Symbolic Unfoldings of Time Petri Nets

Thomas Chatain
joint work with Claude Jard

LSV/ENS Cachan

Petri Nets Course on Unfoldings, Milano, June 25, 2013

Time Petri Nets

- Introduced by Merlin and Farber in 1976
- Specification of real-time concurrent systems
- Time constraints: intervals of possible firing delays
- Strong time semantics
- We consider safe time Petri nets
  - Several undecidable results in the general case...

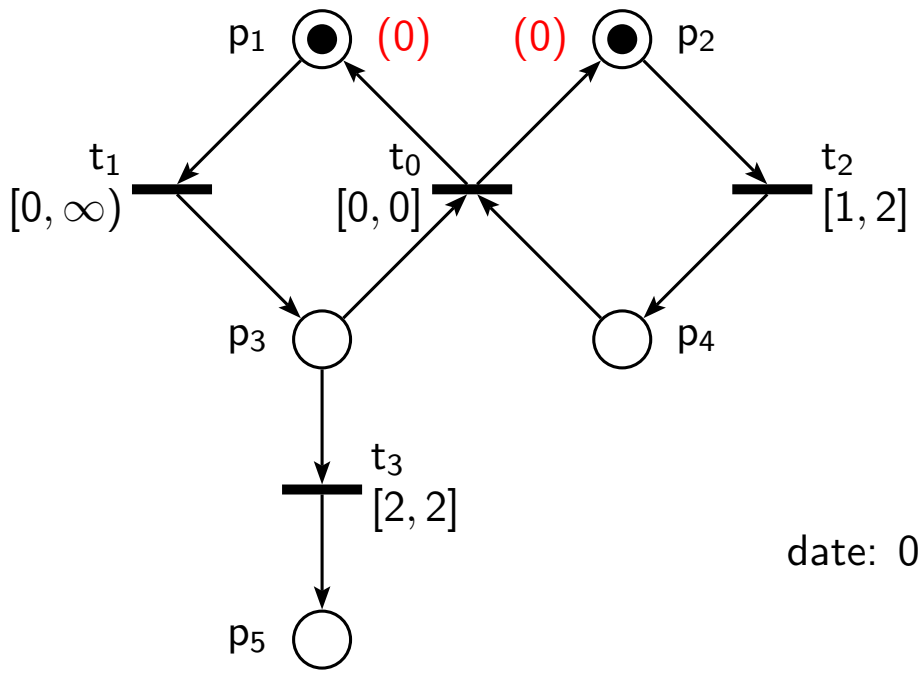
Safe Time Petri Nets: Definition

$\langle P, T, pre, post, efd, lfd \rangle$

- ${}^\bullet t \stackrel{\text{def}}{=} pre(t) \subseteq P$
- $t^\bullet \stackrel{\text{def}}{=} post(t) \subseteq P$
- earliest firing delay:
  $efd : T \longrightarrow Q$
- latest firing delay:
  $lfd : T \longrightarrow Q \cup \{\infty\}$

State $\langle M, dob, \theta \rangle$

- $M \subseteq P$ marking
- θ date
- date of birth:
  $\forall p \in M \quad dob(p) \leq \theta$



Safe Time Petri Nets: Semantics

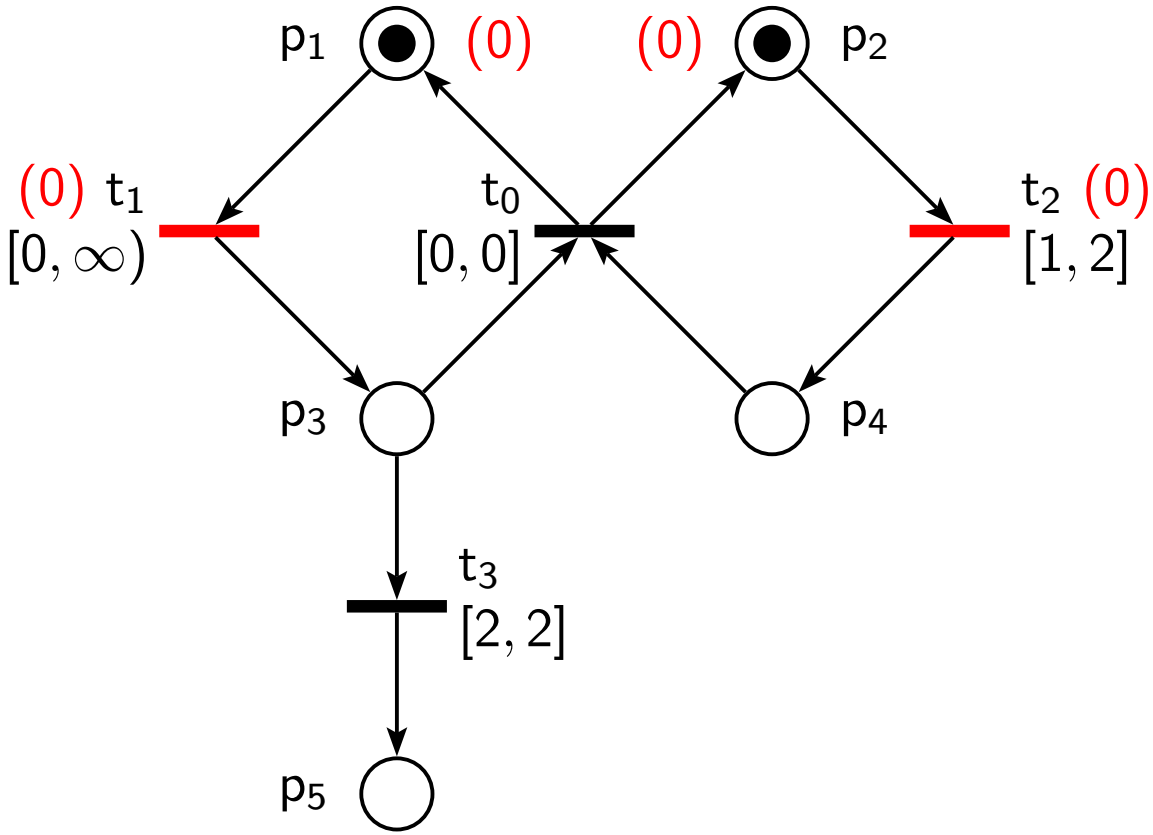
Transition t can fire at time $\theta' \geq \theta$ from state $\langle M, dob, \theta \rangle$ if:

- t is enabled: ${}^\bullet t \subseteq M$;
- the minimum delay is reached:
  $\theta' \geq doe(t) + efd(t)$;
- the enabled transitions do not overtake the maximum delays:
  $\forall t' \in T \quad {}^\bullet t' \subseteq M \implies \theta' \leq doe(t') + lfd(t')$.

where $doe(t) \stackrel{\text{def}}{=} \max_{p \in {}^\bullet t} dob(p)$.

Example of Time Petri Net

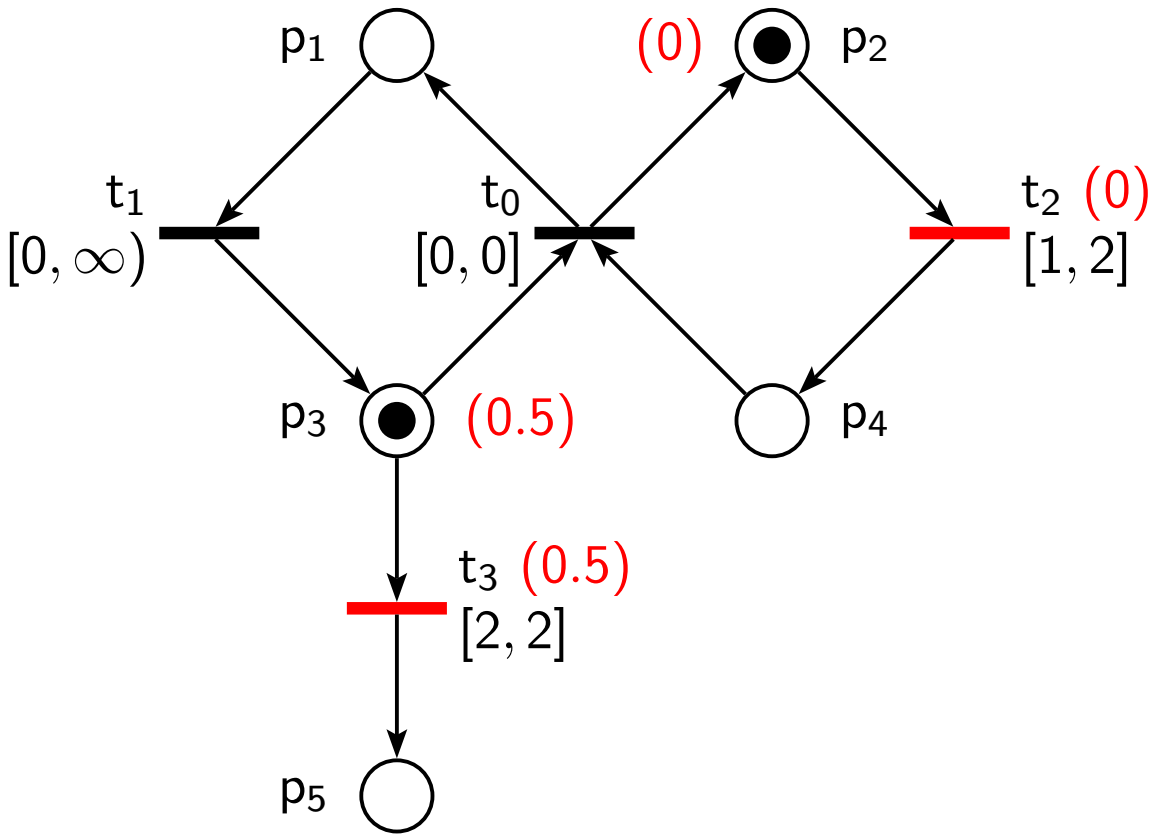
date: 0.0



Example of Time Petri Net

$(t_1, 0.5)$

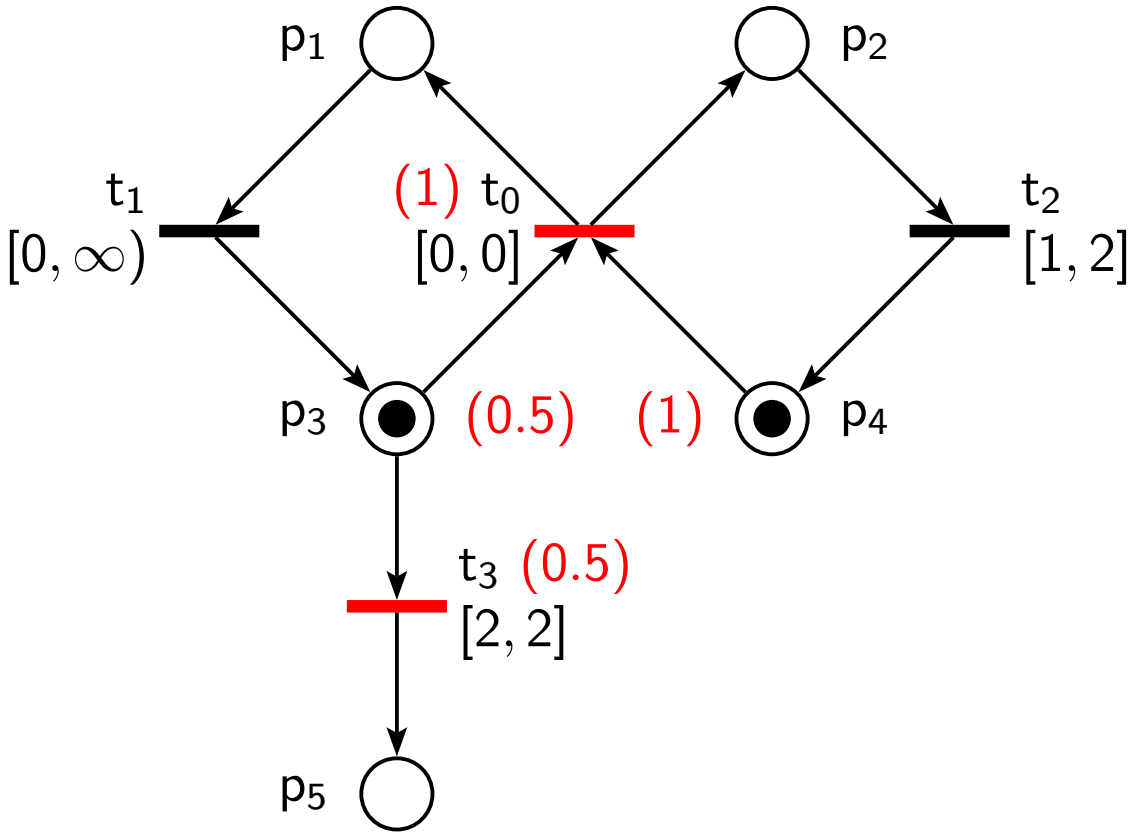
date: 0.5



Example of Time Petri Net

$(t_1, 0.5), (t_2, 1)$

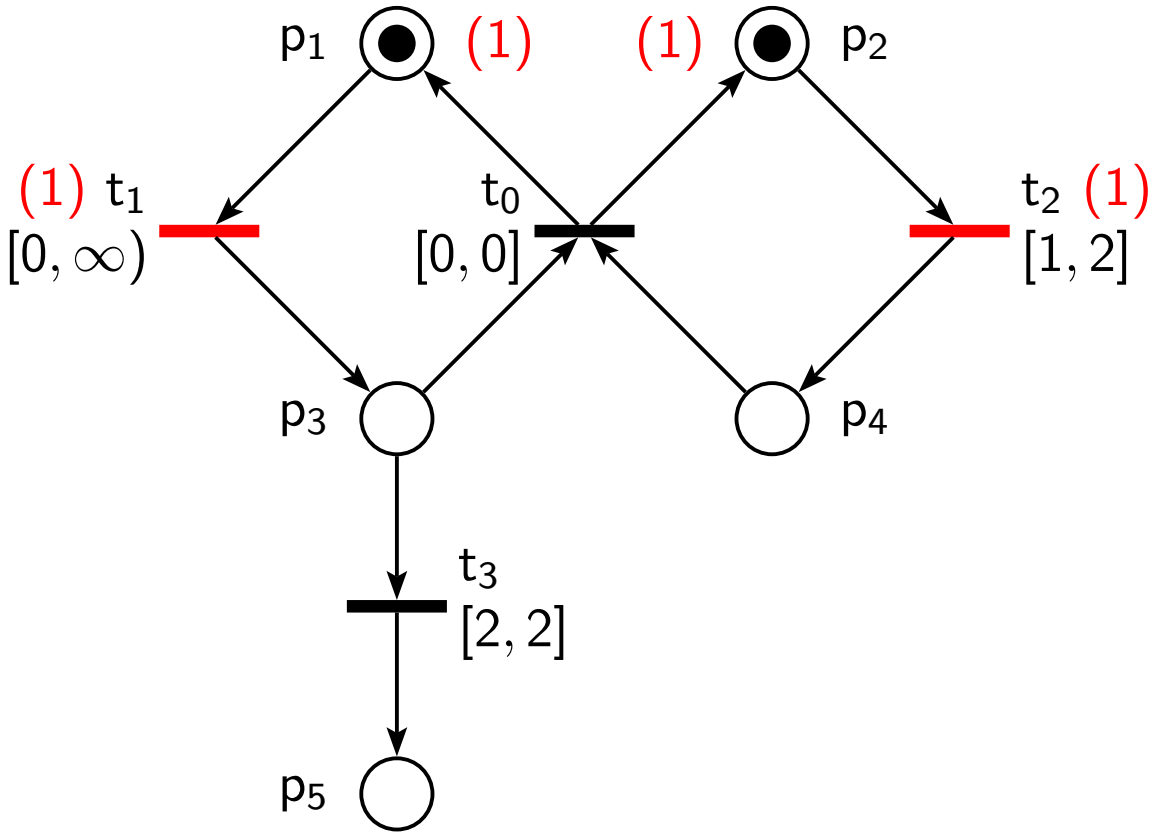
date: 1.0



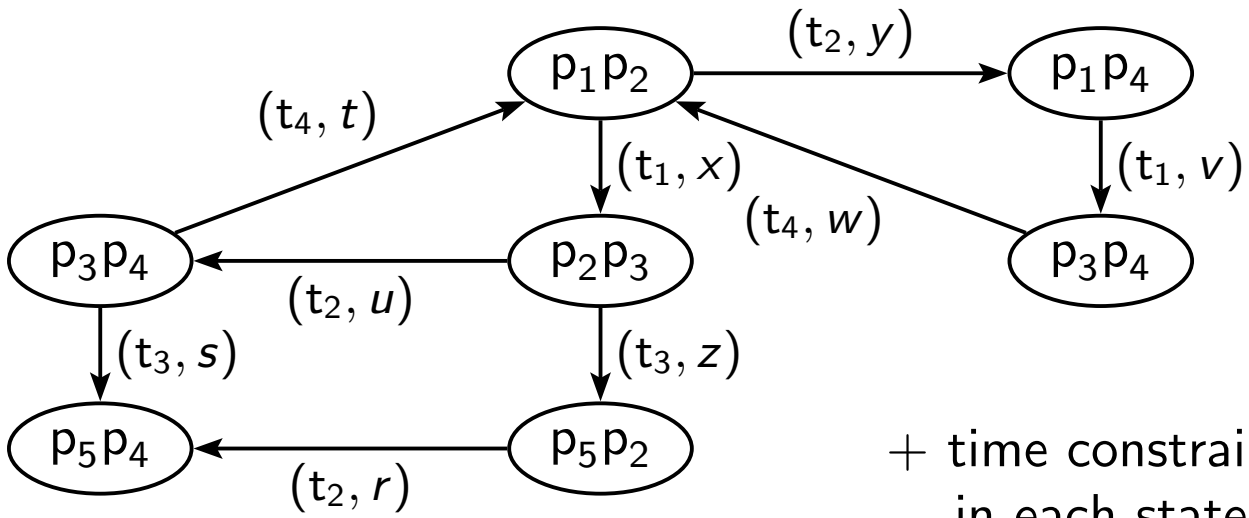
Example of Time Petri Net

$(t_1, 0.5), (t_2, 1), (t_0, 1)$

date: 1.0

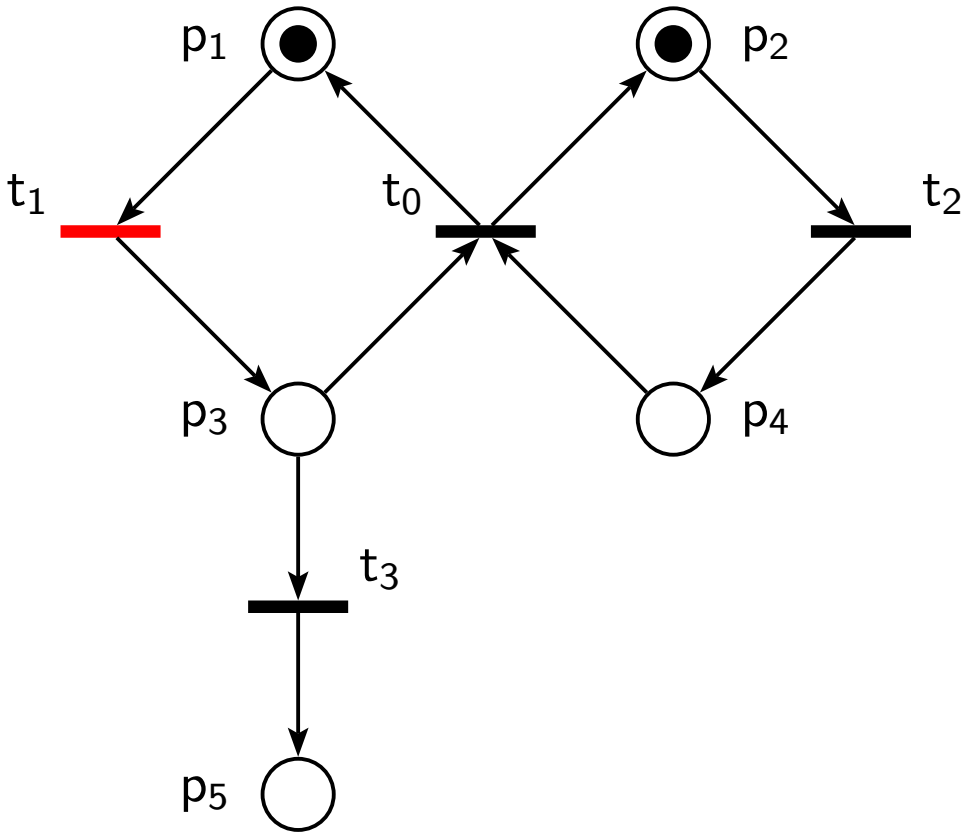


Executions in Interleaving Semantics

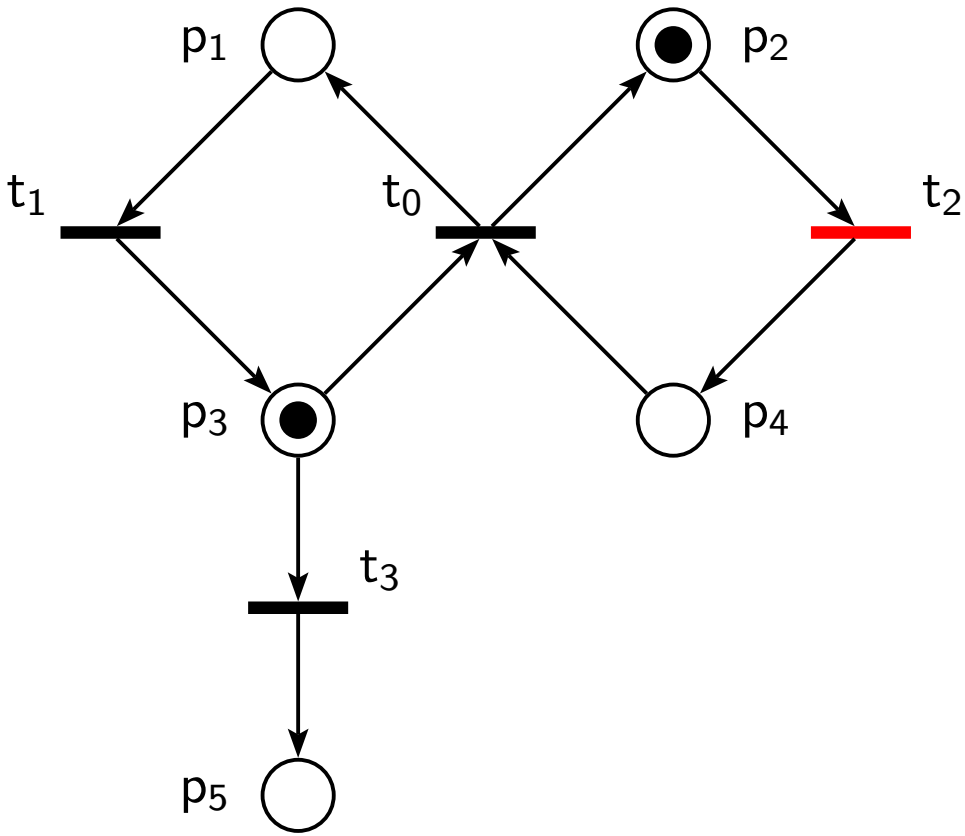


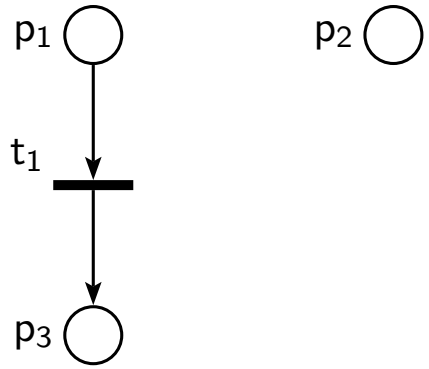
B. Berthomieu and M. Diaz, Modeling and verification of time dependent systems using time Petri nets, IEEE Transactions on Software Engineering, 17(3):259–273, 1991

Processes of Untimed Petri Nets

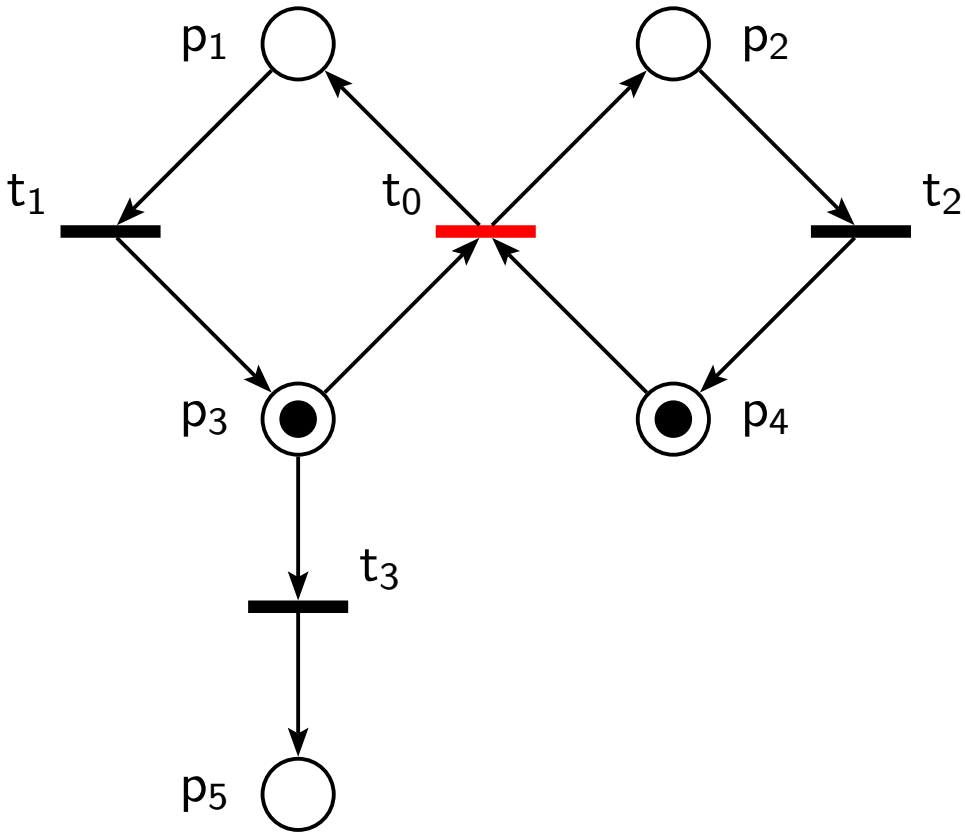


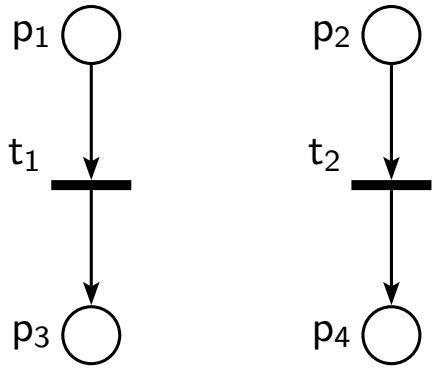
Processes of Untimed Petri Nets



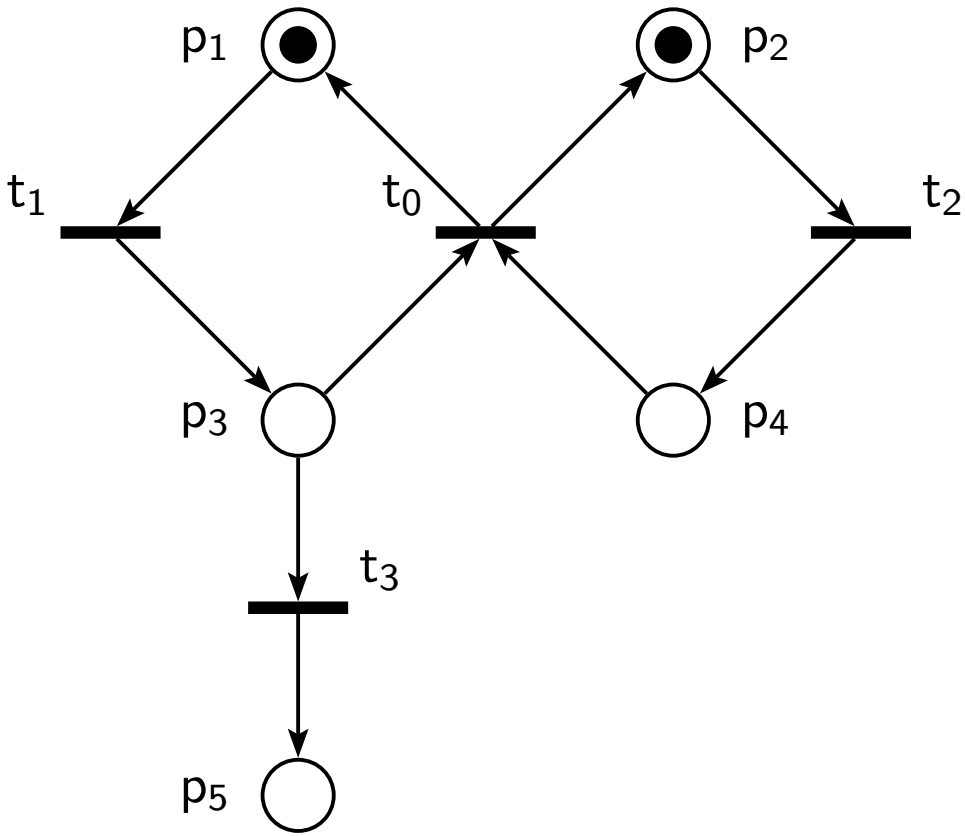


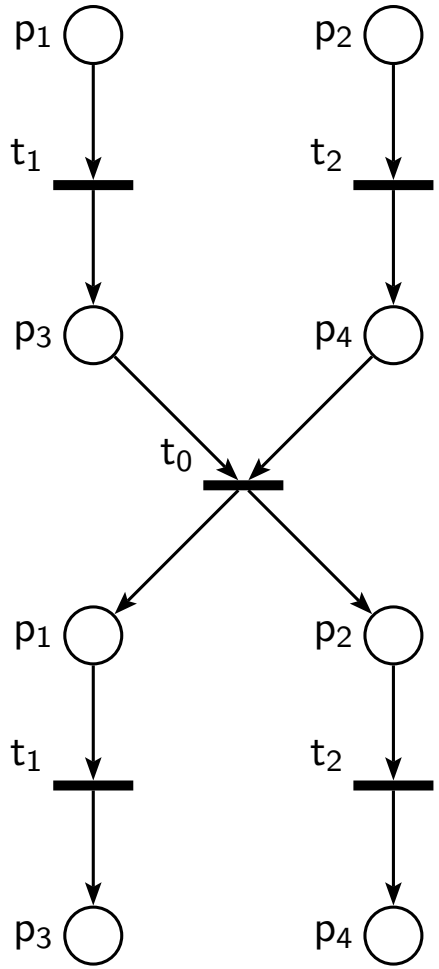
Processes of Untimed Petri Nets



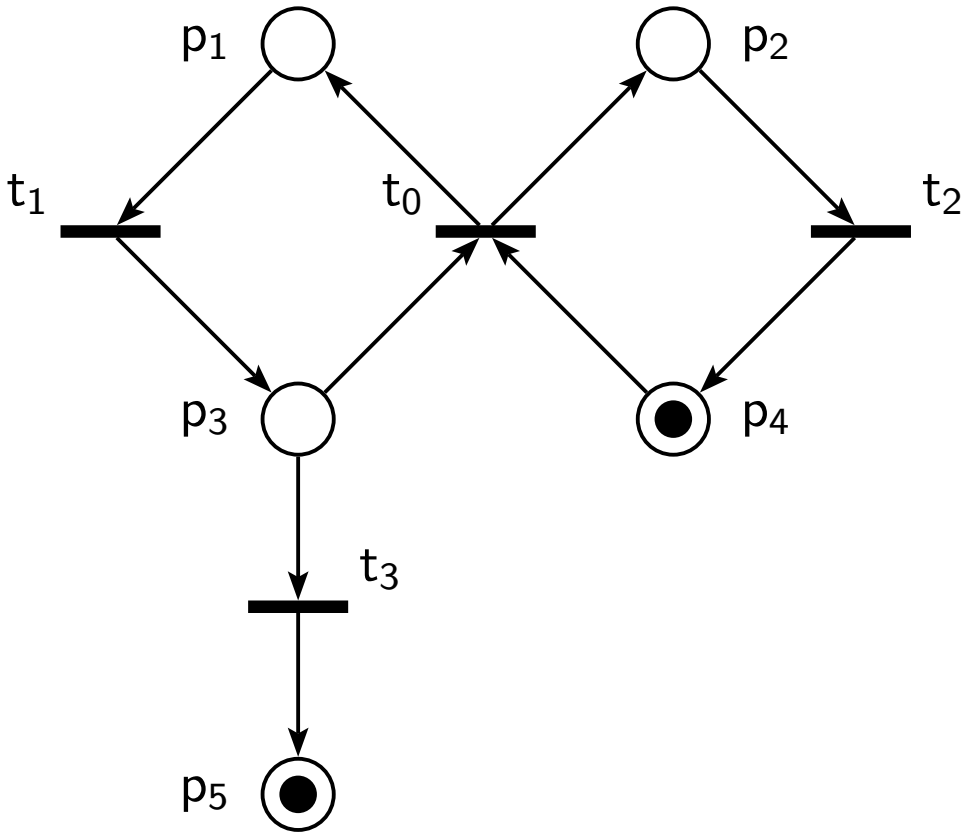


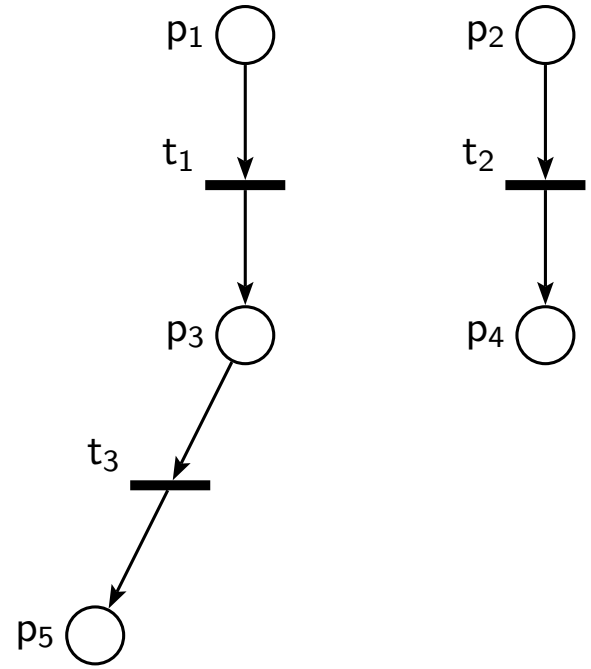
Processes of Untimed Petri Nets



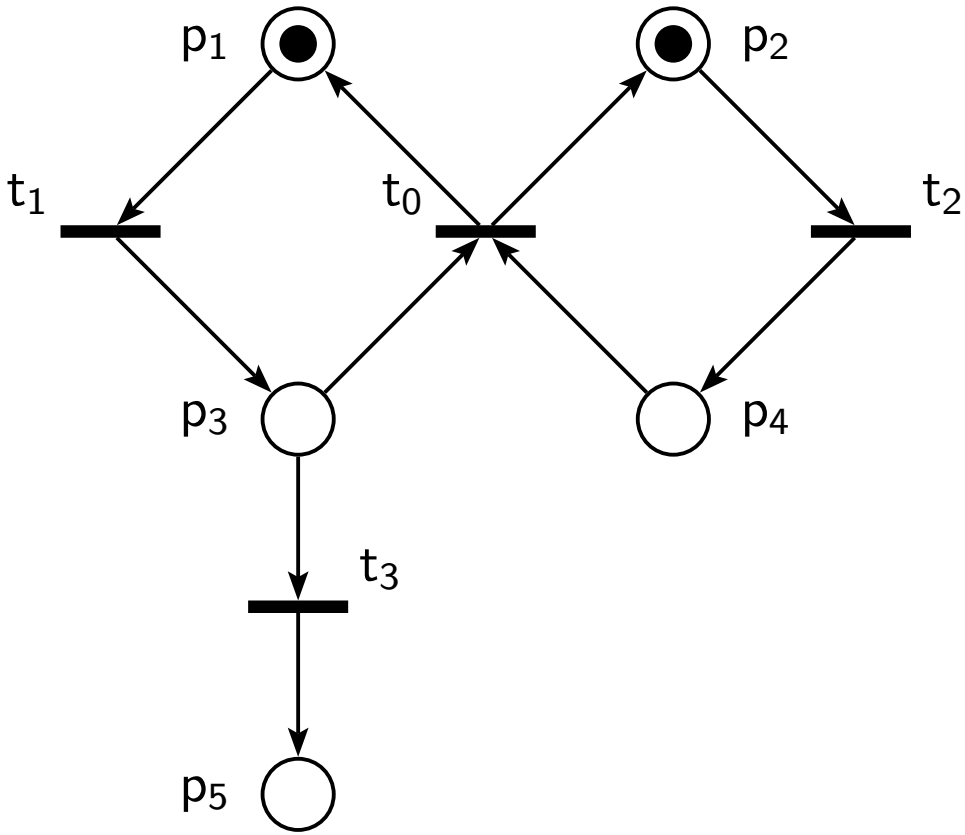


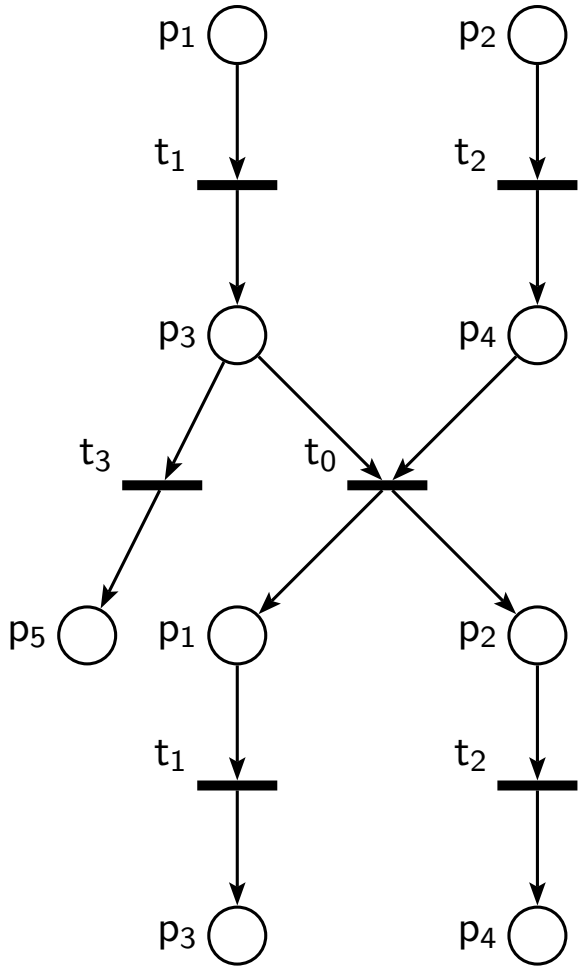
Processes of Untimed Petri Nets





Unfoldings of Untimed Petri Nets

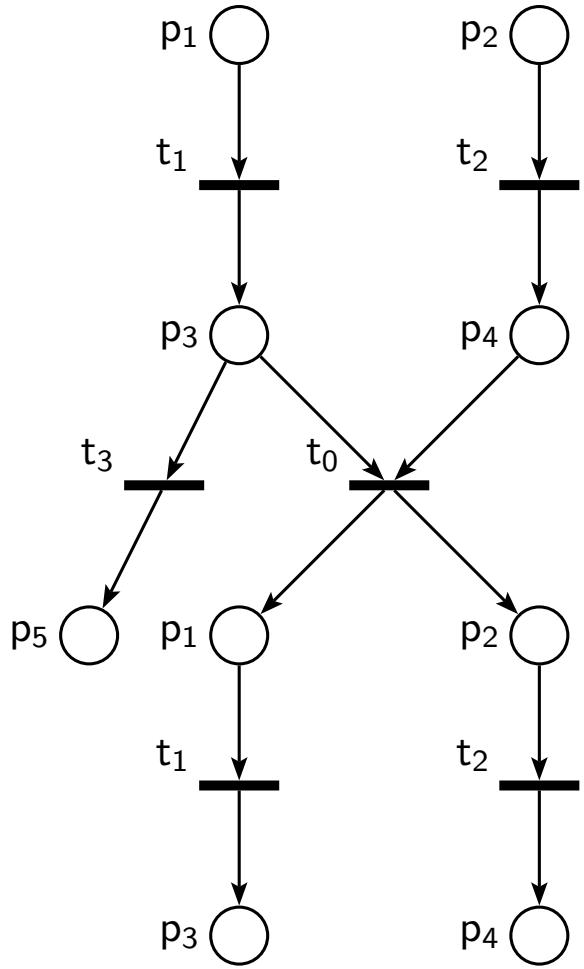




Unfoldings of Untimed Petri Nets

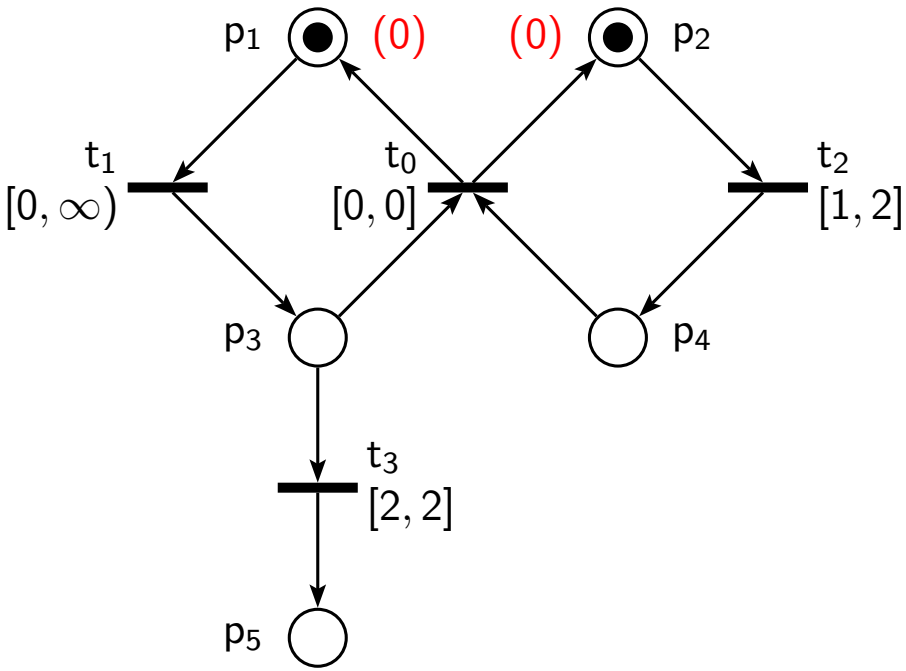
All the processes have been superimposed.

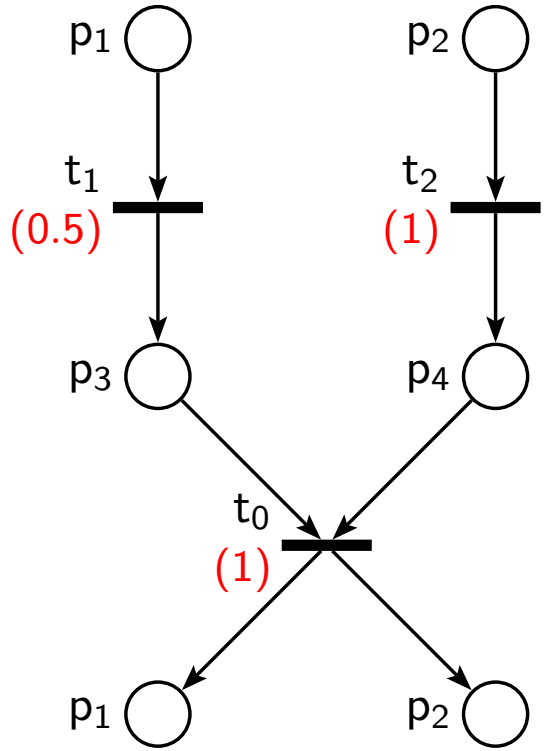
- Given a set E of events, is it a process?
- Given a set E of events, does there exist a process F such that $E \subseteq F$?
- Given a set C of conditions, does there exist a process E such that C are final conditions of E? (useful for construction)



Processes of Time Petri Nets

$(t_1, 0.5), (t_2, 1), (t_0, 1)$





Other dates are possible with the same structure $\rightarrow$ parameters

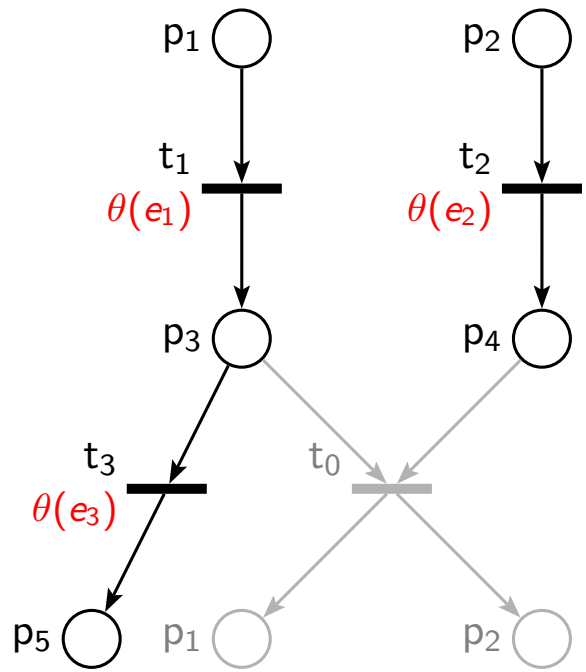
Process of Time Petri Nets – Finding the Possible Dates

Symbolic representation of the dates as parameters.

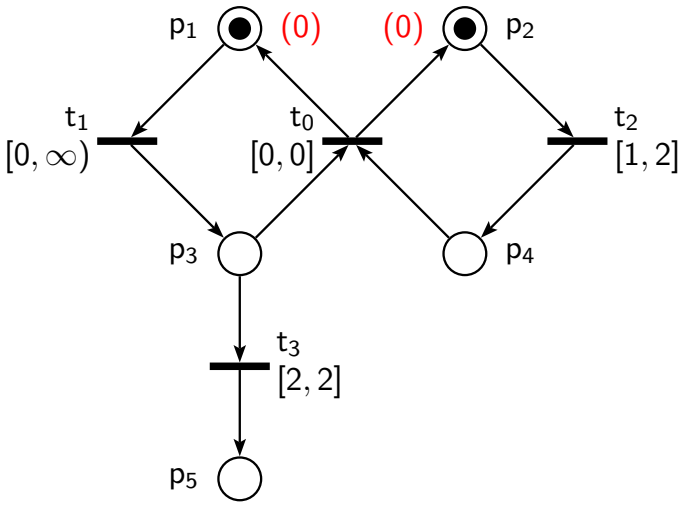
Question: given a process of the underlying Petri net, what are the possible dates for the events?

> Tuomas Aura and Johan Lilius. A causal semantics for time Petri nets. Theoretical Computer Science, 243(2):409-447, (2000)

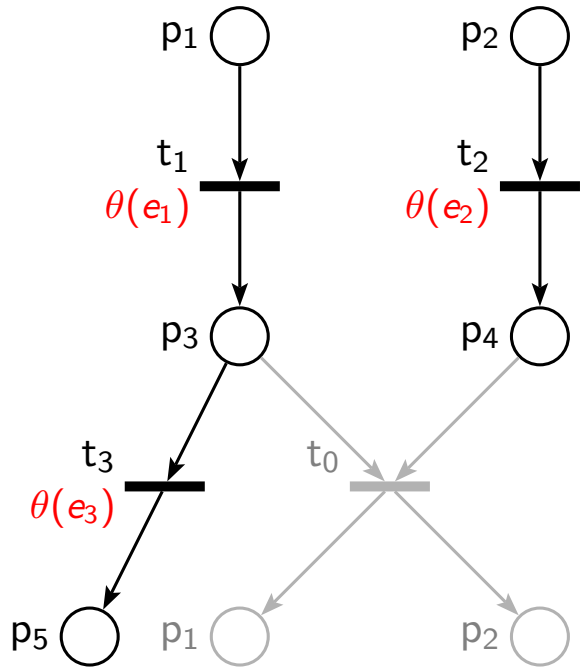
- System of inequalities on the dates of the events.
- Some events that are not in the process must be taken into account.



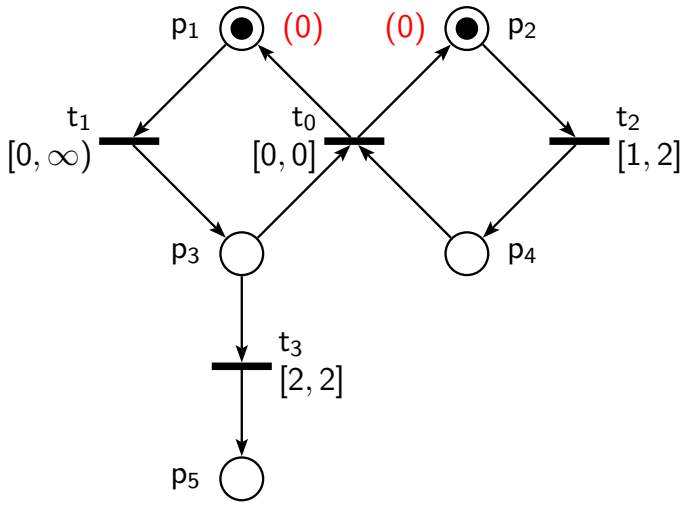
Process of Time Petri Nets – Finding the Possible Dates



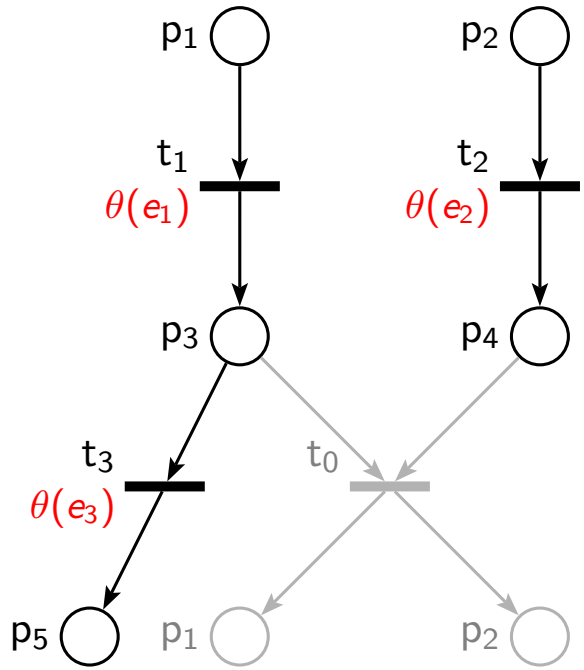
- Firing delays respected
  - $0 \leq \theta(e_1)$
  - $1 \leq \theta(e_2) \leq 2$
  - $\theta(e_3) = \theta(e_1) + 2$



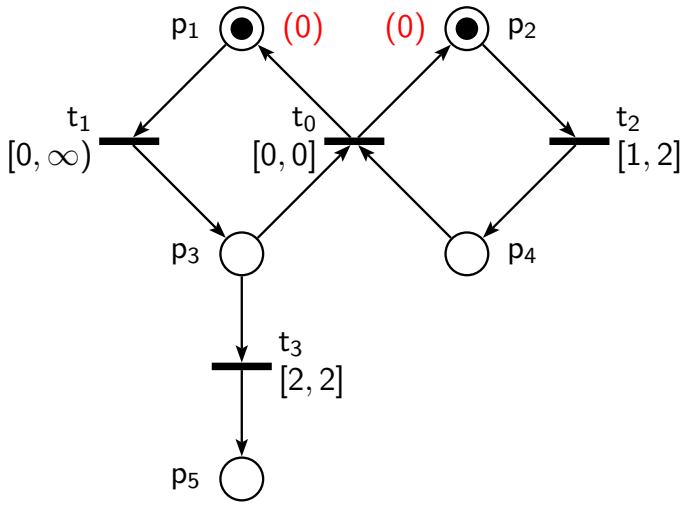
Process of Time Petri Nets – Finding the Possible Dates

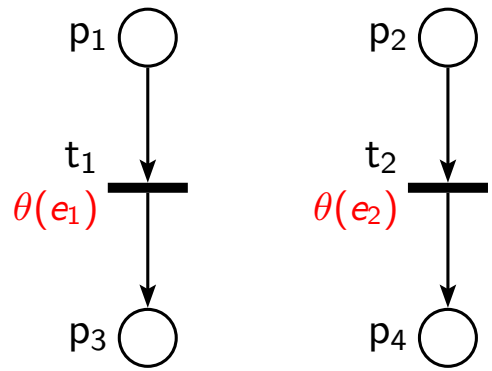


- Firing delays respected
  - $0 \leq \theta(e_1)$
  - $1 \leq \theta(e_2) \leq 2$
  - $\theta(e_3) = \theta(e_1) + 2$
- Disabled events have not overtaken their latest firing delay
  - $\theta(e_3) \leq \max\{\theta(e_1), \theta(e_2)\} + 0$



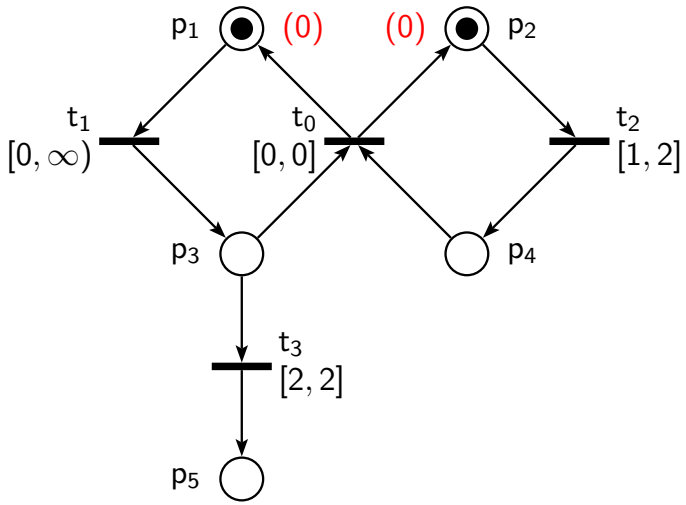
Process of Time Petri Nets – Finding the Possible Dates



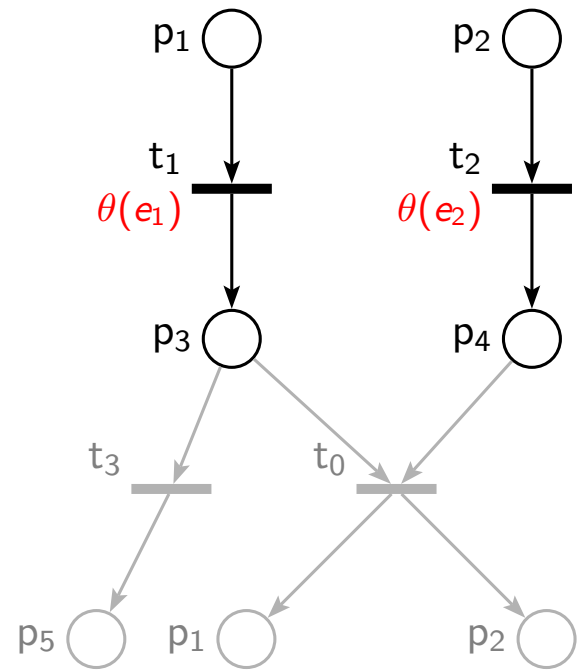


- Firing delays respected
  - $0 \leq \theta(e_1)$
  - $1 \leq \theta(e_2) \leq 2$
- Disabled events have not overtaken their latest firing delay
  - None here

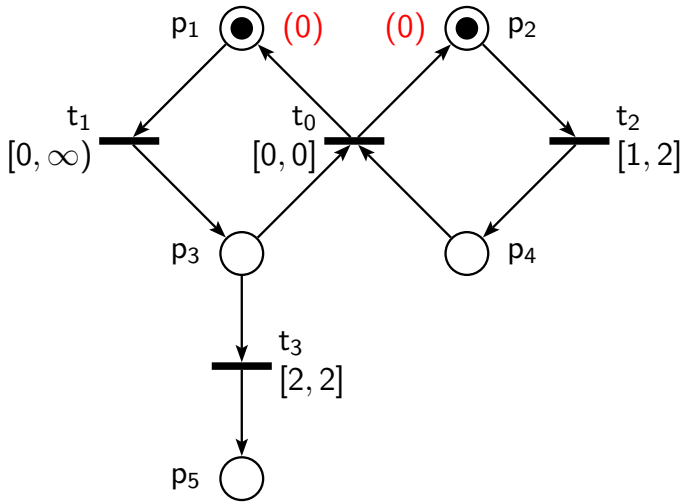
Process of Time Petri Nets – Finding the Possible Dates



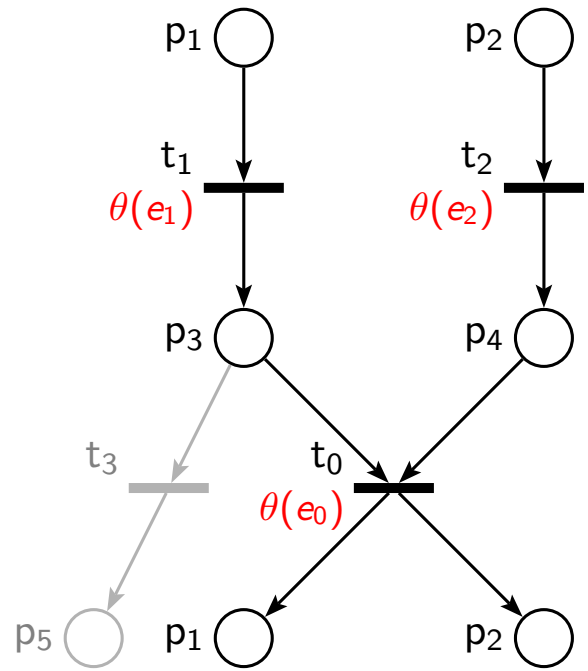
- Firing delays respected
  - $0 \leq \theta(e_1)$
  - $1 \leq \theta(e_2) \leq 2$
- Disabled events have not overtaken their latest firing delay
  - None here
- Events enabled in the final configurations have not fired yet
  - $\max\{\theta(e_1), \theta(e_2)\} \leq \theta(e_1) + 2$ for t_3
  - $\max\{\theta(e_1), \theta(e_2)\} \leq \max\{\theta(e_1), \theta(e_2)\} + 0$ for t_0



Exercise – Finding the possible dates



- Firing delays respected
  - ?
- Disabled events have not overtaken their latest firing delay
  - ?
- Events enabled in the final configurations have not fired yet
  - ?

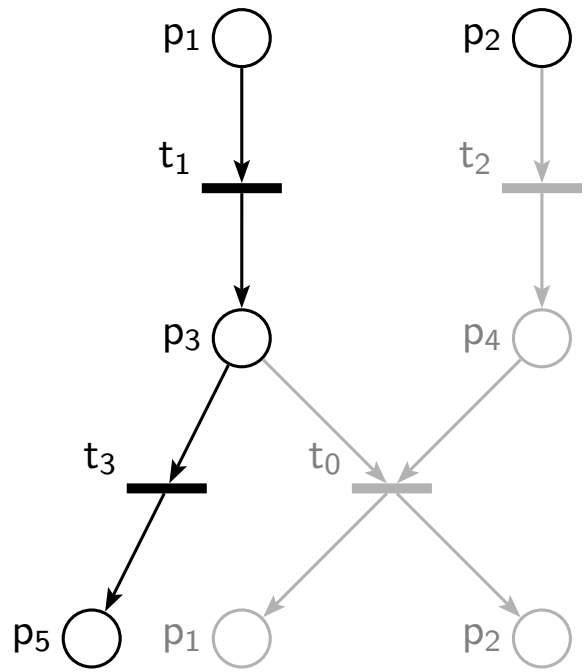


Superimposition of the Processes of a Time Petri Net

Superimposition of all the processes of a time Petri net $\rightarrow$ prefix of the unfolding of the underlying untimed Petri net.

- Given a set E of events, is it a process? $\rightarrow$ answered
- Given a set E of events, does there exist a process F such that $E \subseteq F$?
- Given a set C of conditions, does there exist a process E such that C are final conditions of E? (useful for construction)

Look at what can happen in other parts of the net.
The information that allows a transition to fire is not entirely coded in the corresponding event.
$\rightarrow$ add enough information in the events, but not the global state.



Simple Unfoldings for Time Petri Nets (and Other Models)

Simple cases

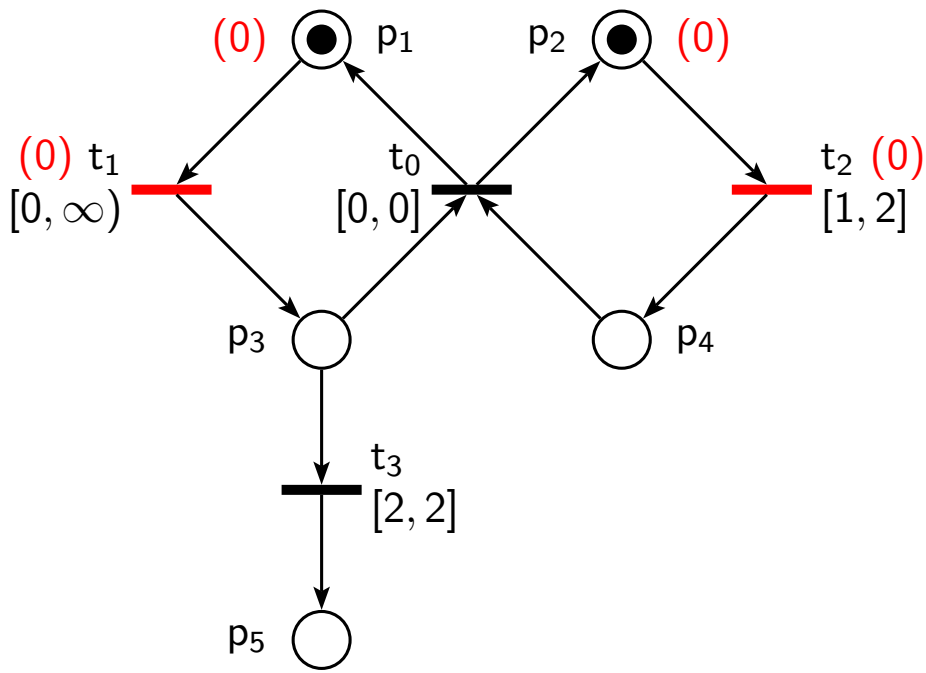
- models without urgency
- models where the choices are local: e.g. extended free choice Time Petri nets

The superimposition of the processes is sufficient for these cases.

Concurrent Operational Semantics for Time Petri Nets

How to simulate a time Petri net without using clocks,
but with as much concurrency as possible?

- Look for local conditions to fire a transition
- Notion of partial state $\langle L, dob, lrd \rangle$
- Partition of P into sets of mutually exclusive places (here $\{p_1, p_3, p_5\}$ and $\{p_2, p_4\}$)
  $\rightarrow$ test the absence of a token
- Executions must map into prefixes of processes



Local Conditions to Fire Transitions

To fire t at θ' from $\langle L, dob, lrd \rangle$, we want (intuitively):

$$\text{for any global state } S \text{ that extends } \langle L, dob, lrd \rangle,$$
$$t \text{ can fire at } \theta' \text{ from } S.$$

We have:

$$t \text{ can fire at } \theta' \text{ from } S$$
$$\text{iff}$$

- t is enabled: ${}^\bullet t \subseteq M$;
- the minimum delay is reached: $\theta' \geq doe(t) + efd(t)$;
- all the enabled transitions in S do not overtake the maximum delays: $\forall t' \in T \quad {}^\bullet t' \subseteq M \implies \theta' \leq doe(t') + lfd(t')$.

$$\text{where } doe(t) \stackrel{\text{def}}{=} \max_{p \in {}^\bullet t} dob(p).$$

Local Firing Condition

Transition t can fire at date θ from the partial state $\langle L, dob, lrd \rangle$ if

- t is enabled: ${}^\bullet t \subseteq L$;
- the minimum delay is reached: $\theta \geq doe(t) + efd(t)$;
- the partial state $\langle L, dob, lrd \rangle$ can remain until θ (whatever happens in other parts of the net)

local stability condition

$$LSC(\langle L, dob, lrd \rangle, \theta)$$

Local Stability Condition

Intuition:

$$LSC(\langle L, dob, lrd \rangle, \theta)$$
iff
for all context S of $\langle L, dob, lrd \rangle$, $\langle L, dob, lrd \rangle$ is stable in S until θ.

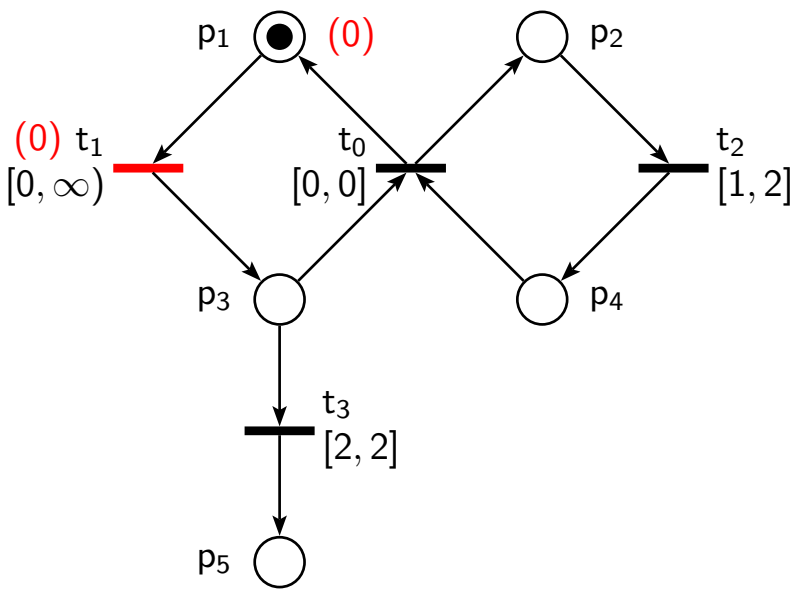
Local Stability Condition

Several choices to define $LSC(\langle L, dob, lrd\rangle, \theta)$:

- trivial choice: $\langle L, dob, lrd\rangle$ is a global state:
  stable until one of the enabled transitions reaches its latest firing delay;
- $\langle L, dob, lrd\rangle$ contains enough information to check that all the transitions that consume tokens in L can wait until θ.

$$\forall t' \in T \quad {}^\bullet t' \cap L \neq \emptyset \implies \begin{cases} & \exists p \in {}^\bullet t' \quad \bar{p} \cap L \neq \emptyset \\ \vee & \theta \leq \max\limits_{p \in {}^\bullet t' \cap L} dob(p) + lfd(t') \end{cases}$$

where $\bar{p}$ denotes the set of places mutually exclusive with p. E.g. $\bar{p_1} = \{p_3, p_5\}$.



$\{(p_1, 0)\}$ is stable forever

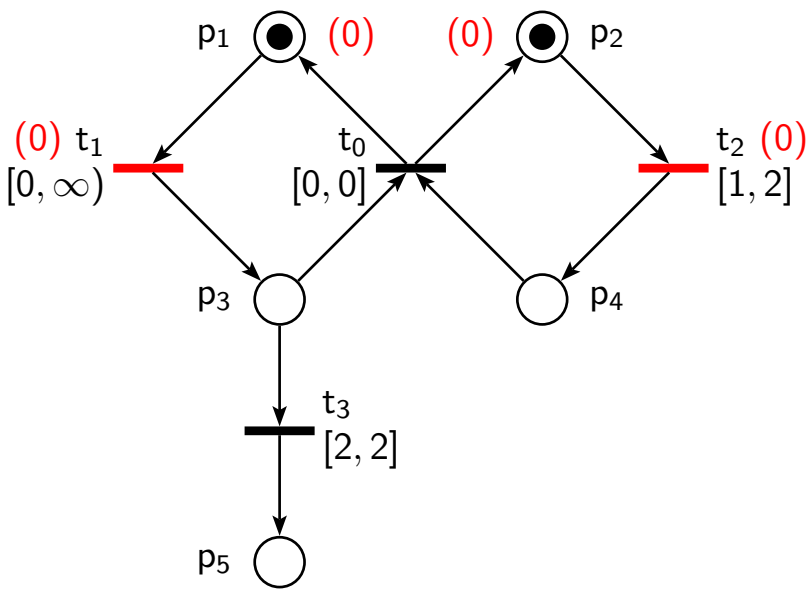
Local Stability Condition

Several choices to define $LSC(\langle L, dob, lrd\rangle, \theta)$:

- trivial choice: $\langle L, dob, lrd\rangle$ is a global state:
  stable until one of the enabled transitions reaches its latest firing delay;
- $\langle L, dob, lrd\rangle$ contains enough information to check that all the transitions that consume tokens in L can wait until θ.

$$\forall t' \in T \quad {}^\bullet t' \cap L \neq \emptyset \implies \begin{cases} & \exists p \in {}^\bullet t' \quad \bar{p} \cap L \neq \emptyset \\ \vee & \theta \leq \max\limits_{p \in {}^\bullet t' \cap L} dob(p) + lfd(t') \end{cases}$$

where $\bar{p}$ denotes the set of places mutually exclusive with p. E.g. $\bar{p_1} = \{p_3, p_5\}$.



$\{(p_1, 0), (p_2, 0)\}$ is stable until $\theta = 2$

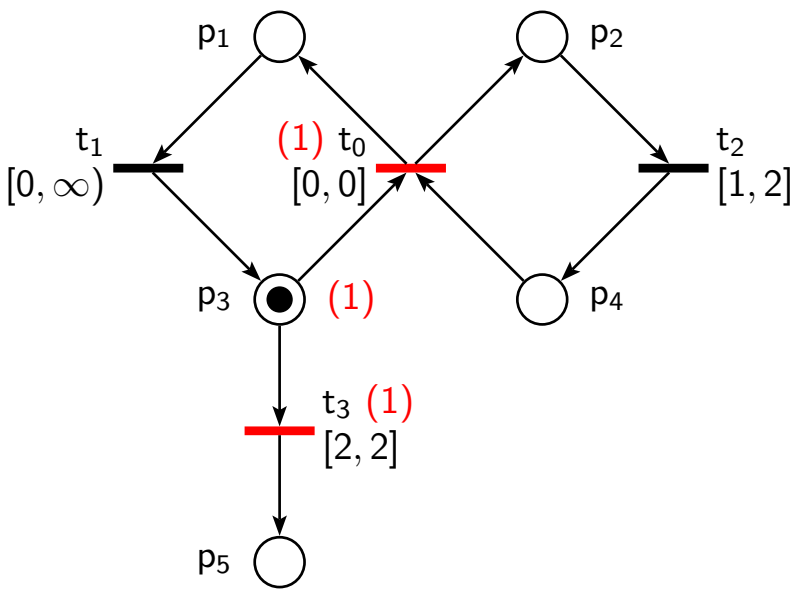
Local Stability Condition

Several choices to define $LSC(\langle L, dob, lrd\rangle, \theta)$:

- trivial choice: $\langle L, dob, lrd\rangle$ is a global state:
  stable until one of the enabled transitions reaches its latest firing delay;
- $\langle L, dob, lrd\rangle$ contains enough information to check that all the transitions that consume tokens in L can wait until θ.

$$\forall t' \in T \quad {}^\bullet t' \cap L \neq \emptyset \implies \begin{cases} & \exists p \in {}^\bullet t' \quad \bar{p} \cap L \neq \emptyset \\ \vee & \theta \leq \max\limits_{p \in {}^\bullet t' \cap L} dob(p) + lfd(t') \end{cases}$$

where $\bar{p}$ denotes the set of places mutually exclusive with p. E.g. $\bar{p_1} = \{p_3, p_5\}$.



$\{(p_3, 1)\}$ is stable until $\theta = 1$

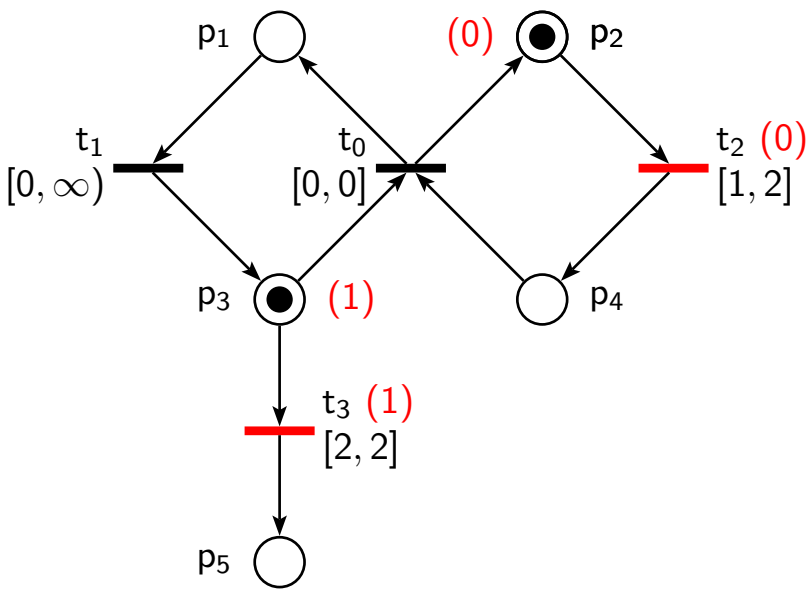
Local Stability Condition

Several choices to define $LSC(\langle L, dob, lrd\rangle, \theta)$:

- trivial choice: $\langle L, dob, lrd\rangle$ is a global state:
  stable until one of the enabled transitions reaches its latest firing delay;
- $\langle L, dob, lrd\rangle$ contains enough information to check that all the transitions that consume tokens in L can wait until θ.

$$\forall t' \in T \quad {}^\bullet t' \cap L \neq \emptyset \implies \begin{cases} & \exists p \in {}^\bullet t' \quad \bar{p} \cap L \neq \emptyset \\ \vee & \theta \leq \max\limits_{p \in {}^\bullet t' \cap L} dob(p) + lfd(t') \end{cases}$$

where $\bar{p}$ denotes the set of places mutually exclusive with p. E.g. $\bar{p_1} = \{p_3, p_5\}$.



$\{(p_3, 1), (p_2, 0)\}$ is stable until $\theta = 2$

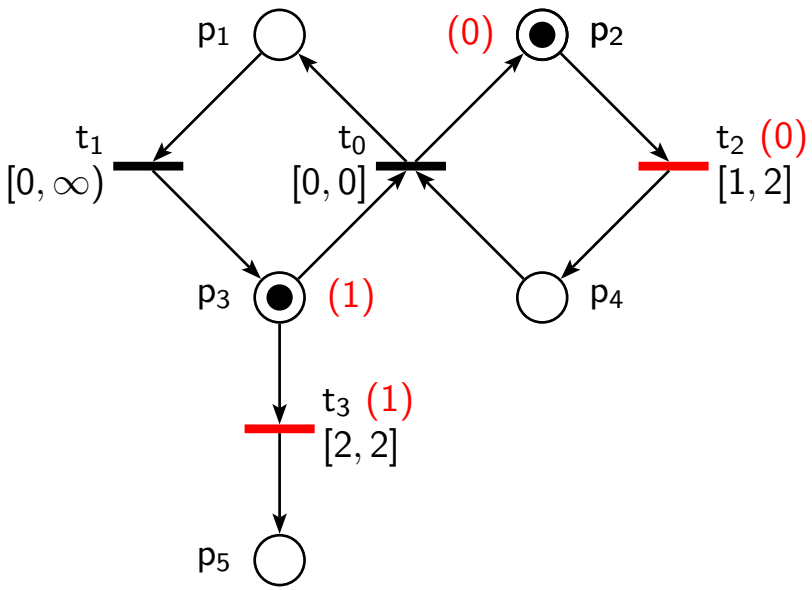
Local Stability Condition

Several choices to define $LSC(\langle L, dob, lrd\rangle, \theta)$:

- trivial choice: $\langle L, dob, lrd\rangle$ is a global state:
  stable until one of the enabled transitions reaches its latest firing delay;
- $\langle L, dob, lrd\rangle$ contains enough information to check that all the transitions that consume tokens in L can wait until θ.

$$\forall t' \in T \quad {}^\bullet t' \cap L \neq \emptyset \implies \begin{cases} & \exists p \in {}^\bullet t' \quad \bar{p} \cap L \neq \emptyset \\ \vee & \theta \leq \max\limits_{p \in {}^\bullet t' \cap L} dob(p) + lfd(t') \end{cases}$$

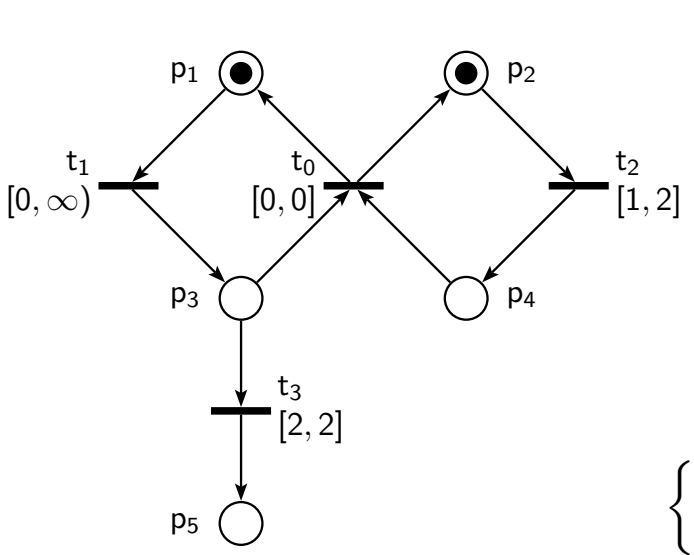
where $\bar{p}$ denotes the set of places mutually exclusive with p. E.g. $\bar{p_1} = \{p_3, p_5\}$.

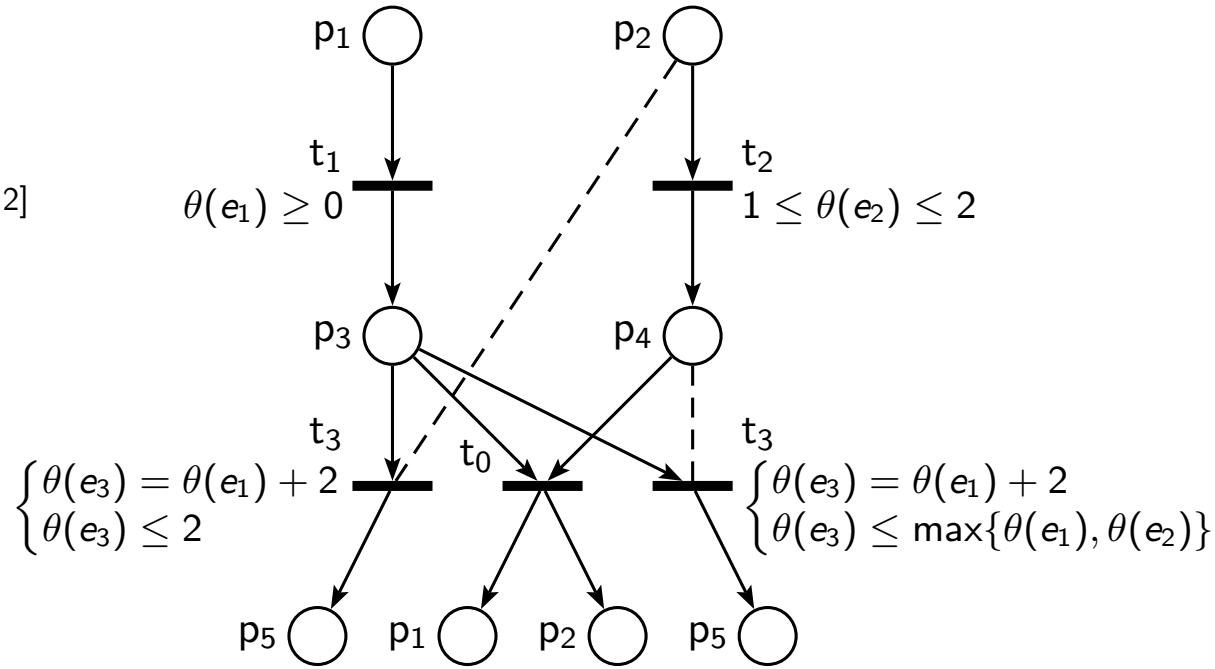


Exercise

Play with local stability conditions!

Symbolic Unfolding





- In symbolic unfoldings: keep track of all the places in L (not only those that are consumed) $\rightarrow$ use read arcs (or consume and rewrite).
- Use only minimal sets L to increase concurrency.
- Redundancy
- Solve constraints on the dates of the events

Conclusion

- Concurrent operational semantics for TPN
- (Parameterized) local stability condition
- Solve constraints on the dates of the events
- Study of the form of the constraints
  $\rightarrow$ finite complete prefix of the unfolding
- If there is no urgency or the choices are local (extended free choice Petri net), the unfolding is simply the superimposition of the processes