

Robustness of Time Petri Nets under Guard Enlargement¹

P.A Reyniers⁽¹⁾ S. Akshay^(2,3,4) L. Hélouët^(3,4) C. Jard^(2,3,4)

⁽¹⁾ Aix-Marseille Université, CNRS, LIF, UMR 7279, Marseille, France

⁽²⁾ ENS Cachan Bretagne, Rennes, France

⁽³⁾ IRISA, Rennes, France

⁽⁴⁾ INRIA Rennes, France

September 2012

Many models have an idealized model of time

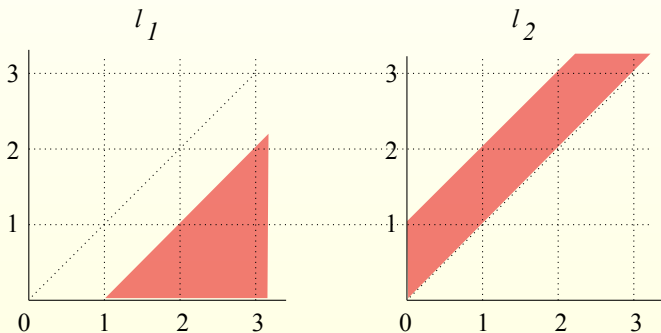
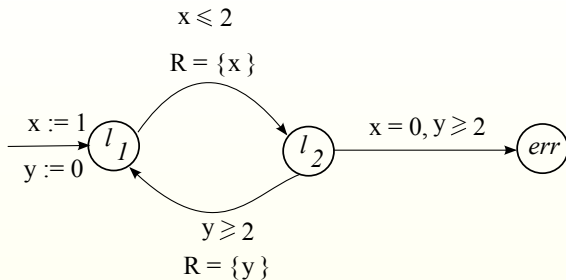
- exact measurement of time,
- exact and instantaneous firing times,
- no clock drift,
- ...

But in the real world :

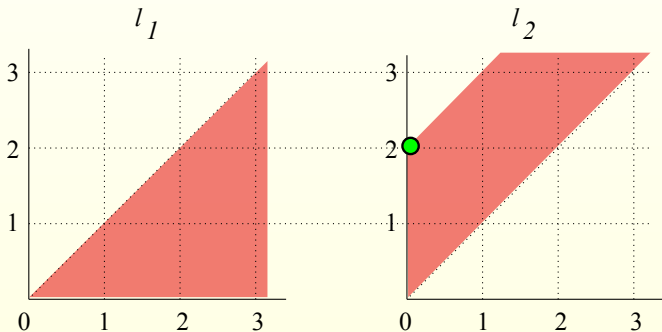
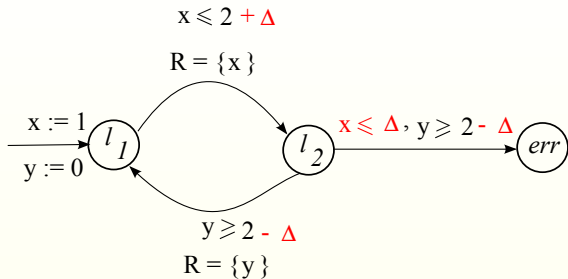
- "firing a after 10 ms" may mean "firing a after 10.0001 ms" (clock imprecision)
- different clocks can measure time on distinct machines with their own pace (clocks drift)
- ...

Ensuring $M \models \phi$ can improve one's confidence, but does it say anything about ϕ in an implementation of M ?

Enlargement in timed automata [Puri00]



Enlargement in timed automata [Puri00]



Enlargement in timed automata

$\mathcal{A} = (\mathcal{L}, \ell_r, X, E, Inv)$

\mathcal{L} locations, X clocks, E : transitions (l, γ, a, R, l') , Inv invariants.

$\llbracket \mathcal{A} \rrbracket$: semantics of \mathcal{A} (Timed Transition System)

$\mathcal{L}(\mathcal{A})$: untimed language of \mathcal{A}

$Reach(\mathcal{A})$: reachable locations of \mathcal{A} .

\mathcal{A}_Δ : Δ -enlarged version of \mathcal{A} .

Lemma 1 (Monotony)

Let \mathcal{A} be a T.A., $\Delta \leq \Delta' \in \mathbb{R}_{\geq 0}$. We have $\llbracket \mathcal{A}_\Delta \rrbracket \preceq \llbracket \mathcal{A}_{\Delta'} \rrbracket$.

Theorem [BouyerMS11]

Let \mathcal{A} be a T.A, S be a subset of locations of \mathcal{A} . One can decide whether there exists $\Delta \in \mathbb{Q}_{>0}$ such that $Reach(\mathcal{A}_\Delta) \cap S = \emptyset$.

Theorem [BouyerMS11]

Robust model checking of ω -regular properties $(\exists? \Delta \in \mathbb{Q}_{>0}, \mathcal{A}_\Delta \models \phi)$ is PSPACE-complete.

Summary of the presentation

Questions :

- Are there similar problems for Time Petri nets ?
- Can we decide similar results ?

Contributions of this work

- Robustness in Time Petri nets (w.r.t enlargement)
- Specific robustness issues due to concurrency
- Robustness issues are in general undecidable for Time Petri nets
- Identify several decidable subclasses of nets for which robustness is guaranteed or decidable.

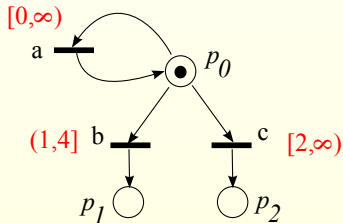
- 1 Time Petri Nets and their Enlargement
- 2 Robustness problems
- 3 Robust translation from TPN to TA
- 4 Robustly bounded TPNs
- 5 Untimed Language Robustness
- 6 Conclusion & Future Work

Time Petri Nets

Time Petri net (over Σ_ε)

$$\mathcal{N} = (\mathcal{P}, \mathcal{T}, \bullet(\cdot), (\cdot)^\bullet, m_0, \Lambda, I)$$

- P finite set of *places*,
- T finite set of *transitions* with $P \cap T = \emptyset$,
- $\bullet(\cdot) \in (\mathbb{N}^P)^T$: *backward* incidence mapping,
- $(\cdot)^\bullet \in (\mathbb{N}^P)^T$ is the *forward* incidence mapping,
- $m_0 \in \mathbb{N}^P$ is the *initial* marking,
- $\Lambda : T \rightarrow \Sigma_\varepsilon$ *labeling function*
- $I : T \mapsto \mathcal{I}(\mathbb{Q}_{\geq 0})$ (time constraint)
 $t \mapsto I(t) = [\alpha(t), \beta(t)]$ (*firing interval, can be open*).



configuration of a TPN : (m, ν)

- $m \in \mathbb{N}^P$: *marking*
- t is *enabled* in m if $m \geq \bullet t$.

$En(m)$ = set of enabled transitions in m .

- $\nu : En(m) \mapsto \mathbb{R}^+$ (valuation)

$\nu(t)$ = time elapsed since transition t was last enabled.

Admissible configurations

$ADM(\mathcal{N}) = \{(m, \nu) \mid \forall t \in En(m), \nu(t) \in I(t)^\downarrow\}$.

Configurations in which no enabled transition violates its upper constraint.

Note 1 : $I(t)^\downarrow = [0, \beta(t))$ or $[0, \beta(t)]$

Note 2 : Some configurations of $ADM(\mathcal{N})$ are not reachable

Time Petri Nets : semantics

Discrete transitions : $(m, \nu) \xrightarrow{t} (m', \nu')$

t can be **fired** from (m, ν) if

- $t \in En(m)$ (usual firing rule of PN)
- $\nu(t) \in I(t)$ (time constraints satisfied).

result of firing : $m' = m - \bullet t + t^\bullet$

t' is **newly enabled** by firing of t from m , (noted $\uparrow enabled(t', m, t)$) iff :

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

- for all t_i , $\nu'(t_i) = \begin{cases} 0 & \text{if } t_i \text{ newly enabled} \\ \nu(t_i) & \text{otherwise} \end{cases}$

Timed transitions : $(m, \nu) \xrightarrow{d} (m, \nu + d)$

d time units can elapse in (m, ν) iff $\forall t \in En(m), \nu(t) + d \in I(t) \downarrow$
time can progress when no clock leaves the firing interval of is associated (enabled) transition.

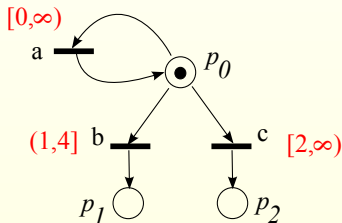
Time Petri Nets : semantics

semantics of a TPN \mathcal{N}

$[[\mathcal{N}]] = (Q, q_0, \rightarrow)$ where

- $Q = ADM(\mathcal{N}), q_0 = (m_0, \mathbf{0})$
- \rightarrow is defined by :
 - **delay moves** : $(m, \nu) \xrightarrow{d} (m, \nu + d)$
 - **discrete moves** : $(m, \nu) \xrightarrow{\Lambda(t)} (m', \nu')$ iff $(m, \nu) \xrightarrow{t} (m', \nu')$

$\mathcal{L}(\mathcal{N}) =$ untimed language of $[[\mathcal{N}]]$. (Regular in \mathcal{N} bounded)



$\bullet \xrightarrow{a} \bullet \xrightarrow{1} \bullet \xrightarrow{a} \bullet \xrightarrow{2} \bullet \xrightarrow{b} \bullet$
allowed by \mathcal{N}

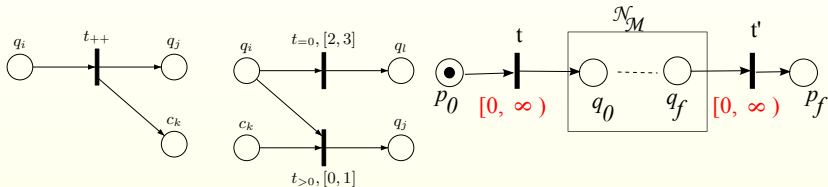
$\bullet \xrightarrow{a} \bullet \xrightarrow{1} \bullet \xrightarrow{a} \bullet \xrightarrow{5} \bullet \xrightarrow{c} \bullet$
not allowed by \mathcal{N}
(urgency)

Time Petri Nets : undecidability

Theorem 1 (PN UNDEC)

Boundedness, Reachability, coverability of a marking are undecidable for TPNs

Proof idea : Encode a counter machine \mathcal{M} with a TPN $\mathcal{N}_{\mathcal{M}}$



Reachability

$m(p) = : \begin{cases} 1 & \text{if } p = p_f \\ 0 & \text{otherwise} \end{cases}$ coverable

$\Leftrightarrow q_f$ is reachable in $\mathcal{N}_{\mathcal{M}}$

$\Leftrightarrow \mathcal{M}$ can reach q_f /halts

Boundedness

\mathcal{N} is bounded

$\Leftrightarrow \mathcal{N}_{\mathcal{M}}$ is bounded

$\Leftrightarrow \mathcal{M}$ is bounded

Time Petri nets : enlargement

Enlargement in TPNs

Let $\mathcal{N} = (\mathcal{P}, \mathcal{T}, \bullet(\cdot), (\cdot)^\bullet, m_0, \Lambda, I)$
 $i = [\alpha, \beta] \in I$ be an interval, $\Delta \in \mathbb{R}_{\geq 0}$

$$i_\Delta = [\max(0, \alpha - \Delta), \beta + \Delta]$$

$$I_\Delta = \{i_\Delta \mid i \in I\}$$

The **enlargement** of \mathcal{N} by Δ is the net

$$\mathcal{N}_\Delta = (P, T, \bullet(\cdot), (\cdot)^\bullet, m_0, \Lambda, I_\Delta)$$

lemma 2 (Monotony)

Let \mathcal{N} be a TPN and $\Delta \leq \Delta' \in \mathbb{R}_{\geq 0}$. We have $\llbracket \mathcal{N}_\Delta \rrbracket \preceq \llbracket \mathcal{N}_{\Delta'} \rrbracket$.

If \mathcal{N} verifies a safety property for some perturbation Δ_0 , it will also verify this property for any $\Delta \leq \Delta_0$.

Robustness problems for TPNs

Robust Boundedness

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

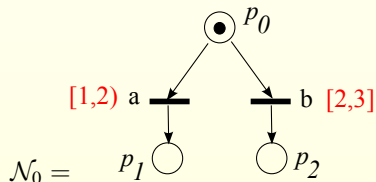
\mathcal{N} is *robustly bounded* if Δ exists

Robust safety :

Given a bounded TPN \mathcal{N} and a marking $m \in \mathbb{N}^P$, does there exist $\Delta \in \mathbb{Q}_{>0}$ s.t., $\text{Reach}(\mathcal{N}_\Delta)$ does not cover m .

Robust Untimed language preservation :

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



$$\mathcal{L}(\mathcal{N}_0) = \{a\}$$

For any $\Delta > 0$ $\mathcal{L}(\mathcal{N}_{0\Delta}) = \{a, b\}$

Such situations are easy to check. We can also decide to work with closed intervals.

Robustness problems for TPNs

Robust Boundedness

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

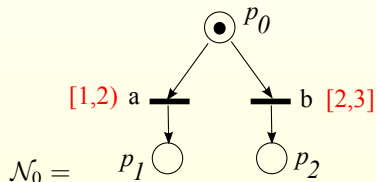
\mathcal{N} is *robustly bounded* if Δ exists

Robust safety :

Given a bounded TPN \mathcal{N} and a marking $m \in \mathbb{N}^P$, does there exist $\Delta \in \mathbb{Q}_{>0}$ s.t., $\text{Reach}(\mathcal{N}_\Delta)$ does not cover m .

Robust Untimed language preservation :

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



$$\mathcal{L}(\mathcal{N}_0) = \{a\}$$

For any $\Delta > 0$ $\mathcal{L}(\mathcal{N}_{0\Delta}) = \{a, b\}$

Such situations are easy to check. We can also decide to work with closed intervals.

Robustness problems for TPNs

Robust Boundedness

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

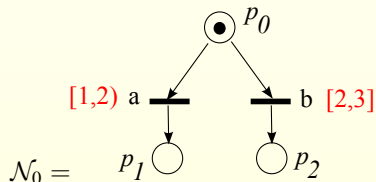
\mathcal{N} is *robustly bounded* if Δ exists

Robust safety :

Given a bounded TPN \mathcal{N} and a marking $m \in \mathbb{N}^P$, does there exist $\Delta \in \mathbb{Q}_{>0}$ s.t., $\text{Reach}(\mathcal{N}_\Delta)$ does not cover m .

Robust Untimed language preservation :

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

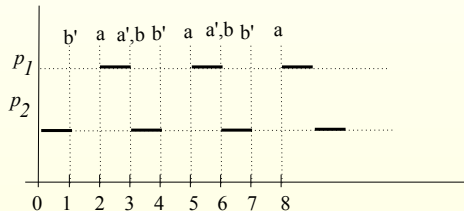
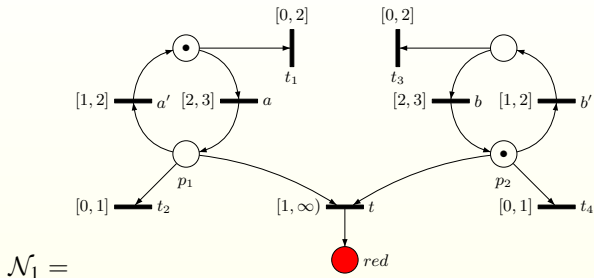


$$\mathcal{L}(\mathcal{N}_0) = \{a\}$$

For any $\Delta > 0$ $\mathcal{L}(\mathcal{N}_{0\Delta}) = \{a, b\}$

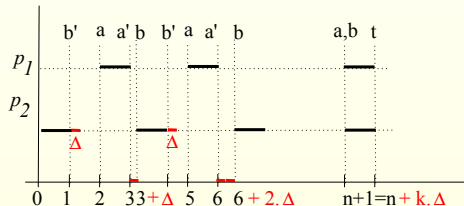
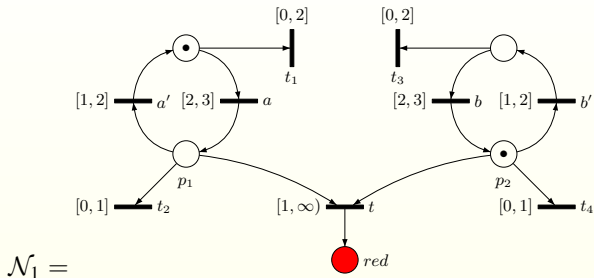
Such situations are easy to check. We can also decide to work with closed intervals.

Robustness problems for TPNs



$red \notin Reach(\mathcal{N}_1)$

Robustness problems for TPNs



$red \notin Reach(\mathcal{N}_1)$

but

For any $\Delta > 0$,

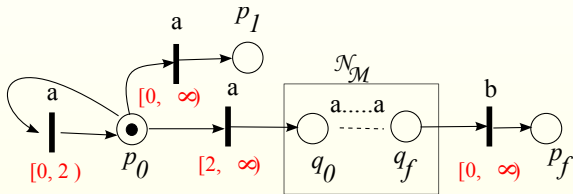
$red \in Reach(\mathcal{N}_{1\Delta})$

Undecidability of Robustness

As boundedness, reachability, etc are undecidable :

Theorem 2 (PN ROB-UNDEC)

Robust boundedness, robust untimed language preservation, and robust safety are undecidable for TPNs



\mathcal{N} is bounded :

\mathcal{N} robustly bounded

$\Leftrightarrow \exists \Delta, \mathcal{N}_\Delta$ bounded

$\Leftrightarrow \mathcal{N}_M$ bounded

$\mathcal{L}(\mathcal{N}) = a^*$:

\mathcal{N} is language robust

$\Leftrightarrow \exists \Delta, \mathcal{L}(\mathcal{N}_\Delta) = a^*.b$

$\Leftrightarrow q_f$ not reachable in \mathcal{M}

Robustness problems for TPNs

A subclass that avoids accumulation due to concurrent loops

Sequential TPNs

A TPN is **sequential** iff :

- $\forall t \in T, I(t)$ is closed
- for any $(m, \nu) \in \text{Reach}(\mathcal{N})$, t, t' fireable from (m, ν)
 t and t' are in conflict, $(\exists p, m(p) < \bullet t(p) + \bullet t'(p))$.

Properties of STPNs

- (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
 $\text{Reach}(\mathcal{N}_\Delta) = \text{Reach}(\mathcal{N})$ and $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$

Can we go beyond sequential nets ?

From TPNs to TA

General Idea :

- fix a set of markings M
- build an automaton \mathcal{A} with M as locations
- Solve Robustness problems using known results on TAs

M and \mathcal{A} must be chosen so that results for \mathcal{A} can be brought back to \mathcal{N}

Definition

Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, \bullet(\cdot), (\cdot)^\bullet, m_0, \Lambda, I)$ be a TPN, $M \subseteq \mathbb{N}^P$ be a set of markings with $m_0 \in M$.

The *M -bounded semantics* of \mathcal{N} , denoted $\llbracket \mathcal{N} \rrbracket|_M$, is the restriction of $\llbracket \mathcal{N} \rrbracket$ to states $\{(m, \nu) \in Q \mid m \in M\}$.

Proposition 2

Let M be a set of markings of a TPN \mathcal{N} containing m_0 .

If $\text{Reach}(\mathcal{N}) \subseteq M$, then $\llbracket \mathcal{N} \rrbracket|_M = \llbracket \mathcal{N} \rrbracket$.

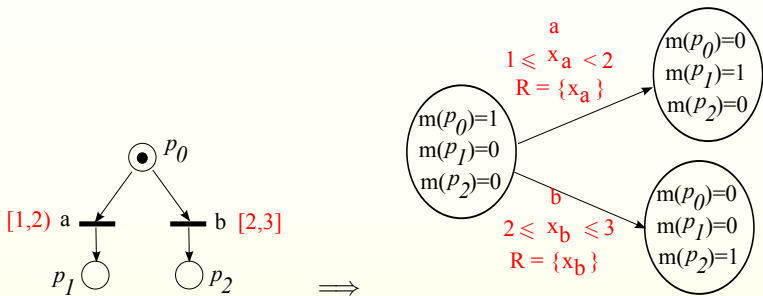
Marking timed automaton (in short)

Let \mathcal{N} be a TPN, $M \subseteq \mathbb{N}^p$ be a finite set of markings with $m_0 \in M$.

Compute $\mathcal{A}_M = (M, m_0, X, \Sigma_\varepsilon, E, Inv)$

- locations = set of markings M
- One clock per transition of T
- invariants = upper bounds of intervals (enabled transitions)
- Transition (l, γ, R, t, l') :
 - $l' = l - \bullet_t + t \bullet$
 - guard $\gamma = x_t \in I_t$
 - reset clocks R : newly enabled transitions

From TPNs to TA



Theorem 3 (TA \approx PN)

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 $\llbracket \mathcal{N}_\Delta \rrbracket|_M \approx \llbracket (\mathcal{A}_M)_\Delta \rrbracket$.

Robustly bounded TPNs

Reminder : checking if \mathcal{N} is Robustly bounded is undecidable.

Reminder 2 : checking if \mathcal{A} is Robust is decidable.

Reminder 3 For fixed finite M we can compute $\mathcal{A} \approx \mathcal{N}$ (on M)

Question : How to fix M ? \mapsto rely on **bounds**

Definition

The class **UB** of TPN whose **underlying net** (i.e. same net without time constraint) is bounded .

Definition

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

Robustly bounded TPNs : Results

Proposition 3

The class UB is a **decidable** subclass of robustly bounded TPNs.

For every $\mathcal{N} \in UB$, one can construct a finite TA \mathcal{A} such that $\llbracket \mathcal{N}_\Delta \rrbracket \approx \llbracket \mathcal{A}_\Delta \rrbracket$ for all $\Delta \geq 0$.

Theorem 4 (RR-decidable)

RR is a decidable subclass of robustly bounded TPNs.

Lemma 4

- The set of robustly bounded TPNs is **recursively enumerable**.
- 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, $\forall 0 \leq \Delta \leq \Delta_0, \llbracket \mathcal{N}_\Delta \rrbracket \approx \llbracket \mathcal{A}_\Delta \rrbracket$.

Robustness results on T.A. transfer to robustly bounded TPNs !

Robustly bounded TPNs

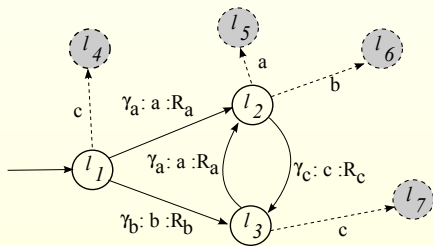
Lemma 3 ("Bounded" Robustness)

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

Proof

Let $\tilde{M} = M \cup \{m' \mid \exists m \in M, t \in T, m' = m - \bullet t + t \bullet\}$ markings reachable from M in at most one-step in the underlying net.

$$\begin{aligned} \text{Reach}(\mathcal{N}_\Delta) \subseteq M &\Leftrightarrow \text{Reach}((\mathcal{A}_{\tilde{M}})_\Delta) \subseteq M \\ &\Leftrightarrow \text{Reach}((\mathcal{A}_{\tilde{M}})_\Delta) \cap (\tilde{M} \setminus M) = \emptyset \text{ (decidable [Bouyer11])} \end{aligned}$$



Theorem 4 obtained by choosing $M = \text{Reach}(\mathcal{N})$

Untimed Language Robustness

Definition

A bounded TPN \mathcal{N} is called **Language-Robust** if $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$ for some $\Delta > 0$.

\mathcal{N} has **distinct labeling** iff $\forall t \neq t' \in T, \Lambda(t) \neq \Lambda(t')$

We denote by :

- **LR** the class of Language-Robust nets
- **LR_≠** (resp. **RR_≠**) the subclass of **LR** (resp. **RR**) with distinct labeling.

Properties

- 1 RR and LR are **incomparable** classes of TPNs w.r.t. set inclusion.
- 2 Membership in **LR** is **undecidable**
- 3 **LR_≠** is strictly contained in **RR_≠**.

theorem 5 (LR-DECIDABLE)

The class LR_{\neq} is decidable, i.e., checking if a distinctly labeled bounded TPN is in LR is decidable.

proof

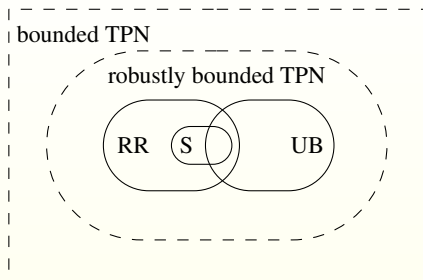
Check if \mathcal{N} is in RR (and therefore in RR_{\neq}) (decidable Thm. 4)

- $\mathcal{N} \notin RR_{\neq} : \implies$ it is not in LR_{\neq} .
- $\mathcal{N} \in RR_{\neq}$: finite set of Reachable markings, and \mathcal{N} Robustly Bounded.
 - build \mathcal{A} which is timed bisimilar to \mathcal{N} for small perturbations. (Lemma 4)
 - \mathcal{A} preserves its untimed language under small perturbations iff \mathcal{N} does.

$$\mathcal{N} \in \mathcal{LR}_{\neq} \iff \mathcal{A} \text{ is language-robust}$$

- as $\mathcal{L}(\mathcal{N})$ regular, resumes to checking an ω -regular property of \mathcal{A} (decidable [Bouyer11])

Conclusion



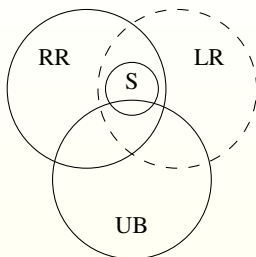
TPN classes

- RR : reach-robust,
- LR : language-robust,
- UB : bounded underlying PNs
- S : sequential bounded TPNs.

TPN classes

- — : decidable
- - - - : undecidable

Conclusion



Without distinct labels

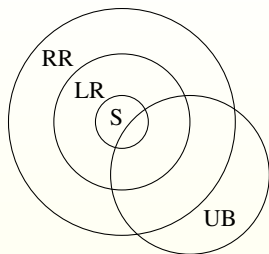
TPN classes

- RR : reach-robust,
- LR : language-robust,
- UB : bounded underlying PNs
- S : sequential bounded TPNs.

TPN classes

- — : decidable
- - - - : undecidable

Conclusion



With distinct labels

TPN classes

- RR : reach-robust,
- LR : language-robust,
- UB : bounded underlying PNs
- S : sequential bounded TPNs.

TPN classes

- — : decidable
- - - - : undecidable

Conclusions

- Robustness in concurrent models slightly differs from robustness in TA
- Robustness for PN uses TA results
- Undecidable without restriction
- robust boundedness decidable for bounded nets.
- Robust safety decidable for robustly bounded net.

Future work

- positive results in unbounded nets
- Address problems specifically due to concurrency
- exploit concurrency to solve robustness

Bibliography



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.



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.



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. Springer, 2008.



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, 2011.



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.



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.



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.

Bibliography (cont.)



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. Springer, 2005.



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.



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.



A. Puri.

Dynamical properties of timed automata.

In *DEDS*, 10(1-2) :87–113, 2000.



O. Sankur.

Untimed language preservation in timed systems.

In *Proc. of MFCS'11*, *LNCS*. Springer, 2011.



M. Swaminathan, M. Fränzle, and J.-P. Katoen.

The surprising robustness of (closed) timed automata against clock-drift.

In *TCS*, pages 537–553. Springer, 2008.

Marking timed automaton

Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, \bullet(\cdot), (\cdot)^\bullet, m_0, \Lambda, I)$ 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 , is

$\mathcal{A}_M = (M, m_0, X, \Sigma_\varepsilon, E, Inv)$, where

- $X = \{x_t \mid t \in T\}$, for each $m \in M$,
- $Inv(m) = \bigwedge_{t \in En(m)} x_t \leq \beta(t)$,
- $m \xrightarrow{g,a,R} m' \in E$ iff there exists $t \in T$ such that
 - $t \in En(m)$,
 - $m' = m - \bullet t + t^\bullet$,
 - g is the constraint $x_t \in I(t)$,
 - $a = \Lambda(t)$
 - $R = \{x_{t'} \mid t' \in \uparrow enabled(t', m, t) = \text{true}\}$

Untimed Language Robustness

Definition

A bounded TPN \mathcal{N} is called Language-Robust if $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$ for some $\Delta > 0$. \mathcal{N} has *distinct labeling* iff $\forall t \neq t' \in T, \Lambda(t) \neq \Lambda(t')$

We denote by :

- LR the class of Language-Robust nets
- LR_{\neq} (resp. RR_{\neq}) the subclass of LR (resp. RR) with distinct labeling.

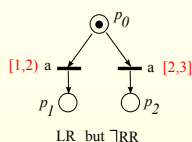
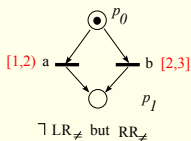
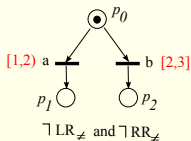
Properties

- 1 RR and LR are incomparable classes of TPNs w.r.t. set inclusion.
- 2 Membership in LR is undecidable
- 3 LR_{\neq} is strictly contained in RR_{\neq} .

Untimed Language Robustness

Proof

- 1 easy counterexamples
- 2 easy corollary of undecidability of language equivalence for TPNs.
- 3 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 $\text{Reach}(\mathcal{N}_{\Delta}) = \text{Reach}(\mathcal{N})$ for the same Δ . The strictness of inclusion follows easily : 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 $\text{Reach}(\mathcal{N})$. Hence, such \mathcal{N} is not in LR_{\neq} , but is still in RR_{\neq} .



Proof of Lemma 4

Lemma 4

- 1 The set of robustly bounded TPNs is recursively enumerable.
- 2 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, $\forall 0 \leq \Delta \leq \Delta_0, \llbracket \mathcal{N}_\Delta \rrbracket \approx \llbracket \mathcal{A}_\Delta \rrbracket$.

proof

- 1 for \mathcal{N} fixed, enumerate **all** sets of markings : For every M , check $\exists? \Delta > 0$ such that $\text{Reach}(\mathcal{N}_\Delta) \subseteq M$ (lemma 3). If the answer is Yes, stop and conclude \mathcal{N} is robustly bounded.
- 2 if \mathcal{N} is robustly bounded, the semi-algorithm stops on finite set of markings M . We can compute Δ_0 such that $\text{Reach}(\mathcal{N}_{\Delta_0}) \subseteq M$. Thus for any $\Delta \leq \Delta_0$, we have
 - $\text{Reach}(\mathcal{N}_\Delta) \subseteq M$ (Lemma 2 - monotony -).
 - $\llbracket \mathcal{N}_\Delta \rrbracket|_M = \llbracket \mathcal{N}_\Delta \rrbracket$ (proposition 2).
 - $\llbracket \mathcal{N}_\Delta \rrbracket|_M \approx \llbracket (\mathcal{A}_M)_\Delta \rrbracket$ (Theorem 3 (TA \approx PN))

Thus, $\forall 0 \leq \Delta \leq \Delta_0, \llbracket \mathcal{N}_\Delta \rrbracket \approx \llbracket (\mathcal{A}_M)_\Delta \rrbracket$

Timed automata

Timed automata (over Σ_ε)

$$\mathcal{A} = (\mathcal{L}, \ell_0, X, E, \text{Inv})$$

- L is a finite set of **locations**, $\ell_0 \in L$ is the *initial location*,
- X is a finite set of **clocks**,
- $\text{Inv} \in \mathcal{C}_{ub}(X)^L$ assigns an **invariant** to each location
- $E \subseteq L \times \mathcal{C}(X) \times \Sigma_\varepsilon \times 2^X \times L$ is a finite set of **edges**. $(\ell, \gamma, a, R, \ell')$ represents a **transition** from location ℓ to location ℓ' labeled by a with constraint γ and reset $R \subseteq X$.

semantics

$\llbracket \mathcal{A} \rrbracket = (Q, q_0, \rightarrow)$ where $Q = \{(\ell, v) \in L \times (\mathbb{R}_{\geq 0})^X \mid v \models \text{Inv}(\ell)\}$, $q_0 = (\ell_0, \mathbf{0})$ and \rightarrow is defined by :

- **delay moves** : $(\ell, v) \xrightarrow{d} (\ell, v + d)$ if $d \in \mathbb{R}_{\geq 0}$ and $v + d \models \text{Inv}(\ell)$;
- **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 $\llbracket \mathcal{A} \rrbracket$ and is denoted by $\mathcal{L}(\mathcal{A})$.

Enlargement (timed automata)

