Romeo: A Tool for Time Petri Nets Analysis

CAV 2005 – Edinburgh

Guillaume Gardey[1] Didier Lime[2] Morgan Magnin[1] Olivier (H.) Roux[1]

[1] IRCCyN, CNRS UMR 6597, Nantes, France
{Guillaume.Gardey|Morgan.Magnin|Olivier-h.Roux}@irccyn.ec-nantes.fr

[2] Aalborg University - CISS, Denmark
didier@cs.aau.dk

Overview

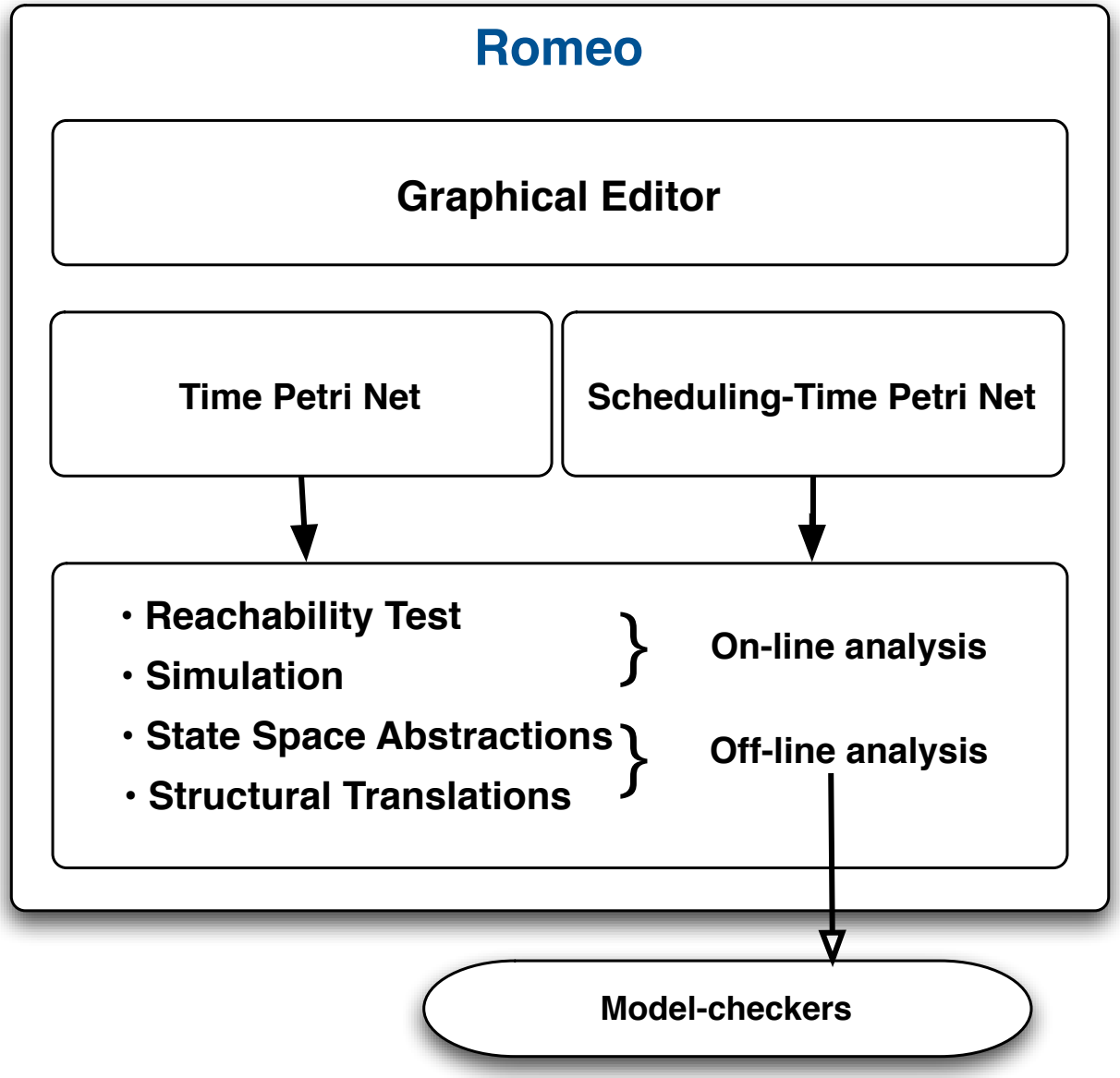
Objectives

- Specification of (preemptive) real-time systems
- Analysis, model-checking

Models

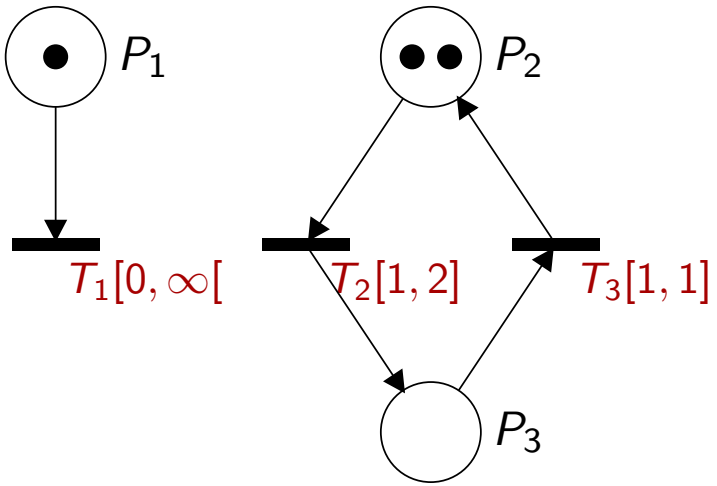
- Time Petri Nets
- Scheduling Time Petri Nets

Overview


Romeo
Graphical Editor
Time Petri Net
Scheduling-Time Petri Net
• Reachability Test
• Simulation
• State Space Abstractions
• Structural Translations
On-line analysis
Off-line analysis
Model-checkers

Time Petri Nets

Time extension of Petri nets.



- No multi-enableness

$$\begin{pmatrix}1\\2\\0\end{pmatrix} \xrightarrow{2\ ut} \xrightarrow{T_2} \begin{pmatrix}1\\1\\1\end{pmatrix} \xrightarrow{1.3\ ut} \xrightarrow{T_2} \begin{pmatrix}1\\0\\2\end{pmatrix} \dots$$

- Strong semantics

State Space Abstractions

ROMEO implements two types of state space computation:

State Space Abstractions

ROMEO implements two types of state space computation:

- Classical state class graph
  - BERTHOMIEU and DIAZ 1991
  - Classical method to compute the state space
  - Untimed language.
  - Preserves *LTL* properties.

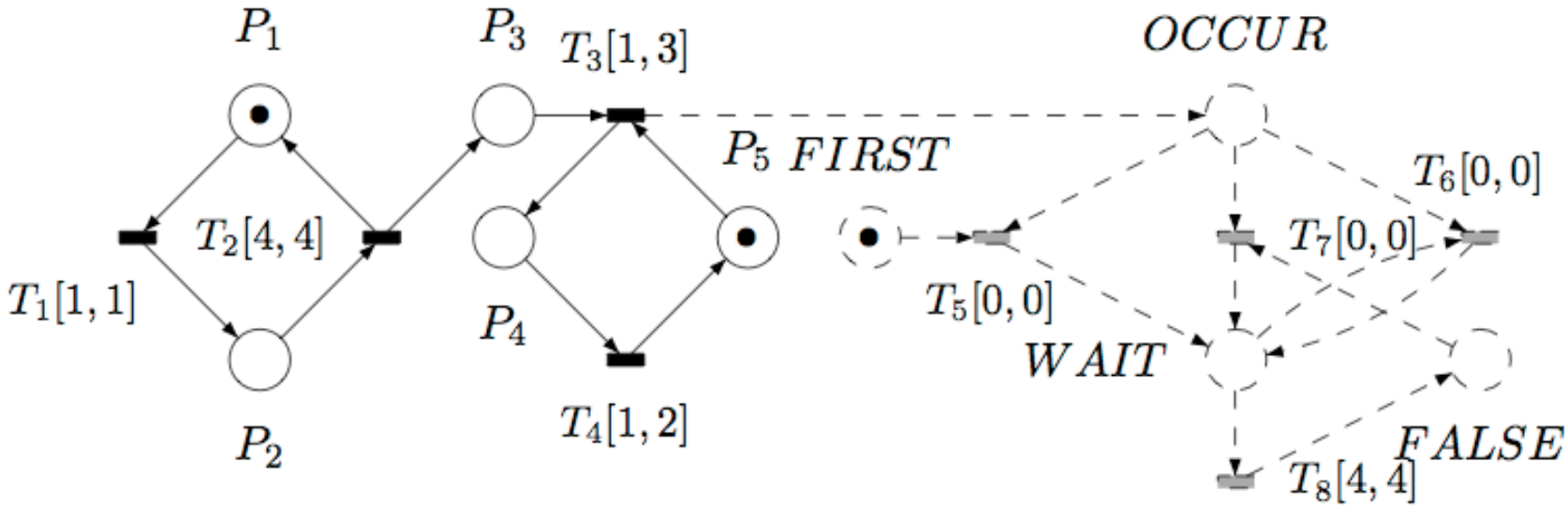
State Space Abstractions

ROMEO implements two types of state space computation:

- Classical state class graph
  - BERTHOMIEU and DIAZ 1991
  - Classical method to compute the state space
  - Untimed language.
  - Preserves *LTL* properties.
- Zone based graph (FORMATS'03)
  - Forward exploration of the state space
  - Efficient method $\Rightarrow$ efficient reachability algorithm for TPN.

Model-checking using Observers

Observer: TPN pattern that does not modify the behaviour of the initial TPN.



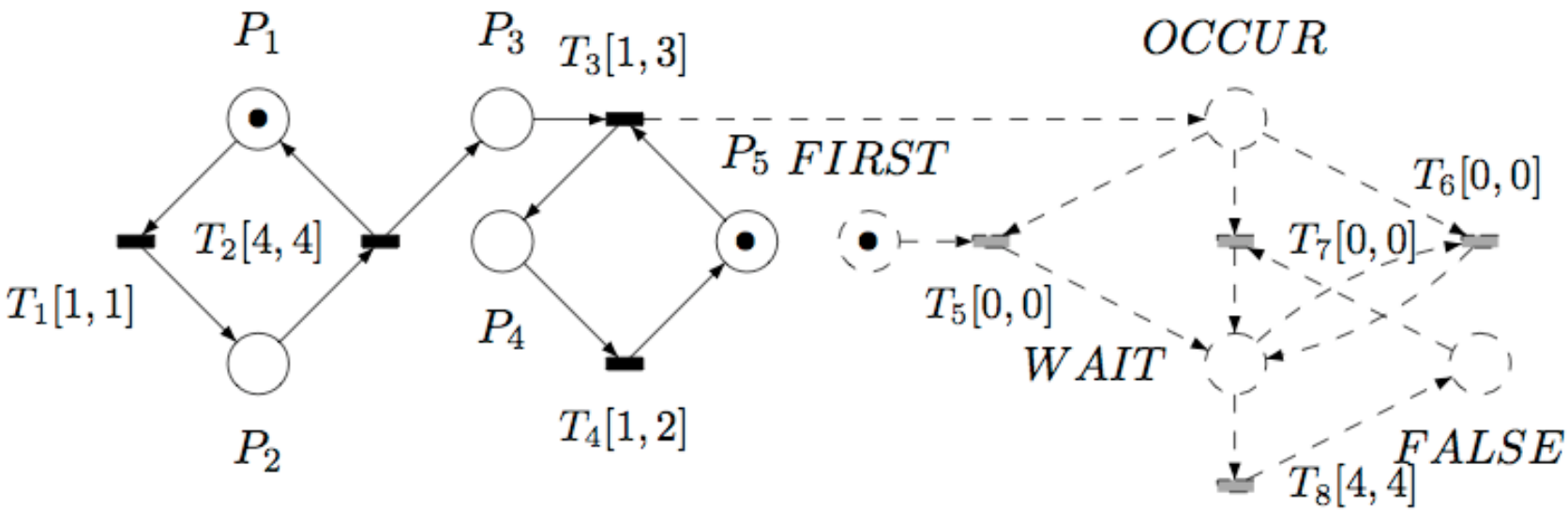
"2 successive occurrences of T_3 always append in less than 4 time units "

$\longrightarrow$ Transform the property to check into an observer:

$\Rightarrow$ reachability test

Model-checking using Observers

Observer: TPN pattern that does not modify the behaviour of the initial TPN.



"2 successive occurrences of T_3 always append in less than 4 time units "

Limitations

Model-checking using translations into Timed Automata

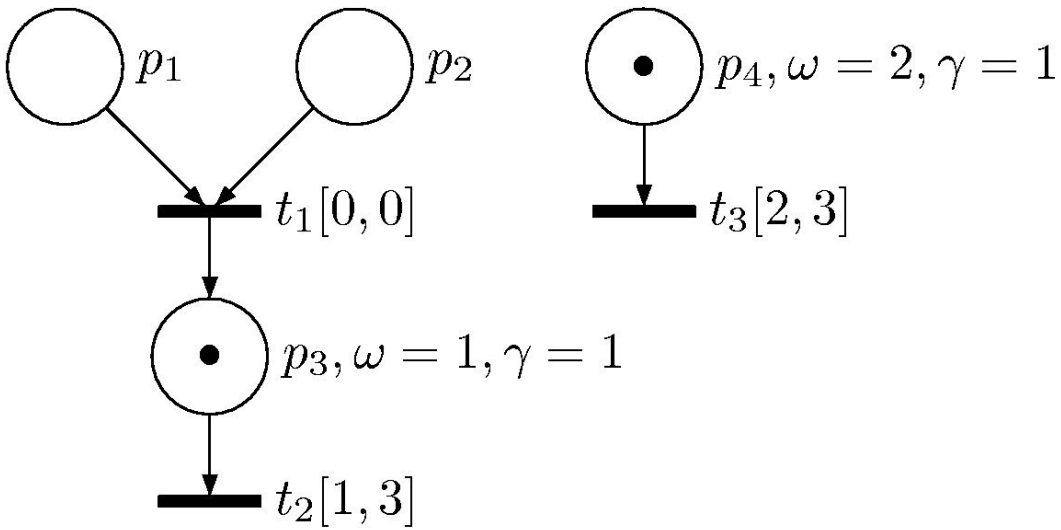
Objectives

- Extend the class of properties
- Use efficient existing model-checkers

Methods

- Structural Translation (AVoCS'04)
  - parallel composition of n Timed Automata
  - optimized to be used with UPPAAL (active clocks)
- State space based translations (jDEDS'05,jTPLP'06)
  - one Timed Automaton
  - small number of clocks

Scheduling Time Petri Nets



- Preemption
- Scheduler: Fixed priority policy

Model-Checking of Scheduling-TPN

- State Space computation (extension of the state class graph)
  - exact: polyhedra (ICATPN'04) , DBM+polyhedra (SoftMC'05)
  - overapproximation (DBM) (FET'03)
- Translation into a Stopwatch Automaton (RTSS'04)
  - Overapproximation (DBM)
  - **but** exact
  - Small number of clocks

Recent Works

Time Petri Nets

- On-the-fly model-checker for a subset of TCTL (EF,EG,AF,AG, bounded liveness)
- Control synthesis for safety properties

Future Works

- System Design
  - Scheduling-TPN: add scheduling policies (Round Robin, Earliest Deadline First...)
  - Inhibitors hyperarcs (Stop and resume clocks)
  - UML (Activity diagram)
- Analysis, Model-checking
  - Discrete Time
  - Full TCTL model-checker

Details

- **Download**
  `http://www.irccyn.ec-nantes.fr/irccyn/d/fr/equipes/TempsReel/logs/software-2-romeo`
- **Contact**
  `romeo@irccyn.ec-nantes.fr`
- **Papers**
  `http://www.irccyn.ec-nantes.fr/~olivier`

Demo: tomorrow 11:00 - 12:00 am

Questions?

Transition Time Petri Net: Definition

Definition

A Transition Time Petri Net (TPN) is a tuple $(P, T, {}^\bullet(.), (.)^\bullet, \alpha, \beta, M_0)$ where:

- $P = \{p_1, p_2, \ldots, p_m\}$, is a non-empty set of places
- $T = \{t_1, t_2, \ldots, t_n\}$, is a non-empty set of transitions
- ${}^\bullet(.) : T \to \mathbb{N}^P$, is the backward incidence function
- $(.)^\bullet : T \to \mathbb{N}^P$, is the forward incidence function
- $M_0 \in \mathbb{N}^P$, the initial marking

Transition Time Petri Net: Definition

Definition

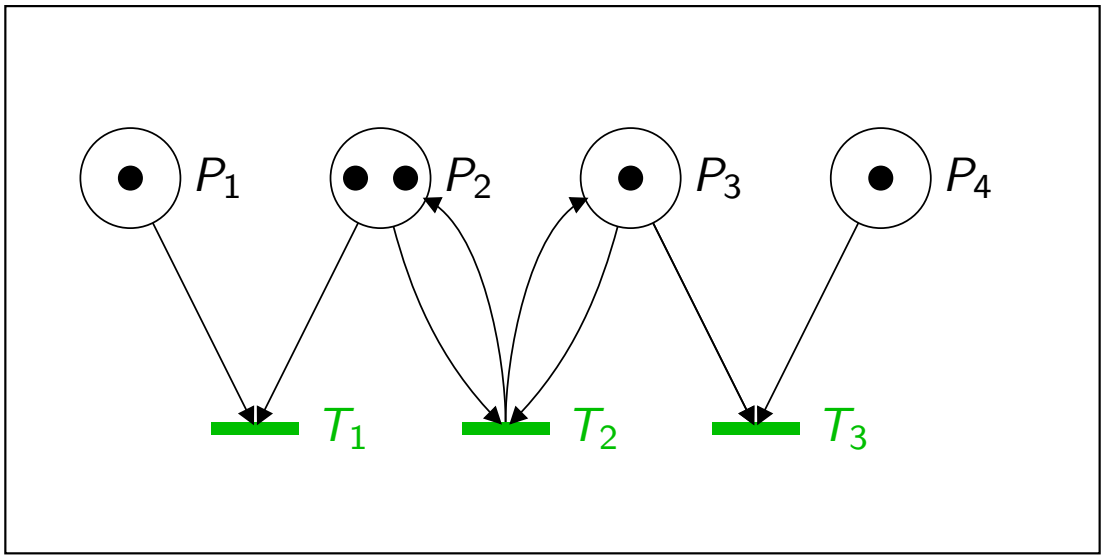
A Transition Time Petri Net (TPN) is a tuple $(P, T, {}^\bullet(.), (.)^\bullet, \alpha, \beta, M_0)$ where:

- $P = \{p_1, p_2, \ldots, p_m\}$, is a non-empty set of places
- $T = \{t_1, t_2, \ldots, t_n\}$, is a non-empty set of transitions
- ${}^\bullet(.) : T \to \mathbb{N}^P$, is the backward incidence function
- $(.)^\bullet : T \to \mathbb{N}^P$, is the forward incidence function
- $M_0 \in \mathbb{N}^P$, the initial marking
- $\alpha : T \to \mathbb{Q}^+$, is the function giving the earliest firing date
- $\beta : T \to \mathbb{Q}^+ \cup \{\infty\}$, is the function giving the latest firing date.

Time Petri Net: Semantics

Definition (Newly enabled transition)

$$enabled\,(M)\,t_i = \begin{cases} t_i \\ t_k \text{ s.t. } M - {}^\bullet t_i \leq {}^\bullet t_k \ \wedge \ M - {}^\bullet t_i + t_i^\bullet \geq {}^\bullet t_k \end{cases}$$

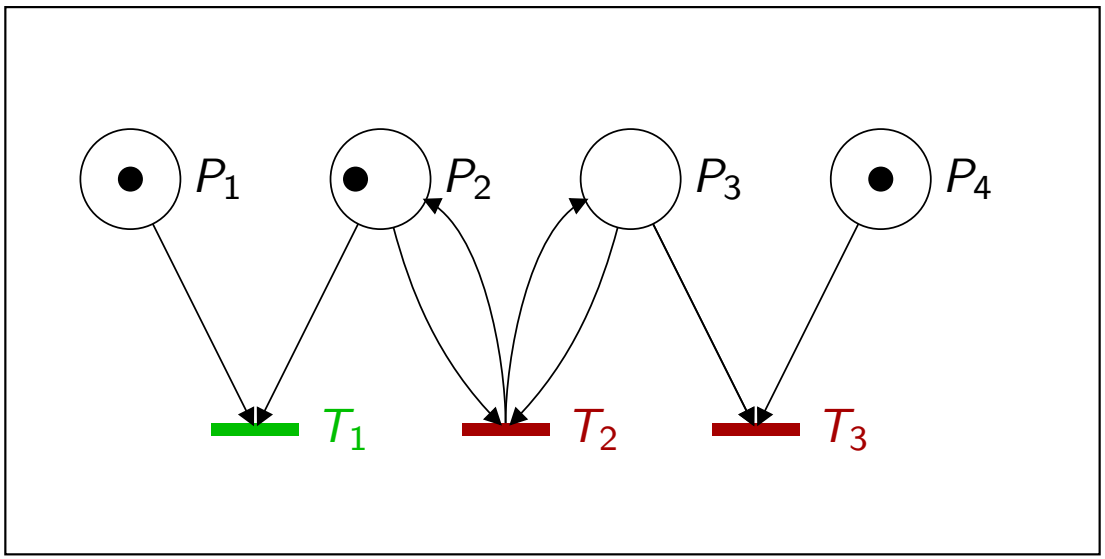


$$enabled\,(M)\,T_2 = \{T_2, T_3\}$$

Time Petri Net: Semantics

Definition (Newly enabled transition)

$$enabled\,(M)\,t_i = \begin{cases} t_i \\ t_k \text{ s.t. } M - {}^\bullet t_i \leq {}^\bullet t_k \ \wedge \ M - {}^\bullet t_i + t_i^\bullet \geq {}^\bullet t_k \end{cases}$$

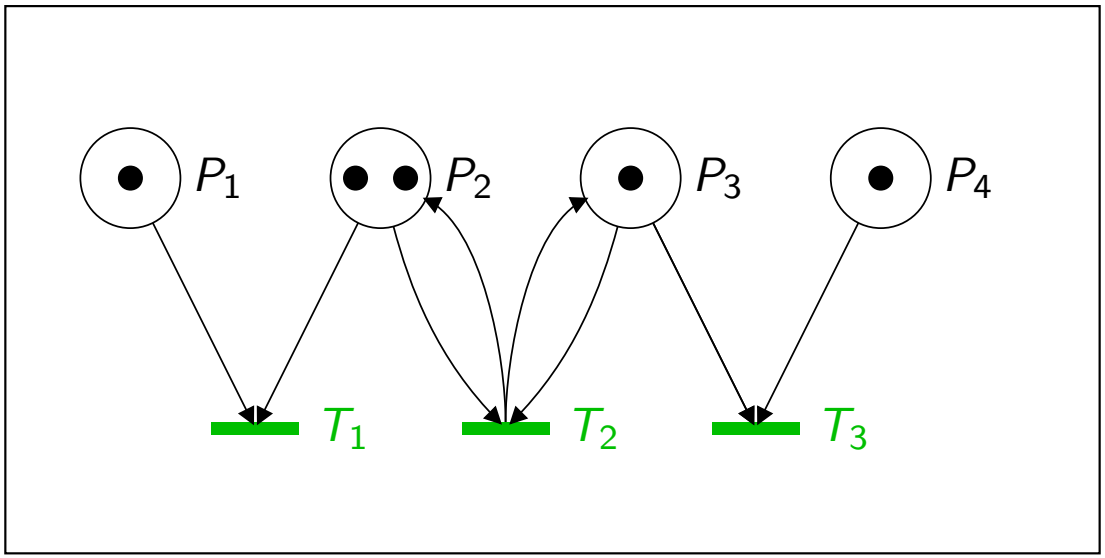


$$enabled\,(M)\,T_2 = \{T_2, T_3\}$$

Time Petri Net: Semantics

Definition (Newly enabled transition)

$$enabled\,(M)\,t_i = \begin{cases} t_i \\ t_k \text{ s.t. } M - {}^\bullet t_i \leq {}^\bullet t_k \ \wedge \ M - {}^\bullet t_i + t_i^\bullet \geq {}^\bullet t_k \end{cases}$$



$$enabled\,(M)\,T_2 = \{T_2, T_3\}$$

Time Petri Net: Semantics

Definition

Timed Transition System $S = (Q, q_0, \rightarrow)$ where :

- $Q = \mathbb{N}^P \times (\mathbb{R}^+)^T$
- $q_0 = (M_0, \bar{0})$
- $\rightarrow \in Q \times (T \cup \mathbb{R}) \times Q$ defined by:
  - **continuous transition** :
    $(M, v) \xrightarrow{e(d)} (M, v')$ iff $\begin{cases} v' = v + d \\ \forall k \in [1, n]\ M \geq^\bullet t_k \Rightarrow v'_k \leq \beta(t_k) \end{cases}$
  - **discrete transition** :
    $(M, v) \xrightarrow{t_i} (M', v')$ iff

$$\begin{cases} M \geq^\bullet t_i \\ M' = M -^\bullet t_i + t_i^\bullet \\ \alpha(t_i) \leq v_i \leq \beta(t_i) \\ \forall k \in [1, n]\ v'_k = \begin{cases} 0 \text{ iff } t_k \in \mathit{enabled}\,(M)\, t_i \\ v_k \text{ otherwise} \end{cases} \end{cases}$$

Undecidability result

Theorem

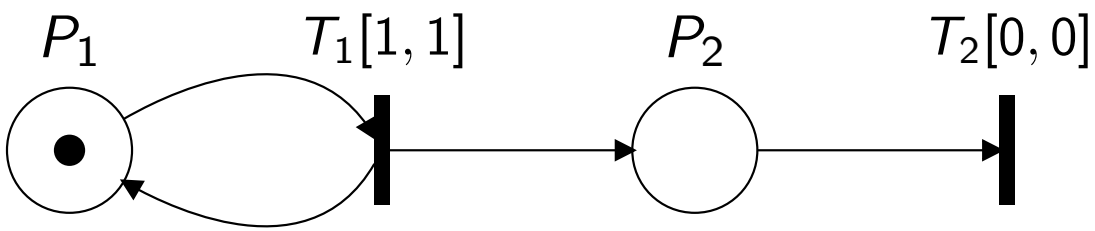
The boundedness of a Transition Time Petri Net is undecidable.

Undecidability result

Theorem

The boundedness of a Transition Time Petri Net is undecidable.

$\longrightarrow$ Does not reduce to the boundedness of the underlying Petri Net.



Undecidability result

Theorem

The boundedness of a Transition Time Petri Net is undecidable.

$\longrightarrow$ Does not reduce to the boundedness of the underlying Petri Net.