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

Pierre-Alain Reynier
Joint work with Patricia Bouyer and Serge Haddad

Réunion ACI Cortos, Persée & Versydis

March 14, 2006

Motivations

Later objective: Efficiently verify parallel composition of TA.

Motivations

Later objective: Efficiently verify parallel composition of TA.

Fact: Parallelism is not well treated in classical tools (Uppaal, Kronos) for TA: the equivalent TA is entirely computed on-the-fly.

Motivations

Later objective: Efficiently verify parallel composition of TA.

Fact: Parallelism is not well treated in classical tools (Uppaal, Kronos) for TA: the equivalent TA is entirely computed on-the-fly.

Idea: Extend a classical method based on unfoldings.

➜ our choice: Mc Millan's algorithm for Petri Nets
(computes a complete finite prefix of the unfolding of a PN.)

Motivations

Later objective: Efficiently verify parallel composition of TA.

Fact: Parallelism is not well treated in classical tools (Uppaal, Kronos) for TA: the equivalent TA is entirely computed on-the-fly.

Idea: Extend a classical method based on unfoldings.

➜ our choice: Mc Millan's algorithm for Petri Nets
(computes a complete finite prefix of the unfolding of a PN.)

Two steps:

1. Find a timed extension of Petri Nets which:
   - expresses efficiently parallel composition of TA.
   - is suitable to extend Mc Millan's algorithm.
2. Extend Mc Millan's algorithm to this model.

Motivations

Later objective: Efficiently verify parallel composition of TA.

Fact: Parallelism is not well treated in classical tools (Uppaal, Kronos) for TA: the equivalent TA is entirely computed on-the-fly.

Idea: Extend a classical method based on unfoldings.

➜ our choice: Mc Millan's algorithm for Petri Nets
(computes a complete finite prefix of the unfolding of a PN.)

Two steps:

1. Find a timed extension of Petri Nets which:
   - expresses efficiently parallel composition of TA.
   - is suitable to extend Mc Millan's algorithm.
2. Extend Mc Millan's algorithm to this model.

Why not Time Petri Nets ?

We could choose the model of Time Petri Nets:
Recent works propose translations from Timed Automata to Time Petri Nets. [Berard et al'05], [BHR'06]

But:

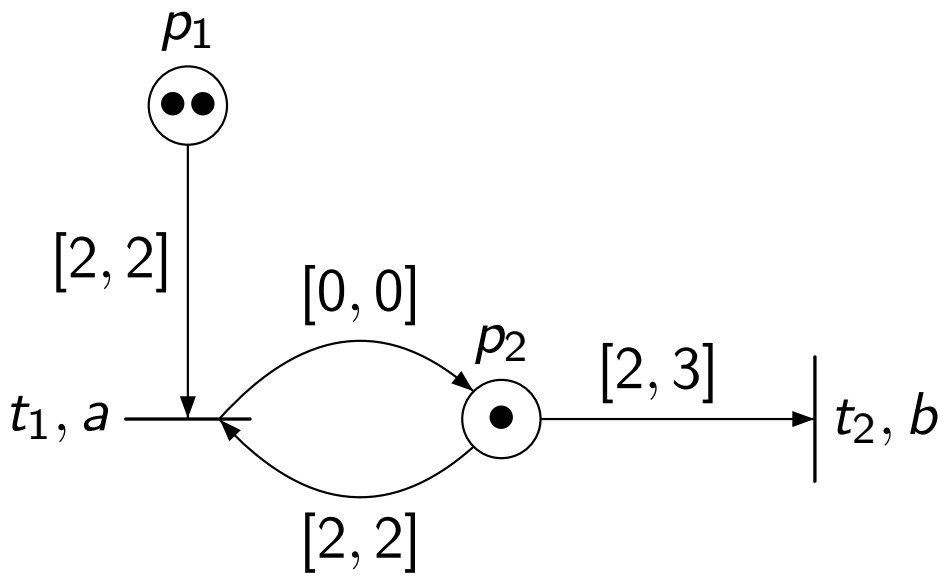
- the "urgent" semantics of TPN is not suitable for applying partial order methods.Indeed we can not let time elapse locally: the upper bound on time elpasing depends on the global configuration [Aura&Lilius-TCS'00],
- we only have decidability results for bounded nets.

➜ We will use the model of Timed Petri Nets (TdPN).

Timed Petri Nets - Syntax

A TdPN is a Petri Net extended in the following manner:

- each token has an age,
- each arc has an interval to add timing constraints on the ages of the tokens which are consumed or produced,
- the transitions carry the labels.



Timed Petri Nets - Semantics

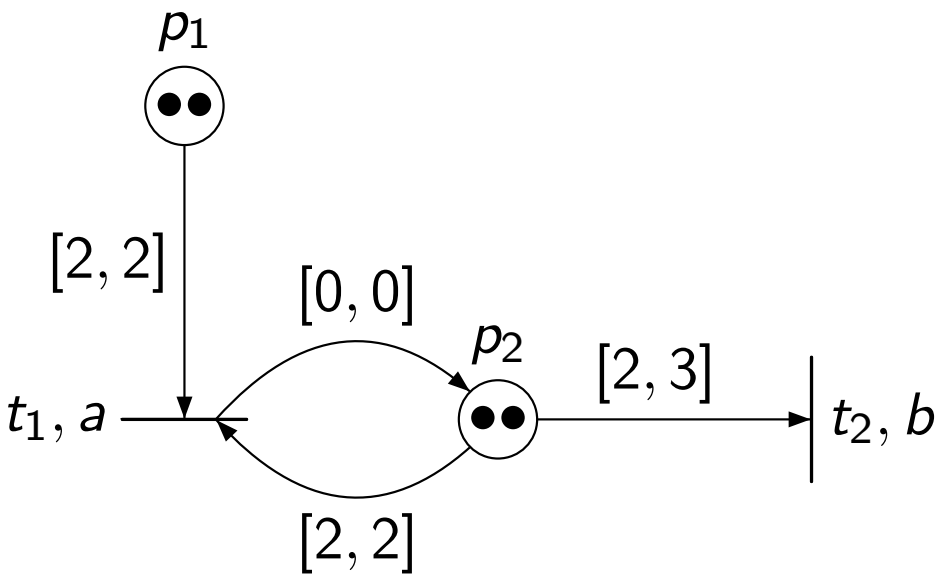
- A transition is enabled if it can consume some tokens whose ages belong to the corresponding interval.
- When firing a transition, it produces tokens whose ages are chosen nondeterministically in the corresponding interval.
- Time can elapse without upper bound and thus produce dead tokens (tokens that can never be used anymore).

Acceptance conditions: conjunctions of formulas $\sum_{i=1}^{n} p_i \bowtie k$ where p_i denotes a place, $k \in \mathbb{N}$ and $\bowtie \in \{\leq, \geq\}$.

- Finite words: condition satisfied by the last marking.
- Infinite words: condition satisfied infinitely often.

We need generalized Büchi conditions (finite set of conditions).

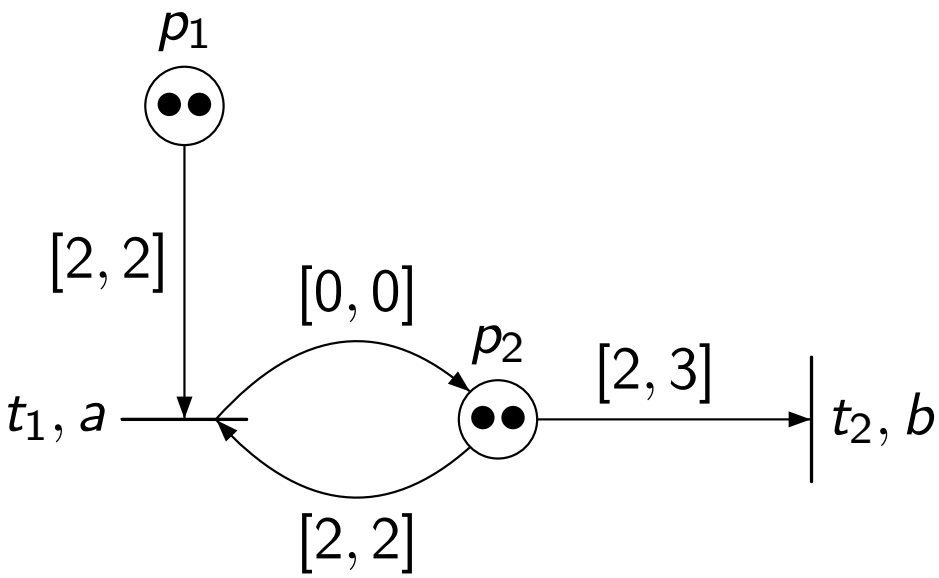
An example



timed word:

p_1	$0, 0$
p_2	$0, 0$

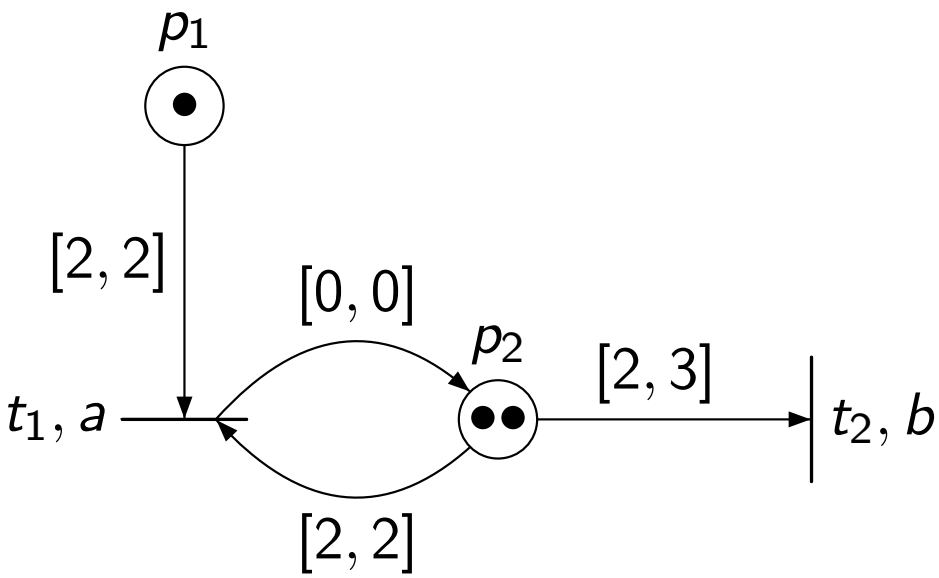
An example



timed word:

$$\begin{array}{c|c} p_1 & 0,0 \\ p_2 & 0,0 \end{array} \xrightarrow{(2)} \begin{array}{c} 2,2 \\ 2,2 \end{array}$$

An example

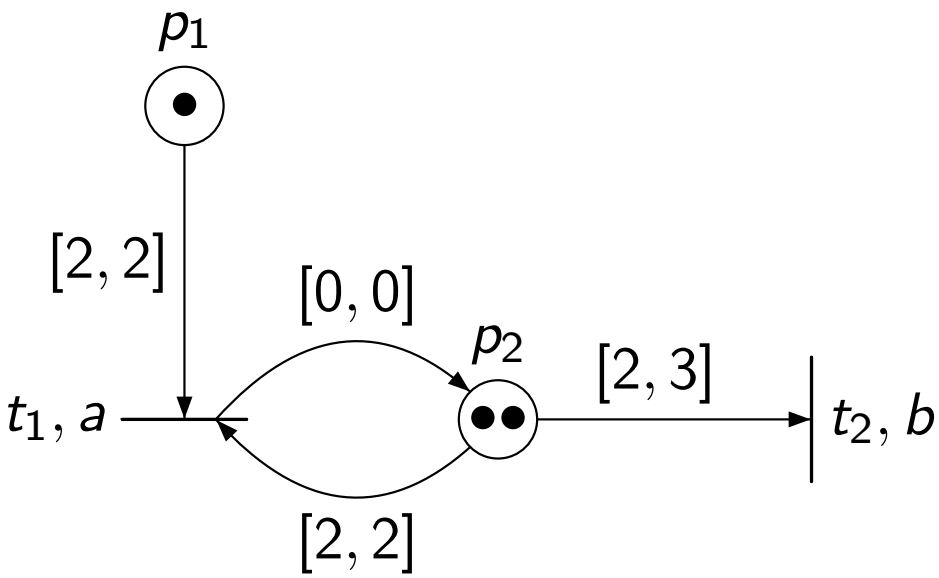


timed word:

$(a, 2)$

$$\begin{array}{c|c} p_1 & 0,0 \\ p_2 & 0,0 \end{array} \xrightarrow{(2)} \begin{array}{c} 2,2 \\ 2,2 \end{array} \xrightarrow{a} \begin{array}{c} 2 \\ 0,2 \end{array}$$

An example

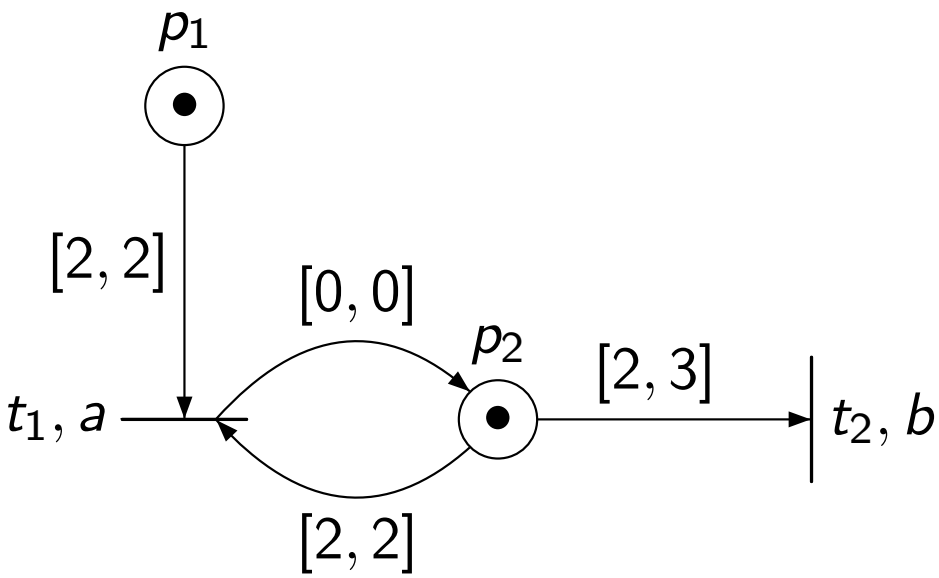


timed word:

$(a, 2)$

p_1	$0, 0$	$\xrightarrow{(2)}$	$2, 2$	$\xrightarrow{a}$	2	$\xrightarrow{(1)}$	3
p_2	$0, 0$		$2, 2$		$0, 2$		$1, 3$

An example

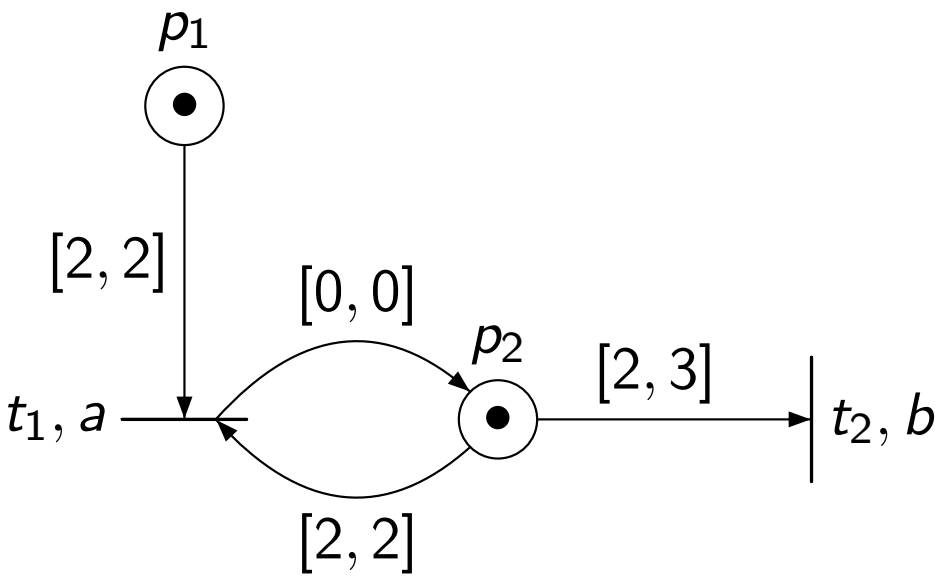


timed word:

$(a, 2)(b, 3)$

p_1	$0, 0$	$\xrightarrow{(2)}$	$2, 2$	$\xrightarrow{a}$	2	$\xrightarrow{(1)}$	3	$\xrightarrow{b}$	3
p_2	$0, 0$		$2, 2$		$0, 2$		$1, 3$		1

An example

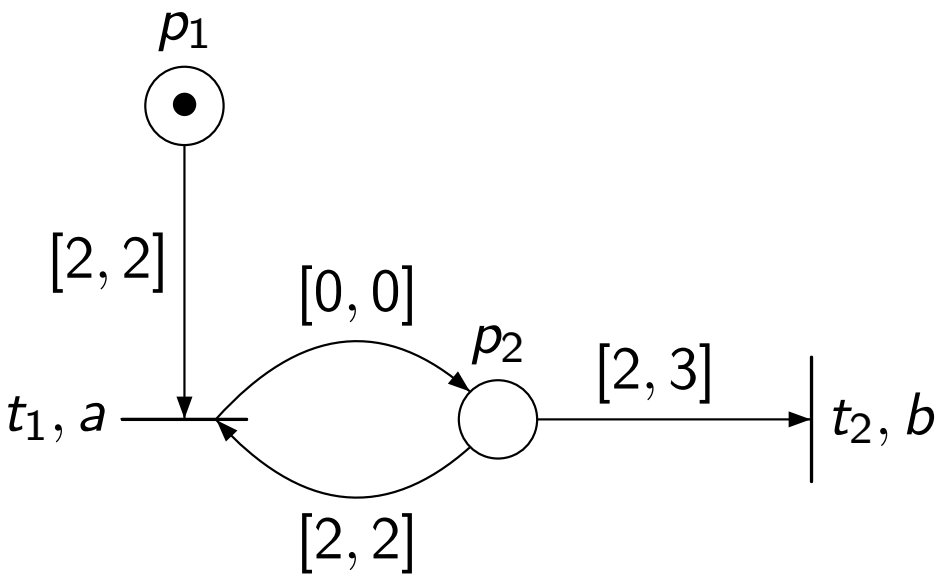


timed word:

$(a, 2)(b, 3)$

p_1	0, 0	$\xrightarrow{(2)}$	2, 2	$\xrightarrow{a}$	2	$\xrightarrow{(1)}$	3	$\xrightarrow{b}$	3	$\xrightarrow{(1)}$	4
p_2	0, 0		2, 2		0, 2		1, 3		1		2

An example



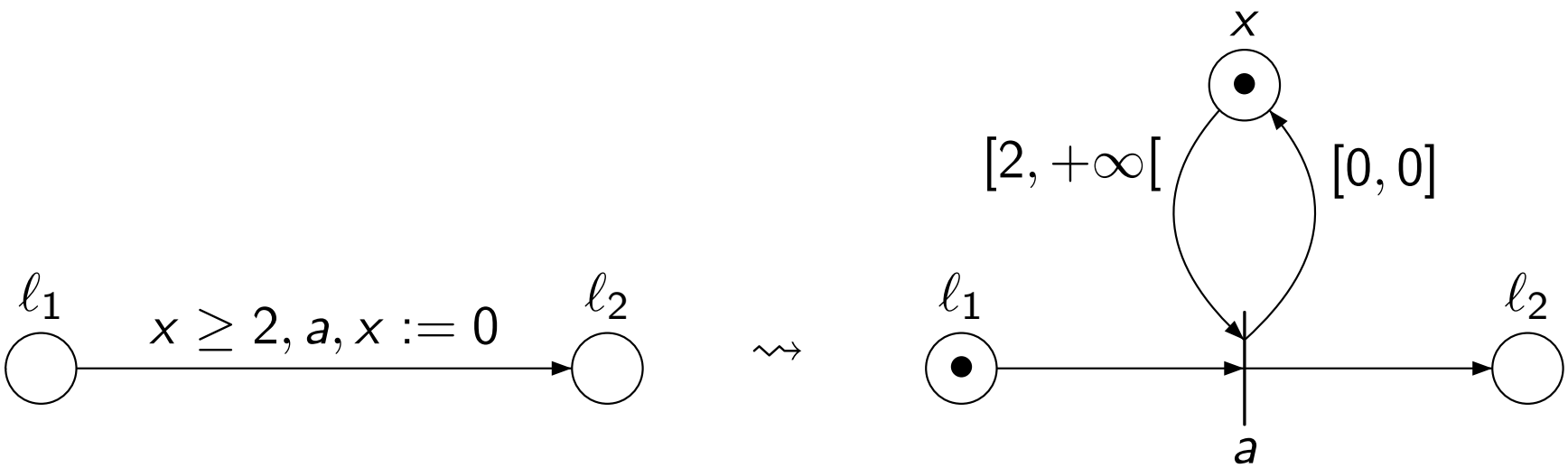
timed word:

$(a, 2)(b, 3)(b, 4)$

p_1	0, 0	$\xrightarrow{(2)}$	2, 2	$\xrightarrow{a}$	2	$\xrightarrow{(1)}$	3	$\xrightarrow{b}$	3	$\xrightarrow{(1)}$	4	$\xrightarrow{b}$	4
p_2	0, 0		2, 2		0, 2		1, 3		1		2		

Encoding a TA into a TdPN: the favorable case

We assume clocks are reset each time they are tested.



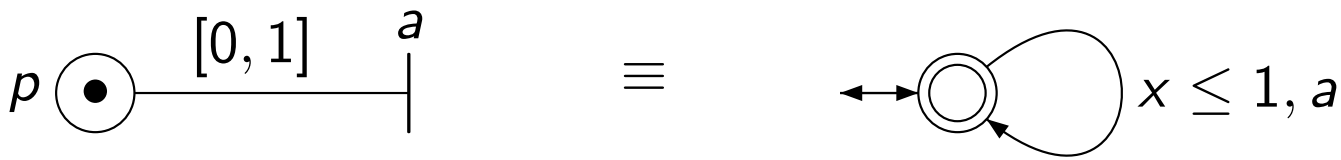
➜ What happens if we do not want to reset clock x?

We add a new kind of arcs: Read Arcs.

Introduction of Read Arcs

Read Arcs = arcs that check the presence and timing of tokens, but without consuming them. (already defined in the untimed framework)

$$Acc = \texttt{true}$$



The accepted languages are

$$\begin{cases} L_1^* = \{(a,\tau_1)\dots(a,\tau_n) \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq 1\} \\ L_1^\omega = \{(a,\tau_1)\dots(a,\tau_n)\dots \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots \leq 1\} \end{cases}$$

$\Rightarrow$ This defines the model of RA-TdPN.

TA $\equiv$ bounded RA-TdPN

We finally get two straightforward translations:

Theorem

Given a TA $\mathcal{A}$, we can build effectively a safe RA-TdPN, equivalent w.r.t. accepted languages of finite and infinite words, whose size is linear in the size of $\mathcal{A}$.

and back...

Theorem

Given a safe RA-TdPN $\mathcal{N}$, we can build effectively a TA, equivalent w.r.t. accepted languages of finite and infinite words, whose size is exponential in the size of $\mathcal{N}$.

Decidability of the coverability

Definition (The coverability Problem)

Given a marking M, is it possible to reach, from the initial marking, a marking M' such that $M \prec M'$?

Theorem

The Coverability Problem is decidable for RA-TdPNs.

Proof: extension of the proof of Abdulla et al in the case of TdPNs. This proof relies on a backward computation of regions. We can show that the discrete Predecessor operator can be extended to deal with Read Arcs.

Study of the expressiveness

The results we obtain depend heavily on the timed language equivalence we consider. We introduce the following ones:

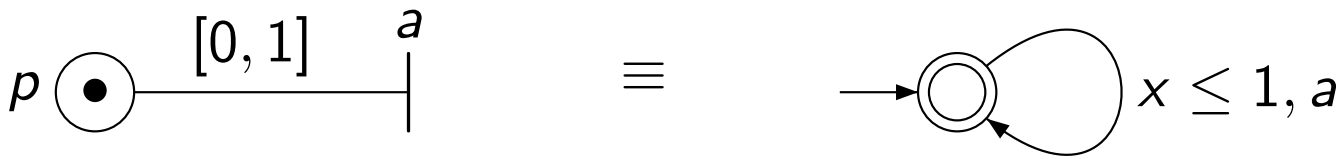
- $*$-equivalence: finite words
- ω-equivalence: infinite words
- ω_{nz}-equivalence: non-Zeno infinite words

We consider three features of (bounded) RA-TdPNs:

- Read Arcs,
- general resets vs 0-resets,
- integral nets.

Infinite timed words: a first discriminating language

$Acc = \texttt{true}$



$$L_1 = \{(a, \tau_1) \dots (a, \tau_n) \dots \mid 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots \leq 1\}$$

Lemma

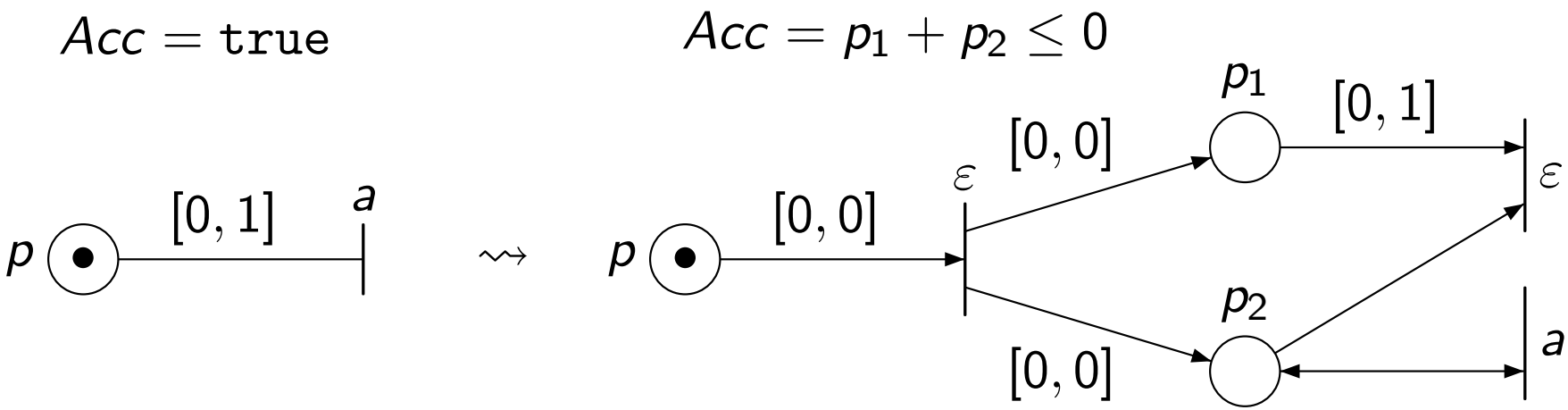
L_1 is recognized by no TdPN (even unbounded).

Corollary

Read Arcs add expressive power w.r.t. ω-equivalence.

Remark 1: L_1 is a Zeno language.
Remark 2: L_1 is accepted by the TA on the right.

Finite timed words



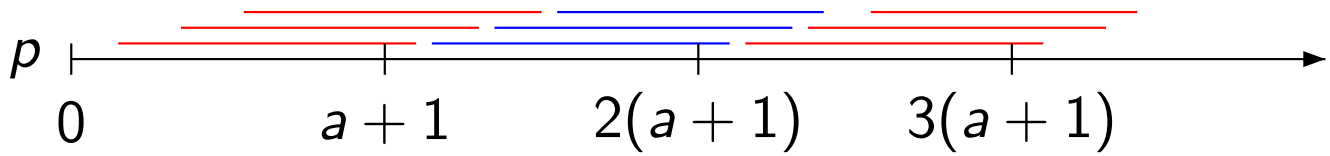
We can generalize this construction and get:

Theorem

Given a (integral, bounded) RA-TdPN, we can build a (integral, bounded) TdPN equivalent w.r.t. $*$-equivalence.

Non-Zeno infinite timed words

The previous acceptance condition becomes:
"infinitely often, the place p is empty." ➜ not always true!



➜ Two copies of p:

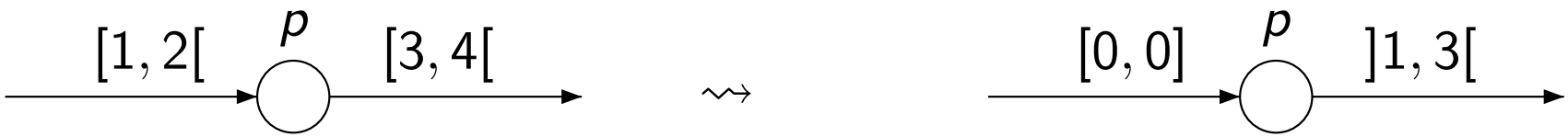
- p_{even} for red tokens,
- and p_{odd} for blue tokens.

➜ with a generalized Büchi condition on p_{even} and p_{odd} , we get:

Theorem

Given a (integral, bounded) RA-TdPN, we can build a (integral, bounded) TdPN equivalent w.r.t. ω_{nz}-equivalence.

Without Read Arcs: case of TdPNs

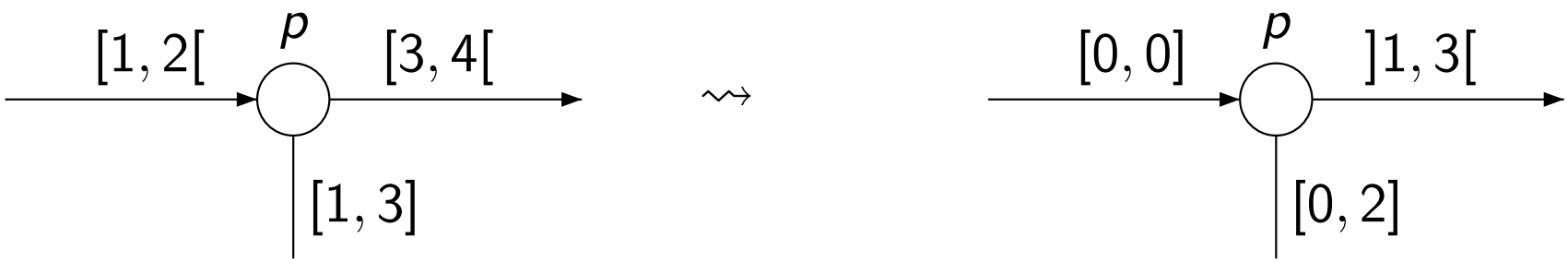


By shifting intervals, we get for TdPNs:

Theorem

Given an (integral, bounded) TdPN with general resets, we can build an (integral, bounded) TdPN with only 0*-resets, equivalent w.r.t.* $*, \omega, \omega_{nz}$*-equivalences.*

This construction is not correct with Read Arcs:



Finite and non-Zeno infinite timed words

By previous results on Read Arcs, we get:

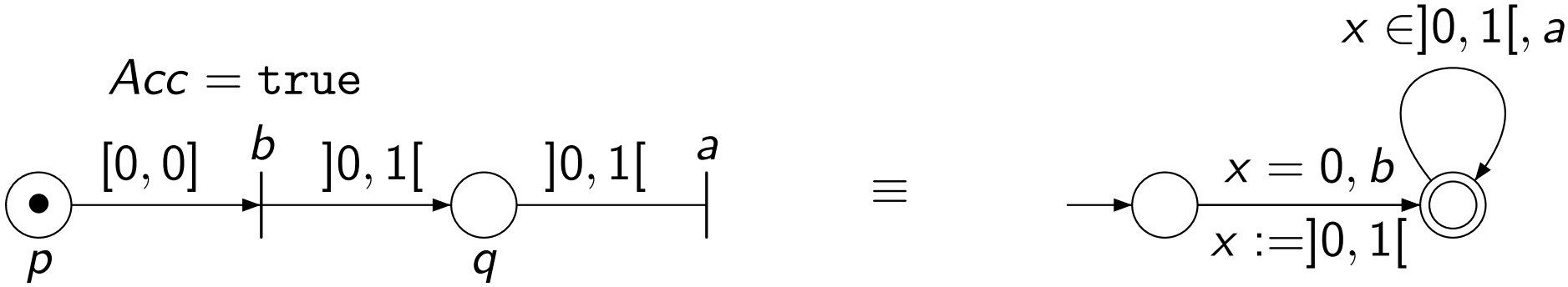
Corollary

Given an (integral, bounded) RA-TdPN with general resets, we can build an (integral, bounded) RA-TdPN with only 0-resets, equivalent w.r.t. $*, \omega_{nz}$-equivalences.

➜ the only remaining case is:

- general resets,
- + Read Arcs,
- + zeno behaviors.

Infinite timed words: a second discriminating language



$$L_2 = \{(b,0)(a,\tau_1)\dots(a,\tau_n)\dots \mid \exists \tau < 1 \text{ s.t. } 0 \leq \tau_1 \leq \dots \leq \tau_n \leq \dots < \tau\}$$

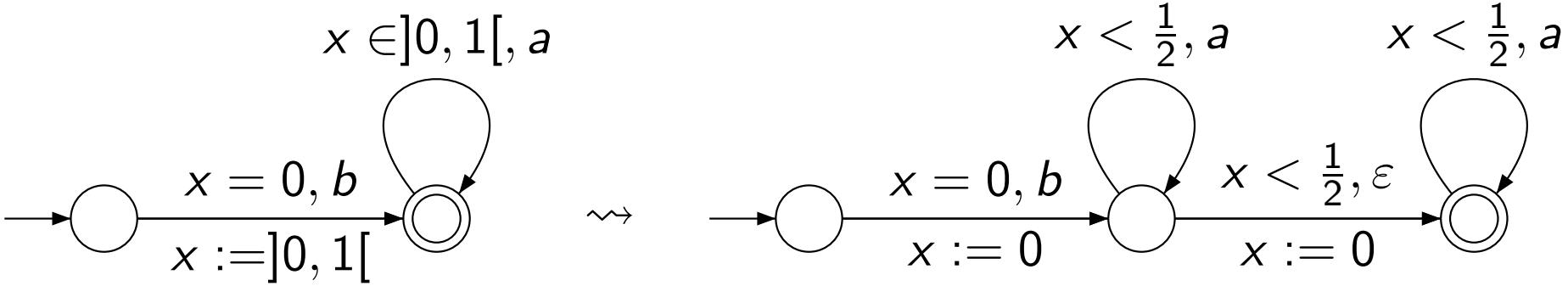
Lemma

L_2 is recognized by no 0-reset integral RA-TdPN.

Remark 1: L_2 is a Zeno language.

Remark 2: L_2 is accepted by the TA on the right.

Infinite timed words: losing the integrality



We can generalize this construction and get:

Theorem

Given a (bounded) RA-TdPN with general resets, we can build a (bounded) RA-TdPN with only 0-resets, equivalent w.r.t. ω-equivalence. This construction does not preserve the integrality of the net.

Summary (1)

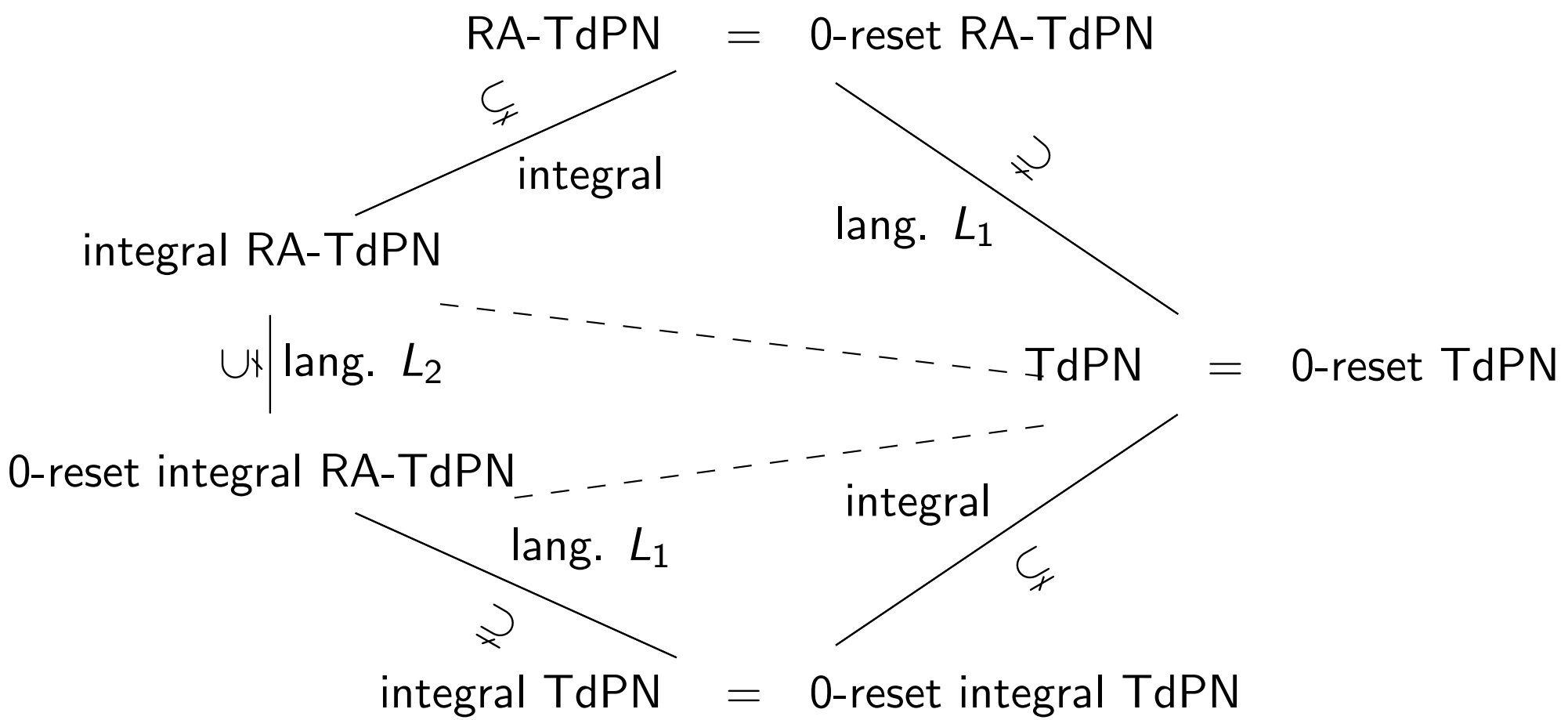
Picture for finite and infinite non-zeno timed words:

$$\text{RA-TdPN} \quad = \quad \text{TdPN} \quad = \quad \text{0-reset TdPN}$$

These equalities also hold for bounded nets and for integral nets.

Summary (2)

Picture for infinite timed words:



Comparison of TA and (bounded) TdPNs

From previous results, we get:

- TA $\equiv_{*,\omega_{nz}}$ bounded TdPN
- TA $>_{\omega}$ bounded TdPN
- TA $<>_{\omega}$ TdPN

➜ This contradicts the intuition that a token is as powerful (for expressiveness) as a clock.

Role of general resets in TA

The role of non-deterministic updates in TA has been studied in [Bouyer et al'04]. We complete these results:

- TA $\equiv_{*,\omega_{nz},\omega}$ $0-$ reset TA
- integral TA $\equiv_{*,\omega_{nz}}$ $0-$ reset integral TA
- integral TA $>_{\omega}$ $0-$ reset integral TA

➜ we have to refine the granularity of the net!

Conclusion

- We have proposed a model which unifies TA and TdPNs,
- We have completely studied the relative expressiveness of TA and TdPNs,
- We have completed the picture on general resets in TA.

As further work, we plan to study the following problems:

- extend other decidability results from TdPNs to RA-TdPNs,
- extend Mc Millan's algorithm from Petri Nets to RA-TdPNs.