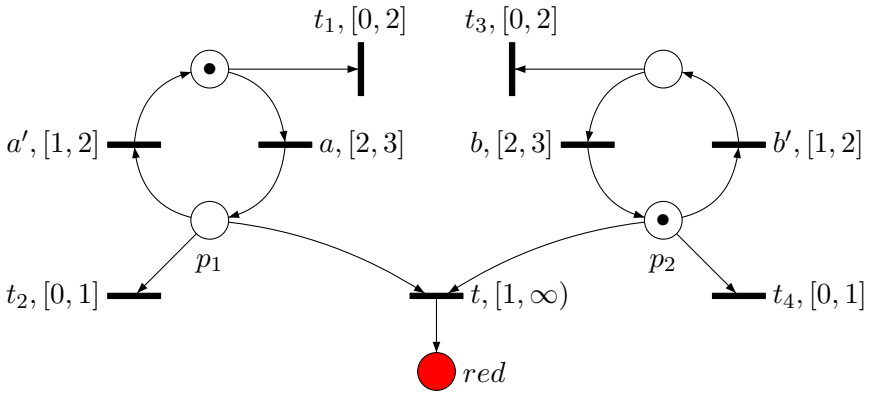


Figure 2. The TPN $\mathcal{N}_1$ exhibiting new discrete behaviors under infinitesimal perturbations.

3.3. An example of accumulated non-robustness

We have already mentioned that neighbor configurations could lead to new behaviors of a TPN under enlargement. These neighbor configurations can be easily detected in bounded nets, and for a fixed horizon, can be avoided in TPNs with closed intervals. The remaining cases concern TPNs in which new behaviors are not neighbors of the reachability set, considered for an unbounded horizon. In this case a new behavior cannot appear directly from a reachable configuration, and there must be several discrete firings before this new behavior is witnessed. Further the number of steps may depend on enlargement Δ: the smaller Δ is, the larger will be the number of steps required. Intuitively, the new behavior is due to an accumulation of clock perturbations, rather than a single clock perturbation.

Puri [23] gave an example of a TA that exhibits accumulations of perturbations. This TA uses two clocks x and y and a location ℓ such that, under the exact semantics, the difference between y and x in location ℓ always lies in the interval $[0, 1]$. Under any enlarged semantics (by some value Δ), this difference lies in the interval $[0, 2]$. There is in addition a transition t leaving location ℓ with a guard that can be satisfied only if the difference between y and x is equal to 2. As a consequence, t yields behaviors in the enlarged semantics which are not present in the exact semantics (for any value of Δ).

Though translations from TA to TPN do exist for the exact semantics (see for instance [4]), it is not guaranteed that they also preserve the enlarged semantics. As a consequence, it is not clear that the TPN resulting from the application of such translations to the TA proposed in [23] will exhibit such accumulations. Instead, we directly build a TPN which exploits the concurrency of Petri nets while using ideas similar to those of Puri.

This TPN, denoted $\mathcal{N}_1$, is depicted on Figure 2. Informally, it is composed of two components performing periodic behaviors around places p_1 and p_2 respectively. These components can be synchronized using transition t if places p_1 and p_2 are marked simultaneously. Intuitively, these components simulate the behavior of clocks x and y of the TA of Puri in the sense that in the exact semantics, places p_1 and p_2 are marked respectively for the k-th time at timestamps τ_k and τ'_k such that $\tau'_k - \tau_k = 1$, and thus t is never fired. On the other hand, in the enlarged semantics, the value of $\tau_k - \tau'_k$ may 'drift' and thus t may be fired. Observe that this example can be simplified using singleton intervals, but we avoid this to show that problems due to accumulation may arise even without singletons.

We give now some details on the executions of $\mathcal{N}_1$. With the usual (exact) semantics, the red state in

$\mathcal{N}_1$ is not reachable as transition t is never fireable. Indeed, one can verify that any run of $\mathcal{N}_1$ which does not fire transitions t_1, t_2, t_3 or t_4 always fires transition a (resp. a', b', b) at time $3k+2$ (resp. $3k+3$, $3k+1$, $3k+3$), for some integer k. By observing the time intervals of transitions t, a' and b', one can deduce that to be able to fire transition t, one has to fire simultaneously the transitions a and b, which is impossible.

Consider the net $(\mathcal{N}_1)_\Delta$, for some positive Δ. We will prove that in this case, it is possible to fire simultaneously transitions a and b. In $(\mathcal{N}_1)_\Delta$, one can delay the firing of transitions a and a' by up to Δ time units. As a consequence, it is easy to verify that after n iterations of the loop aa', the timestamp of the firing of the last occurrence of a can be delayed by up to $(2.n-1).\Delta$ time units (w.r.t. the firing date of this occurrence of a under non-enlarged semantics). Similarly, transitions b and b' can occur earlier, and after the n^{th} iteration of the loop $b'b$, the last occurrence of b can be advanced by $2.n.\Delta$ time units (w.r.t. the firing date of this occurrence of b under non-enlarged semantics). Hence, after a number n of occurrences of loops aa' and $b'b$ with $n \geq \frac{1+\Delta}{4.\Delta}$, it can be the case that the firing date of the n^{th} occurrences of a and b are identical. Place p_1 and p_2 can remain marked for 1 time unit, and the red place is reachable in $(\mathcal{N}_1)_\Delta$, for any positive Δ.

3.4. Sequential TPNs

The accumulation in the above example was due to concurrent loops in the TPN. When we disallow such concurrency, we obtain a very simple class of *sequential TPNs* which is a strict subclass of timed automata. We state their properties in detail here as they will be useful in later proofs. Also this exhibits a clear way to distinguish the relative power of TPNs and TA. A TPN $\mathcal{N}$ is *sequential* if it satisfies the following property: for any reachable configuration (m, ν), and for any transitions $t, t' \in T$ that are fireable from (m, ν) (i.e. such that $t, t' \in En(m)$, $\nu(t) \geq \alpha(t)$ and $\nu(t') \geq \alpha(t')$), t and t' are in conflict, i.e. there exists a place p such that $m(p) < {}^\bullet t(p) + {}^\bullet t'(p)$. The following lemma states robustness properties of sequential TPNs and their relation to timed automata.

Lemma 3.5. We have the following properties:

- (i) Checking whether a bounded TPN $\mathcal{N}$ is sequential is decidable.
- (ii) If $\mathcal{N}$ is a sequential bounded TPN, then it can be translated into a timed automaton which resets every clock on each transition.
- (iii) If $\mathcal{N}$ is sequential, then there exists $\Delta \in \mathbb{Q}_{>0}$ such that $\mathsf{Reach}(\mathcal{N}_\Delta) = \mathsf{Reach}(\mathcal{N})$ and $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$.

Proof:
Decidability follows from the construction of the state class graph, which is possible as the TPN is bounded. Clearly, this can be done in time linear in the size of the state class graph. The second and third properties follow from the observation that in a sequential TPN, each time a discrete transition is fired, each transition that is enabled in the new/resulting marking is newly enabled. Thus, all the clocks are reset and this implies property (ii). Further, since clocks are reset, there is intuitively no memory in clock values. Considering $\Delta < \frac{1}{2}$ to ensure that exactly the same transitions are enabled, we prove by induction on the length of runs that the configurations reached immediately after a discrete transition are the same in $[\![\mathcal{N}]\!]$ and in $[\![\mathcal{N}_\Delta]\!]$.

Consider a run $\rho = (m_0, \nu_0) \xrightarrow{d_1,a_1} (m_1, \nu_1) \dots (m_{n_1}, \nu_{n-1}) \xrightarrow{d_n,a_n} (m_n, \nu_n)$ in $[\![\mathcal{N}_\Delta]\!]$. We prove by induction on the length of ρ that every valuation ν_i verifies $\nu_i(t) = 0$ for all $t \in \mathit{En}(m_i)$, and that there exists a run $\rho' = (m_0, \nu_0) \xrightarrow{d'_1,a_1} (m_1, \nu_1) \dots (m_{n_1}, \nu_{n-1}) \xrightarrow{d_n,a_n} (m_n, \nu_n)$ in $[\![\mathcal{N}]\!]$ which only differs in the time elapsing, but which is such that the configurations reached after each discrete action are the same. The base case (ρ has length 0) of the induction is trivial. Consider a new step $(m_n, \nu_n) \xrightarrow{d_n,a_n} (m, \nu)$ in $[\![\mathcal{N}_\Delta]\!]$. By definition, there exists a transition $t \in T$ which verifies the following conditions:

- $t \in \mathit{En}(m_n)$,
- t is labeled by a_n,
- $\forall t' \in \mathit{En}(m_n), \nu_n(t') + d_n \leq \beta(t') + \Delta$,
- $\nu_n(t) + d_n \geq \alpha(t) - \Delta$

By induction property, we have $\nu_n(t') = 0$ for all $t' \in \mathit{En}(m_n)$. As a consequence, we can deduce that $\alpha(t) - \Delta \leq d_n \leq \min\{\beta(t') \mid t' \in \mathit{En}(m_n)\} + \Delta$. As transitions have rational bounds, one can choose a small rational value for Δ such that $\alpha(t) - \Delta \leq d_n \leq \min\{\beta(t') \mid t' \in \mathit{En}(m_n)\} + \Delta$ implies the inequality $\alpha(t) \leq \min\{\beta(t') \mid t' \in \mathit{En}(m_n)\}$.

For instance letting $\gamma = \min\{\beta(t') - \alpha(t) \mid \alpha(t) < \beta(t')\}$, any value of $\Delta < \frac{\gamma}{2}$ guarantees this implication. If all intervals attached to transitions of the net have integral bounds, taking $\Delta < \frac{1}{2}$ suffices.

We thus pick $d'_n = \alpha(t)$, which ensures:

- $\forall t' \in \mathit{En}(m_n), \nu'_n(t') + d'_n = \alpha(t) \leq \beta(t')$,
- $\nu'_n(t) + d'_n \geq \alpha(t)$

As a consequence, we have $(m_n, \nu'_n) \xrightarrow{d'_n,a_n} (m, \nu')$ in $[\![\mathcal{N}]\!]$. Thanks to the property of being sequential, we can observe that every transition that is enabled in the new marking m' is newly enabled by the firing of the discrete transition t. In particular, this implies $\nu'(t') = 0$ for every transition $t' \in \mathit{En}(m)$, and in particular $\nu' = \nu$. The expected properties on $\mathsf{Reach}(\mathcal{N}_\Delta)$ and $\mathcal{L}(\mathcal{N}_\Delta)$ then directly follow. □

4. Undecidability results

We use the TPN $\mathcal{N}_1$ of Figure 2 to prove undecidability of all the robustness problems considered for bounded TPNs.

Theorem 4.1. The problems of (i) robust boundedness, (ii) robust untimed language preservation, (iii) robust safety and (iv) subset-markings robustness are undecidable for bounded TPNs.

Proof:
To prove undecidability, we combine the standard construction of a TPN from a Minsky machine with the example TPN $\mathcal{N}_1$ from Figure 2 and Lemma 3.5 on sequential TPNs.

For the sake of completeness, we start by briefly recalling the Minsky machine reduction. A Minsky machine $\mathcal{M}$ (which w.l.o.g. we assume deterministic) is defined by a finite set of states q_i with $0 \leq$

$i \leq n$, where q_0 is the initial state and q_n the final one. There are no transition rules from q_n. The machine contains two counters c_1 and c_2 and transition rules corresponding either to incrementations ($q_i \xrightarrow{c_k++} q_j$) or to decrementations with test to zero ($q_i \xrightarrow{c_k--} q_j$ if $c_k > 0$, and $q_i \to q_l$ otherwise). As the machine is deterministic, it has a single execution. It is well known that the reachability of state q_n is undecidable, so boundedness of c_1 and c_2 along the unique execution of $\mathcal{M}$ is also undecidable.

(a) Incrementation (b) Decrementation

Figure 3. Encoding instruction of a Minsky machine $\mathcal{M}$ into a TPN $\mathcal{N}_\mathcal{M}$.

The machine $\mathcal{M}$ is encoded into a TPN $\mathcal{N}_\mathcal{M}$ as follows: we consider a set of places $P = \{q_i\} \cup \{c_1, c_2\}$. Initial marking is $\{q_0\}$. Transitions are represented on Figure 3. In the TPN of Figure 3(b), firing intervals are used to enforce priorities between transitions: if decrementation is possible (meaning that places q_i and c_k are non-empty, which enables transition $t_{>0}$) then it must occur between 0 and 1 time units. Note that in this situation, transition $t_{=0}$ is also enabled, as place q_i is filled. However, as the firing interval of $t_{=0}$ is $[2, 3]$, transition $t_{>0}$ fires first, that is decrementation is prioritary when places q_i and c_k are non-empty. We make two observations. First, as $\mathcal{N}_\mathcal{M}$ simulates exactly executions of $\mathcal{M}$, $\mathcal{N}_\mathcal{M}$ is bounded iff $\mathcal{M}$ is, and $\mathcal{N}_\mathcal{M}$ covers marking $m = \{q_n\}$ iff $\mathcal{M}$ reaches state q_n. Second, in every reachable configuration, exactly one of the places $\{q_i, 0 \leq i \leq n\}$ contains a token. As a consequence, the net $\mathcal{N}_\mathcal{M}$ is sequential.

We then combine the TPNs $\mathcal{N}_1$ from Figure 2 and $\mathcal{N}_\mathcal{M}$ as depicted on Figure 4 to obtain the TPN $\mathcal{N}_2$. More precisely, $\mathcal{N}_2$ is a sequential composition of $\mathcal{N}_1$ and $\mathcal{N}_\mathcal{M}$, defined as the disjoint union of nets $\mathcal{N}_1$ and $\mathcal{N}_\mathcal{M}$ plus a new place f, and two new transitions t_s and t_f. Place red of $\mathcal{N}_1$ is connected to place q_0 of $\mathcal{N}_\mathcal{M}$ through a transition t_s. Place q_n of $\mathcal{N}_\mathcal{M}$ is connected to place f through transition t_f. First note that $\mathcal{N}_2$ is a bounded TPN: without perturbation, transition t (in $\mathcal{N}_1$) is never fired, and thus the set of reachable markings is finite. Second, we label transition t_f by a and every other transition by ε [2]. As $\mathcal{N}_\mathcal{M}$ is sequential, by Lemma 3.5(iii) it follows that

- (1) $\mathcal{N}_2$ is robustly bounded iff $\mathcal{N}_\mathcal{M}$ is bounded, and
- (2) $\mathcal{N}_2$ robustly preserves its untimed language iff $\mathcal{N}_\mathcal{M}$ does not cover marking m.

We note that for (2), $\mathcal{N}_\mathcal{M}$ may not be bounded (if $\mathcal{M}$ is not bounded), however the statement still holds since Lemma 3.5(iii) does not require the boundedness assumption.

Now, from the undecidability of halting and boundedness of Minsky machines, it follows that the problems we considered are also undecidable. We remark that the above proof also shows that robust safety is undecidable, as $\mathcal{N}_2$ covers marking $\{f\}$ iff $\mathcal{N}_\mathcal{M}$ covers marking m.

[2]The reduction can be adapted to avoid the use of ε by labeling every other transition by b, and adding a gadget which can perform arbitrarily many b's. It can however not be adapted to the setting of injective labeling, see Section 7.

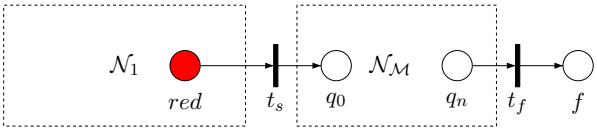


Figure 4. TPN $\mathcal{N}_2$ obtained by combining $\mathcal{N}_1$ and $\mathcal{N}_\mathcal{M}$.

Finally, the net $\mathcal{N}_2$ defined in Figure 4 also shows that subset markings robustness problem is undecidable for bounded TPNs. To see this consider the following problem instance: is the set of markings of $\mathcal{N}_2$ of place f (i.e., $X = \{f\}$) equivalent under some positive enlargement by some Δ? We know that f is reachable if and only if the Minsky machine modeled by $\mathcal{N}_\mathcal{M}$ terminates its computation. Hence the subset marking robustness problem is undecidable. □

Before we proceed, let us consider a special case of the SMR problem, where we fix $X = P$, that is, choose the subset to be the set of all places. Then the subset marking problem is reduced to the question of whether the set of reachable markings of the considered net is robust under enlargement, that is if $\mathsf{Reach}(\mathcal{N}) = \mathsf{Reach}(\mathcal{N}_\Delta)$. We call this the **robustness of reachable markings problem**. In Section 6, we will prove that this problem, unlike the many problems considered so far, is decidable for all bounded TPNs. However, for unbounded nets, even this problem is undecidable as we now demonstrate.

Proposition 4.2. The robustness of reachable markings problem is undecidable for unbounded TPNs.

Proof:
Again, we will show that there exists a reduction from Minsky machines halting problem. Consider the net $\mathcal{N}_\mathcal{M}$ encoding a deterministic Minsky machine as in Figure 3, and consider the place q_n of $\mathcal{N}_\mathcal{M}$. It is undecidable whether q_n can be marked, and the Minsky machine encoding is an unbounded, sequential and robust net. Now connect to q_n a pair of transitions t, t' and two places p, p', such that ${}^\bullet t = q_n, {}^\bullet t' = q_n, t^\bullet = p, t'^\bullet = p'$. Let us define $I(t) = [1, 2]$ and $I(t') = (2, 3]$. Intuitively, we connect the net $\mathcal{N}_0$ of Figure 1 to the final place q_n of $\mathcal{N}_\mathcal{M}$. Recalling that $\mathcal{N}_\mathcal{M}$ is deterministic and sequential, we have that only markings of places p, p' can vary under enlargement. As $\mathcal{N}_0$ does not have a robust set of markings, it is clear that the obtained net preserves its set of markings under enlargement if and only if q_n is unreachable. So, the robustness of $\mathsf{Reach}(\mathcal{N})$ is undecidable in general for unbounded TPNs. □

5. A robust translation from TPN to TA

As robustness issues were first studied for timed automata, and several translations of TPN into TA exist in literature, it is natural to study which of these translations are compatible with robustness. A way to reduce robustness problems for TPNs to robustness problems for TA is to show that an existing timed bisimulation between TPN and its TA translation is preserved under perturbation. We now present a translation which verifies this property.

This construction is close to the marking class timed automaton construction of [13] but different in two aspects. First, in the TA built in [13], for efficiency reasons the number of clocks is reduced by using clock sharing techniques of [21], which may increase the number of locations. For ease of

presentation, we do not consider this optimization, but our results also apply for this setting. Second, the construction of [13] was only stated for TPN whose underlying Petri net (i.e., the Petri net obtained by ignoring the timing information in the given TPN) is bounded. We present the construction in a more general framework: we consider a TPN $\mathcal{N}$ which is not necessarily bounded and we consider as input a finite set of markings M. The construction is then restricted to the set M, and we can prove that it is correct for the set of behaviors of $\mathcal{N}$ which always remain within M. In the sequel, we will instantiate M depending on the context. For TPNs whose underlying PN is bounded, the construction of [13] is recovered by letting M be the set of reachable markings of this PN. We begin with a definition and a proposition that can be infered immediately:

Definition 5.1. Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, m_0, \Lambda, I)$ be a TPN, $M \subseteq \mathbb{N}^P$ be a set of markings such that $m_0 \in M$, and let $[\![\mathcal{N}]\!] = (Q, q_0, \to)$ be the semantics of $\mathcal{N}$. The *M-bounded semantics of $\mathcal{N}$*, denoted $[\![\mathcal{N}]\!]_{|M}$, is defined as the restriction of the TTS $[\![\mathcal{N}]\!]$ to the set of states $\{(m, \nu) \in Q \mid m \in M\}$.

Proposition 5.2. Let M be a set of markings of a TPN $\mathcal{N}$ containing the initial marking. If $\mathsf{Reach}(\mathcal{N}) \subseteq M$, then $[\![\mathcal{N}]\!]_{|M} = [\![\mathcal{N}]\!]$.

Now, let $\mathcal{N} = (P, T, \Sigma_\varepsilon, {}^\bullet(.), (.)^\bullet, m_0, \Lambda, I)$ be a TPN, and $M \subseteq \mathbb{N}^P$ be a finite set of markings such that $m_0 \in M$. The *marking timed automaton of $\mathcal{N}$ over M*, denoted $\mathcal{A}_M$, is defined as $\mathcal{A}_M = (M, m_0, X, \Sigma_\varepsilon, E, Inv)$, where $X = \{x_t \mid t \in T\}$, for each $m \in M$, $Inv(m) = \bigwedge_{t \in En(m)} x_t \leq \beta(t)$, and there is an edge $m \xrightarrow{g,a,R} m' \in E$ iff there exists $t \in T$ such that $t \in En(m)$, $m' = m - {}^\bullet t + t^\bullet$, g is defined as the constraint $x_t \in I(t)$, $a = \Lambda(t)$ and $R = \{x_{t'} \mid \uparrow\!\mathit{enabled}(t', m, t)\}$. With this we have the following theorem:

Theorem 5.3. Let $\mathcal{N}$ be a TPN, M be a finite set of markings containing the initial marking of $\mathcal{N}$, and $\mathcal{A}_M$ be the marking timed automaton of $\mathcal{N}$ over M. Then for all $\Delta \in \mathbb{Q}_{\geq 0}$, we have $[\![\mathcal{N}_\Delta]\!]_{|M} \approx [\![(\mathcal{A}_M)_\Delta]\!]$.

Proof:

We prove by induction that the following relation $\mathcal{R}$ is a timed bisimulation. Let (m, ν) denote a state of the TTS $[\![\mathcal{N}_\Delta]\!]_{|M}$, i.e. $(m, \nu) \in Adm(\mathcal{N}_\Delta)$ with $m \in M$. Similarly, let (ℓ, v) denote a state of $[\![(\mathcal{A}_M)_\Delta]\!]$. We define $(m, \nu)\mathcal{R}(\ell, v)$ if and only if $m = \ell$, and $\forall t \in En(m), \nu(t) = v(x_t)$. First, initial configurations are in $\mathcal{R}$. We then have to consider how pairs $\big((m, \nu), (\ell, v)\big) \in \mathcal{R}$ evolve with respect to different kinds of moves:
delay moves: Let $d \in \mathbb{R}_{\geq 0}$. We have $(m, \nu) \xrightarrow{d} (m, \nu + d)$ iff $\forall t \in En(m), \nu(t) + d \leq \beta(t) + \Delta$. As $\forall t \in En(m), v(x_t) = \nu(t)$, this is equivalent to $\forall t \in En(m), v(t) + d \leq \beta(t) + \Delta$, which itself is equivalent to $v \models Inv(\ell) + \Delta$, which is the invariant of location ℓ in $(\mathcal{A}_M)_\Delta$. This is the condition under which there exists a delay move $(\ell, v) \xrightarrow{d} (\ell, v + d)$ in $[\![(\mathcal{A}_M)_\Delta]\!]$. Thus the result holds for delay moves.
discrete moves: Consider a discrete move $(m, \nu) \xrightarrow{a} (m', \nu')$ in $[\![\mathcal{N}_\Delta]\!]_{|M}$. Such a discrete move exists iff $m, m' \in M$, and there exists a transition $t \in T$ such that:

1. $t \in En(m)$

2. $m' = m - {}^\bullet t + t^\bullet$
3. $\nu(t) \in I_\Delta(t)$ where $I_\Delta(t)$ denotes the Δ-enlargement of interval $I(t)$
4. $\Lambda(t) = a$
5. for any $t' \in En(m')$, we have $\nu'(t') = 0$ if $\uparrow \mathit{enabled}(t', m, t) = \texttt{true}$, and $\nu'(t') = \nu(t)$ otherwise.

Conditions 1-5 imply the existence of a transition $m \xrightarrow{g,a,R} m'$ in $\mathcal{A}_M$, where g is defined as the constraint $x_t \in I(t)$, and R as the set of clocks of newly enabled transitions. As $t \in En(m)$, we have $\nu(t) = v(t)$, and thus the transition can be fired in $[\![(\mathcal{A}_M)_\Delta]\!]$, and we have $(\ell, v) \xrightarrow{a} (m', v')$ where $v' = v[R]$. One can then check that for any transition $t' \in En(m')$, we have $v'(t') = \nu'(t')$. There are two cases, if t' is newly enabled, then the clock value is 0 both in the TA and in the TPN. Otherwise, t' is not newly enabled, and we have $v'(t') = v(t') = \nu(t') = \nu'(t')$.

Conversely, considering a discrete move in $[\![(\mathcal{A}_M)_\Delta]\!]$, one can similarly prove the existence of a corresponding move in $[\![\mathcal{N}_\Delta]\!]_{|M}$. □

Other TA constructions. The construction proposed in [21] builds a state class timed automaton incrementally using a forward exploration of reachable markings of a bounded TPN. Gardey et al [16] use a similar forward-reachability technique to build the reachable state space of TPN, where equivalence classes for clock valuations are encoded as zones. However, as in TPN $\mathcal{N}_1$ of Figure 1, new configurations in an *enlarged semantics* might be reached after accumulation of small delays. Hence, new reachable markings are not necessarily obtained in one enlarged step from a configuration in the non-enlarged semantics. Thus, forward techniques as in [21, 16] cannot be directly extended to obtain enlarged semantics and we need a more syntactic translation which builds an over-approximation of the reachable markings (of the TPN) as in Theorem 5.3.

Cassez et al [11] propose a different syntactic translation from unbounded TPNs by building a timed automaton for each transition, and then synchronizing them using a supervisor. The resulting timed automaton is bisimilar to the original model, but states contain variables, and hence the automaton may have an unbounded number of locations. It may be possible to extend this approach to address robustness problems, but as we focus on bounded TPNs, we leave this for future work.

6. Robustly bounded TPNs

This section focuses on the class of robustly bounded TPNs. By Theorem 4.1, we know that checking membership in this class is undecidable. We present two decidable subclasses, as well as a semi-decision procedure for the whole class. We first consider the subclass of TPNs whose *underlying Petri net* is bounded:

Proposition 6.1. The set of TPNs whose underlying net is bounded is a decidable subclass of robustly bounded TPNs. Further, for each net $\mathcal{N}$ of this class, one can construct a finite timed automaton $\mathcal{A}$ such that $[\![\mathcal{N}_\Delta]\!] \approx [\![\mathcal{A}_\Delta]\!]$ for all $\Delta \geq 0$.

The decidability follows from that of boundedness for (untimed) Petri nets [20]. The second part of the above proposition follows from Theorem 5.3.

We now exhibit another subclass of robustly bounded TPNs whose underlying Petri nets can be unbounded. In fact, this class is incomparable with the above defined subclass. The following technical result is central in our approach:

Lemma 6.2. Let $\mathcal{N}$ be a TPN, and M be a finite set of markings. Determining whether there exists $\Delta > 0$ such that $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M$ is decidable.

Proof:
Call $\widetilde{M} = M \cup \{m' \mid \exists m \in M, t \in T, m' = m - {}^\bullet t + t^\bullet\}$ the (finite) set of markings reachable from M in at most one-step in the underlying Petri net. Let $\mathcal{A}_{\widetilde{M}}$ be the marking timed automaton of $\mathcal{N}$ over $\widetilde{M}$, and let $\Delta \geq 0$. We claim:

$$\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M \iff \mathsf{Reach}((\mathcal{A}_{\widetilde{M}})_\Delta) \subseteq M$$

To prove this equivalence, we consider successively the two implications. For the direct implication, suppose that $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M$. By Proposition 5.2 and Theorem 5.3, we obtain $[\![\mathcal{N}_\Delta]\!] \approx [\![(\mathcal{A}_{\widetilde{M}})_\Delta]\!]$. This yields the result as there is a bijection between transitions of $[\![\mathcal{N}_\Delta]\!]$ and those of $[\![(\mathcal{A}_{\widetilde{M}})_\Delta]\!]$. Conversely, suppose that $\mathsf{Reach}((\mathcal{A}_{\widetilde{M}})_\Delta) \subseteq M$. By contradiction, suppose that $\mathsf{Reach}(\mathcal{N}_\Delta) \not\subseteq M$. Thus, there exists a run $\rho = (m_0, \nu_0) \xrightarrow{d_1,t_1} (m_1, \nu_1) \dots \xrightarrow{d_n,t_n} (m_n, \nu_n)$ of $[\![\mathcal{N}_\Delta]\!]$ such that $m_n \notin M$. W.l.o.g., we assume that $m_i \in M$ for any $i < n$. This entails that $m_i \in \widetilde{M}$ for all i. But then, as we have $[\![\mathcal{N}_\Delta]\!]_{|\widetilde{M}} \approx [\![(\mathcal{A}_{\widetilde{M}})_\Delta]\!]$ by Theorem 5.3, this entails that the "same" run ρ also exists in $[\![(\mathcal{A}_{\widetilde{M}})_\Delta]\!]$. This is a contradiction with $\mathsf{Reach}((\mathcal{A}_{\widetilde{M}})_\Delta) \subseteq M$.

Now, determining whether there exists $\Delta > 0$ such that the right hand side of the previous equivalence holds is decidable thanks to Proposition 3.1. □

We consider the following subclass of bounded TPNs:

Definition 6.3. A bounded TPN $\mathcal{N}$ is called Reach-Robust if $\mathsf{Reach}(\mathcal{N}_\Delta) = \mathsf{Reach}(\mathcal{N})$ for some $\Delta > 0$. We denote by ***RR*** the class of Reach-Robust TPNs.

RR is the class of bounded TPNs whose set of reachable markings is invariant under some guard enlargement. It is easy to see that these nets are robustly bounded. Recalling the result of Lemma 3.5-iii), we can immediately state that bounded sequential nets form a subclass of RR nets. More interestingly, checking membership in this class is decidable, i.e., given a bounded TPN $\mathcal{N}$ we can decide if there is a positive guard enlargement under which the set of reachable markings remains unchanged. This follows from Lemma 6.2, by instantiating the finite set of markings M with $\mathsf{Reach}(\mathcal{N})$:

Theorem 6.4. RR is a decidable subclass of robustly bounded TPNs.

We can now address properties of the general class of robustly bounded TPN.

Lemma 6.5. The set of robustly bounded TPNs is recursively enumerable. Moreover, given a robustly bounded TPN $\mathcal{N}$, we can build effectively a timed automaton $\mathcal{A}$ such that there exists $\Delta_0 > 0$ for which, $\forall 0 \leq \Delta \leq \Delta_0$, $[\![\mathcal{N}_\Delta]\!] \approx [\![\mathcal{A}_\Delta]\!]$.

Proof:
Observe that a TPN $\mathcal{N}$ is robustly bounded iff there exists a finite set of markings M and some $\Delta > 0$ such that $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M$. Thus by naively enumerating the *set* of finite sets of markings and applying the algorithm of Lemma 6.2 at each step of the enumeration, we obtain a semi-decision procedure (to check membership) for the class of robustly bounded TPNs. For the second result, observe that if $\mathcal{N}$ is known to be robustly bounded, then this semi-decision procedure terminates and computes a finite set of markings M and there is a value Δ_0 such that $\mathsf{Reach}(\mathcal{N}_{\Delta_0}) \subseteq M$. Therefore, for any $\Delta \leq \Delta_0$, $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M$. By Proposition 5.2, this entails $[\![\mathcal{N}_\Delta]\!]_{|M} = [\![\mathcal{N}_\Delta]\!]$. In addition, by Theorem 5.3, we have $[\![\mathcal{N}_\Delta]\!]_{|M} \approx [\![(\mathcal{A}_M)_\Delta]\!]$ where $\mathcal{A}_M$ is the marking timed automaton of the TPN $\mathcal{N}$. Thus we have $\forall 0 \leq \Delta \leq \Delta_0,\ [\![\mathcal{N}_\Delta]\!] \approx [\![(\mathcal{A}_M)_\Delta]\!]$. □

This result allows us to transfer existing robustness results for timed automata to TPNs. We will illustrate the use of this property in the following section.

7. Untimed language robustness in TPNs

We now consider the robust untimed language preservation problem, which was shown undecidable in general in Theorem 4.1. We show that for the subclass of *distinctly labeled bounded TPNs* (i.e., labels on transitions are all distinct, and different from ε) this problem becomes decidable.

Definition 7.1. A bounded TPN $\mathcal{N}$ is called Language-Robust if $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$ for some $\Delta > 0$. We denote by LR the class of Language-Robust nets and by $\text{LR}_{\neq}$ (resp. $\text{RR}_{\neq}$) the subclass of LR (resp. RR) with distinct labeling.

We first compare the class RR (for which checking membership is decidable by Theorem 6.4) with the class LR (where, as already noted, checking membership is undecidable by Theorem 4.1). We can then observe that:

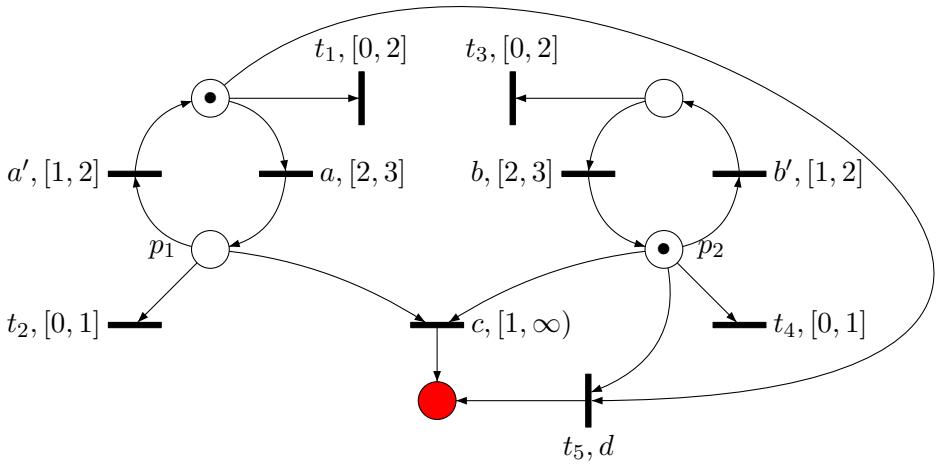


Figure 5. A TPN which is in RR but not LR

Proposition 7.2. (1) The classes RR and LR are incomparable w.r.t. set inclusion. (2) Further, the class $\text{LR}_{\neq}$ is strictly contained in the class $\text{RR}_{\neq}$.

Proof:
We first prove one direction of (1), i.e., RR is not included in LR. Consider the TPN in Figure 5. The set of reachable markings is the same under perturbations so the net is in RR, but the language under perturbation sees the action c which is not seen in the unperturbed net, so this net is not in LR. For the converse direction, it suffices in the net $\mathcal{N}_1$ of Figure 1, to label all transitions by ϵ and then it is in LR (since untimed language is empty) but not in RR since a new place is reachable.

Now for the proof of (2) we have: if $\mathcal{N} \in LR_{\neq}$, then any word $w \in \mathcal{L}(\mathcal{N})$ corresponds to a unique sequence of transitions, and hence leads to a unique marking of $\mathcal{N}$. So if $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$ for some $\Delta > 0$, then $\mathsf{Reach}(\mathcal{N}_\Delta) = \mathsf{Reach}(\mathcal{N})$ for the same Δ. The strictness also follows easily. This inclusion is strict: one can easily design a net $\mathcal{N}$ in which a single transition t is fireable only under enlargement, but producing no new marking outside $\mathsf{Reach}(\mathcal{N})$. Hence, such $\mathcal{N}$ is not in $\text{LR}_{\neq}$, but is still in $\text{RR}_{\neq}$. □

Finally, we show that the problem of robust untimed language preservation becomes decidable under the assumption of distinct labeling:

Theorem 7.3. The class $\text{LR}_{\neq}$ is decidable, i.e., checking if a distinctly labeled bounded TPN is in LR is decidable.

Proof:
We proceed as follows. We first decide by using Theorem 6.4, whether the given distinctly labeled bounded net $\mathcal{N}$ is in RR (and therefore in $\text{RR}_{\neq}$). Now, by Proposition 7.2 if the net is not in $\text{RR}_{\neq}$, then it is not in $\text{LR}_{\neq}$. Otherwise, by Lemma 6.5, we can build a timed automaton $\mathcal{A}$ which is timed bisimilar to $\mathcal{N}$ for small perturbations. This entails that this TA preserves its untimed language under small perturbations iff $\mathcal{N}$ does. Thus we have reduced the problem of checking if $\mathcal{N}$ is in $\text{LR}_{\neq}$ to checking if the timed automaton $\mathcal{A}$ constructed from $\mathcal{N}$ is language-robust. This completes our proof since this problem is decidable for timed automata. More specifically we want to check that $\mathcal{A}$ is in LR, i.e., if there exists $\Delta > 0$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_\Delta)$. In [24] this exact problem is solved for both finite and infinite words but with an additional restriction on the timed automata. Further, it also follows from Proposition 3.1 for general timed automata in the finite words case. That is,

Claim 7.4. Checking if a timed automaton $\mathcal{A}$ is LR is decidable.

Proof:
In [9], it is proved that checking robustness of timed automata with respect to any ω-regular property is decidable. In particular safety properties are decidable, as it is stated in Proposition 3.1. Given a finite timed automaton $\mathcal{A}$, the (untimed) language of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is a regular language. We can build a finite state automaton $\mathcal{C}$ accepting the complement of this language, equipped with final states. Let $\mathcal{B}$ be another timed automaton, and denote by $\mathcal{B} \otimes \mathcal{C}$ the product of $\mathcal{B}$ with $\mathcal{C}$. It is easy to verify that $\mathcal{B} \otimes \mathcal{C}$ never enters a final state of $\mathcal{C}$ iff the (untimed) language of $\mathcal{B}$ is included in that of $\mathcal{A}$. As for any non-negative Δ we have $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_\Delta)$, we obtain that $\mathcal{A}_\Delta \otimes \mathcal{C}$ does not enter the final states of $\mathcal{C}$ iff $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_\Delta)$. As $\mathcal{C}$ is untimed, the two timed automata $\mathcal{A}_\Delta \otimes \mathcal{C}$ and $(\mathcal{A} \otimes \mathcal{C})_\Delta$ are equal. Our problem thus reduces to a robust safety problem for the automaton $\mathcal{A} \otimes \mathcal{C}$. □

This completes the proof of the theorem as detailed above. □

8. Robustness of Subsets of Markings

We now consider the problem of robustness of markings on a subset of places. Given a set of markings M over a set of places P, and a subset of places $X \subseteq P$, we denote by $\Pi_X(M)$ the projection of M onto X. Recall that the subset-markings robustness (SMR) problem is formulated as: given a net $\mathcal{N}$ defined over a set of places P, and a subset of places $X \subseteq P$, is there a value $\Delta > 0$ such that $\Pi_X(\mathsf{Reach}(\mathcal{N})) = \Pi_X(\mathsf{Reach}(\mathcal{N}_\Delta))$?

From Theorem 4.1, we know that the SMR problem is undecidable for bounded TPNs.

Let us now discuss a particularity of SMR problem with respect to other robustness problems described in the paper. It is well known that the marking equivalence and inclusion problems are undecidable for untimed Petri nets [18]. We recall here the definitions of these problems: for a fixed set of places P and a pair of nets $\mathcal{N}_1, \mathcal{N}_2$ defined over P given with their respective initial markings (i.e., $\mathcal{N}_1, \mathcal{N}_2$ are nets with same set of places but with different sets of transitions), is the set of reachable markings of $\mathcal{N}_1$ included in / equal to the set of reachable markings of $\mathcal{N}_2$?

Consider the net of Figure 6, and the SMR problem for places $X = \{x, y, z\}$. One wants to decide whether there exists a value Δ for which the set of markings projected on X is equivalent in $\mathcal{N}$ and in $\mathcal{N}_\Delta$. When place p_1 is filled, the semantics of the net uses transitions $T_1 = \{a, b\}$ and if p_2 is filled, then only transitions $T_2 = \{c, d\}$ are fireable. Similarly, if p_1 is marked, the initial marking in the set of places X is $\{y\}$, while it is equal to $\{x\}$ if p_2 is marked. We know that under any enlargement, place p_2 can be marked, but it can not be marked if the net is not enlarged by some positive Δ. Note that transitions in T_1, T_2 have $[0, \infty)$ time constraint, and can hence be considered as transitions of an untimed net. Hence, the SMR problem on X resumes to checking marking equivalence of the restriction of this net to places in X and transitions $T_1 \cup T_2$ and of a resticion of the net to X with transitions in T_1. That is, markings of $\mathcal{N}$ are robust on X if and only if the markings of $\mathcal{N}_2 = (X, T_2, \{x\})$ are included in those of $\mathcal{N}_1 = (X, T_1, \{y\})$.

This example can be extended, and hence the SMR problem can be used to encode marking inclusion problems for arbitrary sets of places and transitions. One can remark that the net of Figure 6 can be separated into a constrained part, and an unconstrained part (i.e. with $[0, \infty)$ constraints), but that the fact that the subset of places chosen for the SMR problem concerns unconstrained transitions does not change the undecidability result. Hence this undecidability is not only due to timed properties of the net, but also to undecidable results of inclusion of markings for untimed Petri nets [18].

Now, the question is whether the SMR problem is decidable for some subclasses of bounded TPNs.

Proposition 8.1. The subset markings robustness problem:

- always has a positive answer (i.e., is always true) for RR and sequential nets.
- is decidable for Robustly bounded nets.

Proof:
Obviously, for nets in the reach-robust class, the SMR problem is always satisfied, as $\mathsf{Reach}(\mathcal{N}) = \mathsf{Reach}(\mathcal{N}_\Delta)$ for some $\Delta > 0$. Hence for any set $X \subseteq P$, and for the same Δ, we have $\Pi_X(\mathsf{Reach}(\mathcal{N})) = \Pi_X(\mathsf{Reach}(\mathcal{N}_\Delta))$. This property also holds for sequential nets, which are a subclass of RR nets.

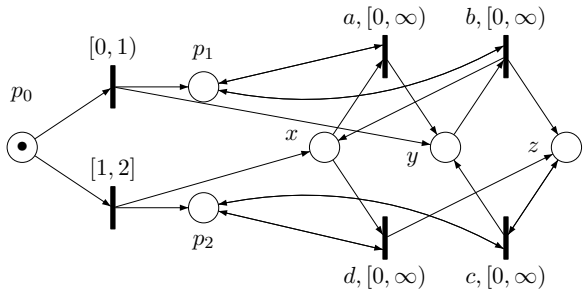


Figure 6. Marking equivalence for a subset of places

Next, when a net is robustly bounded, then by lemma 6.5, one can compute a timed automaton $\mathcal{A}$ such that there exists $\Delta_0 > 0$ for which, $\forall 0 \leq \Delta \leq \Delta_0$, $[\![\mathcal{N}_\Delta]\!] \approx [\![\mathcal{A}_\Delta]\!]$. The construction of this automaton comes with a finite set of markings M such that $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M$ for any $0 \leq \Delta \leq \Delta_0$.

Let us denote by Bad the set of markings that are not equivalent on X to some reachable marking of $\mathcal{N}$, that is $Bad = \{m \in M \mid \nexists m' \in \mathsf{Reach}(\mathcal{N}), \Pi_X(m') = \Pi_X(m)\}$. The SMR problem has a positive answer for $\mathcal{N}$ and X if and only if there exists a positive enlargement $\Delta \leq \Delta_0$ of $\mathcal{A}$ for which the set of markings Bad is not reachable.This safety property can be checked using the results of [9]. □

9. Case Study : regulation of a metro railway system

We propose a case study demonstrating the importance of verifying robustness issues in Petri net models. Petri nets and their colored, timed and stochastic variants have attracted a lot of attention within the context of railway systems. One reason is that the topology of Petri nets mimics that of train networks. One can cite [17, 19] for generic models of trains systems, and the following case studies: [10] considers realizability of time tables for the Bern Metro modeled as a Petri net, [6] proposes a Petri net model that is isomorphic to the real Oslo subway, and [26] defines a Petri net model for the European Train Control System. Though the literature on Petri net models for train systems is not limited to these references, as far as we could find, none of these or other references directly address timed robustness issues. The case study proposed in this section defines a time Petri net model of a pair of metro lines. From this example, natural security and scheduling questions that can be addressed as robustness questions arise. We show that robustness issues can be used to help evaluating robustness of time tables to imprecision, and can help in the regulation of train systems.

Let us consider a simple but illustrative model of regulation of a metro line. In the rest of this section, we will use it to demonstrate several problematic but yet frequently met features of metro lines. Regulation on a metro line consists of ensuring that train departures and arrivals follow (or at least are close to) a strict pre-planned schedule. Any delay in train arrivals forces re-evaluation of the entire schedule. This task is more involved than it seems at first sight and cannot always be ensured by just increasing/decreasing trains speed. For instance, some track portions may be shared among distinct lines and the regulation needs to ensure that shared tracks are occupied by a single train at any instant.

We consider a specific case of a bidirectional metro train network with two lines establishing connections between pairs of distant sites A, B (through intermediate stations $S1, S2, S3$) and C, D (through

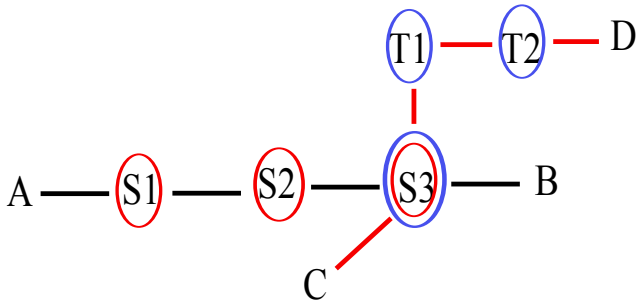
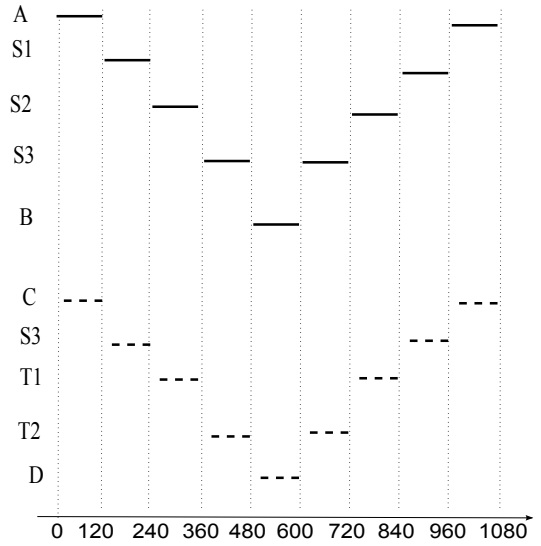


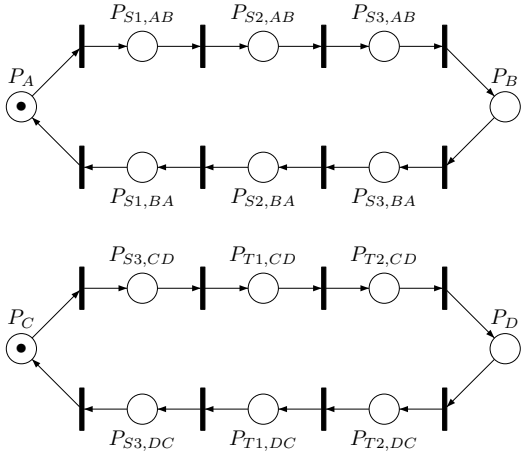
Figure 7. A two-bidirectional lines train network

intermediate stations $S3, T1, T2$) as shown in Figure 7. The two lines share a common portion of track around station $S3$. Each line has a single train and is composed of a single track used in both directions. Further, we assume that all departures and arrivals are scheduled at regular and exact intervals of 2 minutes (120 seconds). Hence, a train occupies (or at least *should* occupy, according to the idealized schedule) a track portion around a station for 2 minutes.

Figure 8(a) represents occupation times for each track portion of the network. Occupation of track portions around stations by the train on AB line is represented by solid black segments, and occupation by train on CD line by dashed segments. One can notice that under a precise timing, the trains on AB and CD pass $S3$ at disjoint intervals: every 18 minutes, the occupation of lines repeats. Hence, $S3$ is occupied by train CD between dates $18.k+2$ and $18.k+4$ min in the C to D direction, and then between



(a) Occupation of stations with perfect time measurement



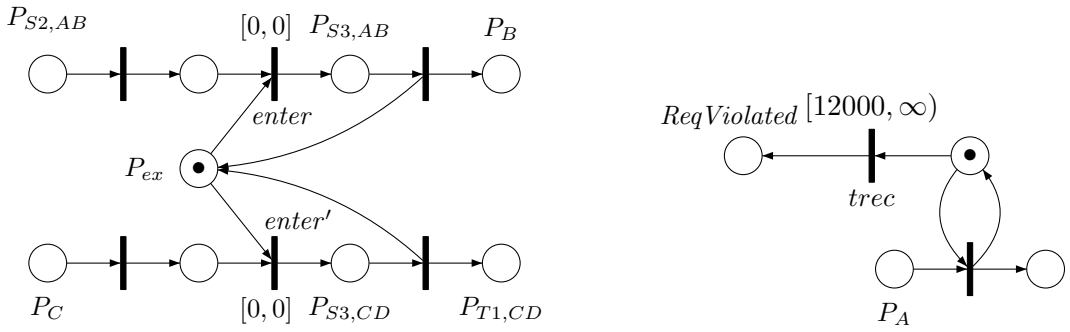
(b) A Petri net model

Figure 8. Behavior and model for precise timing

dates $18.k + 14$ and $18.k + 16$ minutes in the reverse direction, for any $k \in \mathbb{N}$. Similarly, $S3$ is occupied by train AB between dates $18.k+6$ and $18.k+8$ minutes in the A to B direction, and between $18.k+10$ and $18.k+12$ minutes in the reverse direction for any $k \in \mathbb{N}$. We can model this network as a simple time Petri net as illustrated in Figure 8(b) (we will denote this net by $\mathcal{N}$ in the rest of this section). A token in the place $P_{x,AB}$ represents that a train on the way from A to B is at station x (similarly for $P_{x,BA}$ and C, D). Each transition is attached a time interval $[120, 120]$ to depict that the time spent on each portion of the line between two consecutive stations is exactly 2 minutes. One can notice that the net of Figure 8(b) associates one subnet to each line, and that these subnets are two simple independent rings following the lines topologies. One can also notice that the Petri net underlying the TPN $\mathcal{N}$ is robustly bounded. We denote by M its finite set of reachable markings. Now, to avoid collisions at station $S3$, places $P_{S3,AB}$ and $P_{S3,CD}$ as well as place $P_{S3,BA}$ and $P_{S3,DC}$ should be mutually exclusive, i.e., we need to check that the set of markings $M_{bad} = \{m \in M \mid m(P_{S3,AB}) + m(P_{S3,CD}) > 1\} \cup \{m \in M \mid m(P_{S3,BA}) + m(P_{S3,DC}) > 1\}$ is not reachable. This is indeed the case in the absence of imprecisions: one can easily notice that the occupation dates for station S3 by train from the AB and CD are disjoint in both directions. Furthermore, we can check that $[\![\mathcal{N}]\!] = [\![\mathcal{N}]\!]_{|M \setminus M_{bad}}$.

Let us now consider imprecise timings modeled as guard enlargements. Under the enlarged semantics, the safety of the line may depend on the precision of schedules, that is one should check that there exists $\Delta > 0$ such that $\mathsf{Reach}(\mathcal{N}_\Delta) \subseteq M \setminus M_{bad}$. According to the results of Lemma 6.2, this can be effectively checked on our network model, since $M \setminus M_{bad}$ is a finite set of markings. Obviously, any imprecision in timing may result in a collision of trains at station S3, as occurrences of the smallest delay Δ accumulate at each round of trains. Indeed, for any small value Δ, of can find some n such that $18.(n + \Delta) + 2 \leq 18.(n - \Delta) + 16 \leq 18.(n + \Delta) + 4$, i.e. a number of cycles after which trains from lines AB and CD moving in the A to B and C to D directions may both require access to station $S3$. Similar inequations can be written for the reverse directions. This safety verification is rather straightforward for our example, as the equations defining occupancy times of station $S3$ are known. For more complex topologies with mutual exclusion mechanisms, it is not clear that the problem can be solved as satisfiability of some inequations. However, the safety question can still be addressed as a **robust safety** problem, requiring that $\mathsf{Reach}(\mathcal{N}_\Delta)$ does not cover markings in M_{bad}.

Of course, real train networks do not rely on precise timings to guarantee safety of passengers, and mutual exclusion mechanisms are implemented to prevent trains from leaving their station when another train is about to enter the same track. Figure 9(a) shows how a critical section forcing trains to wait before entering a track can be implemented with additional places and transitions. In the precise time setting, a token is always available in the places implementing the critical section whenever needed. However, as soon as small delays modify the behavior of the system, critical sections come into play, forcing trains to wait and completely changing the planned schedule. Let us now try to measure the effects of imprecise timing on the overall behavior of this network. Suppose we have a requirement stating that the time interval between two consecutive trains leaving station A should be at most 20 minutes. Checking that such a requirement is achievable can be done by adding a transition and a pair of places as depicted in Figure 9(b): if a train fails to get back to station A within 20 minutes, transition $trec$ can be fired. Now, with precise timing, it is obvious that this transition $trec$ cannot be fired: the schedule of trains is deterministic, and ensures a departure from station A once every 18 minutes. On our model, this corresponds to firing a transition that resets the clock attached to transition $trec$. However, in an imprecise setting, if the intervals attached to each transition are enlarged by some amount Δ, the train schedules can drift, worsened by mutual exclusion that forces trains to wait. Thus, we would like



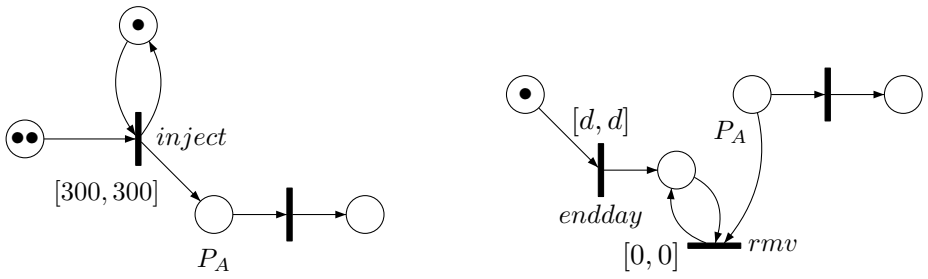
(a) Mutual exclusion (b) Verifying time requirements

Figure 9. A more precise model with mutual exclusion

to check if $trec$ is fireable when guards (on all transitions) are enlarged by some $0 < \Delta < 1$, meaning that the upper bound of 20 minutes between departures from station A is not always met.

Let us denote by $\mathcal{N}'$ the net obtained from $\mathcal{N}$ by adding the places and transitions for mutual exclusion as well as a transition $trec$ and a place $ReqViolated$, as depicted on Figure 9. To check the above property, we first check if place $ReqViolated$ can be marked in the enlargement of $\mathcal{N}'$ by 1. This enlarged net is a bounded time Petri net, and this property can be checked by a construction of its state class graph. If the answer is no, then for any smaller enlargement $\Delta \leq 1$, the place cannot be marked (by Lemma 3.2 since $[\![\mathcal{N}'_\Delta]\!] \preceq [\![\mathcal{N}'_1]\!]$) and hence the 20 minutes upper bound between two trains is respected for any $0 \leq \Delta \leq 1$. If the answer is yes, then timing of trains cannot drift by 1 second without violating the requirement of one train at most every 20 minute at station A. A question that immediately arises is whether there exists a better precision that can guarantee this upper bound on train departures from station A, that is whether there exists Δ smaller than 1 such that $trec$ is never fireable, or equivalently such that place $ReqViolated$ is never marked in $\mathcal{N}'_\Delta$. The net $\mathcal{N}$ is robustly bounded, as transitions do not create new tokens. There are several ways to check validity of the 20 mn upper bound requirement at station A with imprecise timing. The first solution is to reduce the question to a **robust safety** question. Letting m_{fail} be the marking such that $m_{fail}(ReqViolated) = 1$ and $m_{fail}(p) = 0$ for all other places p, satisfying the requirement under enlargement is equivalent to robust safety of $\mathcal{N}$ w.r.t. m_{fail}. Now, if we first check that m_{fail} is not coverable under perfect semantics, the question can also be addressed as a **subset-marking robustness** problem, in which the considered subset of places is $\{ReqViolated\}$. Similarly, the upper bound contraint is violated iff transition $trec$ can be fired. If $trec$ does not appear in any sequence of transitions firings of $\mathcal{N}$ (which can effectively be checked by construction of the state class graph of $\mathcal{N}$), then one can attach an ϵ label to all transitions in $T \setminus \{trec\}$ and any label a to $trec$. The violation of the upper bound requirement can then be checked as a **robust untimed language preservation** problem. If we use a translation to timed automata, then the algorithms of [9] can solve the robustness questions, and allow to compute a value δ_0 such that robustness holds for any value $\Delta \leq \delta_0$.

Other interesting robustness properties can be checked for railway systems modeled as time Petri nets. Usually, metro lines work with more than one train. And of course, the resulting network should remain (robustly) 1-bounded on places modeling tracks, i.e., avoid collisions. Injection of several trains can follow a predetermined schedule, defined as a new part of the TPN. Figure 10(a) shows an injection



(a) Train injection (b) Train removal

Figure 10. A model with many trains

of two trains in the network at station A with an interval of 5 min. With a precise time semantics, the resulting network is 1-bounded on places $X_{AB} = \{P_{x,AB}, P_{x,BA} \mid x \in \{A, S1, S2, S3, B\}\}$ and $X_{CD} = \{P_{y,CD}, P_{y,DC} \mid y \in \{C, S3, T1, T2, D\}\}$, and overall 2-bounded. Now, we may ask whether these bounds are preserved for enlarged semantics. Obviously, on our example, if trains can travel for an unbounded time in a network, a train can always catch up the preceding one even when imprecision is small. Obviously, places in X_{CD} are still 1-bounded after introduction of several trains on line AB, and problems can only arise on places of line AB. Absence of collisions on line AB can be verified as a **subset-markings robustness** problem for the set of places $X = X_{AB}$. Equivalently, one can check absence of collisions on line AB as a **robust safety** problem (one should not cover a marking in which some place in X_{AB} contains 2 tokens).

Note also that metro lines are operated for a bounded duration within one day. Hence, the usual sensible questions are whether the system remains robust during a period that can not exceed one day. When Zeno behaviors are forbidden, this means that one can compute an upper bound on the maximal number K of train departures that can occur within one day, and solve robustness problems under finite horizon, using the techniques of section 3.2, and in particular the construction of lemma 3.4. In particular, an important property to check is that the set of markings remains unchanged on places of X_{AB}. Another possibility is to model explicitly termination of lines operation, and specify the extraction of trains at the end of the day, that is after a fixed delay d. This way, the net deadlocks after some time (no discrete transition can be fired), but there is no need to compute explicitly a bound on the number of train departures. Figure 10(b) shows how to remove trains (tokens) from the network as soon as they arrive at station A at the end of the day. The end of the day is modeled as a transition $endday$, whose firing allows for the removal of a token by transition rmv as soon as it enters place P_A. Clearly, there are small values for enlargement for which this network can still work with bounded place contents for places in X_{AB}. As the network is 2-bounded, one can then compute an equivalent automaton over the set of bounded markings, define a set of bad markings as the set of markings in which a place in X_{AB} contains more than 1 token, and then solve this boundedness problem as a **robust safety** problem. Now, an interesting question is to find the largest value for Δ guaranteeing 1-boundedness of places in X_{AB} within one day. These questions can be addressed building a finite symbolic unfolding of the net as in [12], and the deriving a system of inequations in which firing dates of transitions and Δ are variables, and trying to maximize Δ within this system.

The questions addressed in this case study show that safety, language and subset marking robustness

questions have a practical interest. For simple cases such as the mutual exclusion on station $S3$ for the net of Figure 8(b), the question can be rapidly answered by writing the equations for departure and arrival dates of trains. However questions becomes more complex as soon as mutual exclusion mechanisms introduce combinatorics in the model. So far, all robustness questions for this case study are only settled, and the first question is answered on paper. We plan to develop tools to solve robustness problems for TPNs, as it is unlikely that robustness issues with similar or greater complexity can be addressed without the help of a program.

10. Conclusion

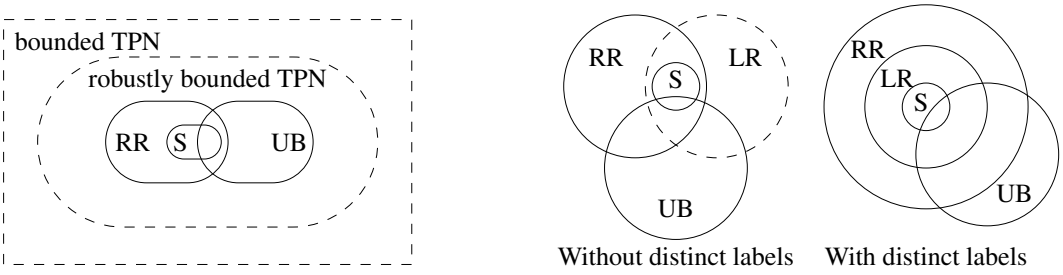


Figure 11. RR stands for reach-robust, LR for language-robust, UB for bounded underlying PNs, S for sequential bounded TPNs. Dotted lines represent that checking membership in the class is undecidable, while solid lines represent that it is decidable.

In this paper, we have launched an investigation into robustness in time Petri nets with respect to guard enlargements. We transferred several positive results from the TA setting to TPNs and showed that some other problems become undecidable in TPNs due to unboundedness. We summarize our results in the diagram in Figure 11 which shows the relative expressiveness of the various classes and their decidability. In the table below, we give a more detailed picture of the decidability results for each problem and each class of TPNs considered. The leftmost column lists the problems addressed while the top row lists the subclasses of nets. A "yes" entry in a cell means that the problem considered at that row is decidable for the class considered in the column, "no" means that it is undecidable, and "g" means that a positive answer to the problem is always guaranteed for the considered class. For instance, robustness of subset-markings is guaranteed for Reach-robust nets.

The undecidability results stem from Theorem 4.1 and Proposition 4.2, that is from the capacity for unbounded TPNs to encode Minsky machines. Robustness of a subset of markings for unbounded nets can also encode reachability problems for Minsky machines, but in addition can encode undecidable marking equivalence problems for untimed Petri nets [18]. Decidability results on robust language preservation with distinct labels follow from Theorem 7.3, which says that one can decide whether a bounded net belongs to $\mathrm{LR}_{\neq}$, i.e. preserves its language under some enlargement.

Problems \ Classes	TPNs	Bounded	Rob. Bounded	RR	UB	S
Robust Boundedness	no	no	g	g	g	g
Robust language preservation (with distinct labels)	no	yes	yes	yes	yes	g
Robust language preservation (without distinct labels)	no	no	yes	yes	yes	g
Robustness of $\mathsf{Reach}(\mathcal{N})$	no	yes	yes	g	yes	g
Robustness of subset-markings	no	no	yes	g	yes	g

For the case of bounded nets without distinct labels, the language preservation problem is undecidable: this is again a consequence of Theorem 4.1, which shows that $\mathrm{LR}_{=}$ is an undecidable class of nets. Once we assume the net to be robustly bounded (or RR or UB which are subsets of robustly bounded TPNs), the "yes" results are proved using Lemma 6.5 and following the same lines as in the proof of Theorem 7.3. Indeed, if a net is robustly bounded, then by Lemma 6.5, one can build a finite automaton $\mathcal{A}$ such that there exists $\Delta_0 > 0$ for which, $\forall 0 \leq \Delta \leq \Delta_0$, $[\![\mathcal{N}_\Delta]\!] \approx [\![\mathcal{A}_\Delta]\!]$. Obviously, the set of locations of this automaton is finite, that is it gives us a set of markings preserved by any enlargement smaller than Δ_0. One can hence compute the regular untimed language recognized by $\mathcal{N}$ and design an untimed automaton $\mathcal{C}$ that recognizes its complement. As in Theorem 7.3, language robustness then reduces to a decidable robust safety problem for $\mathcal{A} \otimes \mathcal{C}$.

Robustness of subset-markings problem is decidable for all subclasses of robustly bounded nets, and guaranteed for Reach Robust and sequential nets, as shown by Proposition 8.1. Decidability of robustness of $\mathsf{Reach}(\mathcal{N})$ for bounded and robustly bounded nets, comes from the fact that RR is a decidable subclass of nets (Theorem 6.4). Robustness of $\mathsf{Reach}(\mathcal{N})$ is guaranteed by definition of RR and bounded sequential nets are a subclass of RR nets. For sequential nets, all robustness properties considered in the table are guaranteed, as demonstrated by Lemma 3.5.

As future work, we would like to show positive results in an unbounded setting. However, we believe that this would require a different approach and new techniques. Another challenge is to address new robustness problems that arise due to concurrency in TPNs. For instance, one may like to check that there is no change in concurrency under enlargement, that is the subset of transitions that can be fired simultaneously remain unchanged. Such issues call for the study of robustness directly on non-interleaved representations.

References

[1] Akshay, S., Hélouët, L., Jard, C., Reynier, P.-A.: Robustness of Time Petri Nets under Guard Enlargement, *Proc. of RP*, 7550, Springer, 2012.

[2] Alur, R., Dill, D.: A Theory of Timed Automata, *In TCS*, **126**(2), 1994, 183–235.

[3] Bérard, B., Cassez, F., Haddad, S., Lime, D., Roux, O. H.: Comparison of Different Semantics for Time Petri Nets, *Proc. of ATVA*, 3707, Springer, 2005.

[4] Bérard, B., Cassez, F., Haddad, S., Lime, D., Roux, O. H.: Comparison of the Expressiveness of Timed Automata and Time Petri Nets, *Proc. of FORMATS*, 3829, Springer, 2005.

[5] Berthomieu, B., Diaz, M.: Modeling and Verification of Time Dependent Systems Using Time Petri Nets, *IEEE Trans. in Software Engineering*, **17**(3), 1991, 259–273.

[6] Bjork, J., Hagalisletto, A.: *Challenges in simulating railway systems using Petri Nets*, Technical report, Precise Modeling and Analysis, Department of Informatics, Univesity of Oslo, 2005.

[7] Bouyer, P., Markey, N., Reynier, P.-A.: Robust Model-Checking of Linear-Time Properties in Timed Automata, *Proc. of LATIN*, 3887, Springer, 2006.

[8] Bouyer, P., Markey, N., Reynier, P.-A.: Robust Analysis of Timed Automata *via* Channel Machines, *Proc. of FoSSaCS*, 4962, Springer, 2008.

[9] Bouyer, P., Markey, N., Sankur, O.: Robust Model-Checking of Timed Automata via Pumping in Channel Machines, *Proc. of FORMATS*, 6919, Springer, 2011.

[10] Burkolter, D. M.: *Capacity of Railways in Station Areas using Petri Nets*, Ph.D. Thesis, Swiss Federal Institute Of Technology, Zurich, Switzerland, 2005.

[11] Cassez, F., Roux, O. H.: Structural translation from Time Petri Nets to Timed Automata, *Journal of Systems and Software*, **79**(10), 2006, 1456–1468.

[12] Chatain, T., Jard, C.: Complete Finite Prefixes of Symbolic Unfoldings of Safe Time Petri Nets, *Petri Nets and Other Models of Concurrency - ICATPN 2006, 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, Turku, Finland, June 26-30, 2006, Proceedings*, 2006.

[13] D'Aprile, D., Donatelli, S., Sangnier, A., Sproston, J.: From Time Petri Nets to Timed Automata:An Untimed Approach, *Proc. of TACAS*, 4424, Springer, 2007.

[14] De Wulf, M., Doyen, L., Markey, N., Raskin, J.-F.: Robust Safety of Timed Automata, *Formal Methods in System Design*, **33**(1-3), 2008, 45–84.

[15] De Wulf, M., Doyen, L., Raskin, J.-F.: Systematic Implementation of Real-Time Models, *Proc. of Formal Methods*, 3582, Springer, 2005.

[16] Gardey, G., Roux, O. H., Roux, O. F.: A Zone-Based Method for Computing the State Space of a Time Petri Net, *Proc. of FORMATS*, 2791, 2003.

[17] Giua, A., Seatzu, C.: Modeling and Supervisory Control of Railway Networks Using Petri Nets, *IEEE T. Automation Science and Engineering*, **5**(3), 2008, 431–445.

[18] Hack, M.: *Decidability Questions for Petri Nets*, Ph.D. Thesis, M.I.T., MIT, CA, USA, 1976.

[19] Janczura, C. W.: *Modelling and Analysis of Railway Network Control Logic using Coloured Petri Nets*, Ph.D. Thesis, School of Mathematics University of South Australia, Adelaide, South Australia, 1998.

[20] Karp, R., Miller, R.: Parallel program chemata, *In JCSS*, **3**, 1969, 147–195.

[21] Lime, D., Roux, O. H.: Model Checking of Time Petri Nets Using the State Class Timed Automaton, *Discrete Event Dynamic Systems*, **16**(2), 2006, 179–205.

[22] Merlin, P. M.: *A Study of the Recoverability of Computing Systems*, Ph.D. Thesis, University of California, Irvine, CA, USA, 1974.

[23] Puri, A.: Dynamical Properties of Timed Automata, *In DEDS*, **10**(1-2), 2000, 87–113.

[24] Sankur, O.: Untimed Language Preservation in Timed Systems, *Proc. of MFCS*, 6907, Springer, 2011.

[25] Swaminathan, M., Fränzle, M., Katoen, J.-P.: The Surprising Robustness of (Closed) Timed Automata against Clock-Drift, *Proc. of TCS*, Springer, 2008, ISBN 978-0-387-09679-7.

[26] Zimmermann, A., Hommel, G.: A Train Control System Case Study in Model-Based Real Time System Desig, *IPDPS*, 2003.