Stochastic Petri Nets

Almost none of the theory

Jonatan Lindén

December 8, 2010

Outline

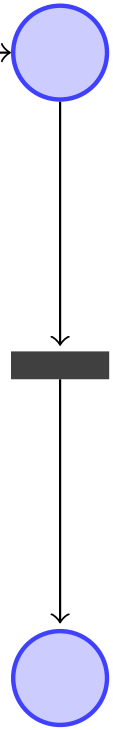
Introduction

- A *Petri net* (PN) is something like a generalized automata.
- A *Stochastic Petri Net* (SPN) – a stochastic extension to Petri nets, for quantitative analysis.
- More expressive than queueing networks and easier to model a wider range of systems.
- Underlying mathematical model : stochastic processes.

Basic example

A PN consists of:

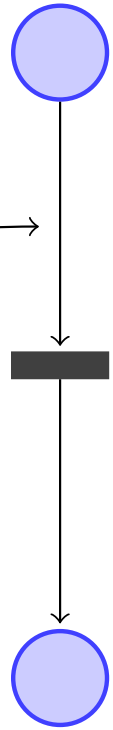
- **Places**



Basic example

A PN consists of:

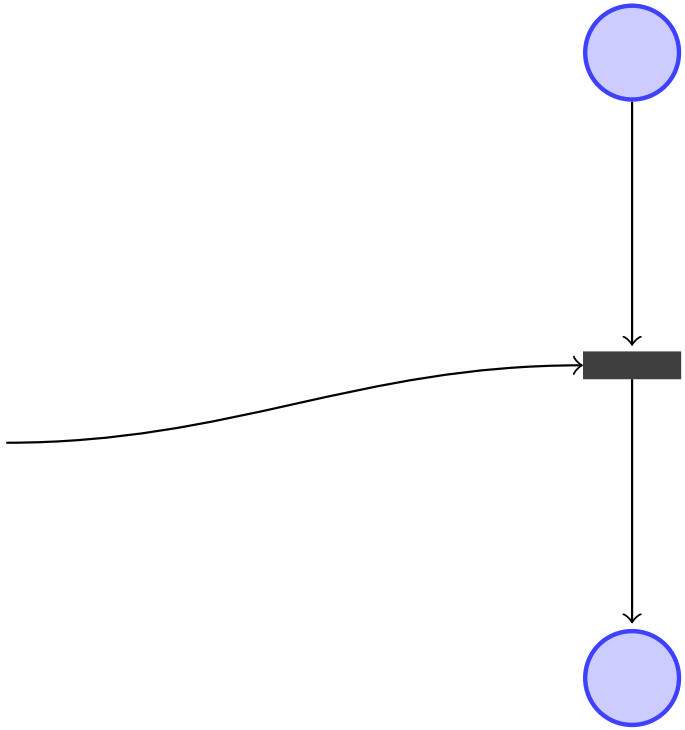
- Places
- **Arcs**



Basic example

A PN consists of:

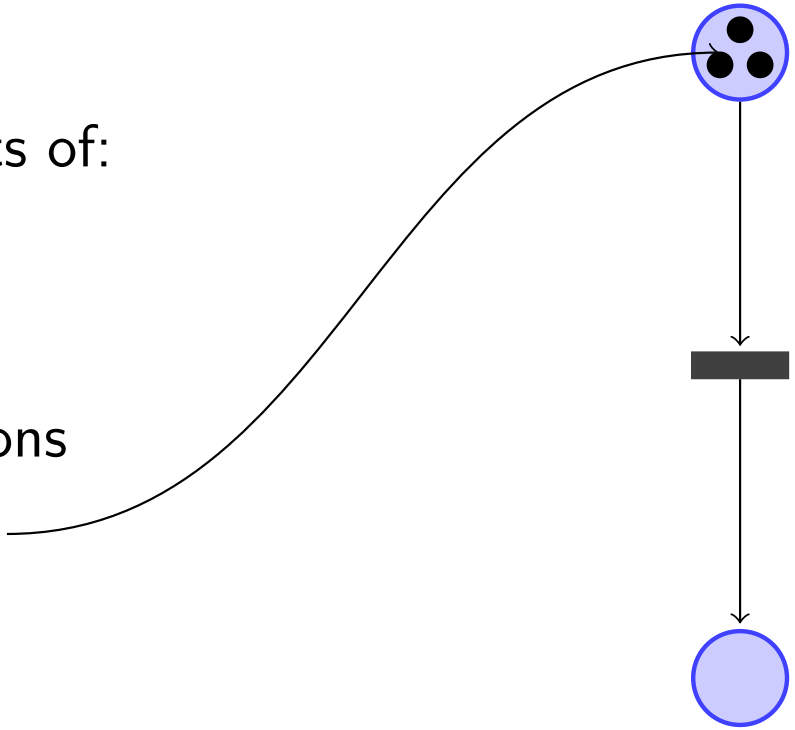
- Places
- Arcs
- **Transitions**



Basic example

A PN consists of:

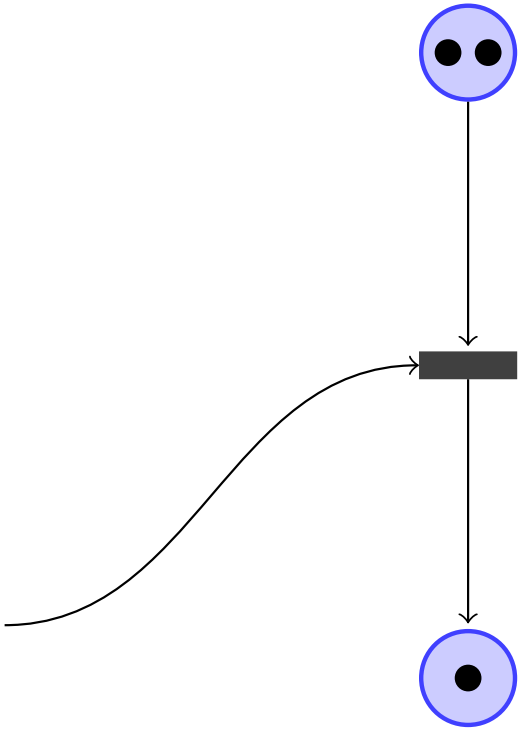
- Places
- Arcs
- Transitions
- **Tokens**



Basic example

A PN consists of:

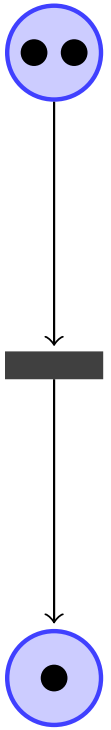
- Places
- Arcs
- Transitions
- Tokens
- **A transition fires**



Basic example

A PN consists of:

- Places
- Arcs
- Transitions
- Tokens
- A transition *fires*



Tokens are moved!

Transitions in SPNs

- All transitions t_i are associated with a *rate* λ_i, possibly marking dependent.
- A transition becomes *enabled* when none of its *input* places is empty.
- When a transition t_i becomes enabled, an exponentially distributed random variate with parameter λ_i is generated.
- When a transitions fires, one token is removed from each input place, and tokens are put into all of its *output* places.
- Each allocation of tokens to the places in the SPN is called a *marking*.

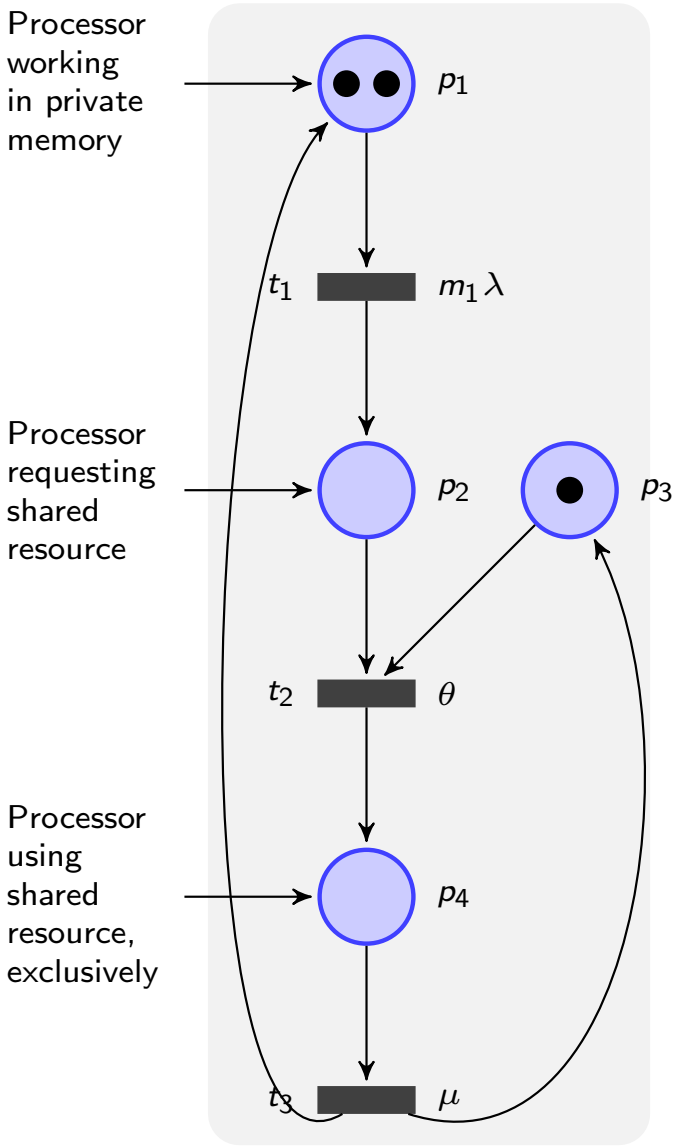
Notation

- M_i - some marking, M_0 - Initial marking.
- t_i - Transition i.
- p_i - Place i.
- m_i - Number of tokens in place i.
- $E(M_i)$ - transitions enabled by marking M_i.
- $G(t_i)$ - the set of markings in which t_i is enabled.

Example

A place can represent some logical state of the system that we want to model.

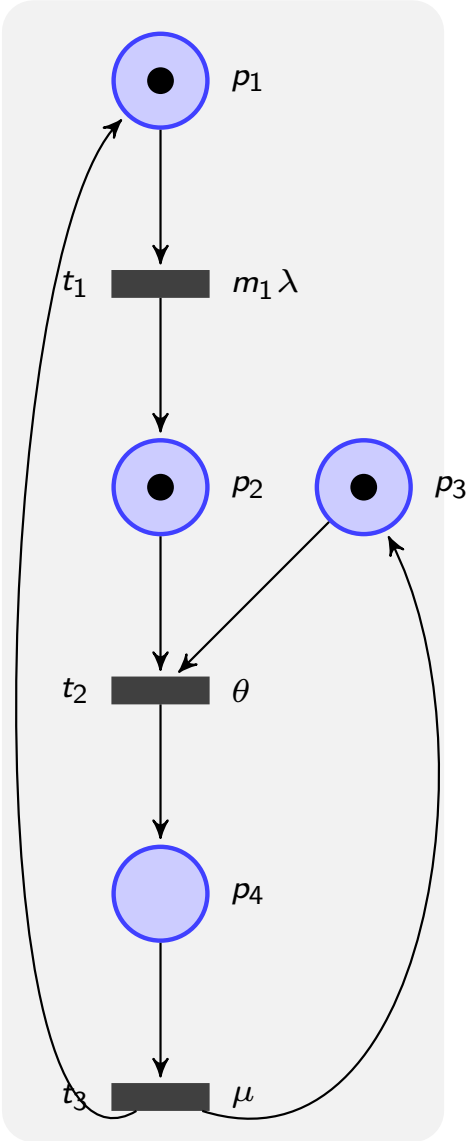
- Initial marking
  $M_0 = \langle 2, 0, 1, 0 \rangle$



Example

A place can represent some logical state of the system that we want to model.

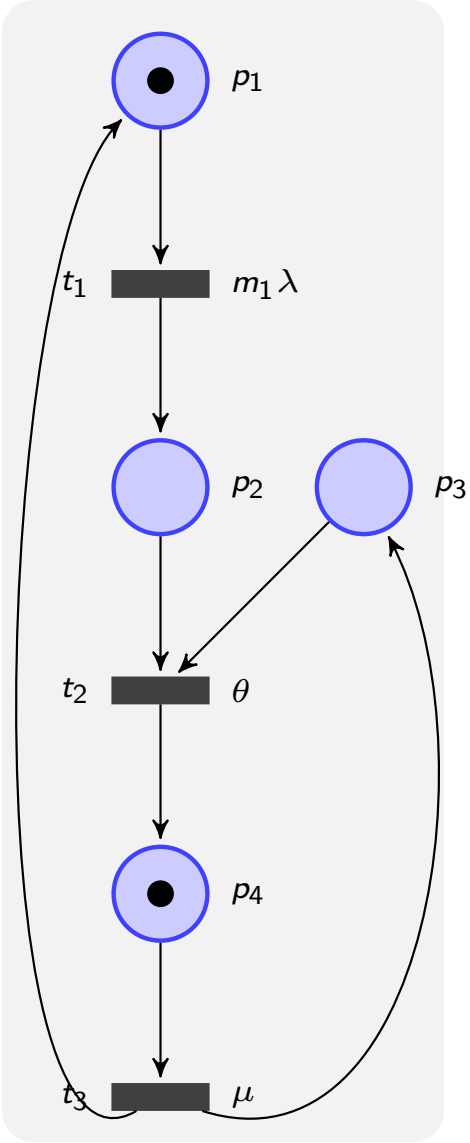
- Initial marking $M_0 = \langle 2, 0, 1, 0 \rangle$
- $M_1 = \langle 1, 1, 1, 0 \rangle$



Example

A place can represent some logical state of the system that we want to model.

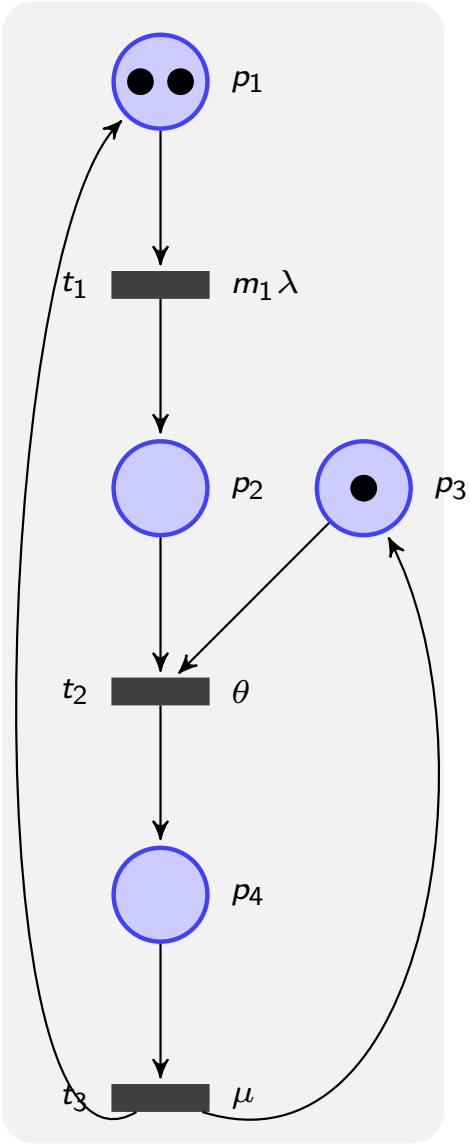
- Initial marking $M_0 = \langle 2, 0, 1, 0 \rangle$
- $M_1 = \langle 1, 1, 1, 0 \rangle$
- $M_2 = \langle 1, 0, 0, 1 \rangle$



Example

A place can represent some logical state of the system that we want to model.

- Initial marking $M_0 = \langle 2, 0, 1, 0 \rangle$
- $M_1 = \langle 1, 1, 1, 0 \rangle$
- $M_2 = \langle 1, 0, 0, 1 \rangle$
- $M_0 = \langle 2, 0, 1, 0 \rangle$



How to solve it

SPN - two approaches to get performance measures:

- Analytical - generate a corresponding CTMC and solve it.
- Simulation - When the state space is too large, or when a more advanced model is necessary (general firing distributions).

SPNs and CTMCs

Fact: SPNs are isomorphic to CTMCs, k-bounded SPNs are isomorphic to finite CTMCs.

SPNs and CTMCs

Fact: SPNs are isomorphic to CTMCs, k-bounded SPNs are isomorphic to finite CTMCs.

- Reachability set $RS = RS(M_0)$ - the markings reachable starting from the initial marking M_0.
- The state space of the CTMC (the marking process) corresponds to the RS.

SPNs and CTMCs

Fact: SPNs are isomorphic to CTMCs, k-bounded SPNs are isomorphic to finite CTMCs.

- Reachability set $RS = RS(M_0)$ - the markings reachable starting from the initial marking M_0.
- The state space of the CTMC (the marking process) corresponds to the RS.
- Home state: $\forall M' \in RS(M_0), M \in RS(M')$
- A k-bounded SPN is said to be ergodic if M_0 is a home state.
- ergodic SPN $\rightarrow$ ergodic CTMC $\rightarrow$ guaranteed steady state solution

Obtaining the CTMC

Convert the SPN to a CTMC, where every marking of the SPN can be associated with a state in the corresponding Markov chain.
We get the generator matrix $\mathbf{Q}$ by:

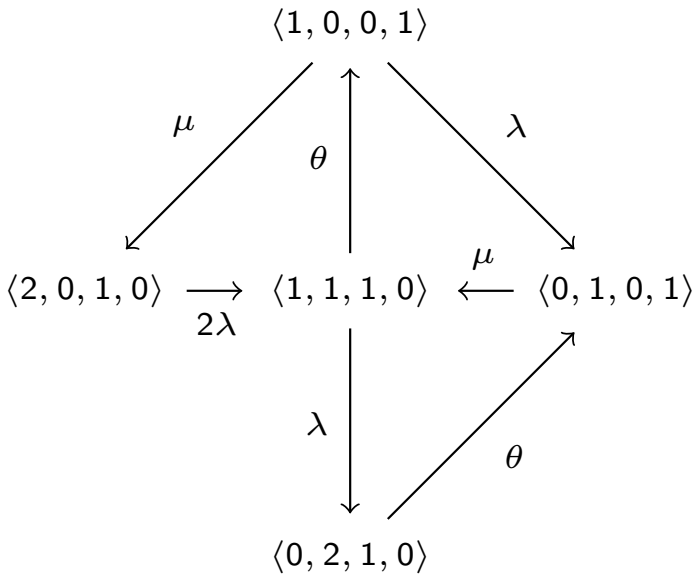
$$q_{ij} = \begin{cases} \sum_{t_k \in E_j(M_i)} \lambda_k(M_i) & i \neq j \\ -\sum_{t_k \in E(M_i)} \lambda_k(M_i) & i = j \end{cases}$$

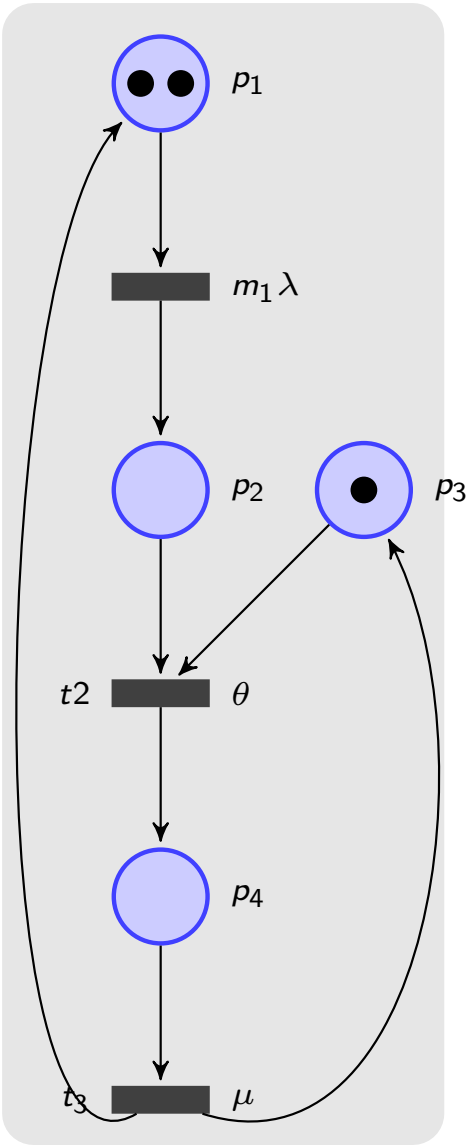
NB. Mean *sojourn* time in any marking (why?):

$$ST_i = \left(\sum_{t_k \in E_j(M_i)} \lambda_k(M_i) \right)^{-1}$$

Obtaining the CTMC

Reminder : each marking is written as $M = \langle m_1, m_2, m_3, m_4 \rangle$.





GSPN

We extend the original SPN model with:

- *Immediate* transitions – transitions that fire immediately, before any timed transitions. (higher priority)
- *Inhibitor* arcs. (will be skipped)

Motivation:

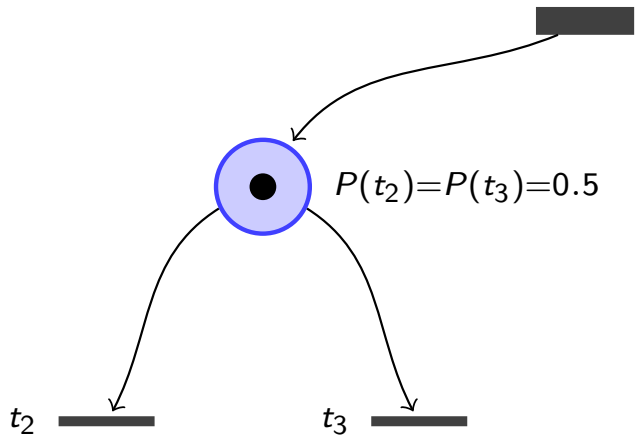
- can be used to model purely logical events or events whose duration is negligible.
- the state space (RS) of the resulting CTMC is reduced (when using an immediate transition instead of a timed transition).
- We increase modeling power.

GSPN, ctd.

- We will call markings $m_i \in G(t_j)$, where t_j is immediate, for *vanishing* states, and we differ those from the *tangible* states (i.e., when the transition is timed).
- The marking process will no longer be a CTMC, but a Semi-Markov process. (The times between transitions have two different distributions)
- It is straightforward to find an EMC, from which it is possible to derive a *reduced* EMC, with a smaller state space.

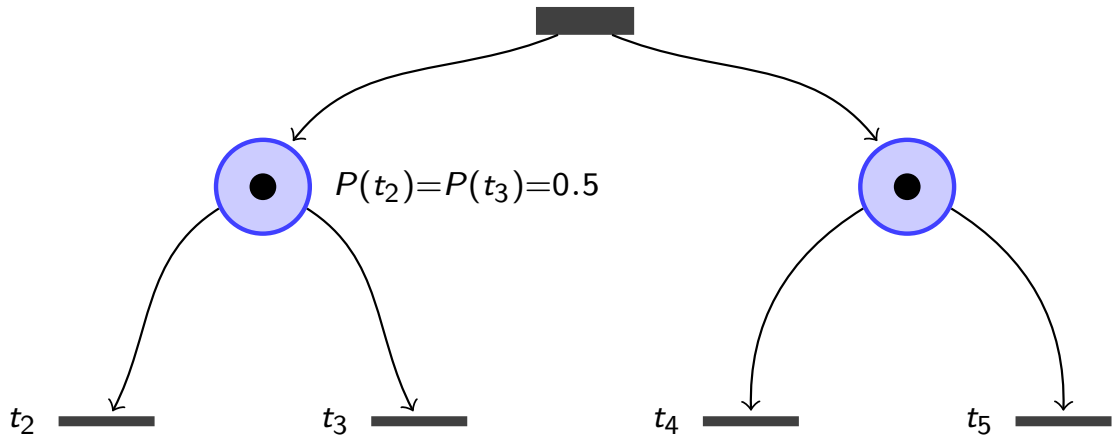
Switching distributions

Whenever two immediate transitions are enabled at the same time, we require some rule that specifies which transition should fire first (possibly marking dependent).



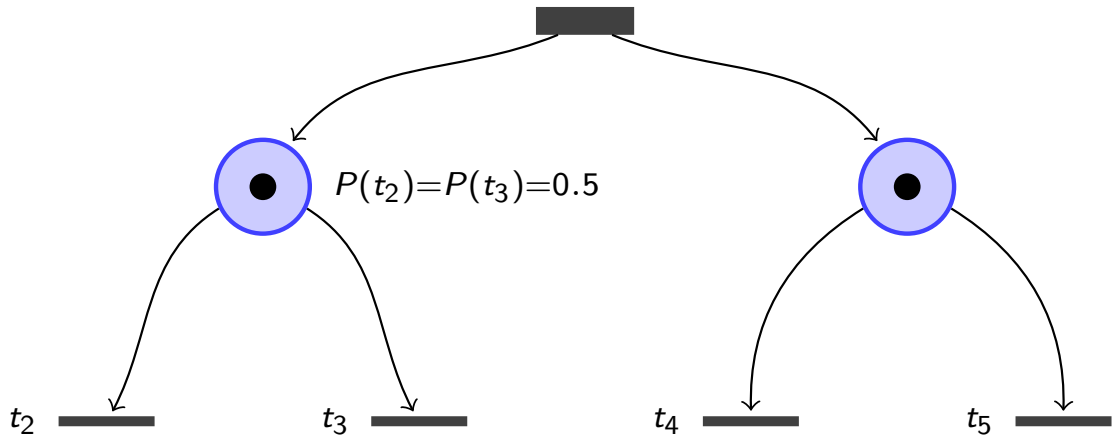
Switching distributions

However, there are some subtleties when defining the switching distributions of the immediate transitions:



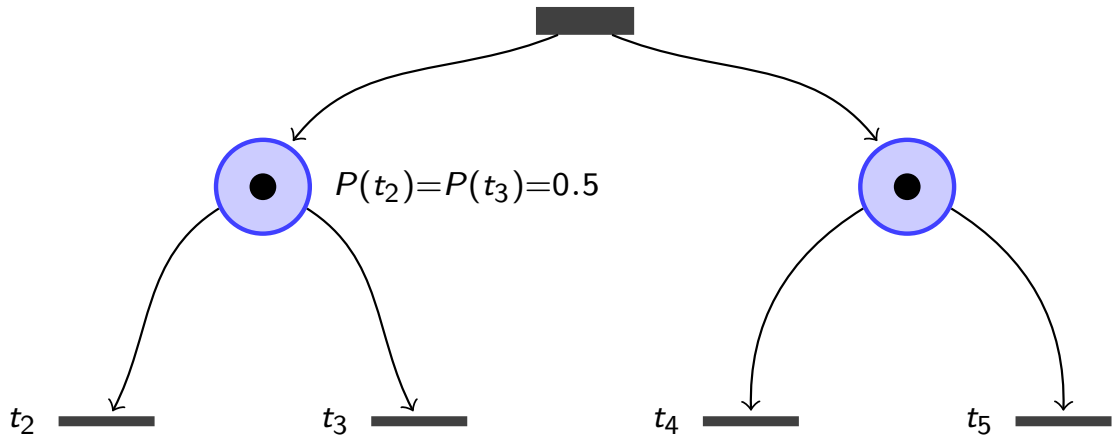
Switching distributions

The probability measure is only well-defined if $P(t_4) = P(t_5) = 0$, which is probably not what we want.



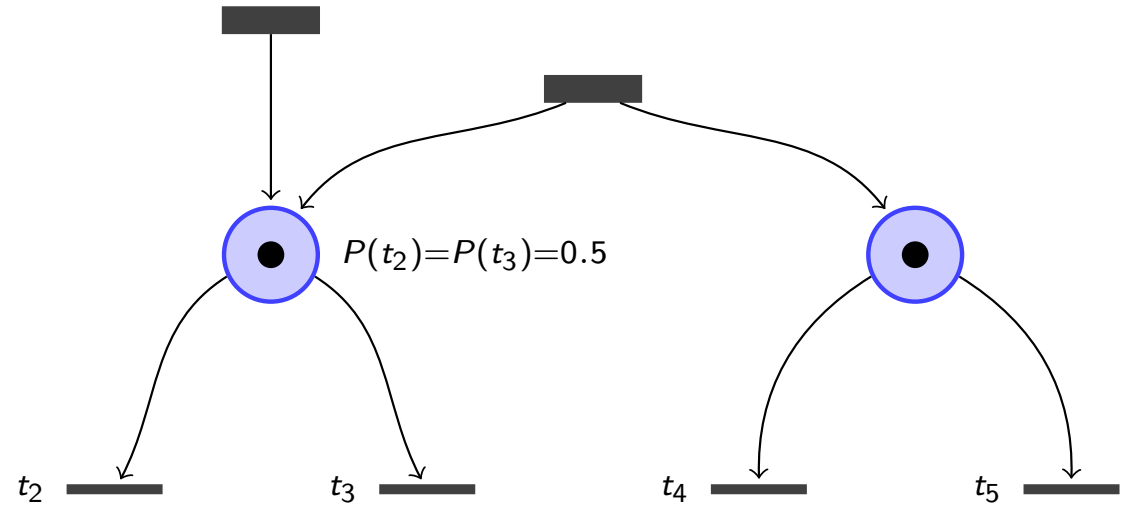
Switching distributions

Hence, knowledge of the marking process is necessary when defining switches.



Switching distributions

Even worse! Solution: define weights instead (as Bengt said).



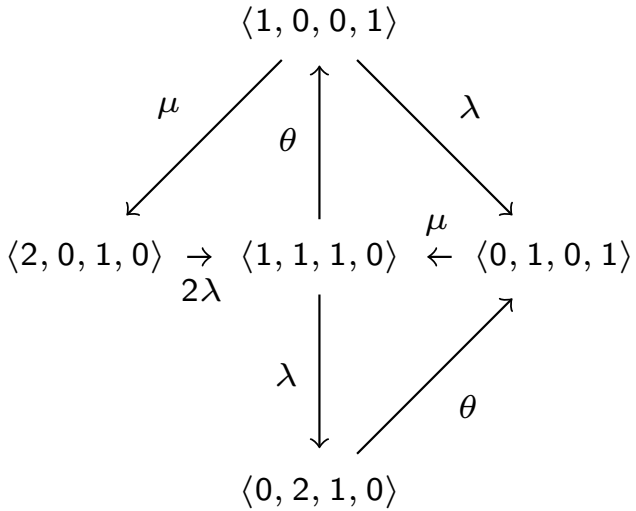
Embedded MC

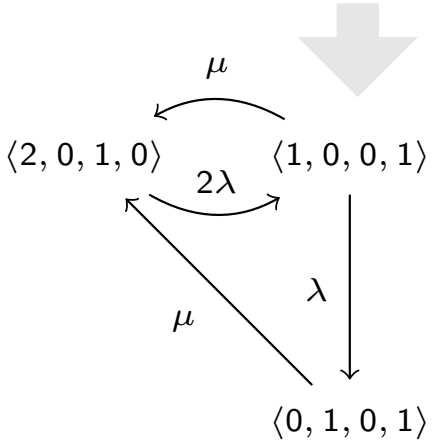
- EMC: Observe a stochastic process $\{X(t), t \in T\}$ only at the moment the state changes.
- Given a GSPN, the obtained process will be a DTMC, with transition matrix

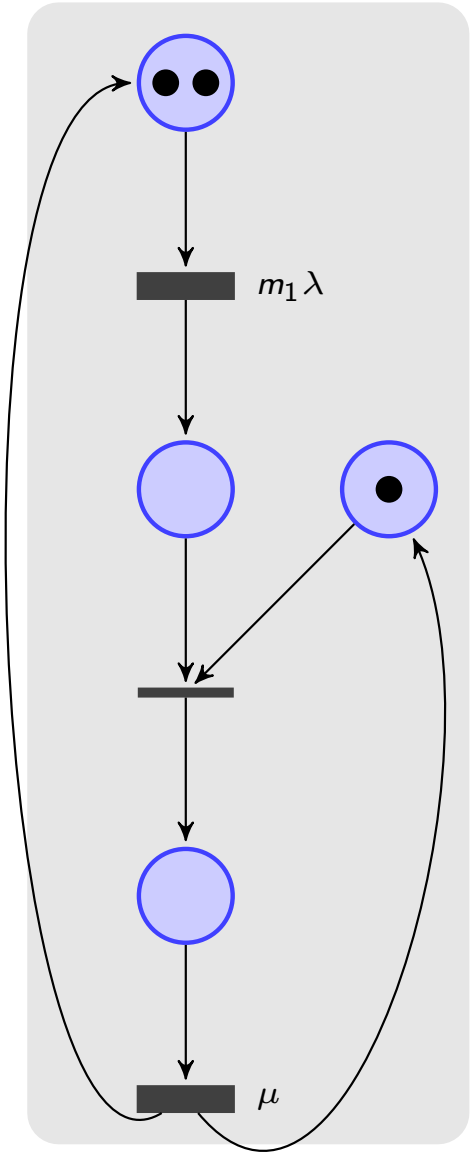
$$u_{ij} = \begin{cases} \sum_{t_k \in E_j(M_i)} \lambda_k / \sum_{t_k \in E(M_i)} \lambda_k & i \neq j \\ 0 & i = j \end{cases}$$

- From the routing probabilities that we get from the EMC and the switching probabilities, the average sojourn time in each state of the marking process, we can find the steady-state probability of the marking process.

Example, revisited.







Performance measures

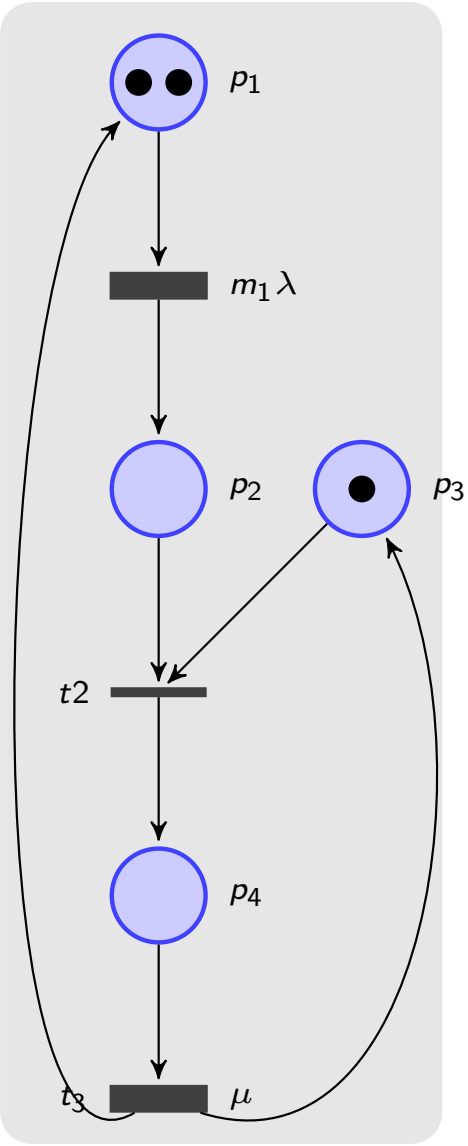
Probability of a certain place/event can be obtained by summing up the probabilites of the corresponding markings:

$$P(p_i) = \sum_{M_i \in S(p_i)} \pi_i$$

where π_i is the steady-state distribution of m_i in the underlying CTMC.
The pmf of the number of tokens in a place p_i gives us the average number of tokens in that place.

Example, ctd.

- Assume $M_0 = \langle 2, 0, 1, 0 \rangle$, $M_1 = \langle 1, 0, 0, 1 \rangle$ and $M_2 = \langle 0, 1, 0, 1 \rangle$.
- We solve the system, and get $P(M_0)$, $P(M_1)$, $P(M_2)$.
- We interpret the places in some way: $P(\text{waiting}) = P(m_2 > 0) = P(M_2)$
- Average number of tokens in m_2, $E(m_2) = 0P(M_0) + 0P(M_1) + P(M_2)$



Throughput and delay

- The *throughput*, or the mean number of firings of a transition t_j :

$$f_j = \sum_{M_i \in G(t_j)} \lambda_j(M_i)\pi_i$$

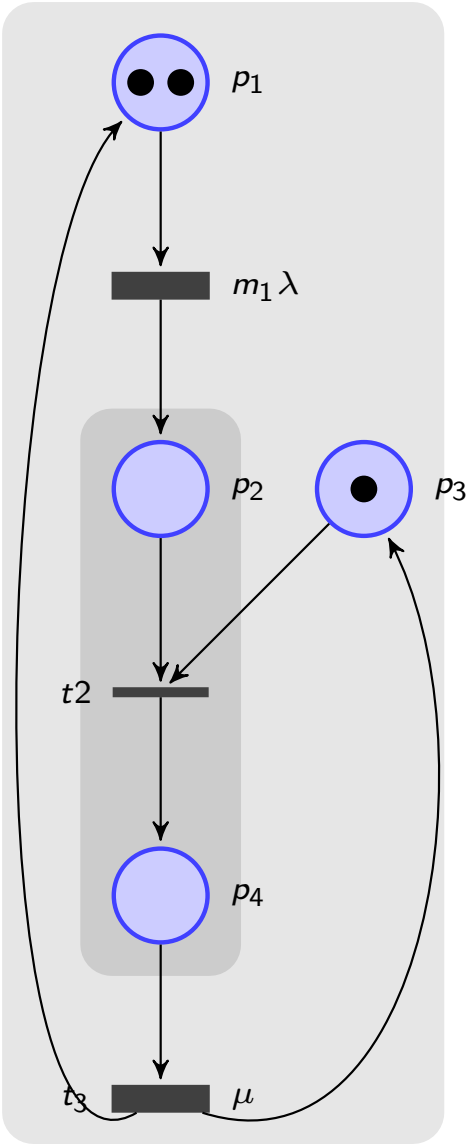
- Little's law gives the average delay of a token passing through a subnet of the model:

$$E(T) = \frac{E(N)}{E(\gamma)}$$

Throughput and delay

- We solve the system, and get $P(M_0)$, $P(M_1)$, $P(M_2)$.
- Rate into the subnet (throughput of transition t_1):
  $E(\gamma) = 2\lambda P(M_0) + \lambda P(M_1)$
- $E(N) = E(m_4) + E(m_2) =$
  $P(M_1) + P(M_2) + P(M_2)$
- Average delay of one token:

$$E(T) = \frac{E(N)}{E(\gamma)} = \frac{P(M_1) + 2P(M_2)}{2\lambda P(M_0) + \lambda P(M_1)}$$



Relation with PN

- All structural results obtained from the underlying PN are valid for the SPN as well.
- Completely deterministic
- **PN**: Adding inhibitor arcs gives them the same modeling power as turing machines (they can "mimic" any turing machine)
  GSPN: Adding inhibitor arcs does not increase the modeling power. (Haas). In fact, any GSPN with one or more inhibitor arc can be converted into an equivalent GSPN without inhibitor arcs. (immediate transitions are used)