Robustness of Time Petri Nets under Guard Enlargement

Sundararaman Akshay, Loic Helouet, Claude Jard, Pierre-Alain Reynier

To cite this version:

Sundararaman Akshay, Loic Helouet, Claude Jard, Pierre-Alain Reynier. Robustness of Time Petri Nets under Guard Enlargement. Reachability Problems, LABRI, Sep 2012, Bordeaux, France. pp.92-106. hal-00879832

HAL Id: hal-00879832
https://hal.inria.fr/hal-00879832

Submitted on 5 Nov 2013

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

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

Robustness of Time Petri Nets under Guard Enlargement

S. Akshay[1,2], Loïc Hélouët[2], Claude Jard[1,2], Pierre-Alain Reynier[3]

[1] INRIA/IRISA Rennes, France
[2] ENS Cachan Bretagne, Rennes, France
[3] Laboratoire d'Informatique Fondamentale de Marseille, France

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.

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 [1] (TA), time Petri nets [13] (TPN) 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, but also develop robust models of systems, that is, 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 [14, 4, 3, 8, 5, 15]. In [9], it is proven that this model of perturbation covers the both issues 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 (TPNs) [13]. 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 two problems. The first is the *robust boundedness* problem, which consists in 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. 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 TPN. For instance, robust safety, that is avoidance of some bad configuration under perturbation, is decidable in TAs, 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 [14], 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 makes a huge difference with 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 two problems we consider are undecidable.

There are several translations from TPN to TA [10, 6, 12, 7]. We study which of these translations can be used to lift robustness results on TA to the model of TPN. In particular, we prove that the marking class timed automaton construction of [7] 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 and language preservation are decidable.

This paper is organized as follows: Section 2 defines the models used throughout the paper. Section 3 introduces our perturbation model for TPNs, and the robust boundedness and language 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, before conclusion. Missing proofs can be found in appendix.

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 $I^{\downarrow} = \{x \mid x \leq y \text{ for some } y \in I\}$, 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 in $\mathbb{R}_{\geq 0}^X$. We note $\mathbf{0}$ the valuation which assigns to every clock $x \in X$ 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 1 (Timed Transition System (TTS)). *A* timed transition system *over Σ_ε is a transition system $S = (Q, q_0, \rightarrow)$, where Q is the set of states, $q_0 \in Q$ is the initial state, and the transition relation $\rightarrow$ 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 $\rightarrow$.*

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 $\rho = q_0 \xrightarrow{d_0} q_0' \xrightarrow{a_0} q_1 \xrightarrow{d_1} q_1' \xrightarrow{a_1} q_2 \ldots \xrightarrow{a_n} q_{n+1}$ where discrete actions and delays alternate, and which starts in the initial configuration. To such a run corresponds a word $a_0 \ldots a_n$ 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 (Time Petri Nets (TPN)). *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, $\Lambda : T \rightarrow \Sigma_\varepsilon$ 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 [13], Time Petri nets (TPNs) associate a time interval to 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 to 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 defined as usual (see for instance [2]). An enabled transition t can be fired if $\nu(t)$ belongs to the interval $I(t)$. The result of this firing is as usual the new marking $m' = m - {}^\bullet t + t^\bullet$. Moreover, some valuations are reset. We say that the a 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. $\nu(t) \in I(t)$, $\forall t' \in \mathit{En}(m - {}^\bullet t + t^\bullet)$, $\nu'(t') = 0$ if $\uparrow\!\mathit{enabled}(t', m, t)$ and $\nu'(t') = \nu(t)$ otherwise

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.

Timed automata: First defined in [1], 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 $\bowtie \in \{<, \leq\}$.

Definition 3 (Timed Automata (TA)). *A* timed automaton $\mathcal{A}$ *over* Σ_ε *is a tuple* $(L, \ell_0, X, E, \mathit{Inv})$ *where* L *is a finite set of* locations, $\ell_0 \in L$ *is the* initial location, X *is a finite set of* clocks, $\mathit{Inv} \in \mathcal{C}_{ub}(X)^L$ *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 $\mathcal{A} = (L, \ell_0, X, E, \mathit{Inv})$ is the TTS $[\![\mathcal{A}]\!] = (Q, q_0, \to)$ where $Q = \{(\ell, v) \in L \times (\mathbb{R}_{\geq 0})^X \mid v \models \mathit{Inv}(\ell)\}$, $q_0 = (\ell_0, \mathbf{0})$ and $\to$ is defined by:

- **delay moves:** $(\ell, v) \xrightarrow{d} (\ell, v + d)$ if $d \in \mathbb{R}_{\geq 0}$ and $v + d \models \mathit{Inv}(\ell)$;
- **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 $\sigma \in \Sigma_\epsilon \cup R$, $q_1 \in Q$, $q_1' \in Q'$ 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 [4, 3, 8]: 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). 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 ($\mathsf{Reach}(\mathcal{A}_\Delta)$ denotes the locations of $\mathcal{A}$ reachable in $[\![\mathcal{A}_\Delta]\!]$):

Proposition 1 ([5]). *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 $\mathsf{Reach}(\mathcal{A}_\Delta) \cap S = \emptyset$.*

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 the desired monotony property holds, entailing 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 1. *Let $\mathcal{N}$ be a TPN and $\Delta \leq \Delta' \in \mathbb{R}_{\geq 0}$. We have $[\![\mathcal{N}_\Delta]\!] \preceq [\![\mathcal{N}_{\Delta'}]\!]$.*

3.1 Problems considered

We now define robustness problems on TPNs in a way which is consistent with the monotony property stated above.

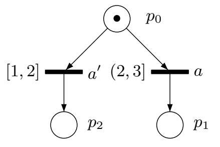
Robust Boundedness: Given a bounded TPN $\mathcal{N}$, does there exist $\Delta \in \mathbb{Q}_{>0}$ such that $\mathcal{N}_\Delta$ is bounded?

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

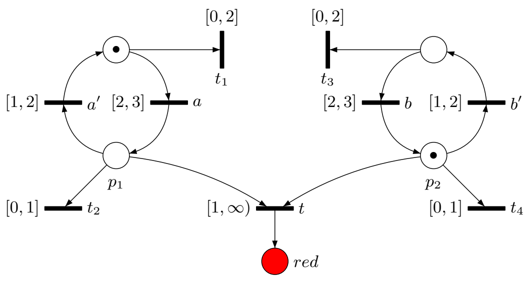
We call a TPN $\mathcal{N}$ *robustly bounded* if there exists $\Delta \in \mathbb{Q}_{>0}$ such that $\mathcal{N}_\Delta$ is bounded. This problem is strongly related to the problem of robust safety asking, given a bounded TPN $\mathcal{N}$ with set of places P, and a marking $m \in \mathbb{N}^P$, whether there exists $\Delta \in \mathbb{Q}_{>0}$ s.t., $\mathsf{Reach}(\mathcal{N}_\Delta)$ 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.

3.2 Examples of non-robust TPNs

Consider the example in Figure 1(a). Due to the open interval and urgency condition (according to the semantics of TPNs, a' has to fire at most 2 time units after enabling), any enlargement of guards would result in reaching place p_1 which is not reachable in



(a) The TPN $\mathcal{N}_0$.



(b) The TPN $\mathcal{N}_1$.

Fig. 1. Two TPNs exhibiting new discrete behaviors under infinitesimal perturbations.

the non-enlarged semantics (from this, we can easily construct examples that are not robustly bounded or robustly safe). In this example, the firing domain of transition a (the set of configurations $\{(p_0, \nu) \mid \nu(a') \in [1, 2]\}$) is a *neighbor* of the reachable configuration $(p_0, \nu(a') = 2)$. By neighbor, we mean that any positive enlargement makes transition a fireable. This is the simplest form of non-robustness which can be easily checked for in bounded TPNs as one can compute a symbolic representation of the reachability set (using the state-class graph construction [2, 12] for instance). Further by requiring that all intervals must be closed, one may avoid this situation. Now assuming there are no transitions whose firing domain is a neighbor of the reachability set, one can prove that under a *bounded time horizon* (as defined for timed automata in [16]) any net is robust, i.e., one can pick a sufficiently small $\Delta > 0$ to ensure that no new behavior occurs.

The remaining case concerns TPNs in which new behaviors are not neighbors of the reachability set, considered for an unbounded time 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 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 [14] gave an example of TA that exhibits accumulations, encoded using time between consecutive resets. However, for TPNs, this encoding does not work since the clocks are always reset when a transition is newly enabled.

We exhibit a TPN where accumulation is due to concurrency in Figure 1(b). This example can be simplified using singleton intervals, but we avoid this to show that accumulation may arise even without singletons. With the usual 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 transition 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 $n \cdot \Delta$ time units. Choosing any $n \geq \frac{1}{\Delta}$, we obtain the result. In particular, the red place is reachable in $(\mathcal{N}_1)_\Delta$, for any positive Δ.

3.3 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 TAs. 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 2. *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 $\textsf{Reach}(\mathcal{N}_\Delta) = \textsf{Reach}(\mathcal{N})$ and $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$.*

Proof (Sketch). 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 $[\![\mathcal{N}_\Delta]\!]$. □

4 Undecidability results

We use the TPNs of Figure 1 to prove undecidability of robustness and untimed language preservation for bounded TPNs.

Theorem 1. *The problems of robust boundedness and robust untimed language preservation are undecidable for bounded TPN.*

Proof (Sketch). To prove undecidability, we combine the standard construction of a TPN from a Minsky machine (see Appendix B) with the gadget from Figure 1 and Lemma 2 on sequential TPNs. The properties of the Minsky machine we use are: given a Minsky machine $\mathcal{M}$, one can build a TPN $\mathcal{N}_\mathcal{M}$ such that $\mathcal{N}_\mathcal{M}$ is bounded iff $\mathcal{M}$ is, and $\mathcal{N}_\mathcal{M}$ covers some marking m iff $\mathcal{M}$ reaches its final state q_n. Moreover, the TPN $\mathcal{N}_\mathcal{M}$ is sequential (it encodes the behavior of $\mathcal{M}$ which is sequential).

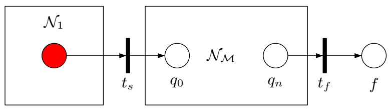


Fig. 2. TPN $\mathcal{N}_2$ obtained by combining $\mathcal{N}_1$ and $\mathcal{N}_\mathcal{M}$.

We combine the TPNs $\mathcal{N}_1$ from Figure 1 and $\mathcal{N}_\mathcal{M}$ as depicted on Figure 2 to obtain the TPN $\mathcal{N}_2$. First note that $\mathcal{N}_2$ is a bounded TPN: without perturbation, transition t 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 [4]. As $\mathcal{N}_\mathcal{M}$ is sequential, by Lemma 2(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 state 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 2(iii) does not require the boundedness assumption.

Thanks to undecidability of halting and boundedness of Minsky machines, the problems we considered are undecidable 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. □

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 [7] but different in two aspects. First, for efficiency reasons [7] reduce the number of clocks of the TA they build, and therefore use clock sharing techniques of [12], 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 [7] 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 [7] 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 4. *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*

[4] 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 abitrarily many b's. It can however not be adapted to the setting of injective labeling, see Section 7.

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 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 $R = \{x_{t'} \mid t' \in \uparrow\!\mathit{enabled}(t', m, t) = \mathtt{true}\}$. With this we have the following theorem:

Theorem 2. *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 (Sketch). We can prove by induction that the following relation R is a timed bisimulation. Let (m, ν) (resp. (ℓ, v)) denote a state of the TTS $[\![\mathcal{N}_\Delta]\!]_{|M}$, i.e. $(m, \nu) \in Adm(\mathcal{N}_\Delta)$ with $m \in M$ (resp. denote a state of $[\![(\mathcal{A}_M)_\Delta]\!]$). We define $(m, \nu) R (\ell, v)$ if and only if $m = \ell$, and $\forall t \in En(m), \nu(t) = v(x_t)$. □

Other TA constructions. The construction proposed in [12] builds a state class timed automaton incrementally using a forward exploration of reachable markings of a bounded TPN. Gardey et al [10] 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 [12, 10] 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 2.

Cassez et al [6] 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 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 3. *The set of TPN 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 [11]. The second part of the above proposition follows from Theorem 2.

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 3. *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 2 and Theorem 2, 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 $\rho = (m_0, \nu_0) \xrightarrow{d_1, t_1} (m_1, \nu_1) \ldots \xrightarrow{d_n, t_n} (m_n, \nu_n)$ 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 2, 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 1. □

We consider the following subclass of bounded TPNs:

Definition 5. *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. 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 3, by instantiating the finite set of markings M with $\mathsf{Reach}(\mathcal{N})$:

Theorem 3. *RR is a decidable subclass of robustly bounded TPNs.*

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

Lemma 4. *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 (sketch). 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 3 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 2, this entails $[\![\mathcal{N}_\Delta]\!]_{|M} = [\![\mathcal{N}_\Delta]\!]$. In addition, by Theorem 2, 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 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 6. *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 LR$_{\neq}$ (resp. 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 3) with the class LR (where, as already noted, checking membership is undecidable by Theorem 1). We can then observe that:

Proposition 4. *The classes RR and LR are incomparable w.r.t. set inclusion. Further, the class LR$_{\neq}$ is strictly contained in the class RR$_{\neq}$.*

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

Theorem 4. *The class 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 3, whether the given distinctly labeled bounded net $\mathcal{N}$ is in RR (and therefore in RR$_{\neq}$). Now, by Proposition 4 if the net is not in RR$_{\neq}$, then it is not in LR$_{\neq}$. Otherwise, by Lemma 4, 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 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 [15] this exact problem is solved for both finite and infinite words but with an additional restriction on the timed automata. Further, it also follows (see Appendix 8, Proposition 6) from Proposition 1 for general timed automata in the finite words case. □

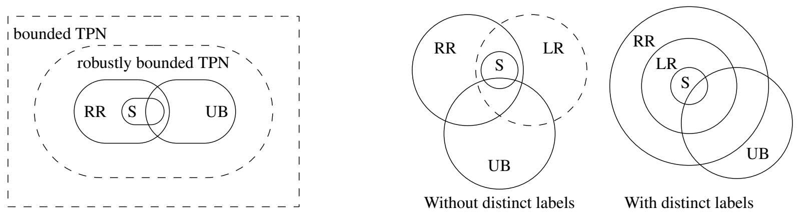


Fig. 3. RR stands for reach-robust, LR for language-robust, UB for bounded underlying PNs, S for sequential bounded TPNs. Dotted lines represent undecidable and solid lines decidable classes.

8 Conclusion

We summarize our results in the above diagram (by a decidable/undecidable class we mean that membership in that class is decidable/undecidable). 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. As future work, we would like to show positive results in an unbounded setting and we believe that this would require a different approach and new techniques.

References

1. R. Alur and D. Dill. A theory of timed automata. *In TCS*, 126(2):183–235, 1994.
2. B. Berthomieu and M. Diaz. Modeling and verification of time dependent systems using time Petri nets. *IEEE Trans. in Software Engineering*, 17(3):259–273, 1991.
3. P. Bouyer, N. Markey, and P.-A. Reynier. Robust analysis of timed automata *via* channel machines. In *Proc. of FoSSaCS'08*, volume 4962 of *LNCS*, pages 157–171.
4. P. Bouyer, N. Markey, and P.-A. Reynier. Robust model-checking of linear-time properties in timed automata. In *Proc. of LATIN'06*, volume 3887 of *LNCS*, pages 238–249, 2006.
5. P. Bouyer, N. Markey, and O. Sankur. Robust model-checking of timed automata via pumping in channel machines. In *Proc. of FORMATS'11*, volume 6919 of *LNCS*, pages 97–112.
6. F. Cassez and O. H. Roux. Structural translation from time petri nets to timed automata. *Journal of Systems and Software*, 79(10):1456–1468, 2006.
7. D. D'Aprile, S. Donatelli, A. Sangnier, and J. Sproston. From time Petri nets to timed automata:An untimed approach. In *TACAS'07*, volume 4424 of *LNCS*, pages 216–230, 2007.
8. M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robust safety of timed automata. *Formal Methods in System Design*, 33(1-3):45–84, 2008.
9. M. De Wulf, L. Doyen, and J.-F. Raskin. Systematic implementation of real-time models. In *Formal Methods (FM'05)*, volume 3582 of *LNCS*, pages 139–156.
10. G. Gardey, O. H. Roux, and O. F. Roux. A zone-based method for computing the state space of a time Petri net. In *Proc. of FORMATS'03*, volume 2791 of *LNCS*, pages 246–259, 2003.
11. R. Karp and R. Miller. Parallel program chemata. *In JCSS*, 3:147–195, 1969.
12. D. Lime and O. H. Roux. Model checking of time petri nets using the state class timed automaton. *Discrete Event Dynamic Systems*, 16(2):179–205, 2006.
13. P. M. Merlin. *A Study of the Recoverability of Computing Systems*. PhD thesis, University of California, Irvine, CA, USA, 1974.
14. A. Puri. Dynamical properties of timed automata. *In DEDS*, 10(1-2):87–113, 2000.
15. O. Sankur. Untimed language preservation in timed systems. In *Proc. of MFCS'11*, LNCS. Springer, 2011.
16. M. Swaminathan, M. Fränzle, and J.-P. Katoen. The surprising robustness of (closed) timed automata against clock-drift. In *TCS 2008*, pages 537–553. Springer.

Appendix

A : Proof of Lemma 2

Proof. We prove that for any $\mathit{\Delta} < \frac{1}{2}$, we have $\mathsf{Reach}(\mathcal{N}_\Delta) = \mathsf{Reach}(\mathcal{N})$ and $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$. Let $\mathit{\Delta}$ be a non-negative real number such that $\mathit{\Delta} < \frac{1}{2}$. Consider a run $\rho = (m_0, \nu_0) \xrightarrow{d_1,a_1} (m_1, \nu_1) \ldots (m_{n_1}, \nu_{n-1}) \xrightarrow{d_n,a_n} (m_n, \nu_n)$ 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 $\rho' = (m_0, \nu_0) \xrightarrow{d'_1,a_1} (m_1, \nu_1) \ldots (m_{n_1}, \nu_{n-1}) \xrightarrow{d_n,a_n} (m_n, \nu_n)$ 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,
- $\forall t' \in \mathit{En}(m_n), \nu_n(t') + d_n \leq \beta(t') + \mathit{\Delta}$, - $\nu_n(t) + d_n \geq \alpha(t) - \mathit{\Delta}$

By induction property, we have $\nu_n(t') = 0$ for all $t' \in \mathit{En}(m_n)$. As a consequence, we can deduce that $\alpha(t) - \mathit{\Delta} \leq d_n \leq \min\{\beta(t') \mid t' \in \mathit{En}(m_n)\} + \mathit{\Delta}$. As transitions have integral bounds, and $\mathit{\Delta} < \frac{1}{2}$, one can verify that this implies the inequality $\alpha(t) \leq \min\{\beta(t') \mid t' \in \mathit{En}(m_n)\}$. We thus pick $d'_n = \alpha(t)$, which ensures:

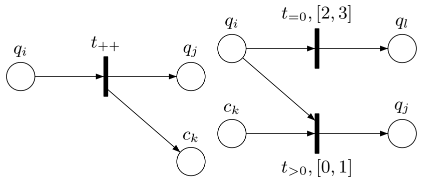
– $\forall t' \in \mathit{En}(m_n), \nu'_n(t') + d'_n = \alpha(t) \leq \beta(t')$,
– $\nu'_n(t) + d'_n \geq \alpha(t)$

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 property on $\mathsf{Reach}(\mathcal{N}_\Delta)$ and $\mathcal{L}(\mathcal{N}_\Delta)$ then directly follows. □

B: Proof of Theorem 1

Proof. To prove undecidability, we use the standard construction of a TPN from a Minsky machine. We recall it briefly for the sake of completeness. A Minsky machine $\mathcal{M}$ (which w.l.o.g. we assume deterministic) is defined by a finite set of state 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 $q_i \rightarrow q_l$ 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 boundednes of c_1 and c_2 along the unique execution of $\mathcal{M}$ is also undecidable.

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 4. 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 $\{q_n\}$ iff M reaches state q_n. Second, in



(a) Incrementation (b) Decrementation

Fig. 4. Encoding instruction of a Minsky machine $\mathcal{M}$ into a TPN $\mathcal{N}_{\mathcal{M}}$.

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 combine the TPNs $\mathcal{N}_1$ and $\mathcal{N}_{\mathcal{M}}$ as depicted on Figure 2 to obtain the TPN $\mathcal{N}_2$. First note that $\mathcal{N}_2$ is a bounded TPN: without perturbation, transition t 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. As $\mathcal{N}_{\mathcal{M}}$ is sequential, we can easily prove that:

- $\mathcal{N}_2$ is robustly bounded iff $\mathcal{N}_{\mathcal{M}}$ is bounded
- $\mathcal{N}_2$ is robustly safe w.r.t. marking $\{f\}$ iff $\mathcal{N}_{\mathcal{M}}$ does not cover state q_n
- $\mathcal{N}_2$ robustly preserves its untimed language iff $\mathcal{N}_{\mathcal{M}}$ does not cover state q_n

As a consequence the three problems we considered are undecidable. □

C: Proof of Theorem 2

Proof. We show that the following relation 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 delay 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 $\uparrow$*enabled*$(t', m, t) =$ `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

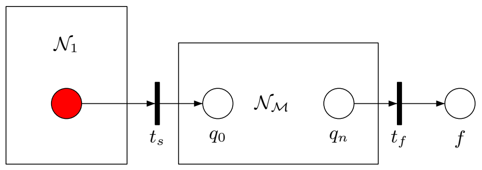


Fig. 5. A TPN which is in RR but not LR

$[\![(\mathcal{A}_M)_\Delta]\!]$, 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 $[\![\mathcal{N}_\Delta]\!]_{|M}$. □

D: Untimed Language Preservation

Proposition 5. *(1) The classes RR and LR are incomparable w.r.t. set inclusion. (2) Further, the class LR$_{\neq}$ is strictly contained in the class 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 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 LR$_{\neq}$, but is still in RR$_{\neq}$. □

Proposition 6. *Checking if a timed automaton $\mathcal{A}$ is LR is decidable.*

Proof. In [5], 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 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 $\mathcal{A} \otimes \mathcal{C}$. □