Robustness of Time Petri Nets under Guard Enlargement

P. A. Reyniers\textsuperscript{(1)}  S. Akshay\textsuperscript{(2,3,4)}  L. Héloüët\textsuperscript{(3,4)}  C. Jard\textsuperscript{(2,3,4)}

\textsuperscript{(1)} Aix-Marseille Université, CNRS, LIF, UMR 7279, Marseille, France
\textsuperscript{(2)} ENS Cachan Bretagne, Rennes, France
\textsuperscript{(3)} IRISA, Rennes, France
\textsuperscript{(4)} INRIA Rennes, France

September 2012

1. Work funded by the ANR IMPRO
Motivations

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 $\phi$ in an implementation of $M$?
Enlargement in timed automata [Puri00]

\[ x \leq 2 \]

\[ R = \{ x \} \]

\[ x := 1 \]
\[ y := 0 \]

\[ y \geq 2 \]

\[ R = \{ y \} \]

\[ x = 0, y \geq 2 \]

\[ \text{err} \]
Enlargement in timed automata [Puri00]

$x \leq 2 + \Delta$

$R = \{ x \}$

$x := 1$

$y := 0$

$x \leq \Delta, y \geq 2 - \Delta$

$R = \{ y \}$

Diagram:

- **$l_1$**
  - Transition: $x := 1$
  - Transition: $y := 0$

- **$l_2$**
  - Transition: $x \leq \Delta, y \geq 2 - \Delta$

- **$err$**
$\mathcal{A} = (\mathcal{L}, \ell, X, E, \text{Inv})$

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

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

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

$\mathcal{A}_\Delta$ : $\Delta$-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 $\mathcal{L}(\mathcal{A}_\Delta) \preceq \mathcal{L}(\mathcal{A}_{\Delta'})$.

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 $\text{Reach}(\mathcal{A}_\Delta) \cap S = \emptyset$.

Theorem [BouyerMS11]

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

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

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

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

\[\text{En}(m) = \text{set of enabled transitions in } m.\]

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

\[\nu(t) = \text{time elapsed since transition } t \text{ was last enabled.}\]

**Admissible configurations**

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

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 \(\text{ADM}(\mathcal{N})\) are not reachable
Time Petri Nets : semantics

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

t can be fired from \((m, \nu)\) if

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

result of firing : \(m' = m - \bullet t + t\cdot\)

\(t'\) is newly enabled by firing of \(t\) from \(m\), (noted \(\uparrow\text{enabled}(t', m, t)\)) iff :

\[
t' \in En(m - \bullet t + t\cdot) \land ((t' \not\in En(m - \bullet t)) \lor t = t')
\]

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 \((m, \nu)\) iff \(\forall t \in En(m), \nu(t) + d \in I(t)\)

time can progress when no clock leaves the firing interval of is associated (enabled) transition.
Time Petri Nets: semantics

**semantics of a TPN** $\mathcal{N}$

$\llbracket \mathcal{N} \rrbracket = (Q, q_0, \rightarrow)$ where

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

$L(\mathcal{N}) = \text{untimed language of } \llbracket \mathcal{N} \rrbracket$. *(Regular in $\mathcal{N}$ bounded)*

---

**Diagram**

- $\bullet \xrightarrow{a} \bullet \xrightarrow{1} \bullet \xrightarrow{a} \bullet \xrightarrow{2} \bullet \xrightarrow{b} \bullet$ 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)*
Theorem 1 (PN UNDEC)

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

Proof idea: Encode a counter machine $M$ with a TPN $N_M$

Reachability

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

$\iff q_f$ is reachable in $N_M$

$\iff M$ can reach $q_f$/halts

Boundedness

$N$ is bounded

$\iff N_M$ is bounded

$\iff M$ is bounded
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, \( \Delta \in \mathbb{R}_{\geq 0} \)

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

\[
I_\Delta = \{i_\Delta | i \in I\}
\]

The enlargement of \( \mathcal{N} \) by \( \Delta \) 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 \([\mathcal{N}_\Delta] \preceq [\mathcal{N}_{\Delta'}]\).

If \( \mathcal{N} \) verifies a safety property for some perturbation \( \Delta_0 \), 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 $\mathcal{N}_\Delta$ is bounded?

$\mathcal{N}$ is *robustly bounded* if $\Delta$ 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 $\mathcal{L}(\mathcal{N}_\Delta) = \mathcal{L}(\mathcal{N})$?

$\mathcal{N}_0 = \begin{array}{c}
\bullet \\
\downarrow a \\
\bullet \\
\downarrow b \\
\circ \end{array}$

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

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 $\mathcal{N}_\Delta$ is bounded?
$\mathcal{N}$ is robustly bounded if $\Delta$ exists.

Robust safety:
Given a bounded TPN $\mathcal{N}$ and a marking $m \in \mathbb{NP}$, does there exists $\Delta \in \mathbb{Q}_{>0}$ 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 $\mathcal{N}_\Delta$ is bounded?
$\mathcal{N}$ is *robustly bounded* if $\Delta$ exists

Robust safety:
Given a bounded TPN $\mathcal{N}$ and a marking $m \in \mathbb{N}^P$, does there exists $\Delta \in \mathbb{Q}_{>0}$ 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

\[ N_1 = \]

\begin{align*}
\mathcal{N}_1 & = \\
\end{align*}

\begin{align*}
\text{red} & \not\in \text{Reach}(N_1)
\end{align*}
Robustness problems for TPNs

\[ N_1 = \]

\[ a', [2, 3] \quad a \quad [0, 2] \quad t_1 \]

\[ p_1 \]

\[ 1, 2 \]

\[ b', [2, 3] \quad b' \quad [0, 2] \quad t_3 \]

\[ p_2 \]

\[ 0, 1 \]

\[ t_2 \]

\[ [0, 1, \infty ) \quad t \]

\[ red \]

\[ a', a, b, b', a', b \]

\[ red \notin \text{Reach}(N_1) \]

but

For any \( \Delta > 0 \),

\[ red \in \text{Reach}(N_{1, \Delta}) \]
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.

\[ L(N) = a^* : \]
\[ N \text{ is language robust} \]
\[ \iff \exists \Delta, L(N_\Delta) = a^* . b \]
\[ \iff q_f \text{ not reachable in } M \]

\[ N \text{ is bounded :} \]
\[ N \text{ robustly bounded} \]
\[ \iff \exists \Delta, N_\Delta \text{ bounded} \]
\[ \iff N_M \text{ bounded} \]
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 $(m, \nu)$ $t$ 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 $A$ with $M$ as locations
- Solve Robustness problems using known results on TAs

$M$ and $A$ must be chosen so that results for $A$ can be brought back to $N$

Definition

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

The $M$-bounded semantics of $N$, denoted $[N]_M$, is the restriction of $[N]$ to states $\{(m, \nu) \in Q \mid m \in M\}$.

Proposition 2

Let $M$ be a set of markings of a TPN $N$ containing $m_0$.

If $\text{Reach}(N) \subseteq M$, then $[N]_M = [N]$. 
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** $(l, \gamma, R, t, l')$ :

- $l' = l - \cdot t + t^*$
- guard $\gamma = x_t \in I_t$
- reset clocks $R :$ newly enabled transitions
Theorem 3 (TA≈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 \( [\mathcal{N}_\Delta]_M \approx [(\mathcal{A}_M)_\Delta] \).
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.
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 $A$ such that $[\mathcal{N}_\Delta] \approx [A_\Delta]$ 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 $A$ such that there exists $\Delta_0 > 0$ for which, $\forall 0 \leq \Delta \leq \Delta_0$, $[\mathcal{N}_\Delta] \approx [A_\Delta]$.

Robustness results on T.A. transfer to 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 ($\Delta$ can be effectively computed).

Proof

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

$\text{Reach}(\mathcal{N}_\Delta) \subseteq M \iff \text{Reach}(\mathcal{A}_{\tilde{M}}) \subseteq M$

$\iff \text{Reach}(\mathcal{A}_{\tilde{M}}) \cap (\tilde{M} \setminus M) = \emptyset$ (decidable [Bouyer11])

Theorem 4 obtained by choosing $M = \text{Reach}(\mathcal{N})$
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**.
The class \( LR \neq \) is decidable, i.e., checking if a distinctly labeled bounded TPN is in LR is decidable.

\textbf{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 \( A \) which is timed bisimilar to \( \mathcal{N} \) for small perturbations. (Lemma 4)
  - \( A \) preserves its untimed language under small perturbations iff \( \mathcal{N} \) does.

\[ \mathcal{N} \in LR \neq \iff A \text{ is language-robust} \]

- as \( \mathcal{L}(\mathcal{N}) \) regular, resumes to checking an \( \omega \)-regular property of \( A \) (decidable [Bouyer11])
**TPN classes**
- RR: reach-robust,
- LR: language-robust,
- UB: bounded underlying PNs
- S: sequential bounded TPNs.

**TPN classes**
- ---: decidable
- - - -: undecidable
TPN classes
- RR : reach-robust,
- LR : language-robust,
- UB : bounded underlying PNs
- S : sequential bounded TPNs.

TPN classes
- — : decidable
- - - : undecidable
**Conclusion**

TPN classes

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

- —: 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
B. Berthomieu and M. Diaz.
Modeling and verification of time dependent systems using time Petri nets.

Robust model-checking of linear-time properties in timed automata.

Robust analysis of timed automata via channel machines.

Robust model-checking of timed automata via pumping in channel machines.

F. Cassez and O. H. Roux.
Structural translation from time petri nets to timed automata.

D. D’Aprile, S. Donatelli, A. Sangnier, and J. Sproston.
From time Petri nets to timed automata: An untimed approach.

Robust safety of timed automata.
M. De Wulf, L. Doyen, and J.-F. Raskin.
Systematic implementation of real-time models.

A zone-based method for computing the state space of a time Petri net.

D. Lime and O. H. Roux.
Model checking of time petri nets using the state class timed automaton.

A. Puri.
Dynamical properties of timed automata.

O. Sankur.
Untimed language preservation in timed systems.

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

Let $\mathcal{N} = (P, T, \Sigma_\varepsilon, \cdot (.), \cdot, m_0, \Lambda, I)$ 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 $A_M = (M, m_0, X, \Sigma_\varepsilon, E, \text{Inv})$, where

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

A bounded TPN $\mathcal{N}$ is called Language-Robust if $L(\mathcal{N}_\Delta) = L(\mathcal{N})$ 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 $\Delta$. 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 $RR \neq$. 

\[
\begin{array}{c}
\begin{array}{c}
\text{[1,2)} \ a
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{[1,2)} \ a
\end{array}
\end{array}
\]

- LR \neq and RR \neq
- LR \neq but RR \neq
- LR but 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, $\forall 0 \leq \Delta \leq \Delta_0$, $[\mathcal{N}_\Delta] \approx [\mathcal{A}_\Delta]$.

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 $\Delta_0$ 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 -).
   - $[\mathcal{N}_\Delta]_{|M} = [\mathcal{N}_\Delta]$ (Proposition 2).
   - $[\mathcal{N}_\Delta]_{|M} \approx [((\mathcal{A}_M)_\Delta)]$ (Theorem 3 (TA $\approx$ PN)).

Thus, $\forall 0 \leq \Delta \leq \Delta_0$, $[\mathcal{N}_\Delta] \approx [((\mathcal{A}_M)_\Delta)]$.
Timed automata

Timed automata (over $\Sigma_\varepsilon$)

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

semantics

$[A] = (Q, q_0, \rightarrow)$ where $Q = \{(\ell, v) \in L \times (\mathbb{R}_{\geq 0})^X \mid v \models Inv(\ell)\}$,
$q_0 = (\ell_0, 0)$ and $\rightarrow$ is defined by:
- delay moves: $(\ell, v) \xrightarrow{d} (\ell, v + d)$ if $d \in \mathbb{R}_{\geq 0}$ and $v + d \models Inv(\ell)$;
- 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 $A$ is defined as that of $[A]$ and is denoted by $L(A)$. 
Enlargement (timed automata)

\[
x \leq 2 + \Delta
\]

\[
R = \{ x \}
\]

\[
x := 1
\]
\[
y := 0
\]

\[
x \leq \Delta, \ y \geq 2 - \Delta
\]

\[
R = \{ y \}
\]

\[
y \geq 2 - \Delta
\]