Probabilistic automata = compositional version of MDPs

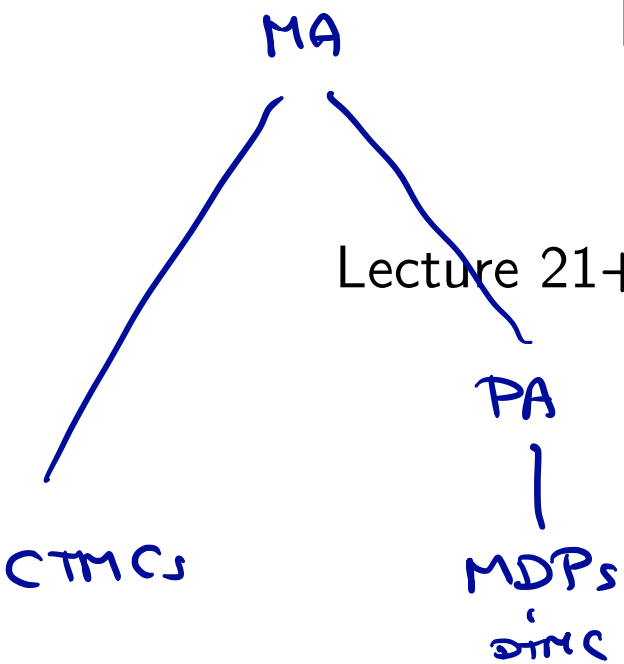
Markov Automata

similar to prob. automata

⇒ compositional modeling of CTMCs

Joost-Pieter Katoen

RWTH AACHEN UNIVERSITY



Lecture 21+22 course MVPS 2018/19

Overview

Introduction

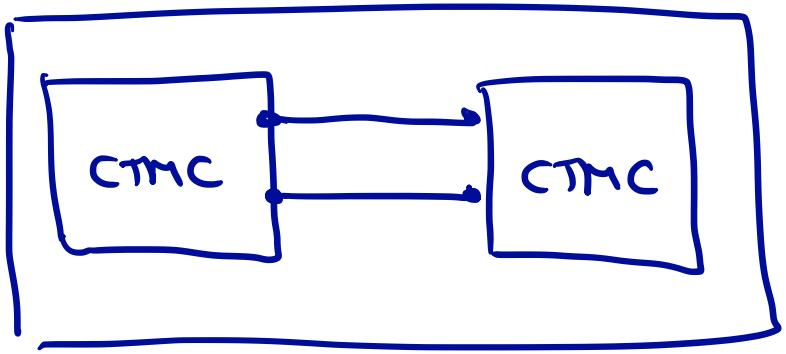
Beautiful theory
- What are Markov Automata?
- Concurrent composition and hiding
- Bisimulation
- Analysis algorithms

The usage for high-level modeling languages
- Process algebra
- Generalized Stochastic Petri Nets

Today: Markov Automata

Aim:

- compositional modelling of CTMCs



(simple)

A process algebra for CTMCs.

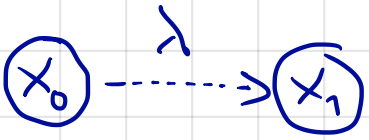
$\lambda_i \in \mathbb{R}_{>0}$

$$T ::= 0 \mid \sum_{i \in I} (\lambda_i).T_i \mid X$$

- 0: (deadlock)
- $\sum_{i \in I}$: finite index set
- X: recursion variable

Syntax

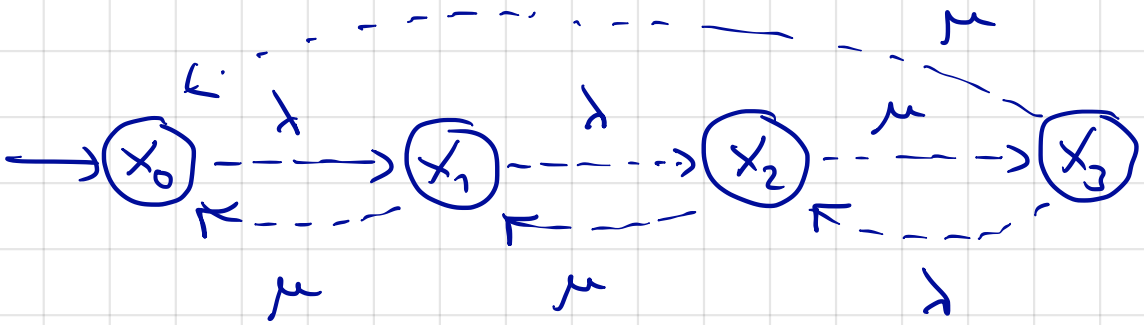
example $X_0 = (\lambda).X_1$



$$X_1 = (\mu).X_0 + (\lambda).X_2$$

$$X_2 = (\mu).X_1 + (\mu).X_3$$

$$X_3 = (\mu).X_0 + (\lambda).X_2$$



Semantics

$$\frac{T \xrightarrow{\lambda} T'}{X \xrightarrow{\lambda} T'} \quad (X = T)$$

$$\sum_{i \in I} (\lambda_i).T_i \xrightarrow{\lambda} T$$

where $\lambda = \sum_{\substack{i \in I \\ T_i = T}} \lambda_i$

$$\underbrace{(\lambda_1).0}_{1} + \underbrace{(\lambda_2).0}_{2} \xrightarrow{\lambda_1 + \lambda_2} 0$$

$|I| = 2 \qquad T_1 = 0 = T_2$

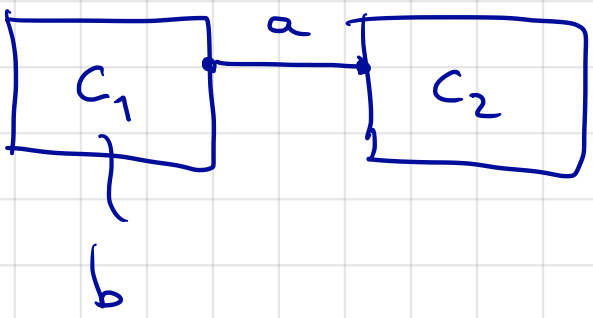
This syntax is pretty restricted: there is no parallel composition.

Q: How to incorporate parallel composition?

for prob. automata:

$||_A$ — synchronise on actions in A

actions $\notin A$ are autonomously executed



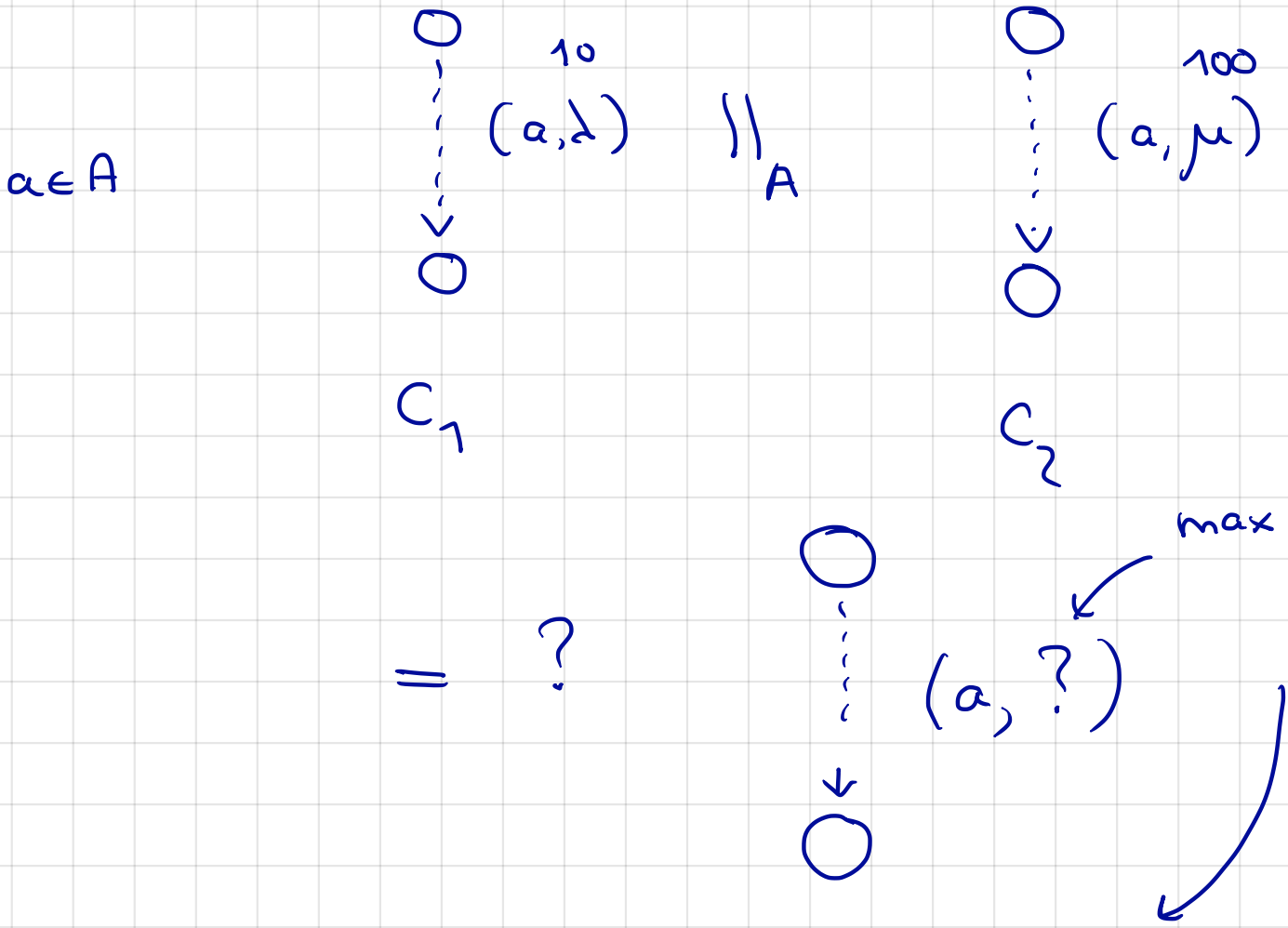
$a \in A$

Q: How to incorporate actions into CTMCs?

Naive solution

pairs of actions + rates (a, λ)

parallel composition



this is not an exponential distribution, since the maximum of two exp. distributions is not an exp. distribution

⇒ reject the naive solution.

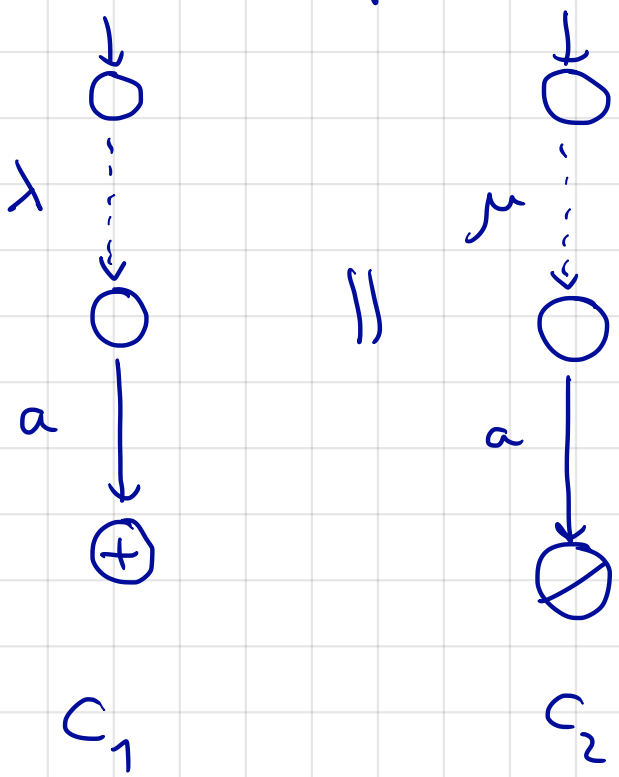
Solution:

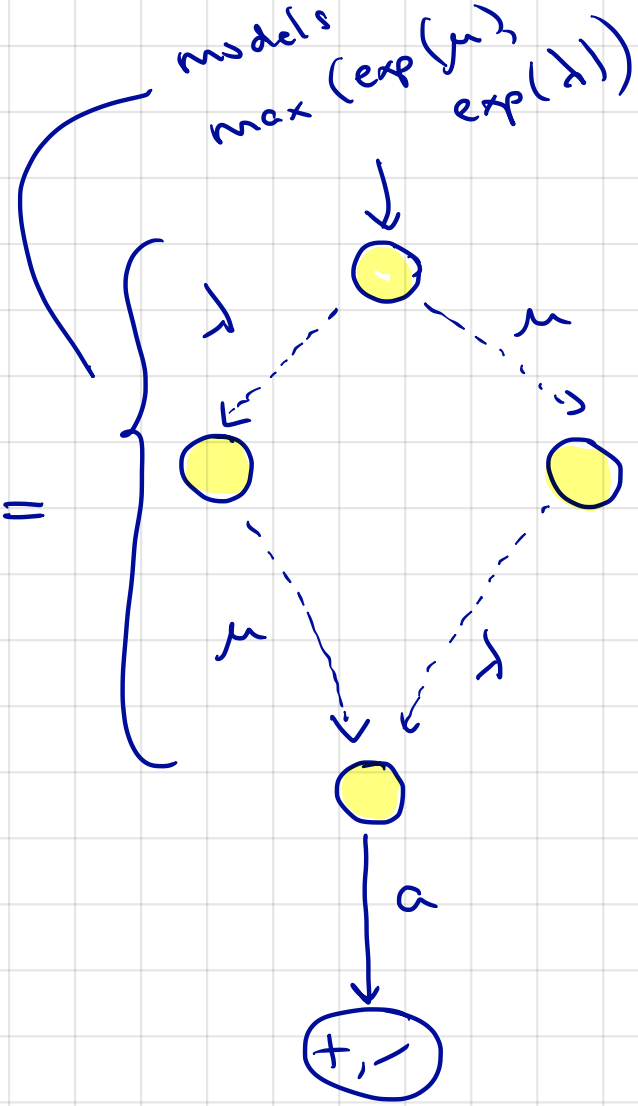
- separate actions

  and

- delays

parallel composition:





$X \sim \exp(\lambda)$

$Y \sim \exp(\mu)$

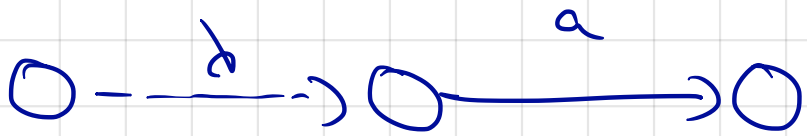
$Z = \max(X, Y)$

$$Pr(Z \leq t) = Pr(\max(X,Y) \leq t)$$

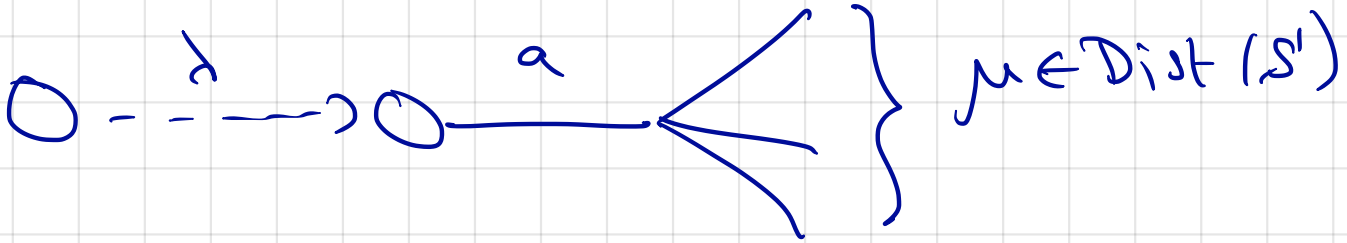
$$= 1 - Pr(\max(X,Y) > t)$$

$$1 - e^{\cdots \lambda} + \cdots .$$

rather than



we use



Today: Markov Automata

The *beauty* of its theory

- The simplicity of the model
- Parallel composition
- Bisimulation
- Quantitative analysis

The *usage* for modeling languages

1. Process algebra
2. Stochastic Petri Nets
3. ..... not today .....
4. Architectural Analysis & Design Language
5. Dynamic Fault Trees
6. Scenario-Aware Dataflow

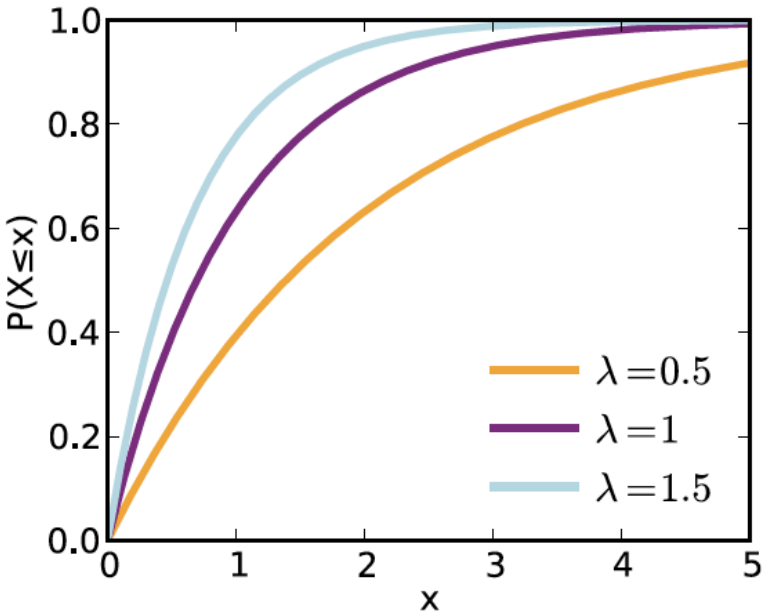
Overview

Introduction

Beautiful theory
- What are Markov Automata?
- Concurrent composition and hiding
- Bisimulation
- Analysis algorithms

The usage for high-level modeling languages
- Process algebra
- Generalized Stochastic Petri Nets

Exponential distributions



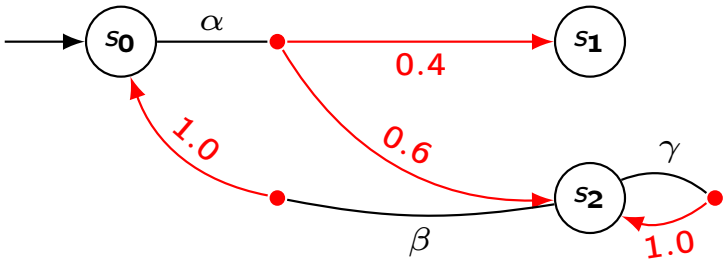
- The cdf of exponentially distributed r.v. X with rate $\lambda \in \mathbb{R}_{>0}$ is:

$$F_X(x) = 1 - e^{-\lambda \cdot x}$$

- The rate λ uniquely determines F_X
- The higher λ, the faster F_X approaches 1
- Unique memoryless continuous distribution
- Expectation = λ^{-1}

A marriage

 and 



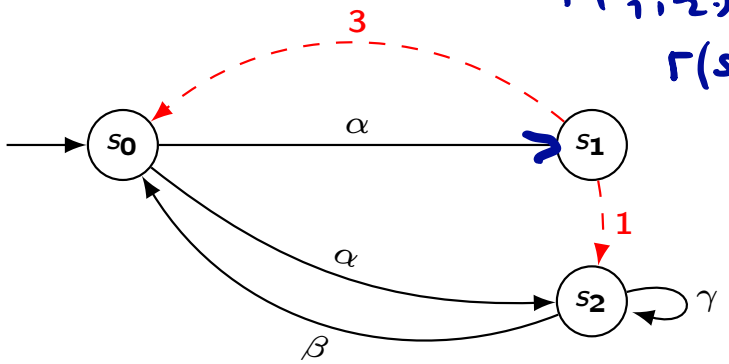
Segala's probabilistic automata

Key: a transition yields a distribution over states

$P(s_1, s_0) = \frac{3}{4}$

$P(s_1, s_2) = \frac{1}{4}$

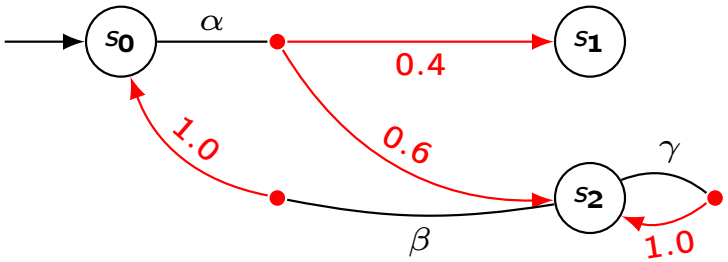
$r(s_1) = 4$



Hermanns' interactive Markov chains

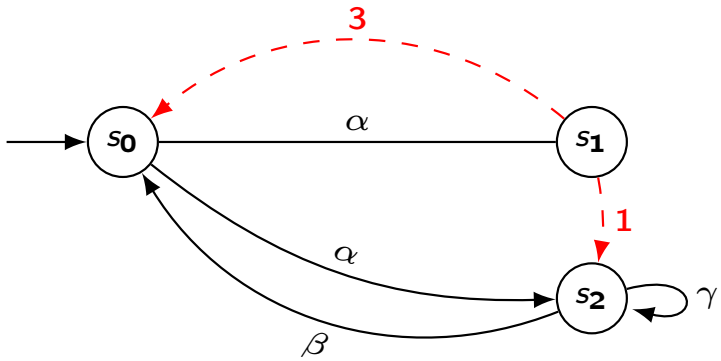
A marriage

 and 



Segala's probabilistic automata

Key: a transition yields a distribution over states

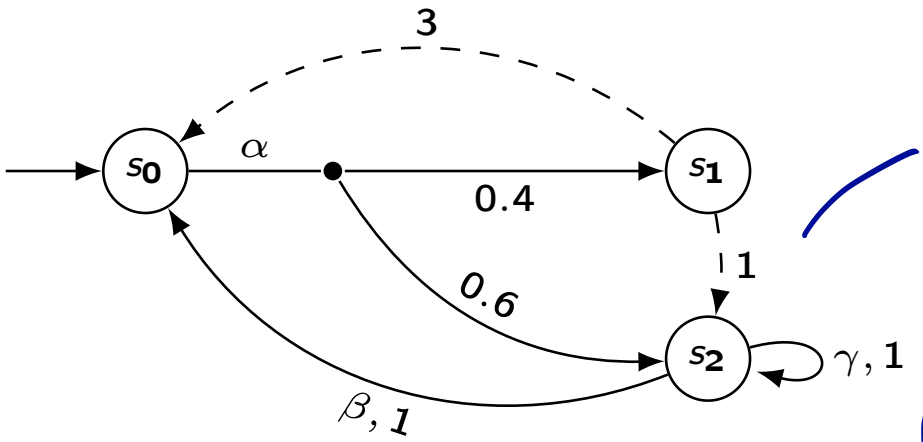


Hermanns' interactive Markov chains

Key: separated action and delay transitions

Markov automata

[Eisentraut *et al*, 2010]



$s_0 \xrightarrow{\alpha} \mu$

$\mu(s_1) = \frac{2}{5}$

$\mu(s_2) = \frac{3}{5}$

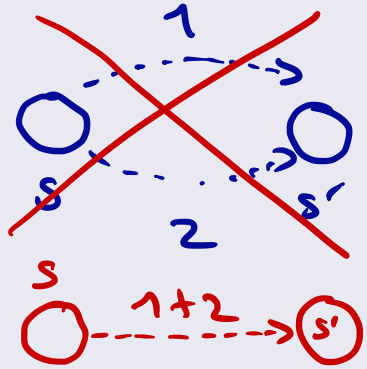
$s_1 \overset{1}{\dashrightarrow} s_2$

cannot synchronise !

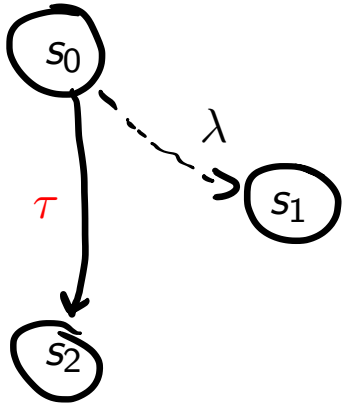
A Markov automaton M is a tuple $(S, Act, \rightarrow, \dashrightarrow, s_0)$ where

- S is a nonempty set of states with initial state $s_0 \in S$
- Act is a set of actions; τ is an internal action
- $\rightarrow \subseteq S \times Act \times Dist(S)$ is a set of action transitions
- $\dashrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ is a set of Markovian transitions

  such that there is at most one $r \in \mathbb{R}_{>0}$ with $s \overset{r}{\dashrightarrow} s'$



Maximal progress assumption



either do τ immediately

or move to s_2 after delay

$\exp(\lambda)$

Maximal progress assumption

$$Pr(\exp(\lambda) \leq 0) = 0$$

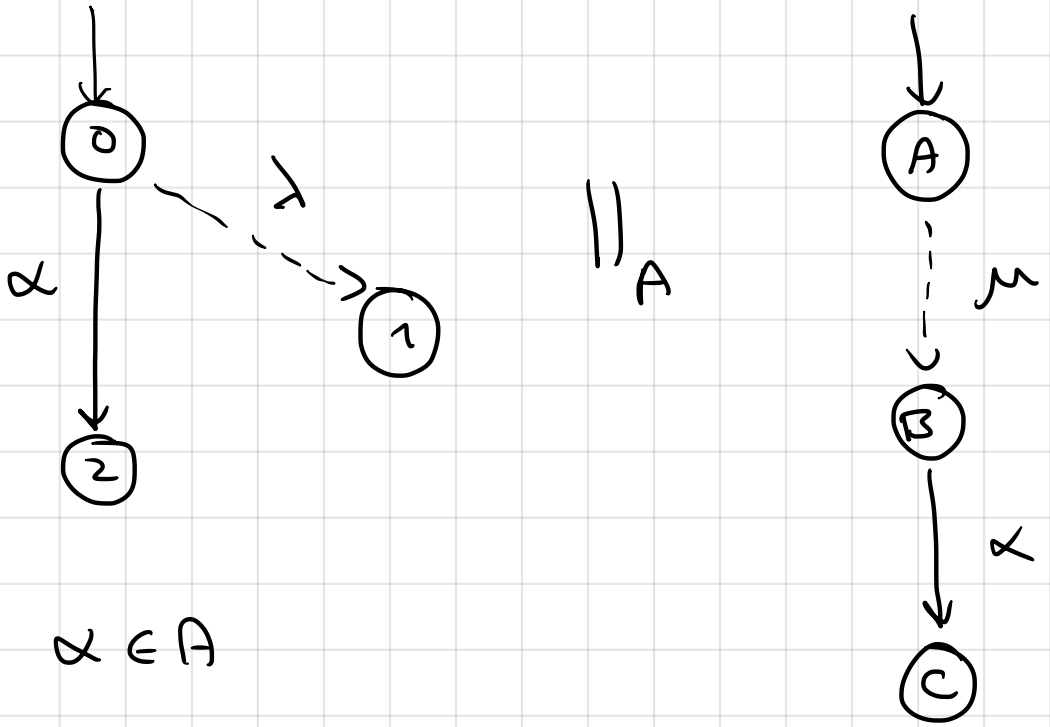
Maximal progress assumption

But as visible actions may be subject to delaying by other components:

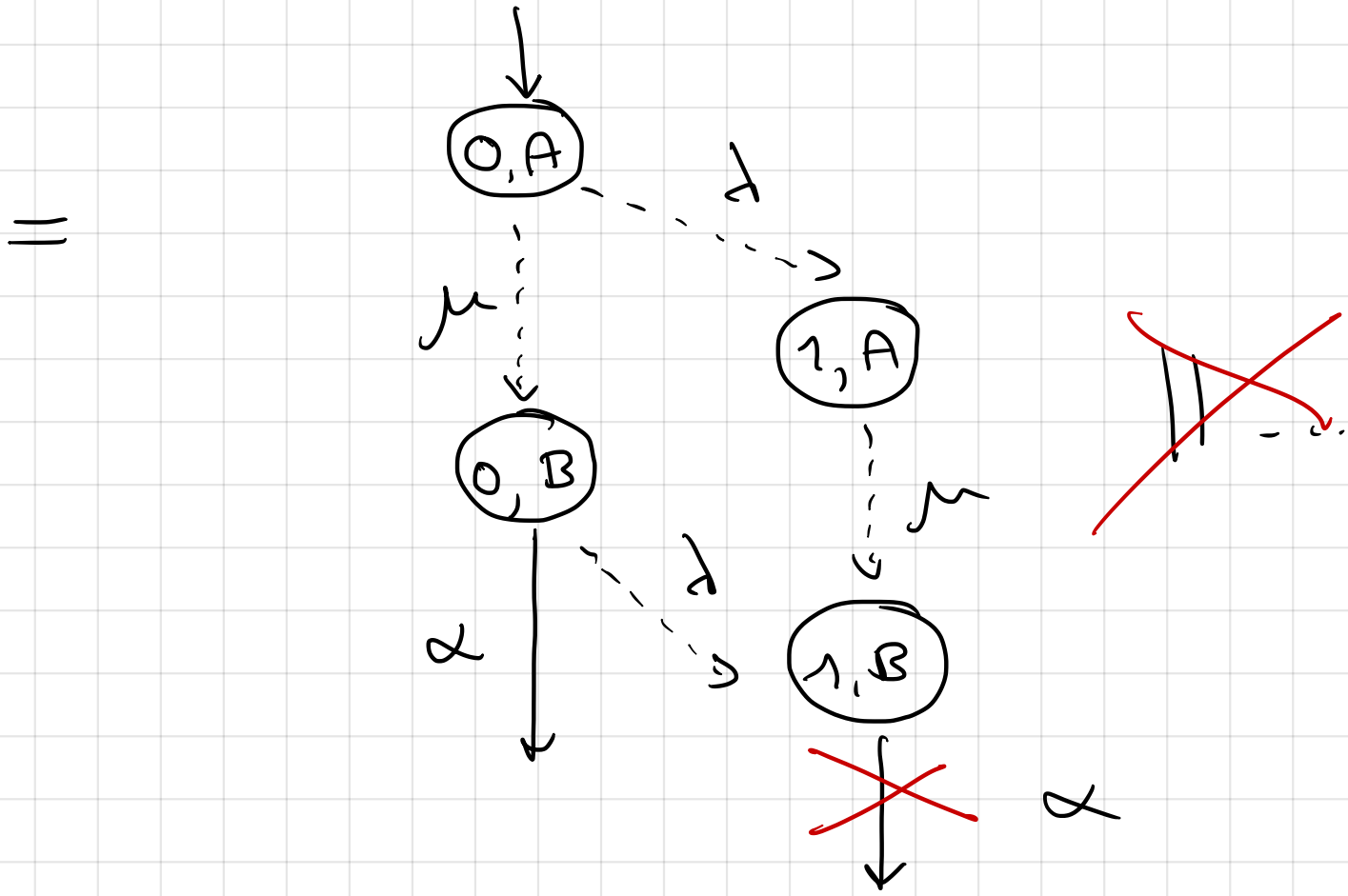
any $\alpha \in Act$, $\alpha \neq \tau$

Maximal progress assumption

But as visible actions may be subject to delaying by other components:


0
1
2
λ
α
α ∈ A
||A
A
B
C
μ
α

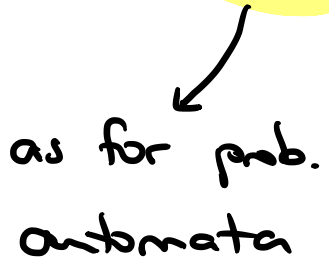
=


0,A
1,A
0,B
1,B
λ
μ
λ
μ
α
α

Concurrent composition

The *composition* of M_1 and M_2 wrt. $A = (Act_1 \cap Act_2) \setminus \{\tau\}$ is:

$$M_1 \,||\, M_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \dashrightarrow, (s_{0,1}, s_{0,2}))$$



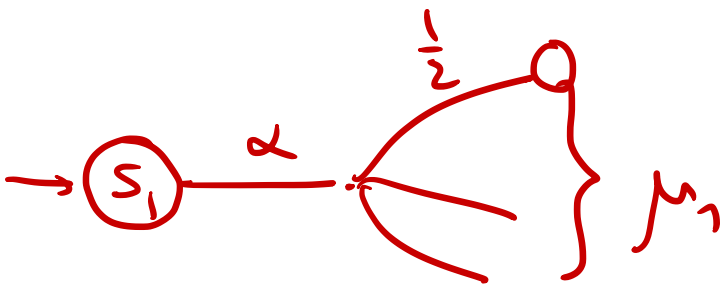
Concurrent composition

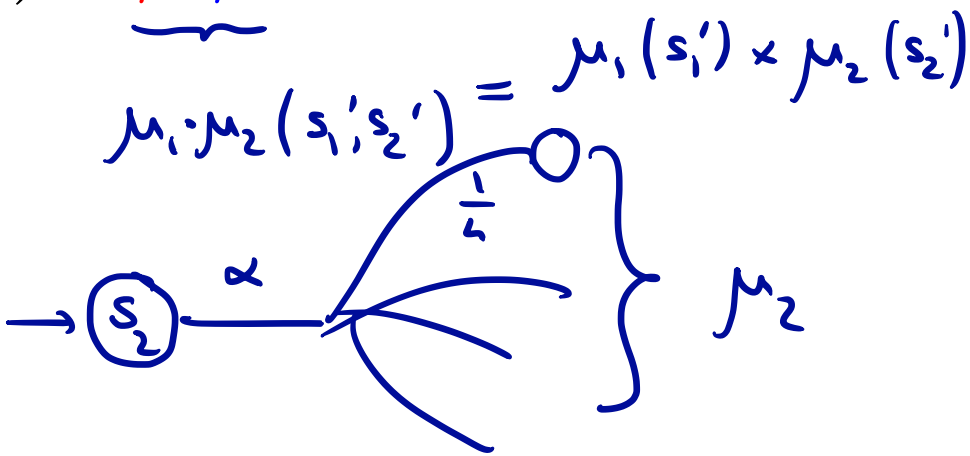
The *composition* of M_1 and M_2 wrt. $A = (Act_1 \cap Act_2) \setminus \{\tau\}$ is:

$$M_1 \parallel_A M_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \dashrightarrow, (s_{0,1}, s_{0,2}))$$

where $\rightarrow$ and $\dashrightarrow$ are defined as the smallest relations satisfying:

$$\text{(SYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } s_2 \xrightarrow{\alpha}_2 \mu_2 \text{ and } \alpha \in A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \mu_2}$$





Concurrent composition

The *composition* of M_1 and M_2 wrt. $A = (Act_1 \cap Act_2) \setminus \{\tau\}$ is:

$$M_1 \,||\, M_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \dashrightarrow, (s_{0,1}, s_{0,2}))$$

where $\rightarrow$ and $\dashrightarrow$ are defined as the smallest relations satisfying:

same as for prob. automata.

$$\text{(SYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } s_2 \xrightarrow{\alpha}_2 \mu_2 \text{ and } \alpha \in A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \mu_2}$$

$$\text{(ASYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } \alpha \notin A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \Delta_{s_2}}$$

$$\frac{s_2 \xrightarrow{\alpha}_2 \mu_2 \quad \alpha \notin A}{(s_1, s_2) \xrightarrow{\alpha} \Delta_{s_1} \cdot \mu_2}$$

$$\Delta_{s_2} = \{ s_2 \mapsto 1 \}$$

$$\Delta_{s_2}(s_2) = 1 \qquad \Delta_{s_2}(s') = 0 \quad s' \neq s_2$$

Concurrent composition

The *composition* of M_1 and M_2 wrt. $A = (Act_1 \cap Act_2) \setminus \{\tau\}$ is:

$$M_1 \,||\, M_2 = (S_1 \times S_2, Act_1 \cup Act_2, \rightarrow, \dashrightarrow, (s_{0,1}, s_{0,2}))$$

where $\rightarrow$ and $\dashrightarrow$ are defined as the smallest relations satisfying:

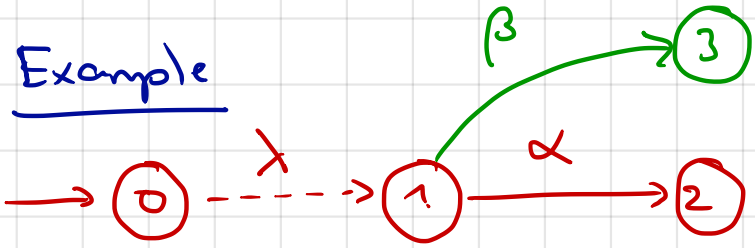
$$\text{(SYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } s_2 \xrightarrow{\alpha}_2 \mu_2 \text{ and } \alpha \in A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \mu_2}$$

$$\frac{s_2 \overset{\lambda}{\dashrightarrow} s_2'}{(s_1, s_2) \overset{\lambda}{\dashrightarrow} (s_1, s_2')}$$

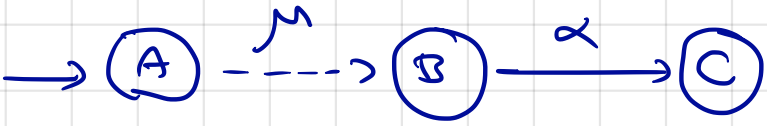
$$\text{(ASYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } \alpha \notin A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \Delta_{s_2}}$$

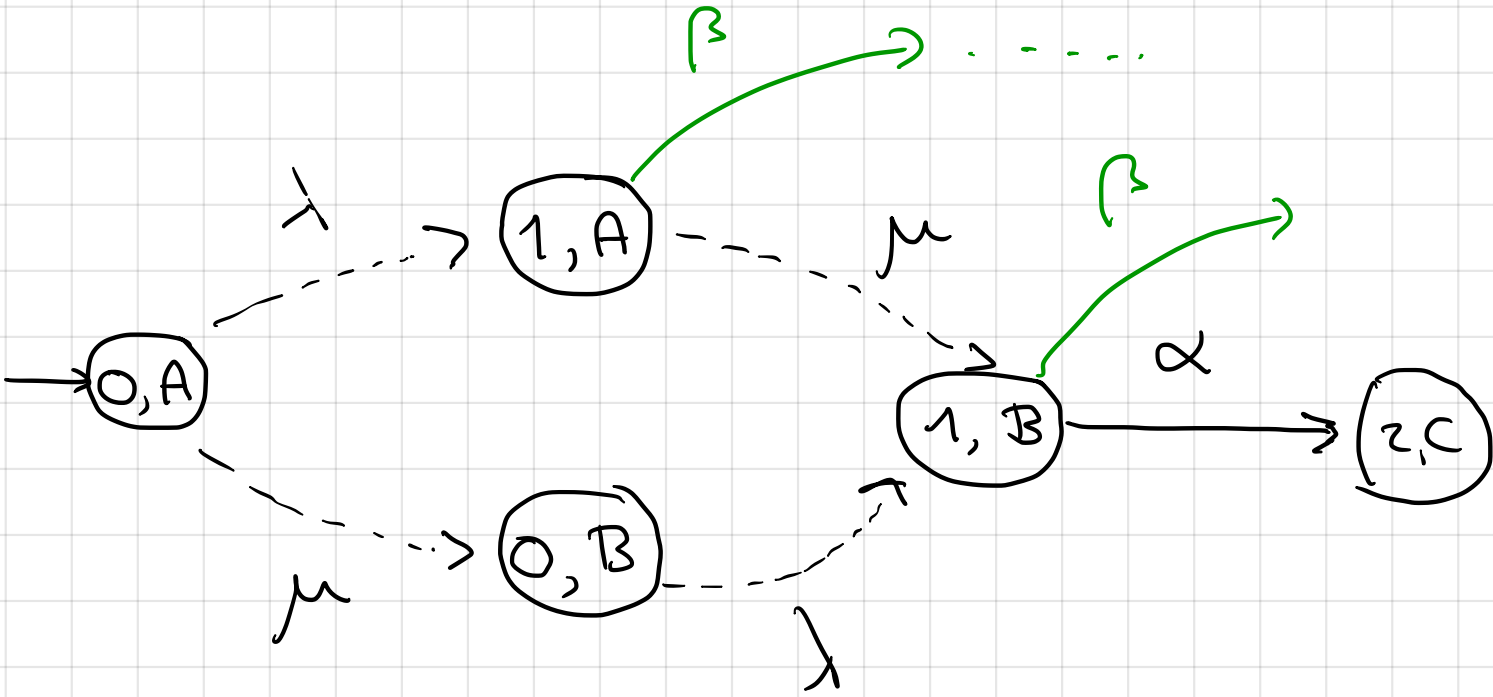
$$\text{(DELAY)}\ \frac{s_1 \overset{\lambda}{\dashrightarrow}_1 s_1'}{(s_1, s_2) \overset{\lambda}{\dashrightarrow} (s_1', s_2)} \quad \text{and} \quad \frac{s_1 \overset{\lambda}{\dashrightarrow}_1 s_1 \text{ and } s_2 \overset{\lambda'}{\dashrightarrow}_2 s_2}{(s_1, s_2) \overset{\lambda+\lambda'}{\dashrightarrow} (s_1, s_2)}$$

Example



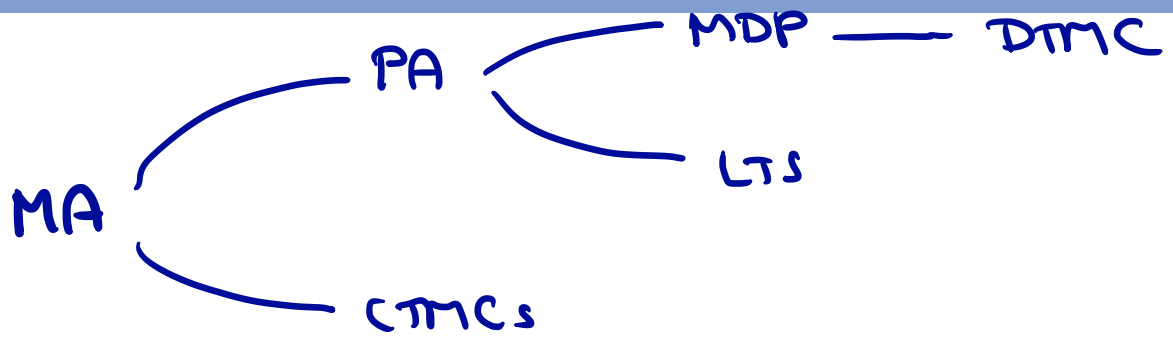
$M_1 \; ||_{\{\alpha\}} \; M_2$





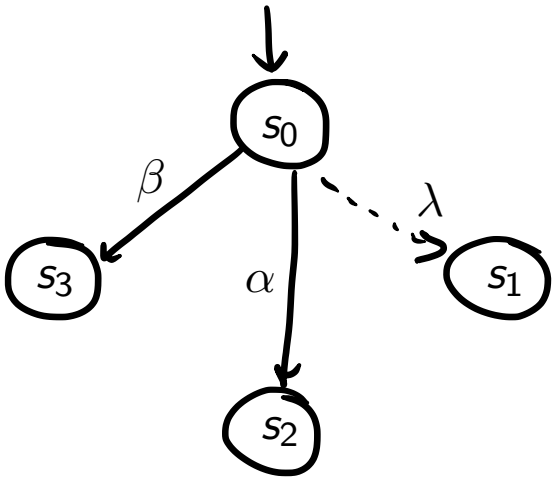
this models $\max(\exp(\mu), \exp(\lambda))$

Compatibility



Parallel composition is backward compatible with parallel composition on probabilistic automata and parallel composition on labeled transition systems.

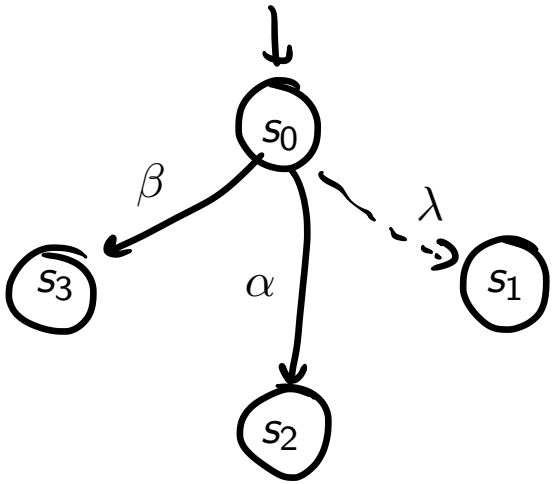
Hiding



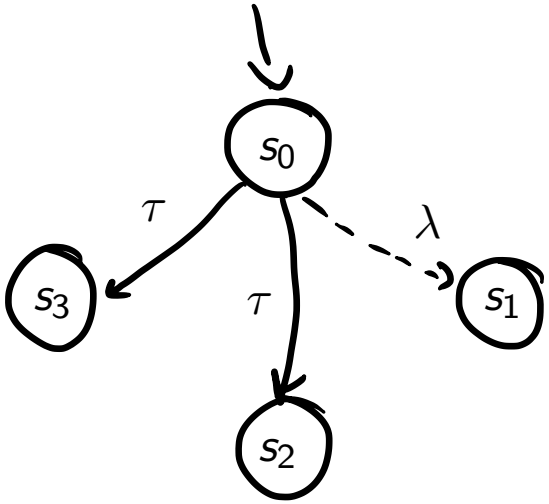
special role of τ

hiding $\{\alpha, \beta\}$

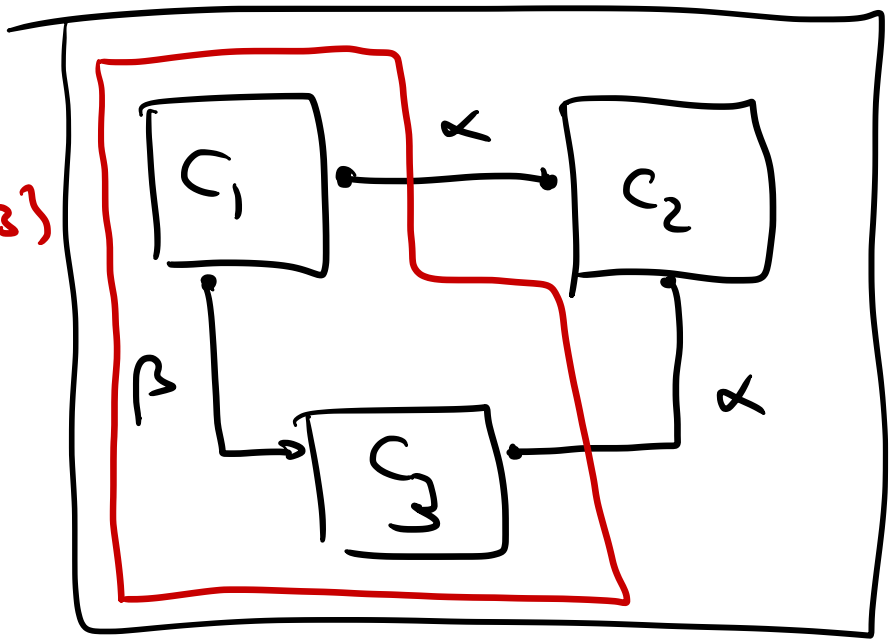
Hiding



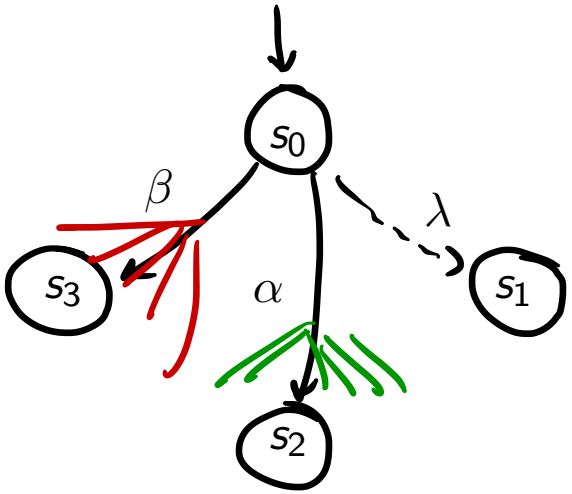
hiding $\{\alpha, \beta\}$ yields



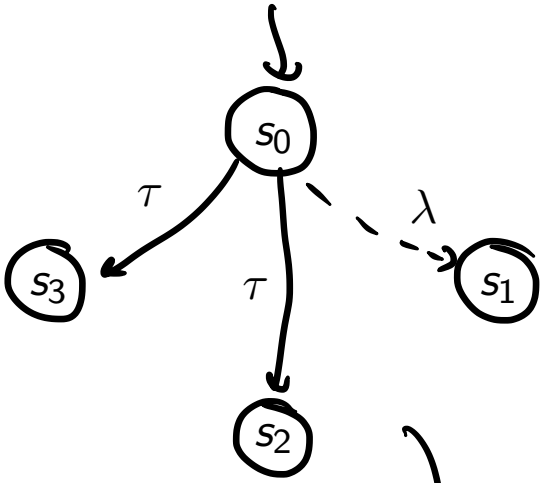
$(C_1 \| C_3) \setminus \{\beta\}$



Hiding

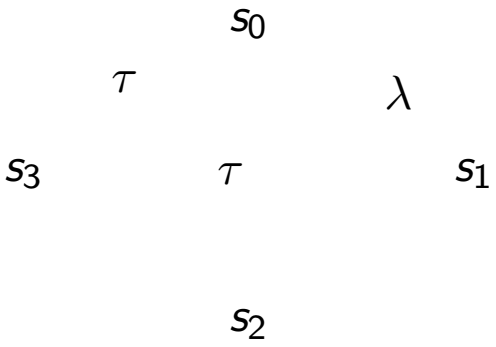


hiding $\{\alpha, \beta\}$ yields

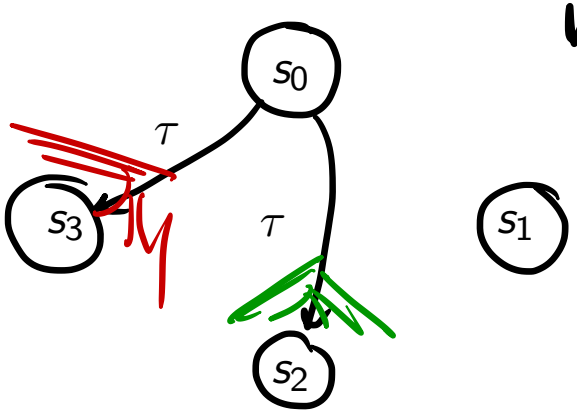


max. progress

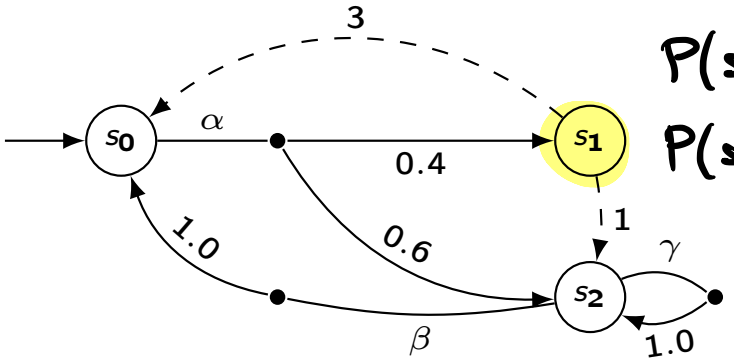
Applying maximal progress reduction yields:



reduces to



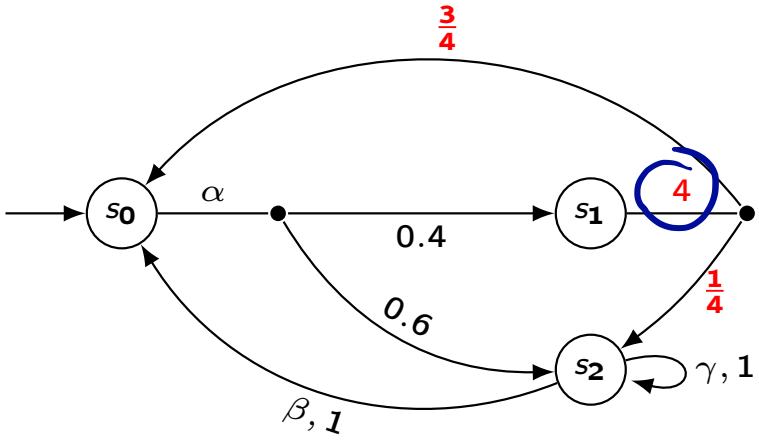
Bisimulation



$r(s_1) = 4$

$P(s_1, s_0) = \frac{3}{4}$

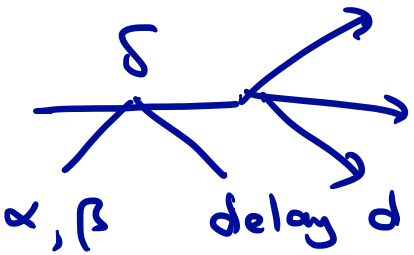
$P(s_1, s_2) = \frac{1}{4}$

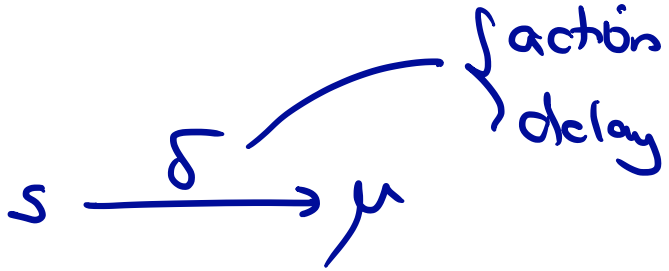


Unifying $\longrightarrow \subseteq S \times Act \times Dist(S)$

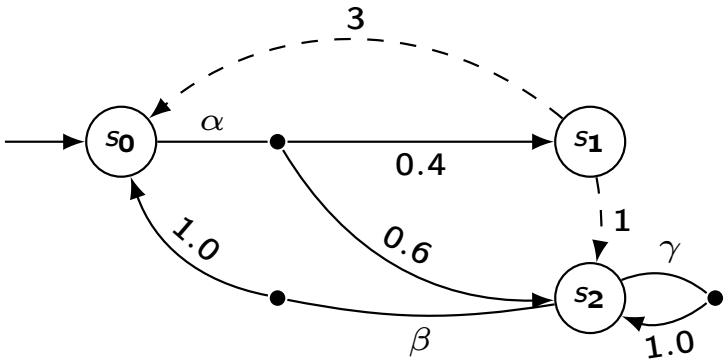
$+$

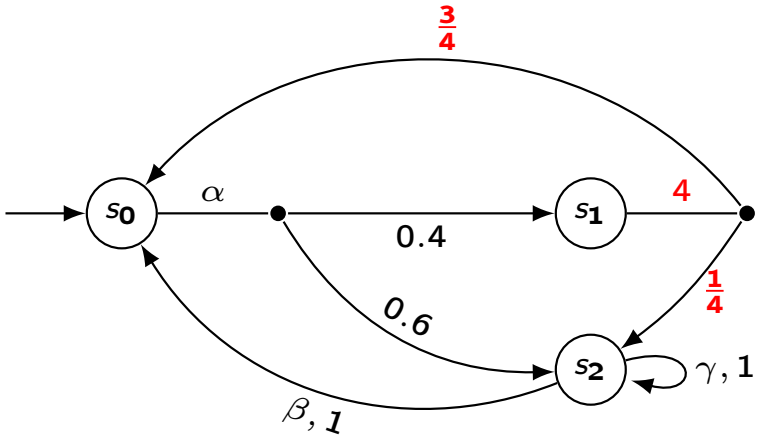
$\dashrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$





Bisimulation

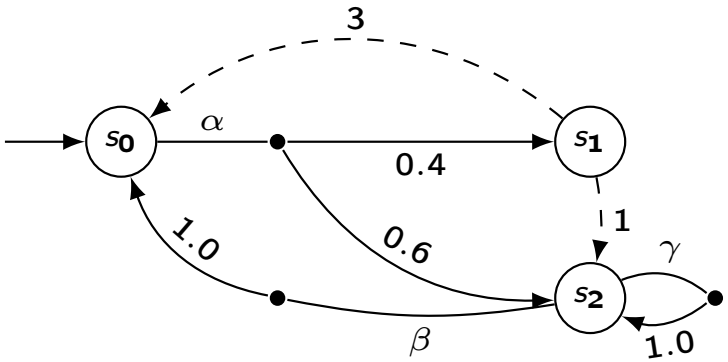


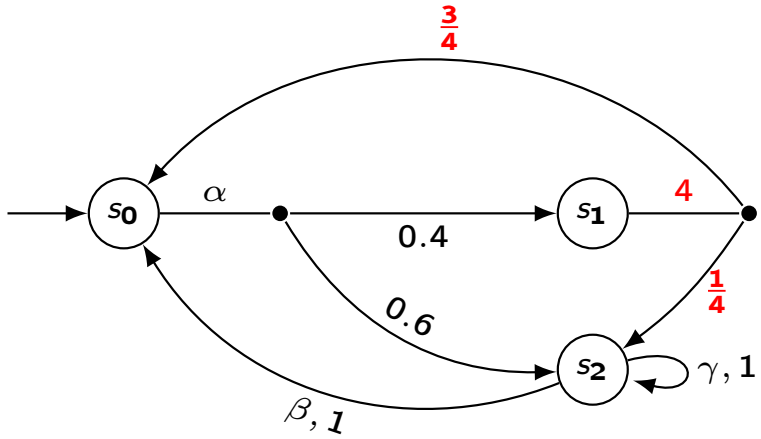


Bisimulation

Equivalence $R \subseteq S \times S$ is a *bisimulation* if for all $(s, t) \in R$:

Bisimulation





Bisimulation

Equivalence $R \subseteq S \times S$ is a *bisimulation* if for all $(s, t) \in R$:

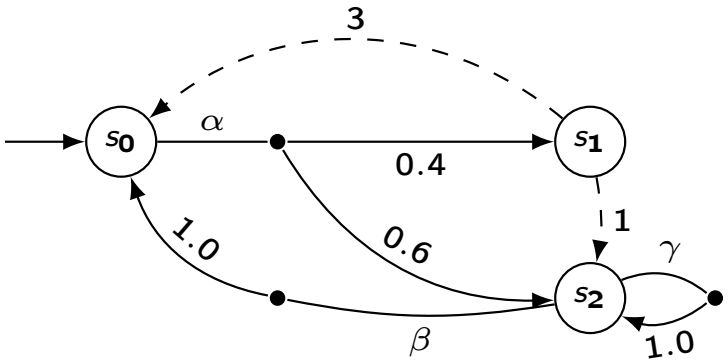
$\forall \delta \in Act \cup \mathbb{R}_{>0}$: $s \xrightarrow{\delta} \mu$ implies $t \xrightarrow{\delta} \nu$ with $\forall C \in S/R : \mu(C) = \nu(C)$.

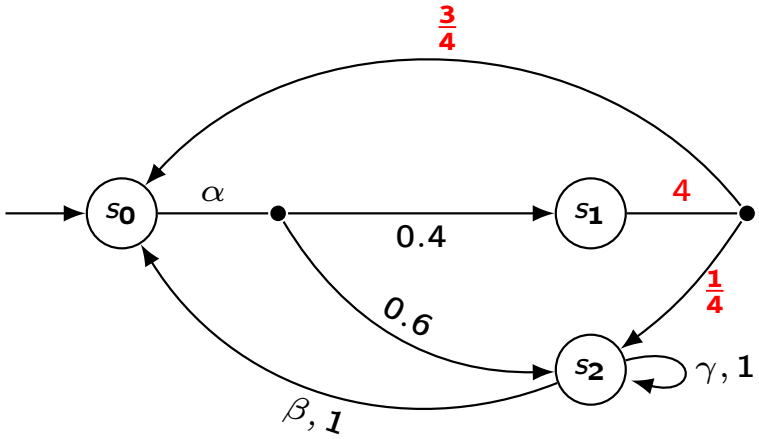
$\rightarrow$ $\dashrightarrow$

$s \xrightarrow{\alpha} \mu$ then $t \xrightarrow{\alpha} \nu$ $\quad \mu(C) = \nu(C) \quad \forall C \in S/R$

$s \overset{\lambda \in \mathbb{R}_{>0}}{\dashrightarrow} s'$ then $t \overset{\lambda}{\dashrightarrow} t'$ $\quad \mu(C) = \nu(C)$

Bisimulation





Bisimulation

Equivalence $R \subseteq S \times S$ is a *bisimulation* if for all $(s, t) \in R$:

$$\forall \delta \in Act \cup \mathbb{R}_{>0}\colon s \xrightarrow{\delta} \mu \text{ implies } t \xrightarrow{\delta} \nu \text{ with } \forall C \in S/R : \mu(C) = \nu(C).$$

Let $\sim$ be the largest bisimulation relation.

Congruence [Eisentraut *et al*, 2010]

$\sim$ is a congruence wrt. parallel composition and hiding.

1) $\sim$ is a congruence wrt. parallel composition

2) " " " wrt. hiding

ad 1): MA M_1 and M_2

$M_1 \sim M_2$ implies $\forall$ MA R. $M_1 \| R \sim M_2 \| R$

ad 2):

$M_1 \sim M_2$ implies $\forall A \subseteq Act.\ M_1 \setminus A \sim M_2 \setminus A$

hide all actions in A

$S = M_1 \| (M_2 \| M_3) \ldots \| M_N$

$M_i \sim M_i'$

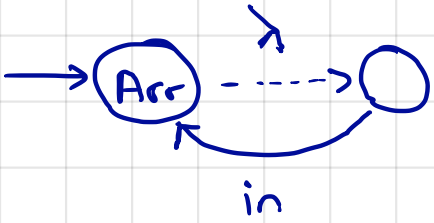
$M_2' \| M_3'$

$S' = M_1 \| (M_2' \| M_3')' \| \ldots \| M_N$

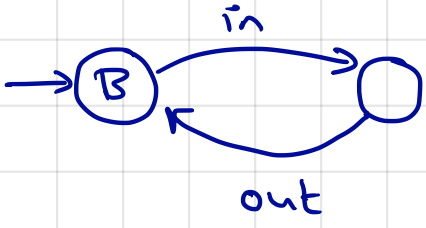
$S \sim S'$

Example

$Arr = (\lambda).\ in.\ Arr$

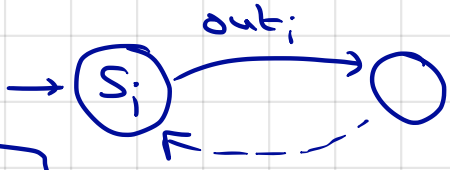


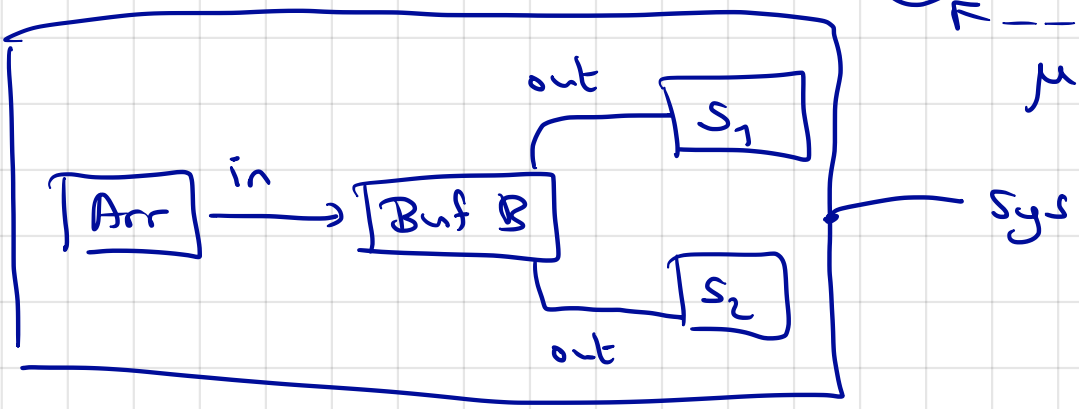
$B = in.\ out.\ B$



$S_i = out_i.\ (\mu).\ S_i$

$(i \in \{1,2\})$



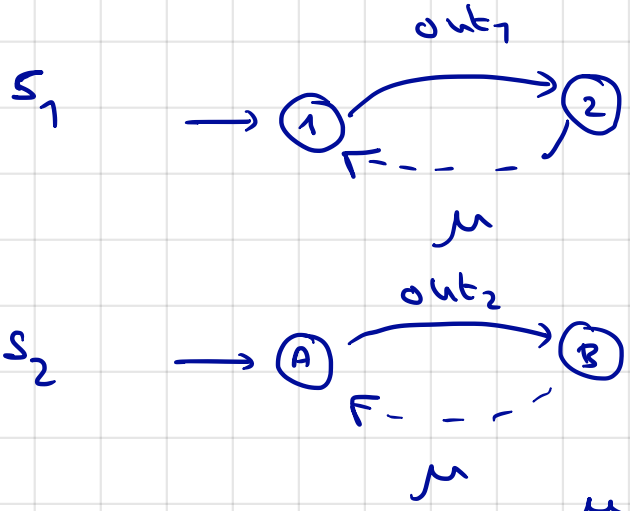


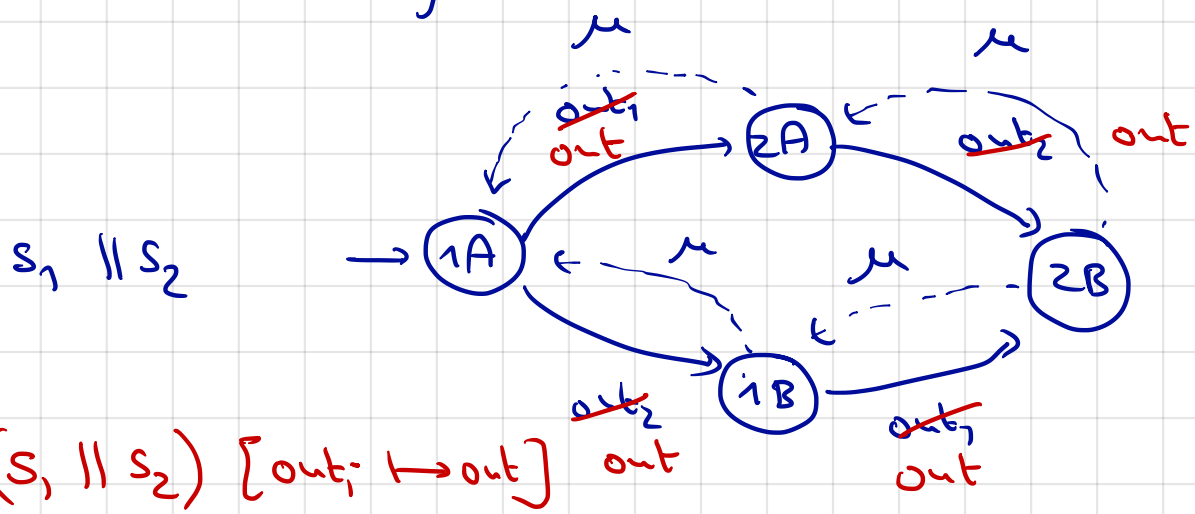
$$(Arr \,||\, B) \,||\, (S_1 \,||\, S_2)\,[out_i \mapsto out]$$

$$Sys = \Big((Arr \,||\, B) \,||\, (S_1 \,||\, S_2)\,[out_i \mapsto out] \Big) \setminus \{in, out\}$$

hiding of in, out

$S_i = out_i . (\mu) . S_i$



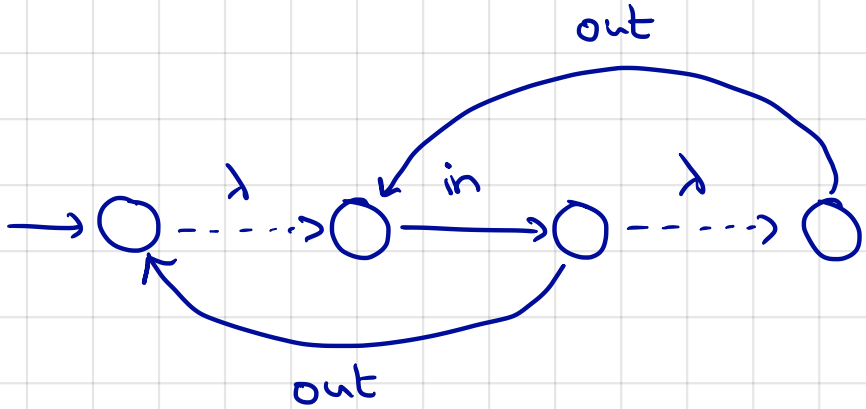


$(S_1 \,||\, S_2)\,[out_i \mapsto out]$

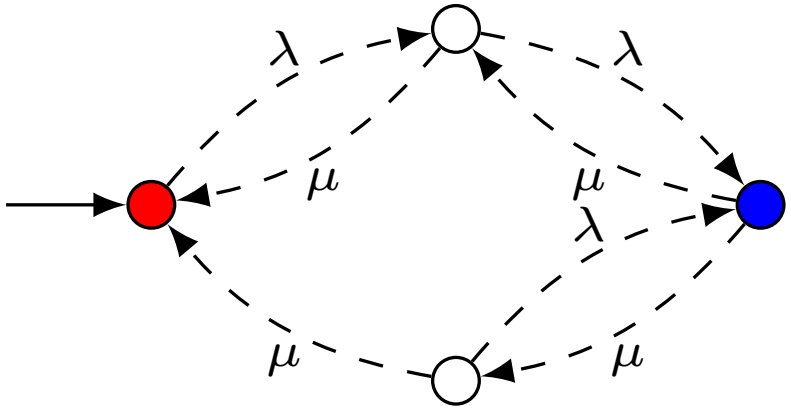
Arr ||B

Arr = $(\lambda) . in . Arr$

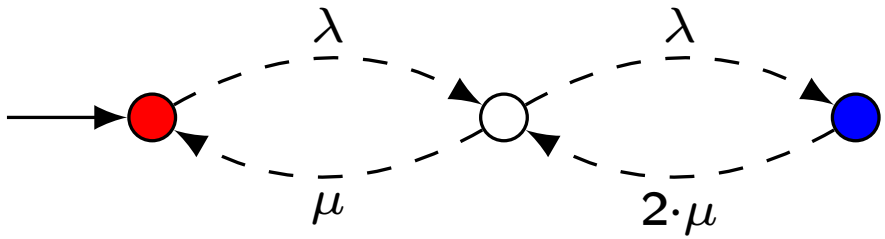
B = $in . out . B$



Bisimulation – Example



is bisimilar to



Compatibility

Bisimulation is backward compatible with bisimulation on probabilistic automata and bisimulation on labeled transition systems.

Weak bisimulation

DTMCs
CTMCs
} in some lectures before

Weak bisimulation

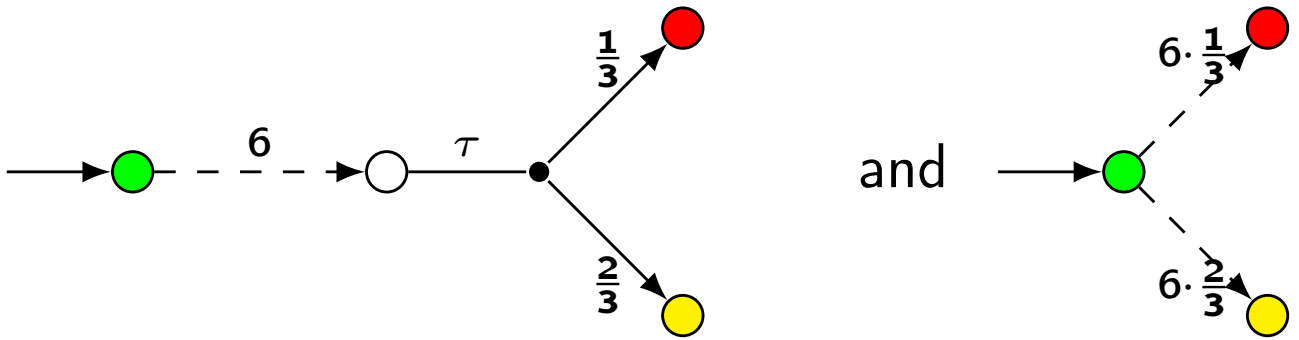
A naive attempt

Equivalence $R \subseteq S \times S$ is a *weak bisimulation* if for all $(s,t) \in R$:

$$\forall \delta \in Act \cup \mathbb{R}_{>0}\colon\ s \xrightarrow{\delta} \mu \text{ implies } t \stackrel{\delta}{\Longrightarrow} \nu \text{ with } \forall C \in S/R : \mu(C) = \nu(C)$$

where $t \stackrel{\delta}{\Longrightarrow} \mu$ means $t \xrightarrow{\tau^*} \xrightarrow{\delta} \xrightarrow{\tau^*} \nu$ (over trees).

This relation is backward compatible but too fine, as it distinguishes:



Weak bisimulation over distributions [Doyen *et al.*, 2008]

skip

Definition 10 (Weak bisimulation [20]). *A symmetric relation $\mathcal{R}$ on subdistributions over S is called a* weak bisimulation *if and only if whenever $\mu_1 \mathcal{R} \mu_2$ then for all $\alpha \in \mathbb{R} \cup \{\varepsilon\}$: $|\mu_1| = |\mu_2|$ and for all $s \in Supp(\mu_1)$ there exist $\mu_2{}^{\rightarrow}, \mu_2{}^{\Delta}$: $(\mu_2{}^{\rightarrow}, \mu_2{}^{\Delta}) \in split(\mu_2)$ and*

(i) $\mu_1(s)\delta_s \ \mathcal{R} \ \mu_2{}^{\rightarrow}$ *and* $(\mu_1 \ominus s) \ \mathcal{R} \ \mu_2{}^{\Delta}$

(ii) whenever $s \xrightarrow{\alpha} \mu_1'$ *for some* μ_1' *then* $\mu_2{}^{\rightarrow} \stackrel{\alpha}{\Longrightarrow}_C \mu''$ *and* $(\mu_1(s) \cdot \mu_1') \ \mathcal{R} \ \mu''$

Two subdistributions μ and γ are weak bisimilar, denoted by $\mu \approx \gamma$, if the pair (μ, γ) is contained in some weak bisimulation.

Weak bisimulation over distributions [Doyen *et al.*, 2008]

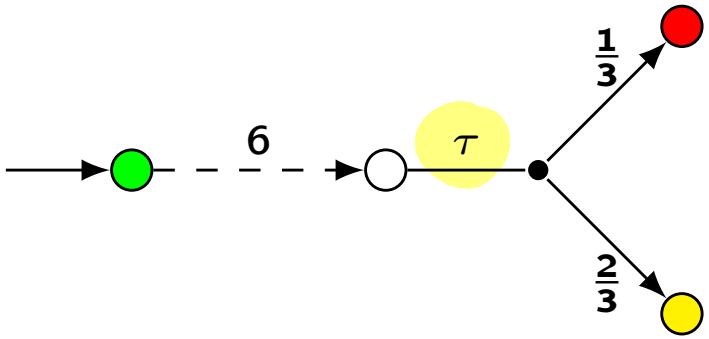
Congruence [Eisentraut et. al., 2010]

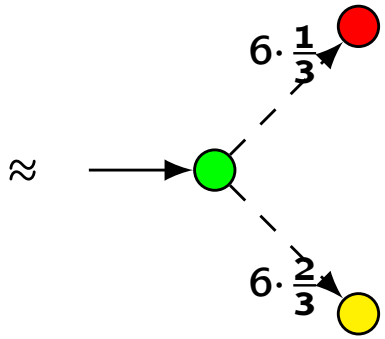
$\approx$ is a congruence wrt. parallel composition and hiding.

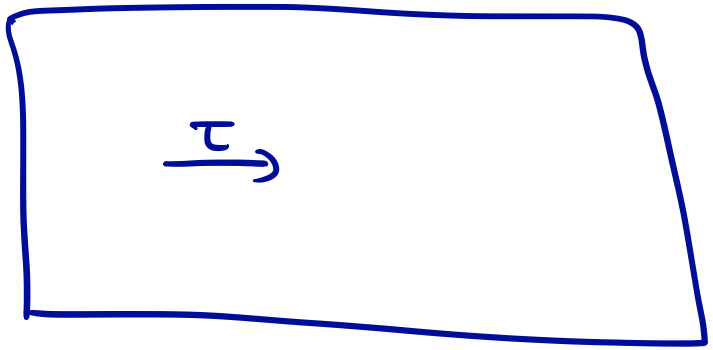
Theorem [Deng & Hennessy, 2011]

$\approx$ is the coarsest "reasonable" notion of weak bisimulation.

Backward incompatibility



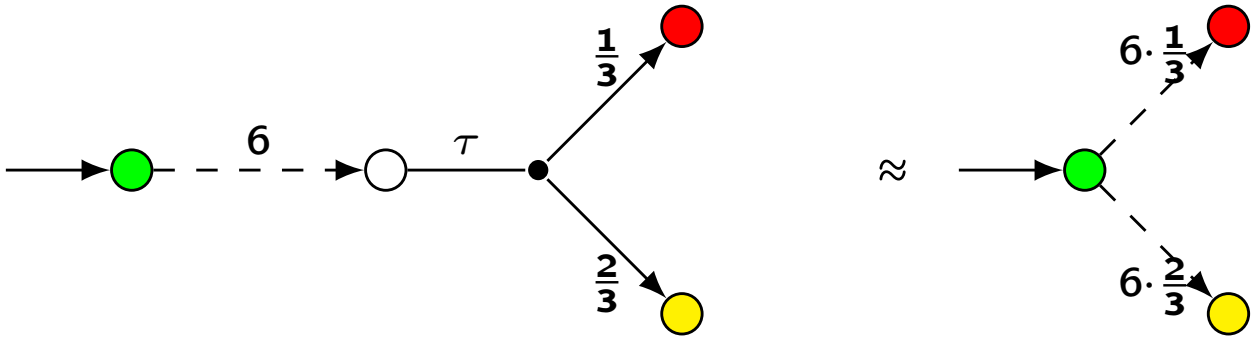




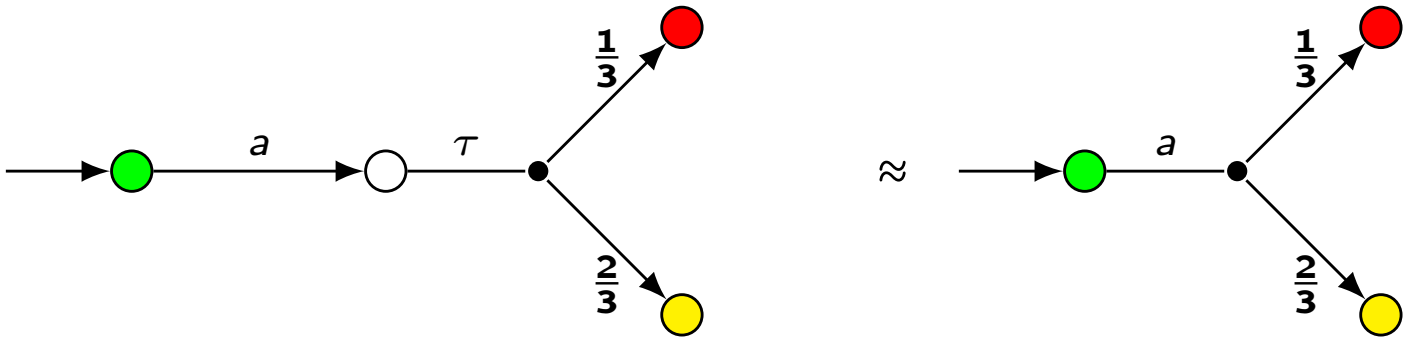
$r(\bullet) = 6 \cdot \frac{1}{3} + 6 \cdot \frac{2}{3} = 6$

$P(\bullet, \bullet) = \frac{6 \cdot \frac{1}{3}}{6} = \frac{1}{3}$

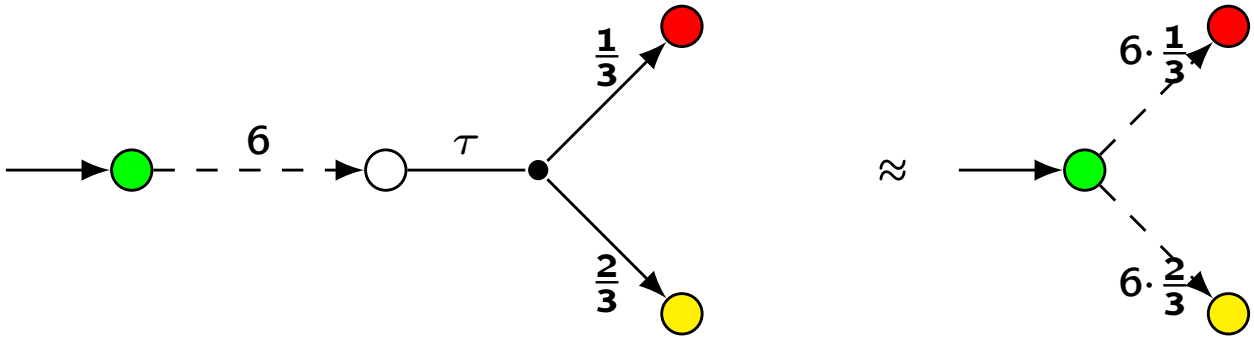
Backward incompatibility



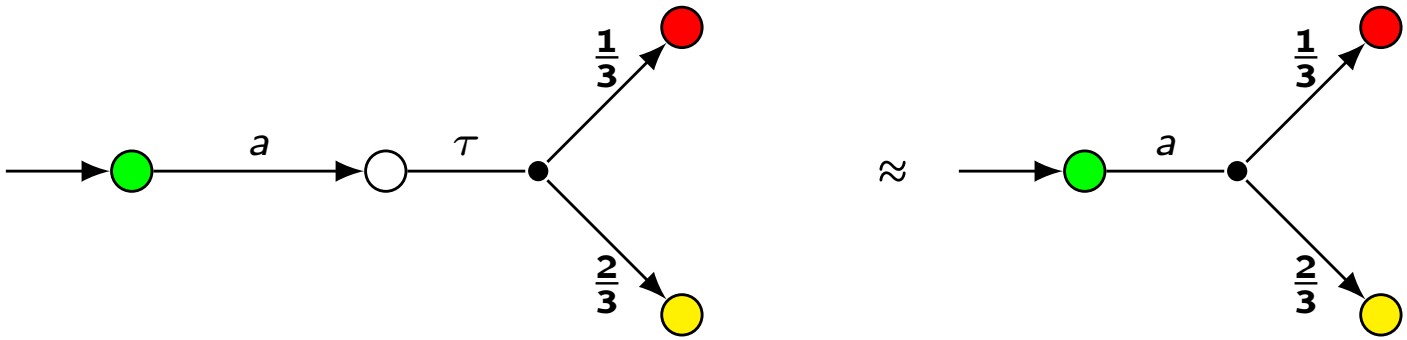
Similarly, one obtains:



Backward incompatibility



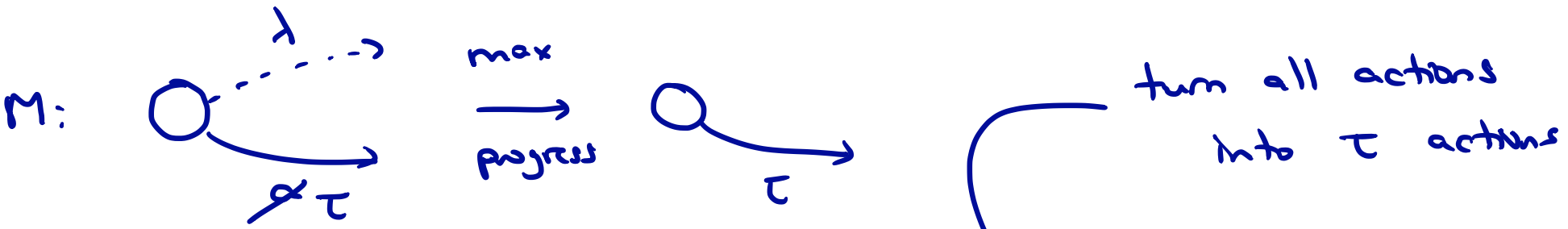
Similarly, one obtains:



But Segala's weak bisimulation distinguishes these PA

slide 15

Analysis



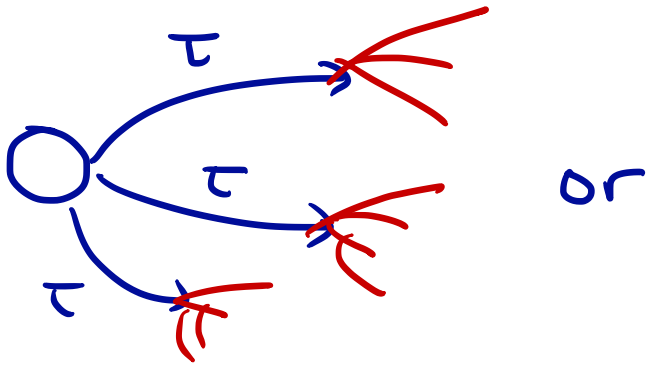
Model to be analysed

Typical structure:

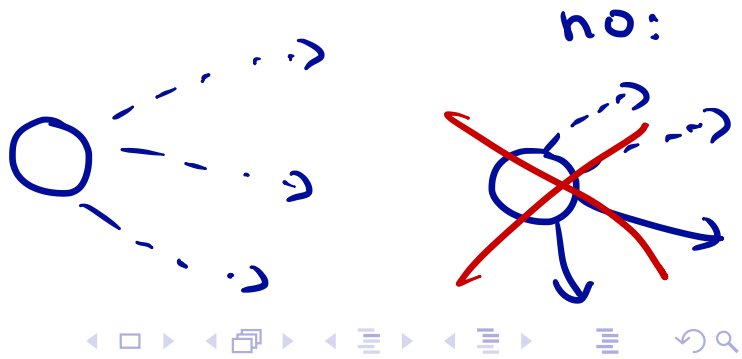
$$M = (M_1 \,||\, M_2 \,||\, \ldots \,||\, M_n) \setminus A$$

where A is the union of all visible actions, i.e., $A = \bigcup_i Act(M_i) - \{\tau\}$.

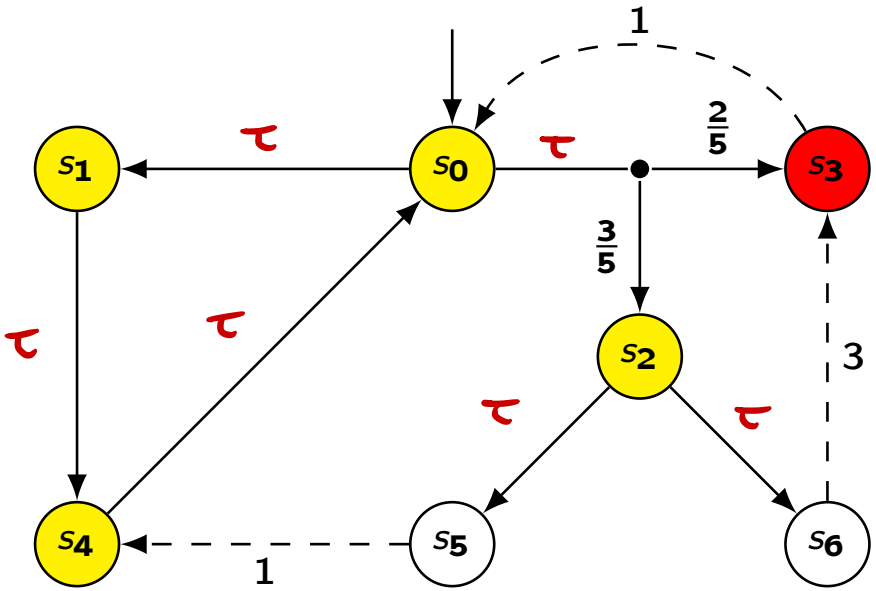
States in M have either only Markovian or only action transitions. No mixtures.



or



Expected time



Example M

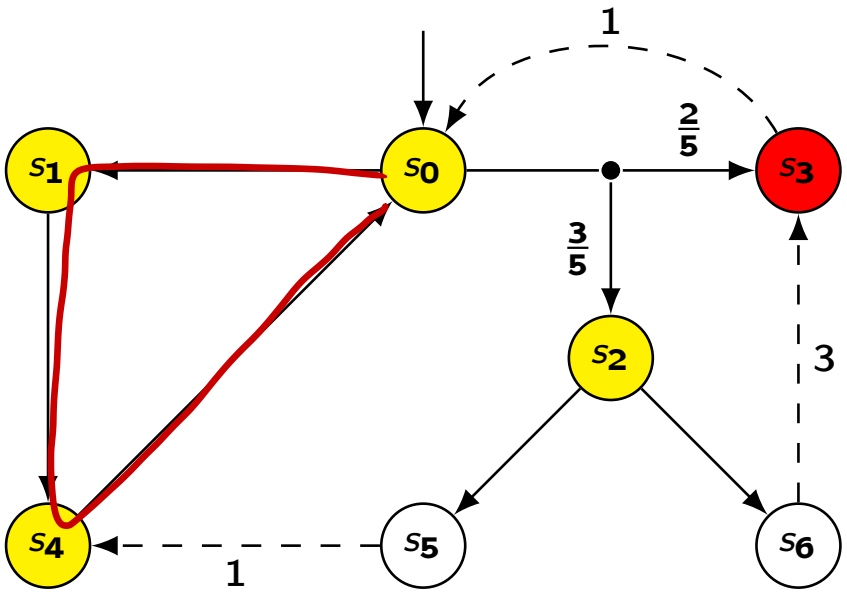
$s_0 \xrightarrow{\tau} \mu_0 \qquad \mu_0(s_1) = 1$

$s_0 \xrightarrow{\tau} \mu_1 \qquad \mu_1(s_2) = \frac{3}{5}$

$\mu_1(s_3) = \frac{2}{5}$

- Expected time from s_0 to s_3?

Expected time

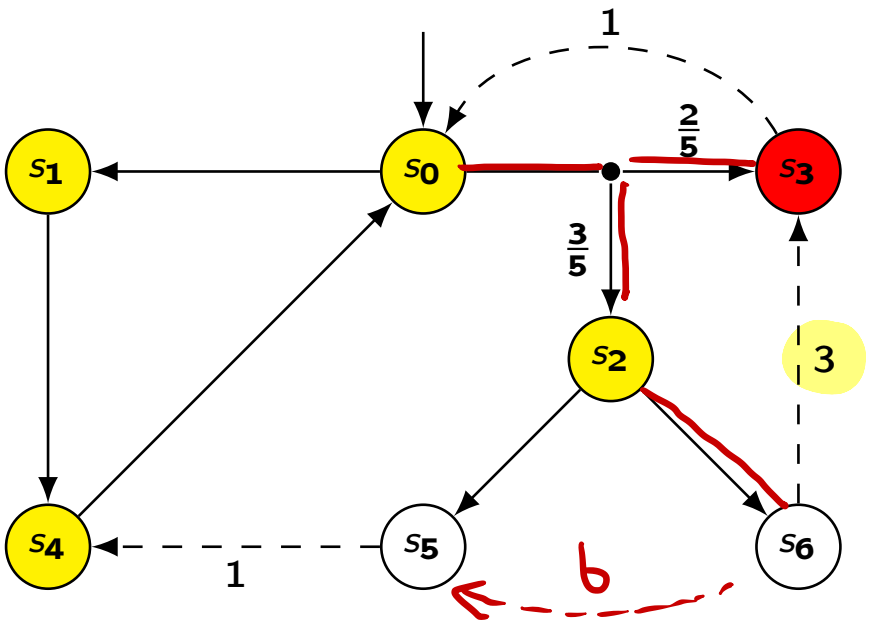


- Expected time from s_0 to s_3?
- Maximally ∞

Non-determinism

The expected time to reach G is not uniquely defined.

Expected time



- Expected time from s_0 to s_3?
- Maximally ∞
- Minimally $\frac{2}{5} \cdot 0 + \frac{3}{5} \cdot \frac{1}{3}$

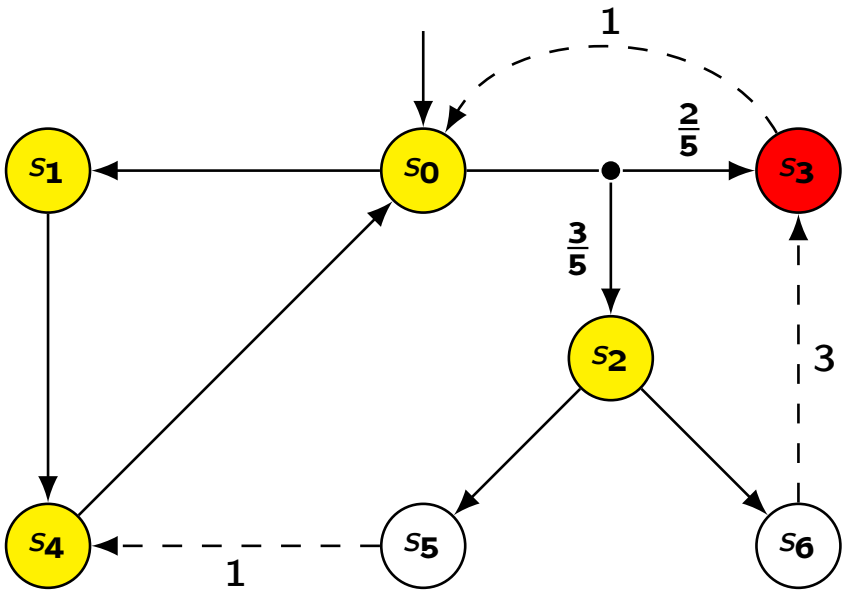
$\frac{2}{5} \cdot 0 + \frac{3}{5} \cdot \frac{3}{3+b} \cdot \frac{1}{3+b}$

Non-determinism

The expected time to reach G is not uniquely defined.
It depends on the choices in states s_0 and s_2.

prob (s_6, s_3)

Expected time



- Expected time from s_0 to s_3?
- Maximally ∞
- Minimally $\frac{2}{5} \cdot 0 + \frac{3}{5} \cdot \frac{1}{3}$

Non-determinism

The expected time to reach G is not uniquely defined.
It depends on the choices in states s_0 and s_2. Approach: consider all possibilities. This yields bounds.

Expected time

Let $eT_P(s, \Diamond G)$ be the expected time to reach G starting from state s under policy P.

Aim:

Determine the minimal expected time until reaching G from s, i.e., $eT_P(s, \Diamond G)$ under the most demonic policy P that prevents the system from reaching G.

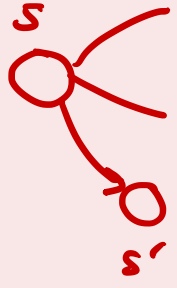
Fixpoint theorem

expected delay in state s

Theorem

$eT^{\min}(s, \Diamond G)$ is the unique fixpoint of the Bellman operator:

$$[L(v)](s) = \begin{cases} \frac{1}{r(s)} + \sum_{s' \in S} \mathbf{p}(s,s') \cdot v(s') & \text{if } s \in MS - G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_\alpha(s') \cdot v(s') & \text{if } s \in PS - G \\ 0 & \text{if } s \in G \end{cases}$$



$[L(v)](s)$ = $v(s)$

$v(s) = eT^{\min}(s, \Diamond G)$

algorithm: à la reachability probabilities in MDPs.

Fixpoint theorem

Theorem

$eT^{\min}(s, \Diamond G)$ is the unique fixpoint of the Bellman operator:

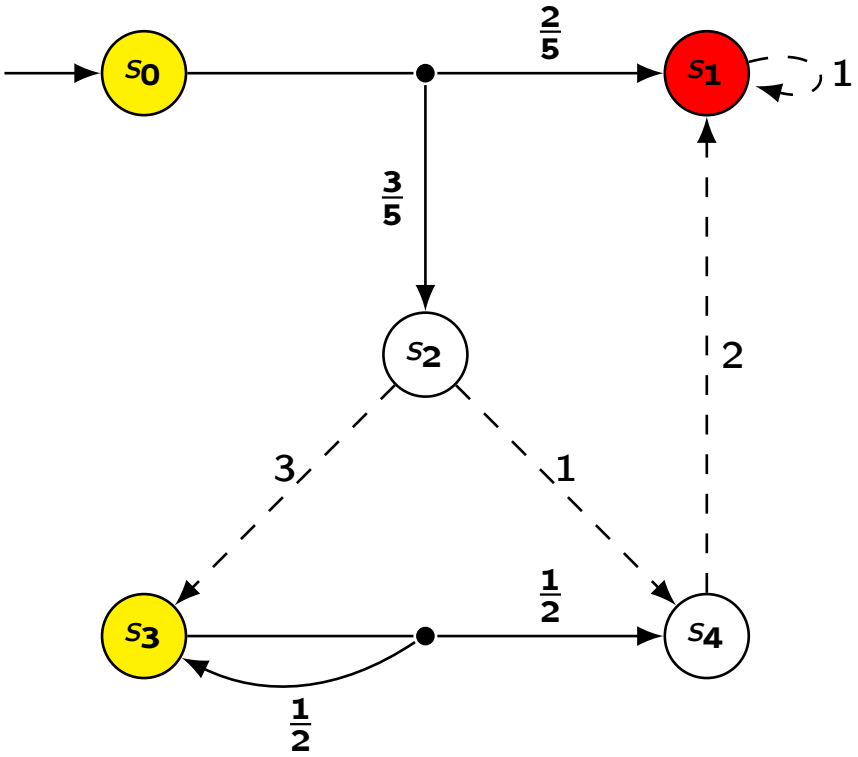
$$[L(v)](s) = \begin{cases} \frac{1}{\mathbf{r}(s)} + \sum_{s' \in S} \mathbf{p}(s, s') \cdot v(s') & \text{if } s \in MS - G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_\alpha(s') \cdot v(s') & \text{if } s \in PS - G \\ 0 & \text{if } s \in G \end{cases}$$

Corollary

$eT^{\min}(s, \Diamond G)$ equals the minimal cost reachability of G of a stochastic shortest path problem (SSP).

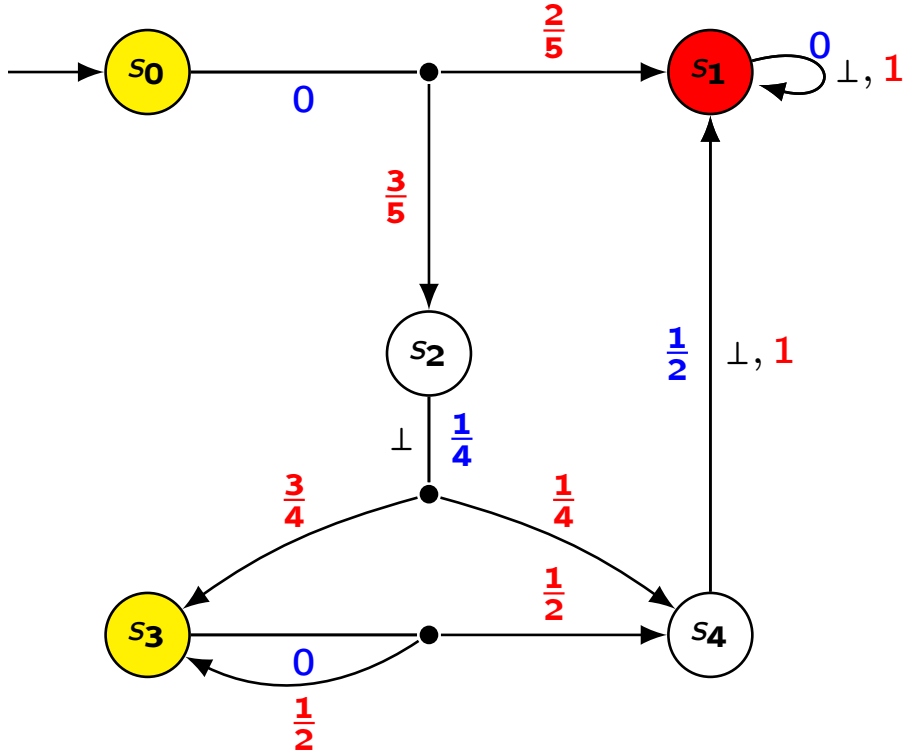
Reduction to SSP problem

Example Markov automaton:



$$\mathbf{p}(s,\sigma,s') = \begin{cases} \frac{R(s,s')}{r(s)} & \text{if } \sigma = \bot \\ \mu(s') & \text{if } s \xrightarrow{\sigma} \mu \\ 0 & \text{otherwise} \end{cases}$$

Its induced SSP instance:



$$c(s,\sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \notin G, \sigma = \bot \\ 0 & \text{otherwise} \end{cases}$$

Expected time analysis: synopsis

Minimal and maximal expected time

1. Make all states in G absorbing
2. Transform the Markov automaton to an SSP problem
3. Solve the SSP problem by linear programming

Positional policies suffice

There is a positional policy that yields $eT^{\min}(s, \Diamond G)$.

Overview

Introduction

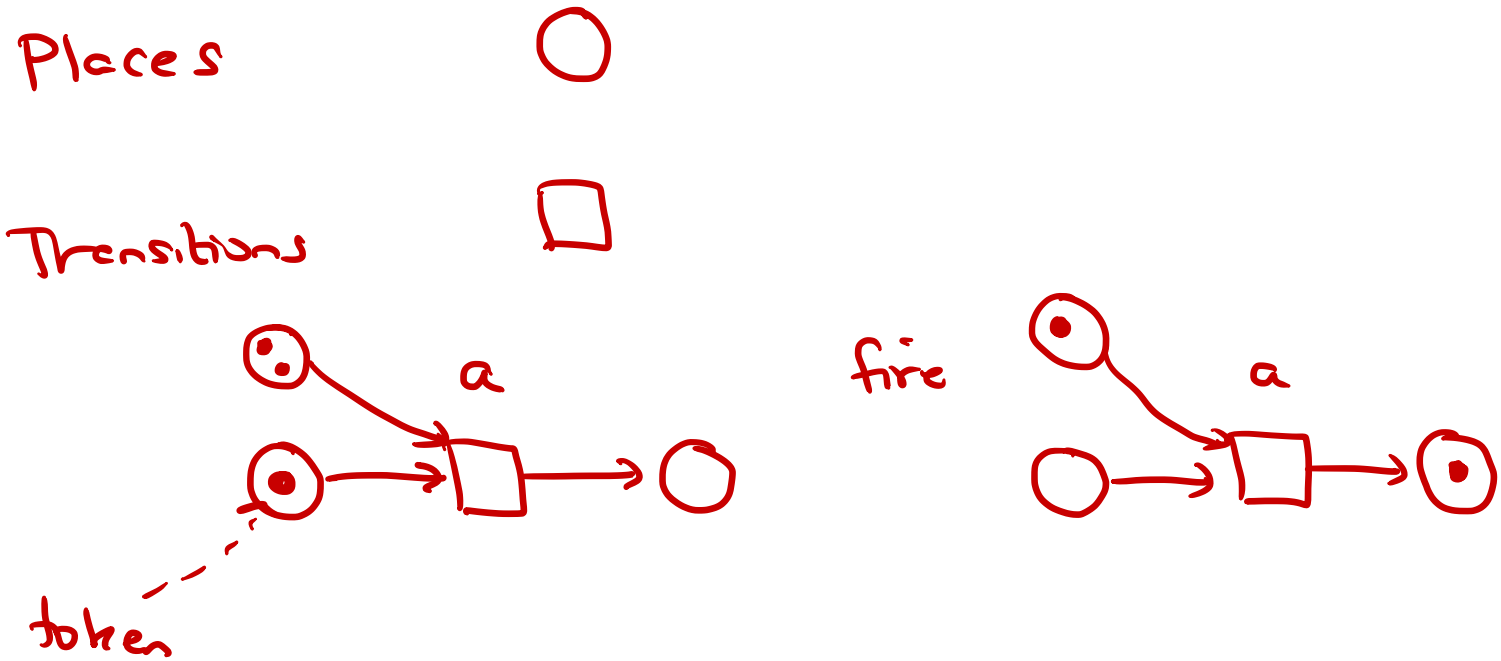
Beautiful theory
- What are Markov Automata?
- Concurrent composition and hiding
- Bisimulation
- Analysis algorithms

The usage for high-level modeling languages
- Process algebra
- Generalized Stochastic Petri Nets

A process algebra for PA

GSPNs: historical perspective

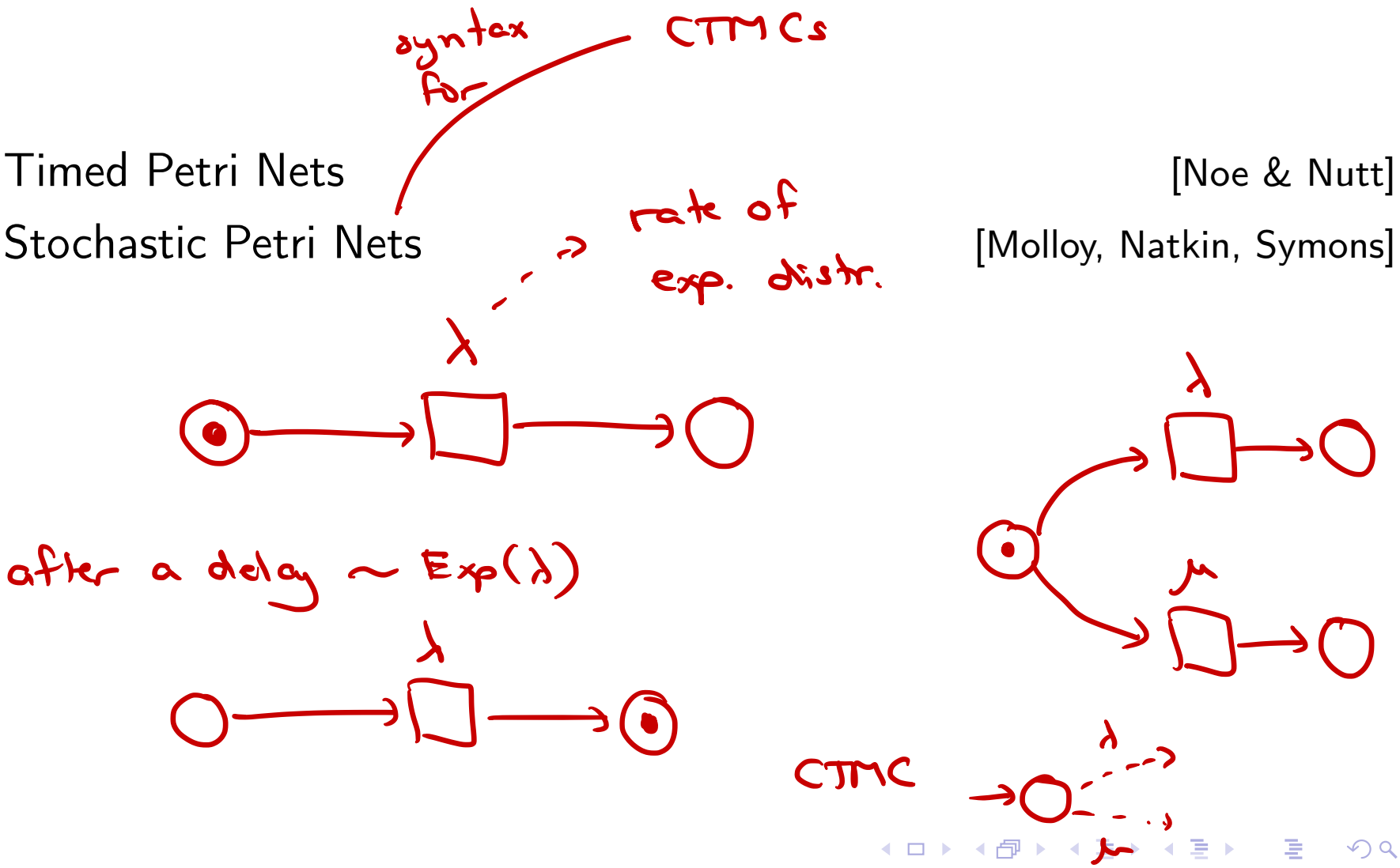
1973 Timed Petri Nets [Noe & Nutt]



GSPNs: historical perspective

1973 Timed Petri Nets [Noe & Nutt]

1980 Stochastic Petri Nets [Molloy, Natkin, Symons]



GSPNs: historical perspective

1973 Timed Petri Nets [Noe & Nutt]

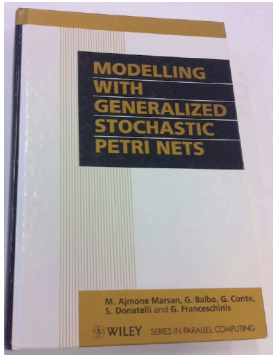
1980 Stochastic Petri Nets [Molloy, Natkin, Symons]

1984 Generalized Stochastic Petri Nets [Ajmone Marsan, Conte & Balbo]

1995 Modeling with Generalized Stochastic Petri Nets [Ajmone Marsan *et al.*]

A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems

MARCO AJMONE MARSAN and GIANNI CONTE
Politecnico di Torino, Turin, Italy
and
GIANFRANCO BALBO
Universita' di Torino, Turin, Italy

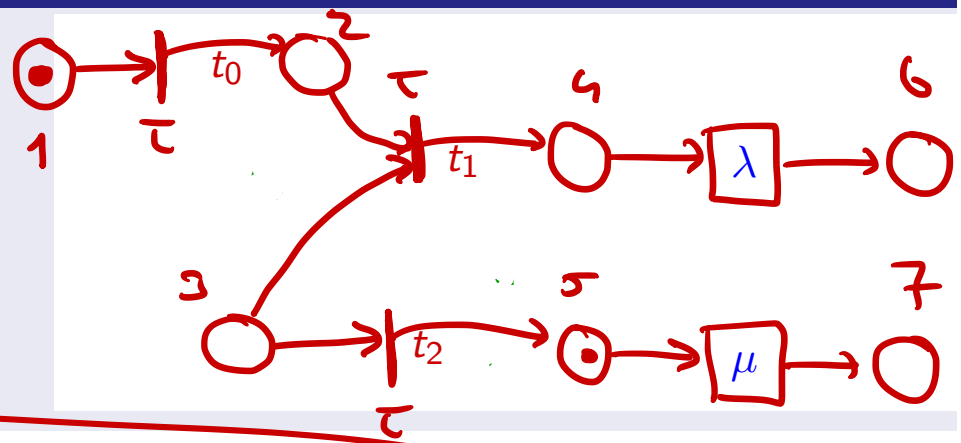


Generalized stochastic Petri nets [Ajmone Marsan et al, 1984]

What is a GSPN?

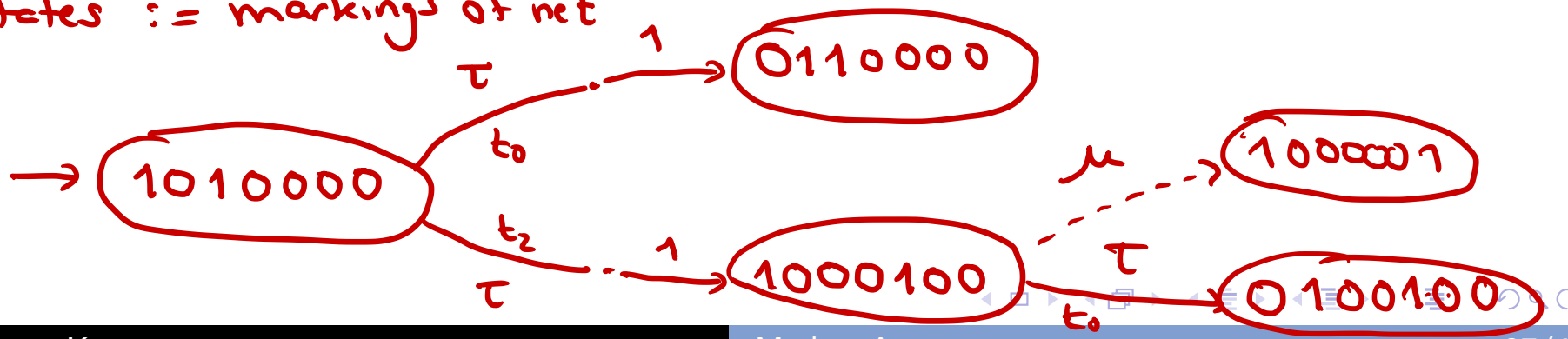
A Petri net with

- Timed transitions
- Immediate transitions
- Natural weights

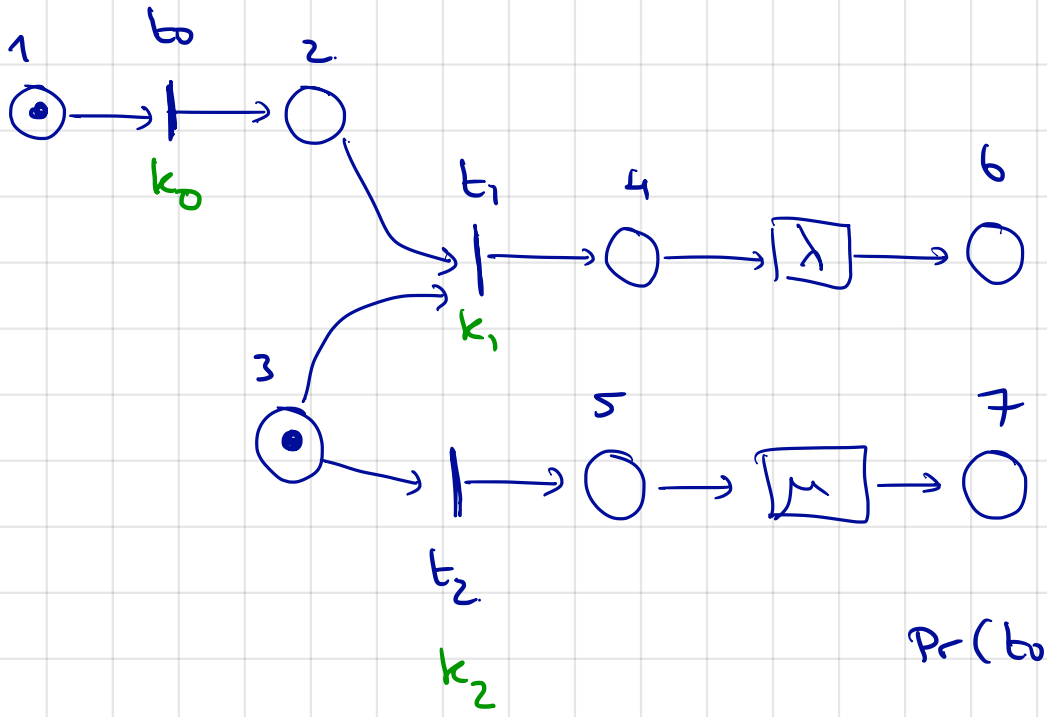


Any GSPN is a Markov automaton!

- states := markings of net

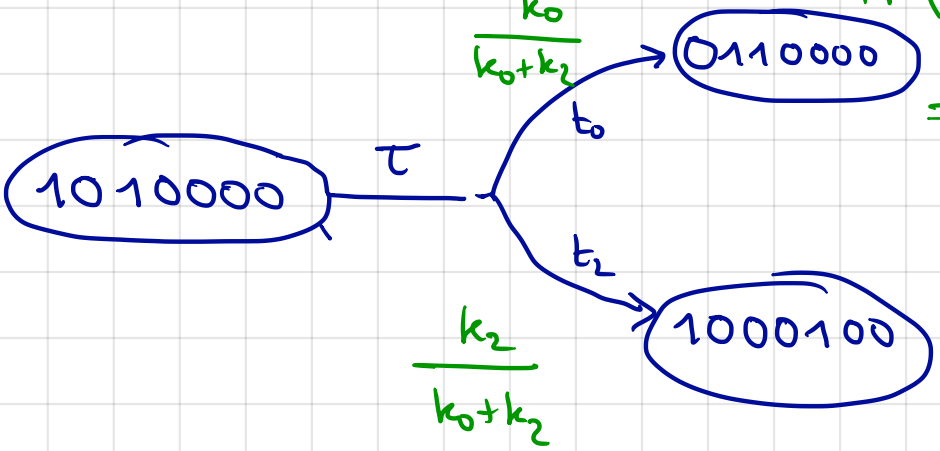


GSPN example



weights $\in \mathbb{N}_{>0}$

$$Pr(t_0 \text{ firing}) = \frac{k_0}{k_0 + k_2}$$



$$Pr(\text{immediate trans } t) = \frac{\text{weight } t}{\text{total weight}}$$

over all enabled imm. transitions

Thus; **Any GSPN is a Markov automaton.**

Generalized stochastic Petri nets [Ajmone Marsan et al, 1984]

What is a GSPN?

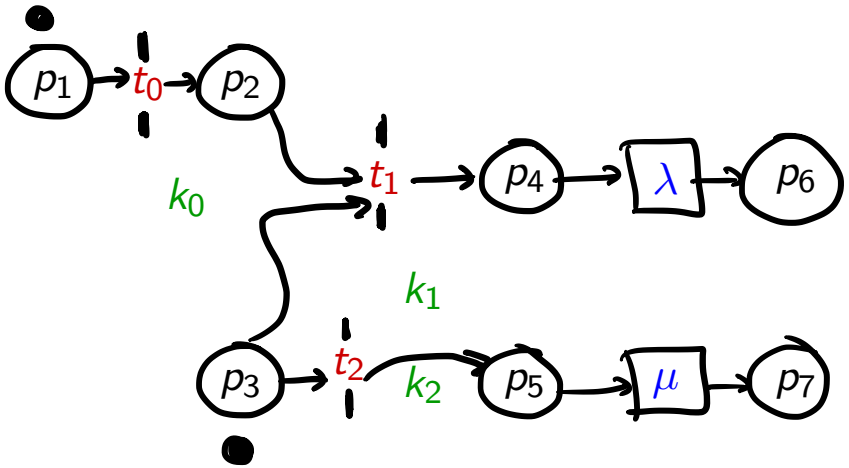
A Petri net with

- Timed transitions
- Immediate transitions
- Natural weights

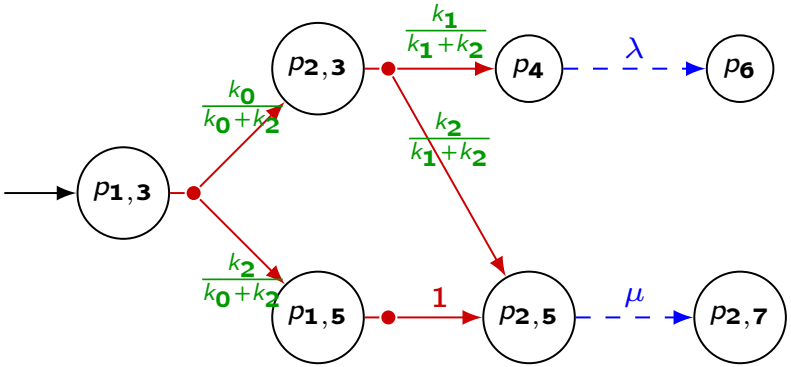
Two-phase semantics

1. Determine enabled transitions and their probability
   - Maximal progress: immediate transitions have priority
2. Determine the underlying stochastic process

GSPN semantics by example

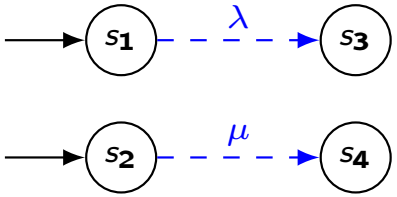


Token game and probabilities



Isn't this a Markov automaton?

Induced stochastic process



Initial distribution $\mu(s_1) = \frac{k_0}{k_0+k_2} \cdot \frac{k_1}{k_1+k_2}$, and

$$\mu(s_2) = \frac{k_2}{k_0+k_2} + \frac{k_0}{k_0+k_2} \cdot \frac{k_1}{k_1+k_2}$$

Isn't this weakly bisimilar?

Well-defined nets

Backward compatibility [Eisentraut *et al.*, 2013]

The MA semantics of a well-defined GSPN is weak bisimilar to its standard GSPN semantics.

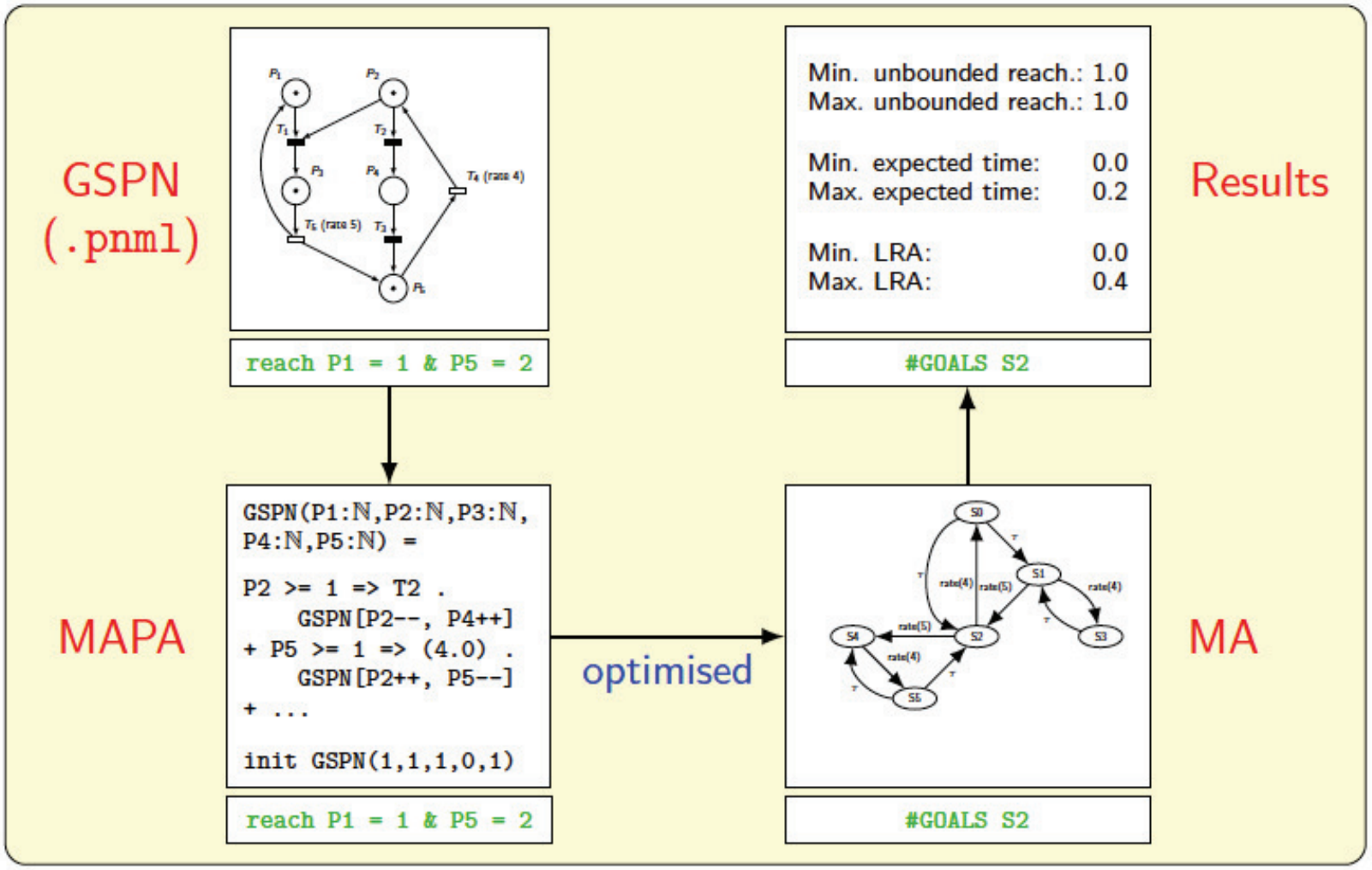
GSPNs go non-deterministic

Advantages of MA semantics

- It is truly simple
- It is intuitive
- It is compositional
- It is backward compatible
- No restrictions on net level

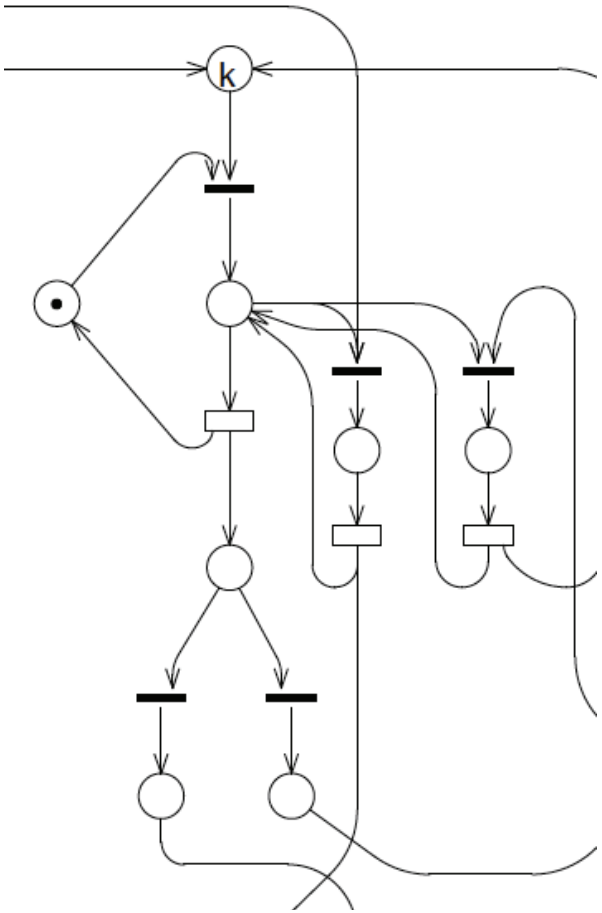
This solves a long-standing open issue in stochastic Petri nets

Tool support



http://wwwhome.cs.utwente.nl/~timmer/mama/ Storm

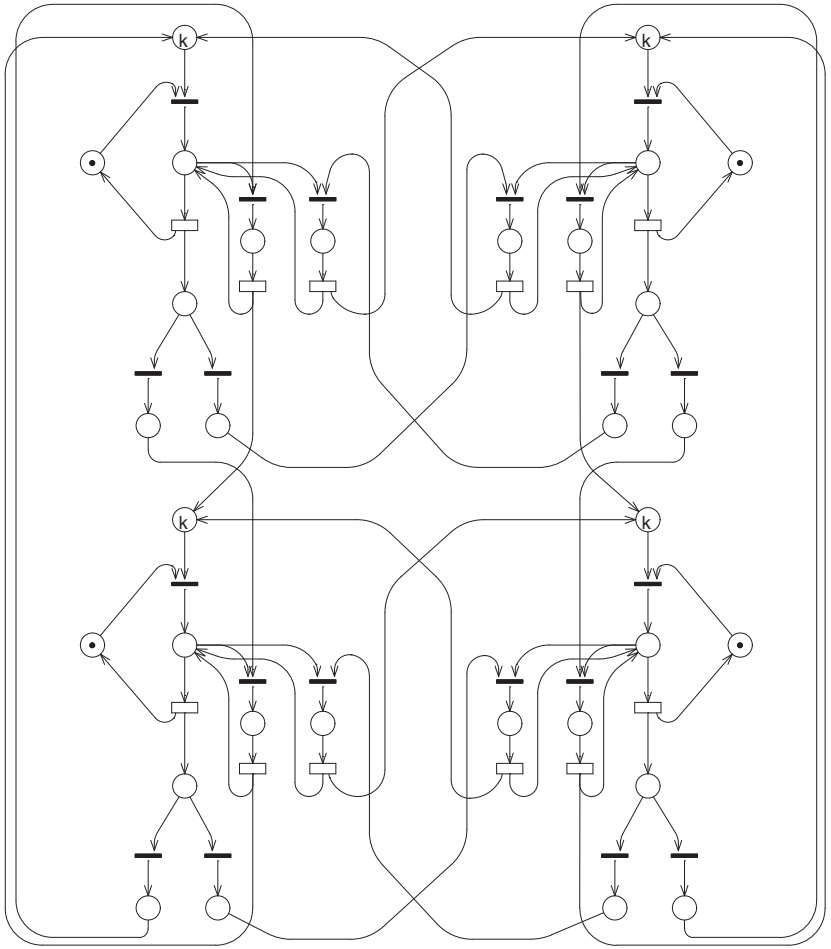
GSPN model of multi-processor system [Ajmone Marsan et. al., 1994]



GSPN of a single processor

- A 2×2 multi-processor grid
- Multi-tasking of k tasks/processor
- Two-phase task execution:
  1. local processing (1)
  2. co-operative processing (10)
- Selection policy for neighbour
- Pipelining of tasks per processor
- Co-operation has priority

Multi-processor system



Presence of immediate transitions excludes usage GSPN tools

Processor throughput

k	# states	# transitions	generation (s)	tp processor 1	tp processor 2	tp processor 4
2	2508	3215	14.5	.9031	ditto	ditto
3	10852	14379	64.7	.9086	ditto	ditto
4	31832	42879	193.0	.9090	ditto	ditto

Scenario one: uniform weight assignment

2	as above	4254	0.8	[.9031,.9055]	[.8585,.9479]	[.9029,.9032]
3	as above	19089	3.2	[.9081,.9089]	[.8633,.9541]	[.9086,.9087]
4	as above	56704	9.8	[.9089,.9091]	[.8636,.9545]	[.9090,.9091]

Scenario two: processor one selects non-deterministically

2	as above	4698	0.6	[.8110,.9956]	ditto	ditto
3	as above	20872	2.7	[.8173,.9998]	ditto	ditto
4	as above	62356	7.9	[.8181,1.0]	ditto	ditto

Scenario three: fully non-deterministic