Formal verification of stochastic systems

Christel Baier
Universität Bonn
Germany

Probability elsewhere ...

- randomized algorithms [Rabin 1960]

  fingerprints, input sampling, breaking symmetry, ...

  models: discrete-time Markov chains, Markov decision process

- performance modelling [Erlang 1907]

  emphasis on steady-state and transient measures

  models: continuous-time Markov chains

- stochastic control theory, operations research [Bellman 1957]

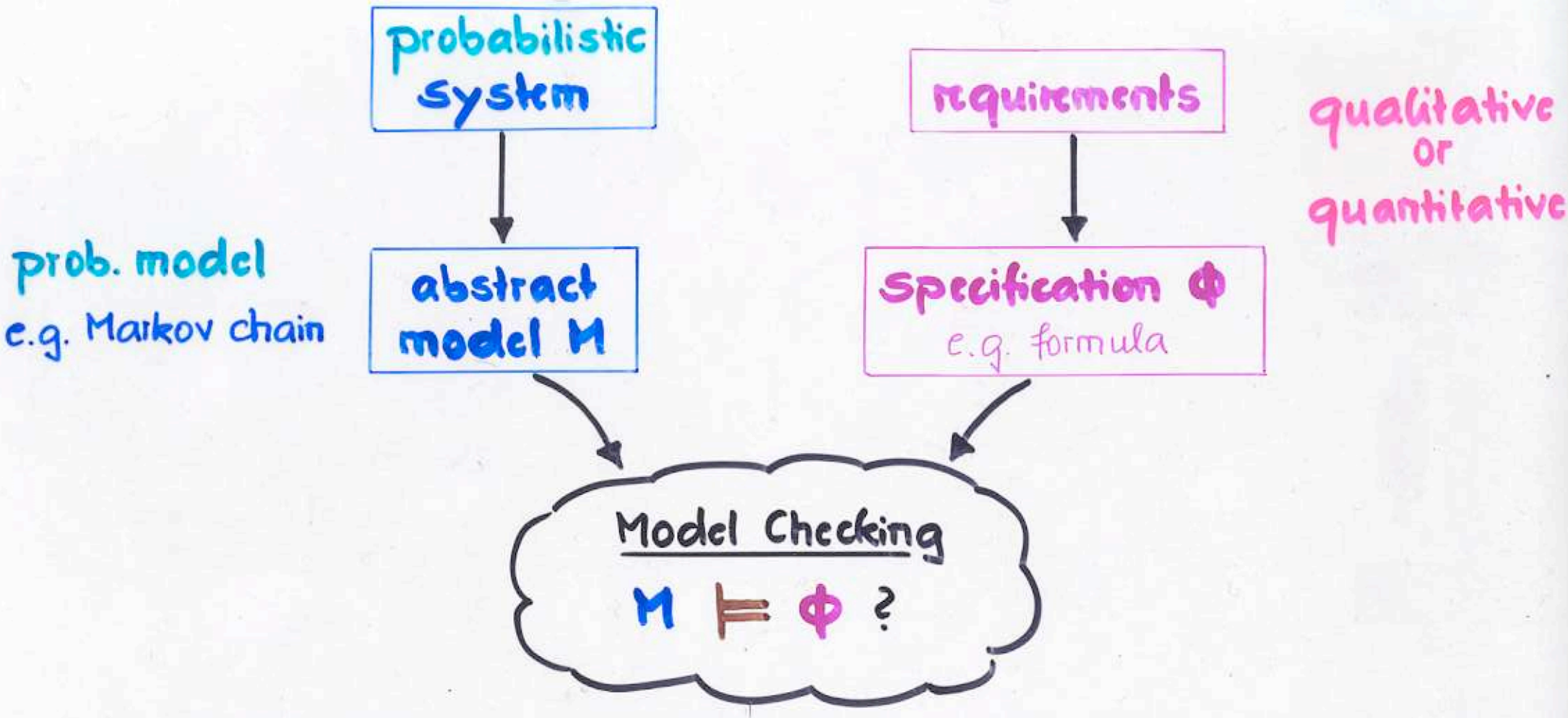
  emphasis on finding optimal policies for average measures

  models: discrete-time Markov decision processes

- modelling biological systems

- security protocols

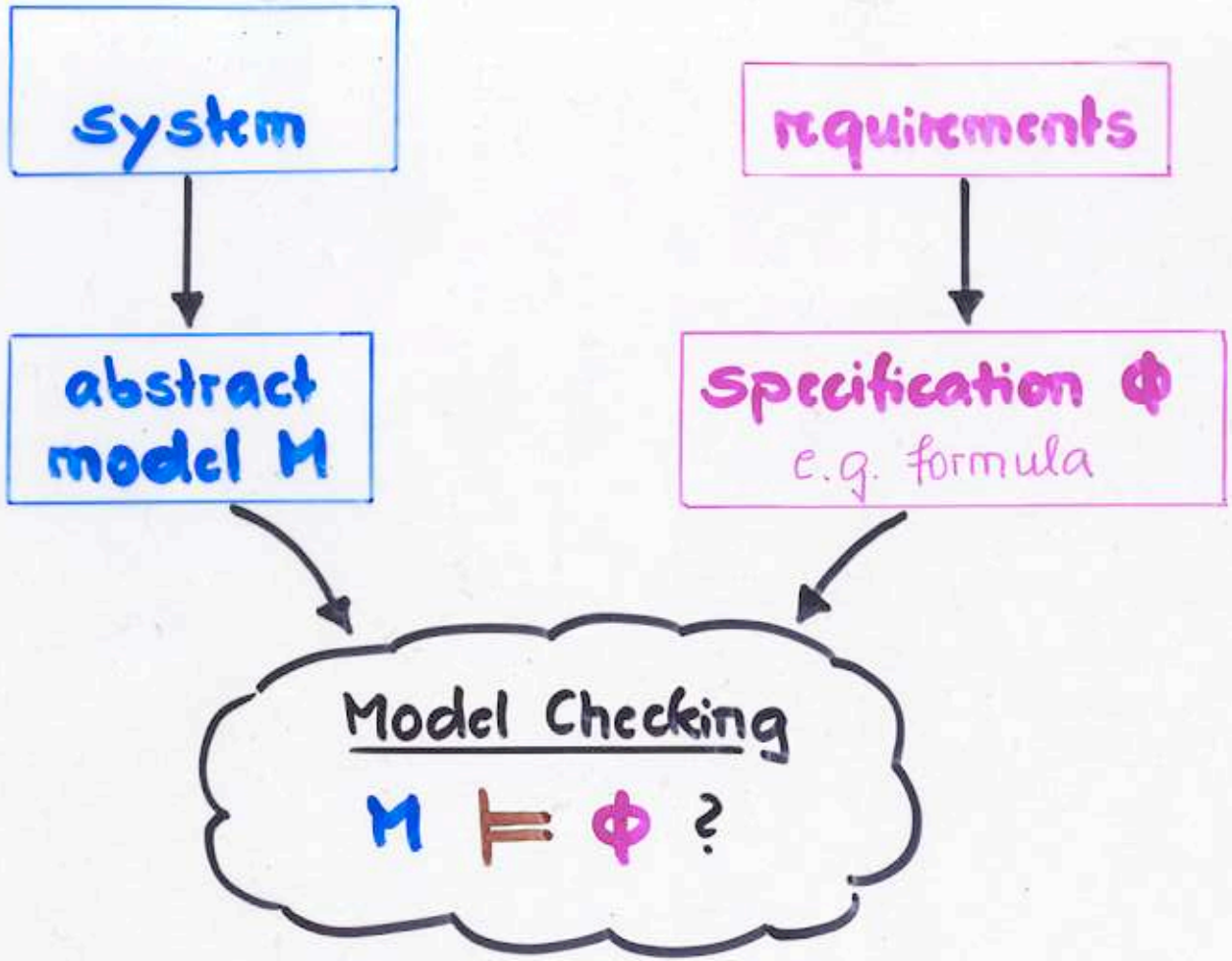
  ...



qualitative properties: a certain condition holds with probability 1

quantitative properties:

lower/upper bounds for probabilities / expected values



Temporal logic ⟹ unambiguous, precise measure specifications

Model checking ⟹ complex measures are automatically computed,
no expert knowledge required

exchanging techniques with application areas (performance modelling, ...)

Tutorial: Formal verification of stochastic systems

Outline:

- Markov chains and probabilistic computation tree logic
- Markov decision processes and PCTL*
- Continuous-time Markov chains and continuous stochastic logic

Probabilistic model checking so far

- termination of probabilistic programs — Hart, Sharir, Pnueli 1983
- qualitative linear time properties for discrete-time Markov models — Vardi, Wolper 1986; Courcoubetis/Yannakakis 1988
- probabilistic computation tree logic for discrete-time Markov models — Hansson, Jonsson 1994; Bianco, de Alfaro 1995
- continuous stochastic logic for continuous-time Markov chains — Aziz, Sanwal, Singhal, Brayton '96; Baier, Katoen, Hermanns 1999
- probabilistic timed automata — Jensen 1996, Kwiatkowska et al '00

⋮

Tools:

PRISM, ETMCC, VESPA, YMER, APNN Toolbox,
TwoTowers, RAPTURE, LIQUOR, ...

Discrete-time (or time-abstract) Markov chains:

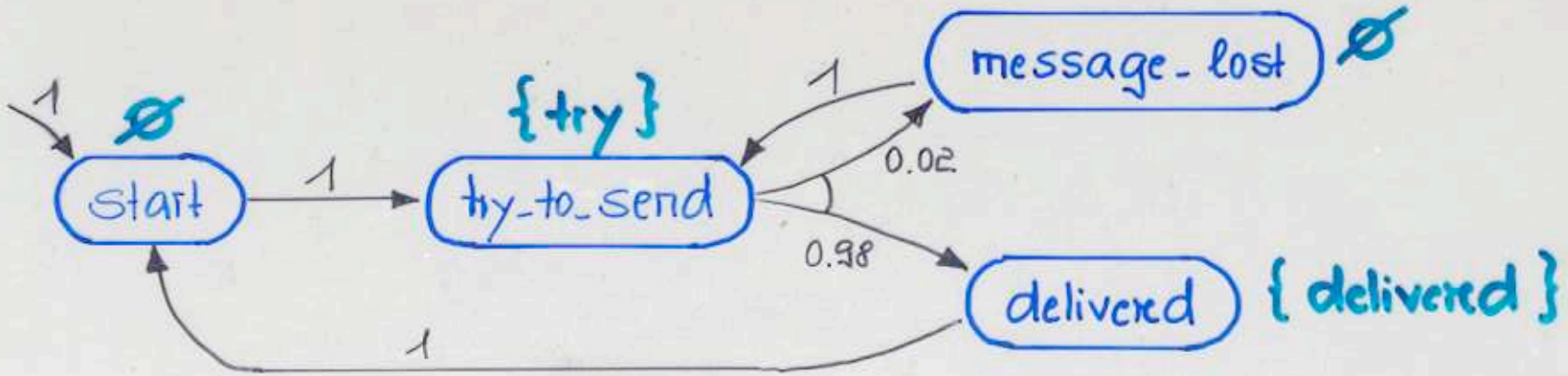
"transition systems with probabilities for the successor states"

$M = (S, P, AP, L)$ + **initial distribution / state**

where
- S set of states (here: finite)
- $P : S \times S \longrightarrow [0,1]$ transition probability matrix

  s.t. $\sum_{s' \in S} P(s,s') = 1$

- AP set of atomic propositions
- $L : S \longrightarrow 2^{AP}$ labelling function



Probability measure of a Markov chain

$$M = (S, P, AP, L, \pi_0)$$

π_0: initial distribution $\pi_0 : S \to [0,1]$

probability measure for sets of infinite paths:

- consider the σ-algebra generated by the cylinder sets

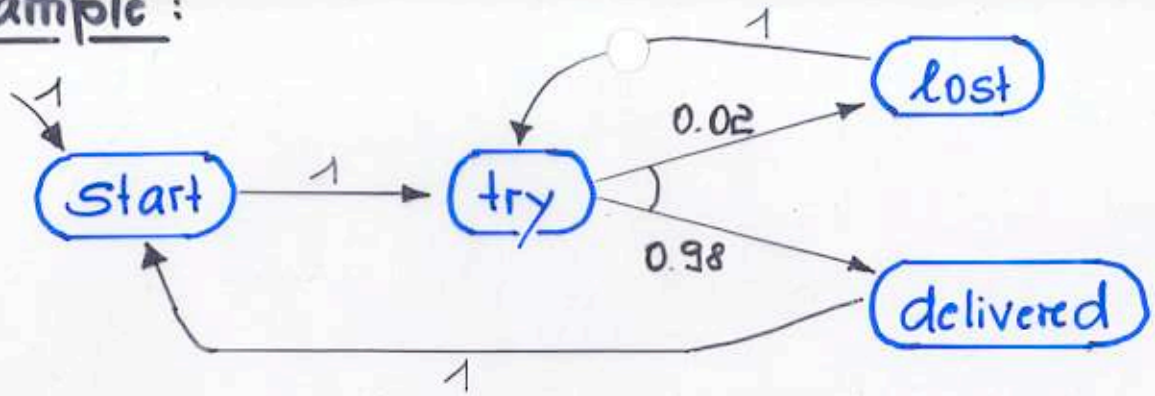
  $\Delta(s_0, \ldots, s_n)$ = set of infinite paths $s_0, \ldots, s_n, s_{n+1}, s_{n+2}, \ldots$

  ($s_0, \ldots, s_n$: finite path)

- probability measure is given by:

$$Pr^M(\Delta(s_0, \ldots, s_n)) = \underbrace{\pi_0(s_0)}_{\text{probability for initial state } s_0} \cdot \underbrace{\prod_{1 \le i \le n} P(s_{i-1}, s_i)}_{\text{probability for the finite path } s_0, \ldots, s_n}$$

Example :



Pr^M("delivered will be reached within the next 5 steps")

$= Pr^M$(start try delivered) + Pr^M(start try lost try delivered)

$= 0.98 + 0.02 * 0.98 = 0.9996$

Pr^M("delivered will be reached eventually")

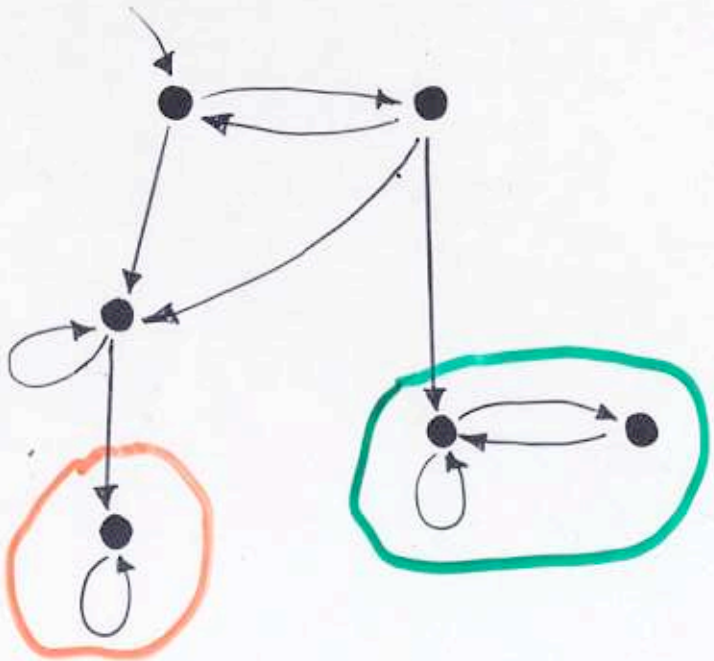
$= \sum_{n \geq 0} Pr^M$(start try (lost try)^n delivered)

$= \sum_{n \geq 0} 0.02^n * 0.98 = 1$

Fundamental property of finite Markov chains:

Almost surely a bottom strongly connected component (BSCC) will be reached and all its states visited infinitely often.

$$Pr^M\{\, s_0 s_1 s_2 \ldots \mid \exists i \geq 0\ \exists \text{ BSCC } C \text{ s.t. } \forall j \geq i.\ s_j \in C \wedge \forall s \in C\ \overset{\infty}{\exists} j.\ s_j = s \,\} = 1$$



2 BSCCs

(arbitrary non-zero probabilities for all edges)

PCTL* (prob. computation tree logic)

[Hansson/Jonsson '94]

State formulae:

$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathbb{P}_{\sim p}(\varphi)$$

(a: atomic proposition)

where $\sim \in \{\leq, <, \geq, >\}$ and $p \in [0,1]$

Path formulae:

$$\varphi ::= \Phi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathrm{X}\varphi \mid \varphi_1 \,\mathrm{U}\, \varphi_2$$

- probabilistic operator $\mathbb{P}_{\ldots}(\varphi)$ replaces the CTL-quantifiers $\exists$, $\forall$
- specifies lower or upper probability bounds for the event given by path-formula φ
- qualitative properties: $\mathbb{P}_{>0}(\varphi)$, $\mathbb{P}_{=1}(\varphi)$
- quantitative properties: $\mathbb{P}_{<0.5}(\varphi)$, $\mathbb{P}_{\geq 0.99}(\varphi)$

PCTL*

State formulae:

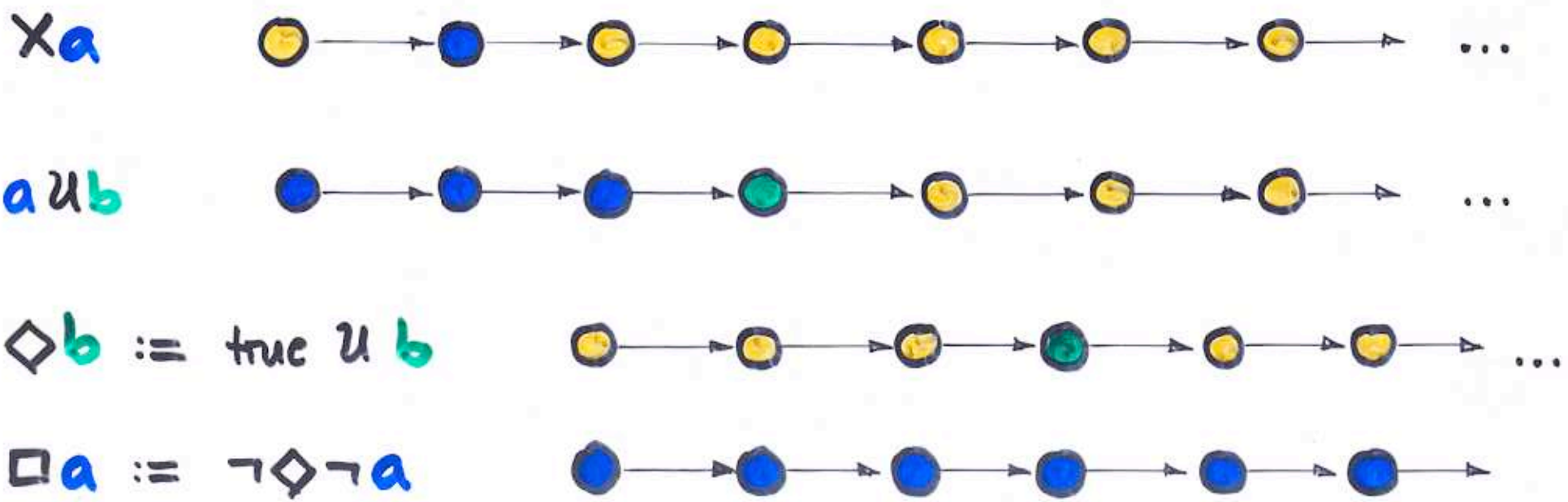
$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathbb{P}_{\geq p}(\varphi) \mid \dots$$

Path formulae:

$$\varphi ::= \Phi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid \varphi_1 \, U \, \varphi_2$$

next step

until



PCTL*:

Interpretation of the state formula over the states of a Markov chain

$s \models \text{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \neg\phi$	iff	$s \not\models \phi$
$s \models \phi_1 \wedge \phi_2$	iff	$s \models \phi_1$ and $s \models \phi_2$
$s \models Pr_{\geq p}(\varphi)$	iff	$Pr^M(s, \varphi) \geq p$

probability measure of the set of paths σ s.t. $\sigma \models \varphi$ when s is viewed as the unique starting state

PCTL* (probabilistic computation tree logic)

state formulae

$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \Pr_{\geq p}(\varphi) \mid \Pr_{\leq p}(\varphi)$$

path formulae

$$\varphi ::= \Phi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid \varphi_1 \, U \, \varphi_2 \mid \Diamond\varphi \mid \Box\varphi$$

(next step) (until) (eventually) (always)

Interpretation of the path formulae:

Let $\sigma = s_0 s_1 s_2 \ldots$ be an infinite path.

$\sigma \models \Phi$ iff $s_0 \models \Phi$

$\sigma \models X\varphi$ iff $s_1 s_2 s_3 \ldots \models \varphi$

$\sigma \models \varphi_1 \, U \, \varphi_2$ iff $\exists i \geq 0$ s.t. $s_i s_{i+1} \ldots \models \varphi_2$

and $\forall 0 \leq j < i : s_j s_{j+1} \ldots \models \varphi_1$

Examples for PCTL* - specifications:

communication protocol:

$$Pr_{\leq 0.001}(\Diamond \text{error})$$

$$Pr_{=1}(\Box(\text{try\_to\_send} \rightarrow Pr_{\geq 0.9}(X \text{ delivered})))$$

$$Pr_{=1}(\Box \text{ try\_to\_send} \rightarrow \neg \text{start } \mathcal{U} \text{ delivered})$$

leader election protocol for n processes in a ring [Itai, Rodeh]

each process chooses a random number in $\{1, \ldots, k\}$ as id.

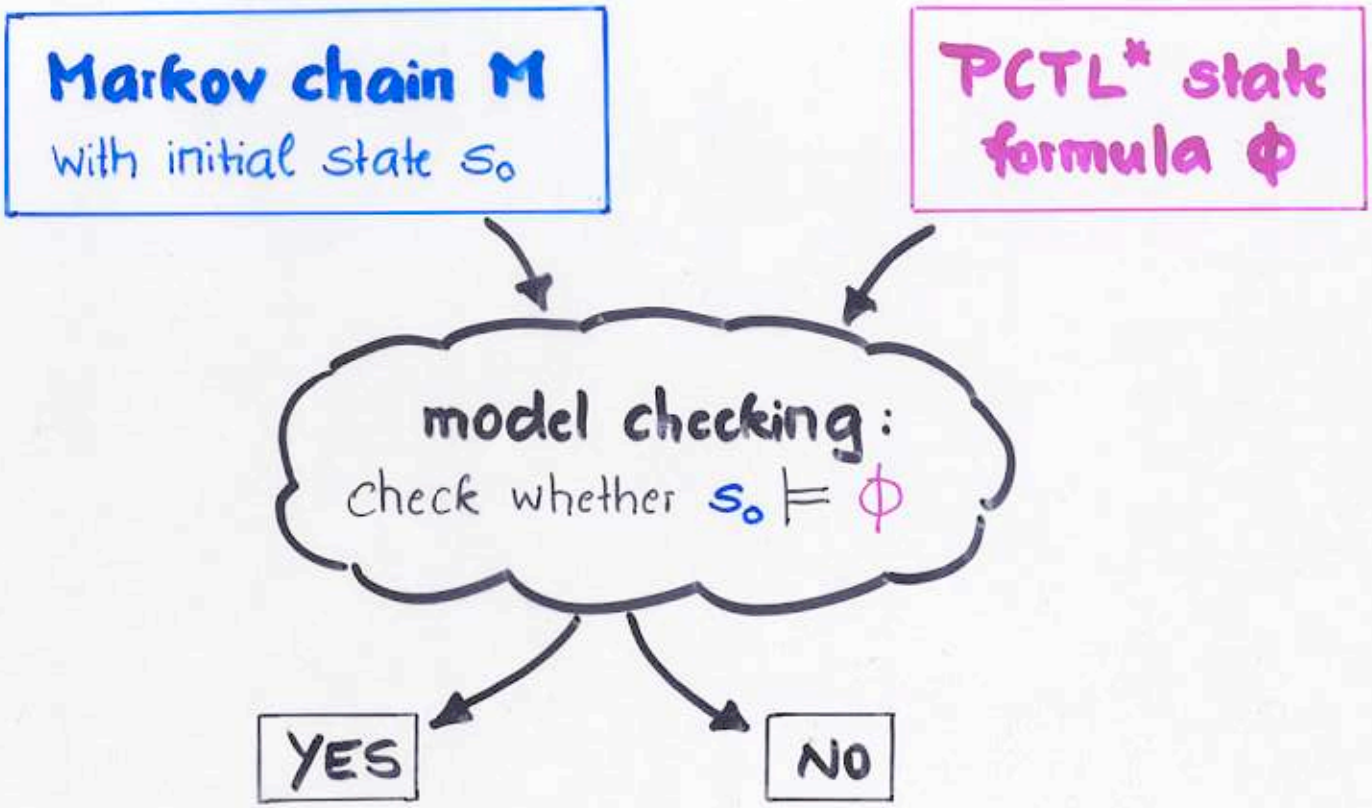
all ids are synchronously passed around the ring.

if there is a unique id then elect the process with the max. unique id,

otherwise repeat

$$Pr_{=1}(\Diamond \text{leader\_elected}), \; Pr_{\geq 0.9}(\bigvee_{i \leq n} X^i \text{ leader\_elected})$$

PCTL* model checking:



Idea: recursively compute $Sat(\psi) = \{ s : s \models \psi \}$ for all sub-state formulae ψ of Φ and

check whether $s_0 \in Sat(\Phi)$

Recursive computation of the satisfaction sets $Sat(\dots)$

$$Sat(true) = S \qquad \text{(state space of } M\text{)}$$

$$Sat(a) = \{ s \in S : a \in L(s) \}$$

$$Sat(\neg \Phi) = S \setminus Sat(\Phi)$$

$$Sat(\Phi_1 \wedge \Phi_2) = Sat(\Phi_1) \cap Sat(\Phi_2)$$

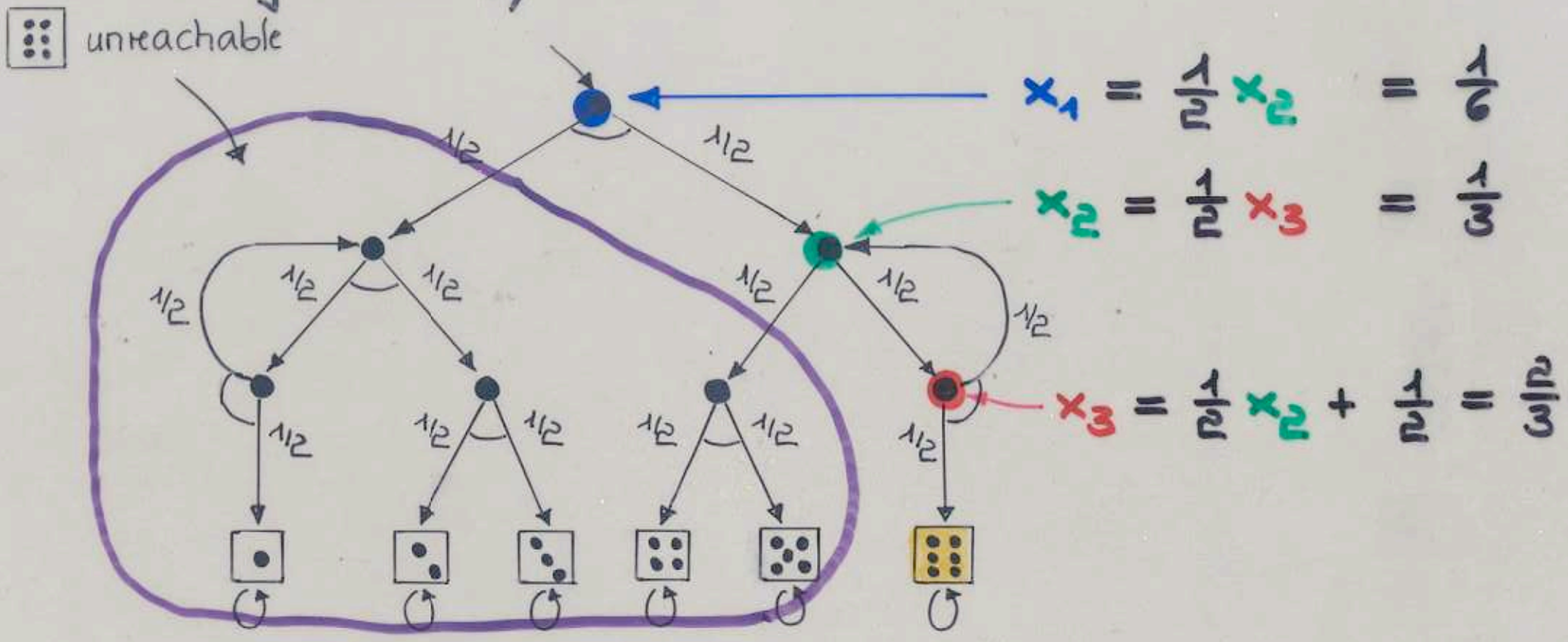
$$Sat(\mathcal{P}_{\geq p}(\varphi)) = \{ s \in S : Pr^M(s, \varphi) \geq p \}$$

Special case: $\varphi = \Diamond \Phi$

compute $x_s = Pr^M(s, \Diamond \Phi)$ by solving a **linear equation system**

Dice example: [Knuth 1970]

Simulating a dice by a coin



$$P_f^M(\diamond\ \boxdot) = x_1 = \frac{1}{6}$$

Computing $x_s = Pr^M(s, \Diamond T)$ for $T \subseteq S$

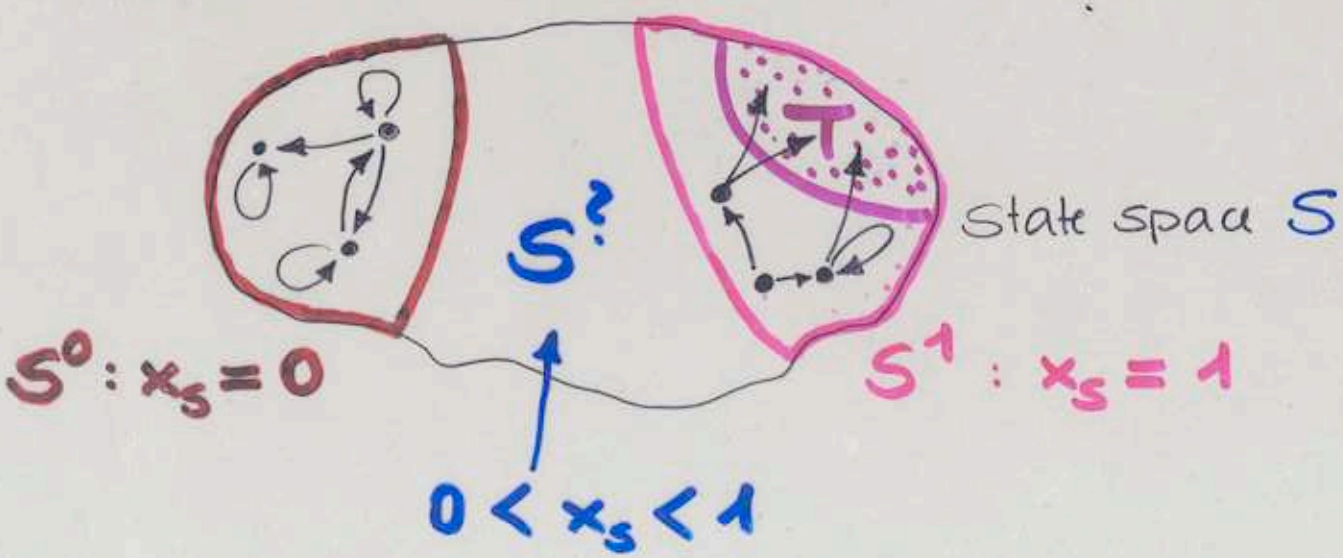
Step 1: Compute S^0 and S^1 by means of graph algorithms

$S^0 = \{ s : T \text{ is not reachable from } s \}$

$S^1 = \{ s : S^0 \text{ is not reachable from } s \}$

Step 2: compute x_s for $s \in S^?$ by solving the linear equ. system:

$$x_s = \sum_{s' \in S^?} P(s,s') \cdot x_{s'} + \sum_{s' \in S^1} P(s,s')$$



$x = Ax + b$

iff

$(I - A)x = b$

↑ non-singular

PCTL

sublogic of PCTL* where only path formulae of the form $X\,\Phi$ and $\Phi_1\,U\,\Phi_2$ are allowed (with state formulae Φ, Φ_1, Φ_2)

model checking:

- recursive computation of Sat(.) for all sub-state formulae
- treatment of $Pr_{\geq p}(X\,\psi)$:

  compute $x_s = \sum_{s' \models \psi} P(s,s')$

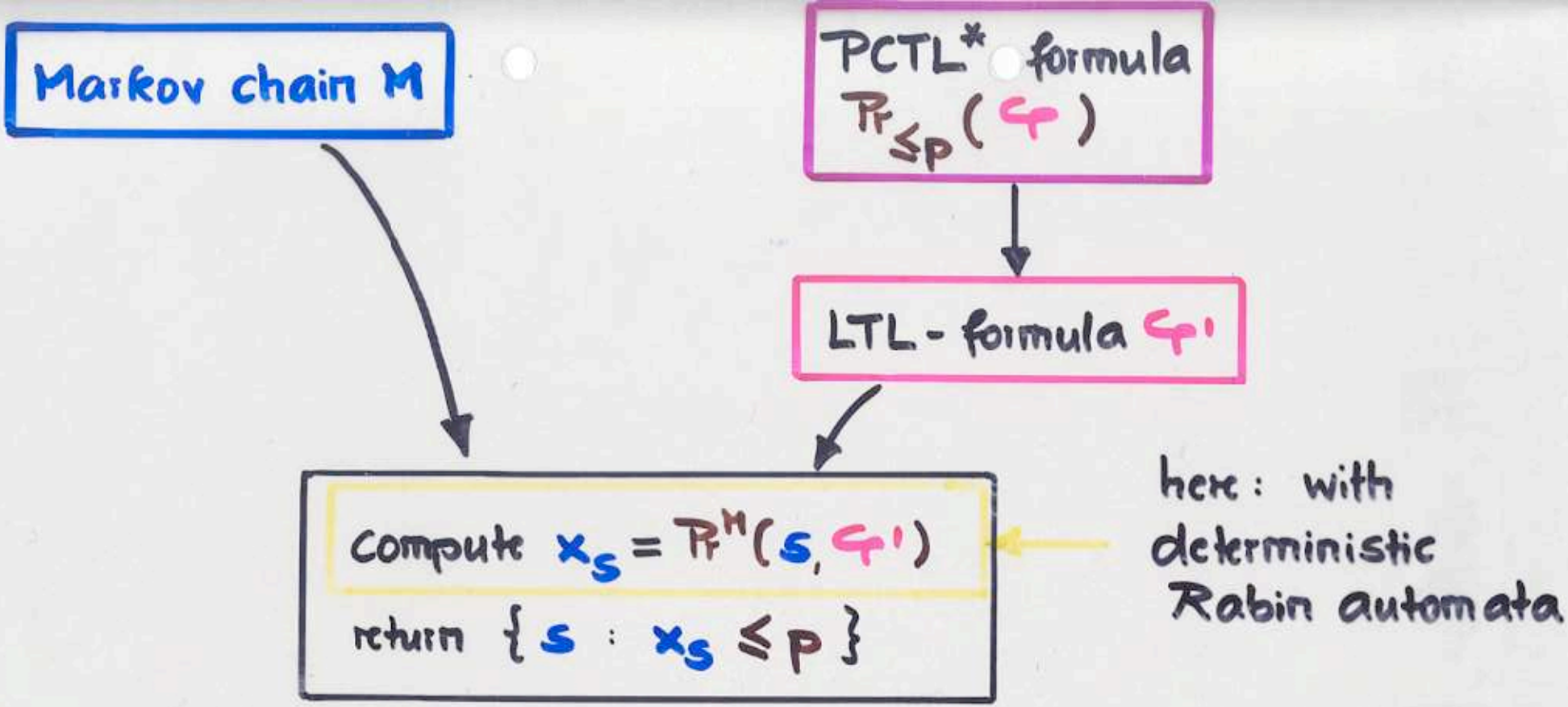
- treatment of $Pr_{\geq p}(\Phi_1\,U\,\Phi_2)$:
  - compute $S^0 = \{ s : s \not\models \exists\, \Phi_1\,U\,\Phi_2 \}$

    and $S^1 = \{ s : \Phi_1\,U\,\Phi_2$ holds with prob. 1 for $s \}$
  - solve the linear equation system

$$x_s = \sum_{s' \in S^?} P(s,s') \cdot x_{s'} + \sum_{s' \models \Phi_2} P(s,s') \quad \text{for } s \in S^?$$

complexity:

$O(\text{poly}(|M|, |\Phi|))$



PCTL* path formula $\rightsquigarrow$ LTL - formula

by replacing each max. state-subformula with a new atomic prop.

e.g. $\Diamond(a \,U\, Pr_{\geq 0.7}(\Box\Diamond b) \wedge X\, Pr_{<0.3}(X\Box c)) \rightsquigarrow \Diamond(a\, U\, d \wedge Xe)$

Deterministic Rabin automata (DRA):

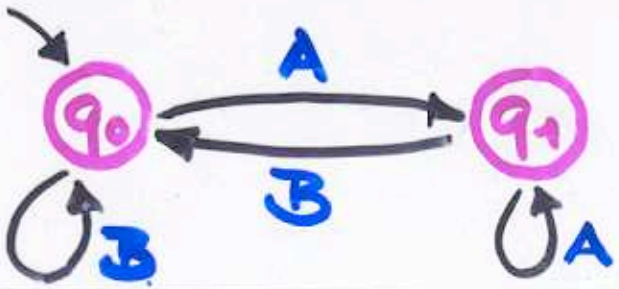
$$\mathcal{A} = (Q, \Sigma, \delta, q_0, \text{Acc})$$

- Q finite state space
- q_0 initial state
- Σ alphabet
- $\delta : Q \times \Sigma \longrightarrow Q$ deterministic transition relation
- acceptance condition $\text{Acc} = \bigvee_{1 \le i \le k} (\Diamond\Box \neg L_i \wedge \Box\Diamond U_i)$

  (where $L_i, U_i \subseteq Q$)

accepted language:

$$\mathcal{L}(\mathcal{A}) = \{ \sigma \in \Sigma^\omega : \text{the run for } \sigma \text{ in } \mathcal{A} \text{ fulfills Acc} \}$$



$$\text{Acc} = \Diamond\Box \neg q_0 \wedge \Box\Diamond q_1$$

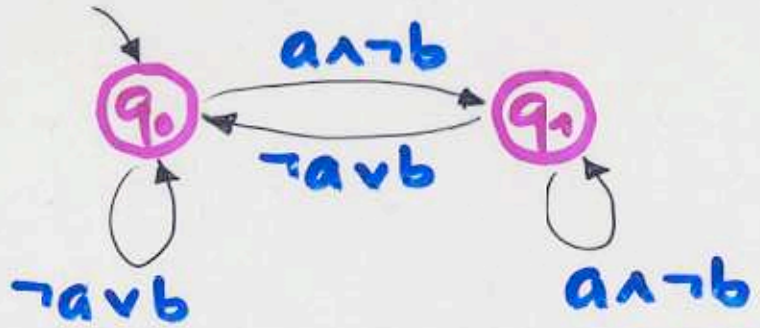
accepted language: $(A+B)^* A^\omega$

Fundamental result:

For each LTL-formula φ there exists a DRA A with the alphabet $\Sigma = 2^{AP}$ s.t. $|A| = O(2^{\exp(|\varphi|)})$ and

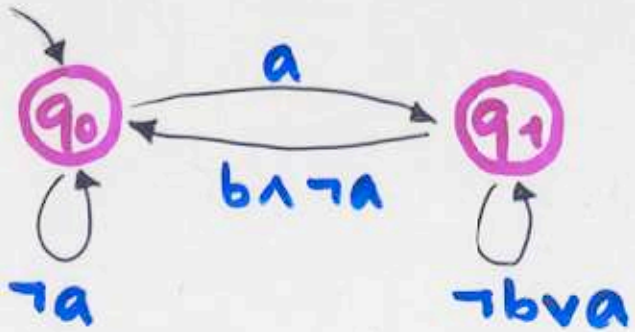
$$\mathcal{L}(A) = \{ \sigma \in \Sigma^{\omega} : \sigma \models \varphi \}$$

Examples: $AP = \{a, b\}$



acc. condition: $\Diamond\Box \neg q_0 \wedge \Box\Diamond q_1$

LTL formula $\Diamond\Box(a \wedge \neg b)$



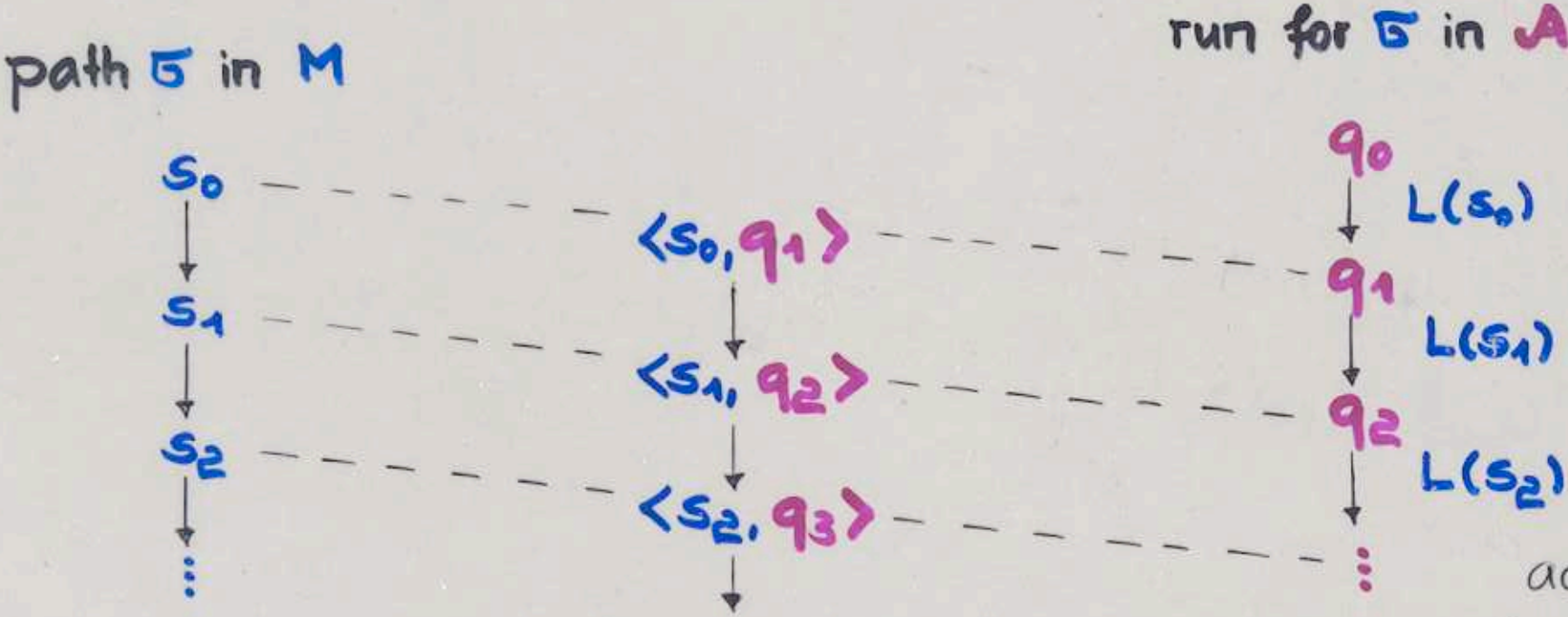
acc. condition: $\Diamond\Box \neg q_1 \wedge \Box\Diamond q_0$

LTL formula $\Box(a \rightarrow \Diamond(b \wedge \neg a)) \wedge \Diamond\Box \neg a$

Markov chain M

dct. Rabin automaton $\mathcal{A}$ for LTL-formula φ

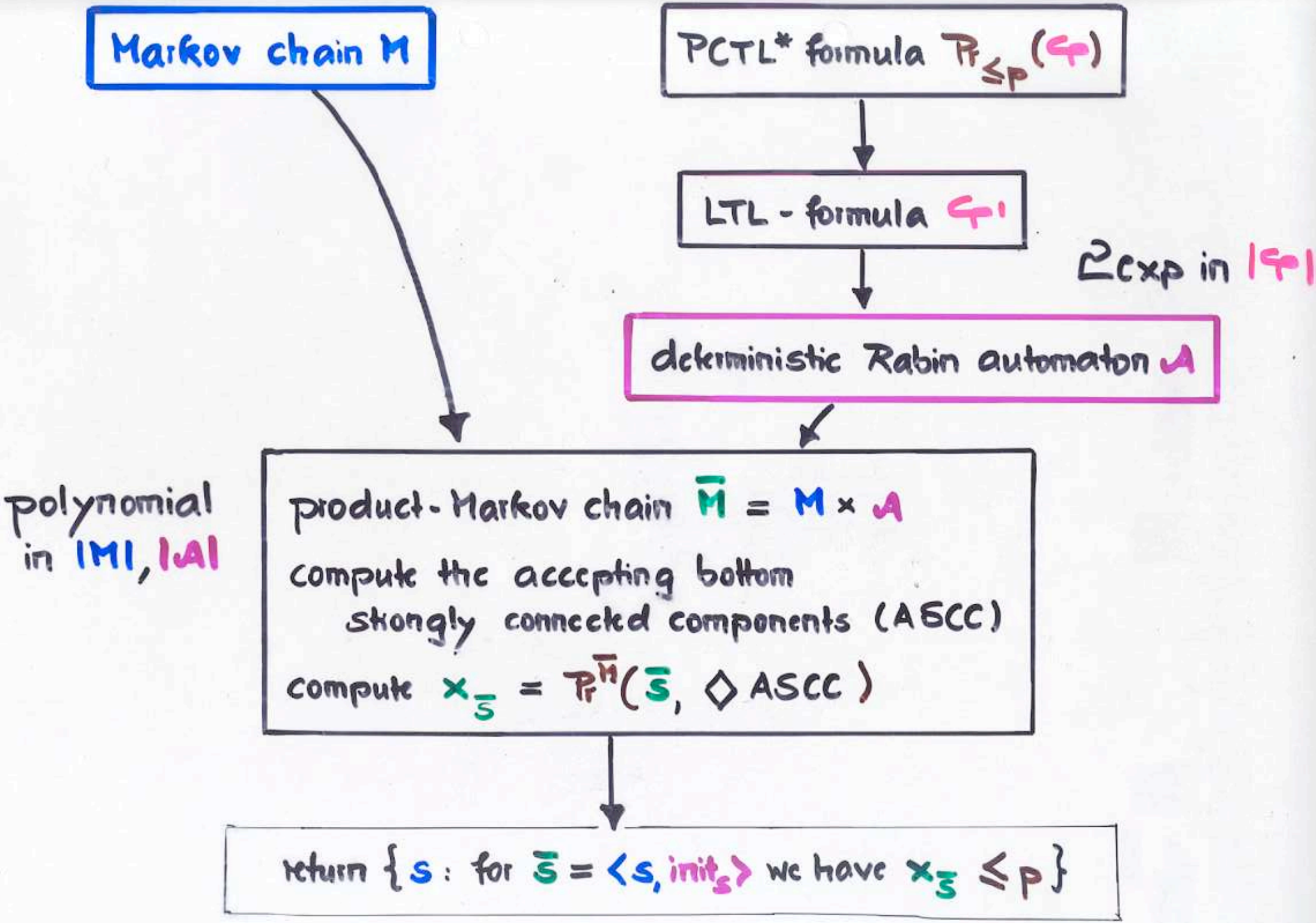
product Markov chain $M \times \mathcal{A}$



acceptance cond. of $\mathcal{A}$

$$Pr^M(s_0, \varphi) = Pr^{M \times \mathcal{A}}\Big(\langle s_0, q_1 \rangle, \bigvee_{1 \le i \le k} (\Diamond\Box \neg L_i \wedge \Box\Diamond U_i)\Big)$$

$$= Pr^{M \times \mathcal{A}}\big(\langle s_0, q_1 \rangle, \Diamond \text{ acc. bottom SCC}\big)$$


Markov chain M
PCTL* formula $Pr_{\leq p}(\varphi)$
LTL - formula φ_1
$2^{exp\ in\ |\varphi_1|}$
deterministic Rabin automaton A
polynomial in |M|, |A|
product-Markov chain $\bar{M} = M \times A$
compute the accepting bottom strongly connected components (ASCC)
compute $x_{\bar{s}} = Pr^{\bar{M}}(\bar{s}, \Diamond ASCC)$
return $\{s : \text{for } \bar{s} = \langle s, init_s \rangle \text{ we have } x_{\bar{s}} \leq p\}$

Markov reward model

- Markov chain with a reward function that assigns a fixed reward $r(s)$ to each state (that will be earned whenever visiting s)

- extension of PCTL* by a reward operator $\mathcal{R}_{\leq r}(\Phi)$

> $s \models \mathcal{R}_{\leq r}(\Phi)$ iff the expected accumulated reward on paths from s to a Φ-state is $\leq r$

- Example: $\mathcal{R}_{\leq 5}(\text{leader\_elected})$
  "the average number of rounds of a leader election protocol is ≤ 5"

- <u>model checking</u>: compute the expected accumulated rewards by solving the linear equation system:

$$x_s = r(s) + \sum_{s'} P(s,s') \, x_{s'} \quad \text{if } s \not\models \Phi$$

and $x_s = 0$ for $s \models \Phi$

Tutorial: Formal verification of stochastic systems

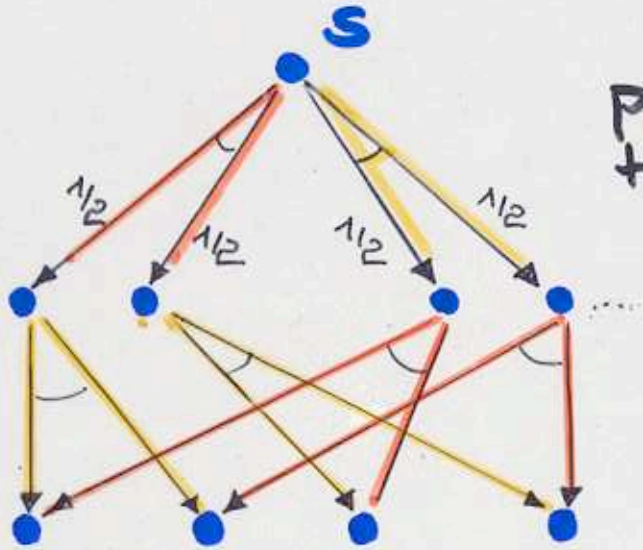
Outline:

- Markov chains and PCTL*
- Markov decision processes and PCTL*
- continuous-time Markov chains and continuous stochastic logic

Markov decision processes (MDP):

- extend Markov chains by nondeterminism
- needed for modelling asynchronous distributed systems by interleaving

process 1 tosses a coin



process 2 tosses a coin

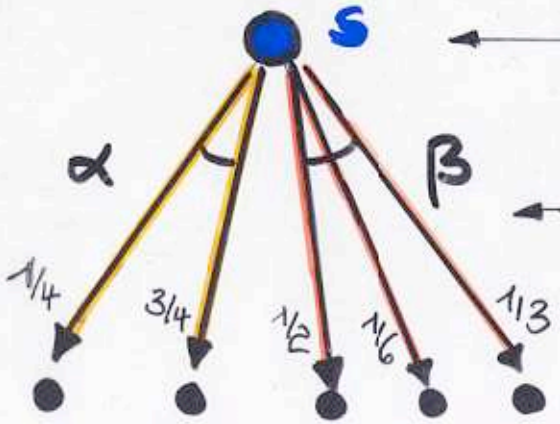
- also needed for abstraction purposes and representing the interface with an unpredictable environment (e.g. human user)

Markov decision process (MDP):

$M = (S, Act, P, AP, L)$ + initial state/distribution

- S finite state space
- Act finite set of actions
- $P : S \times Act \times S \longrightarrow [0,1]$ transition probability function

  s.t. $\forall s \in S \; \forall \alpha \in Act : \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$



← non-det. choice

← prob. choice

↑ α enabled in s

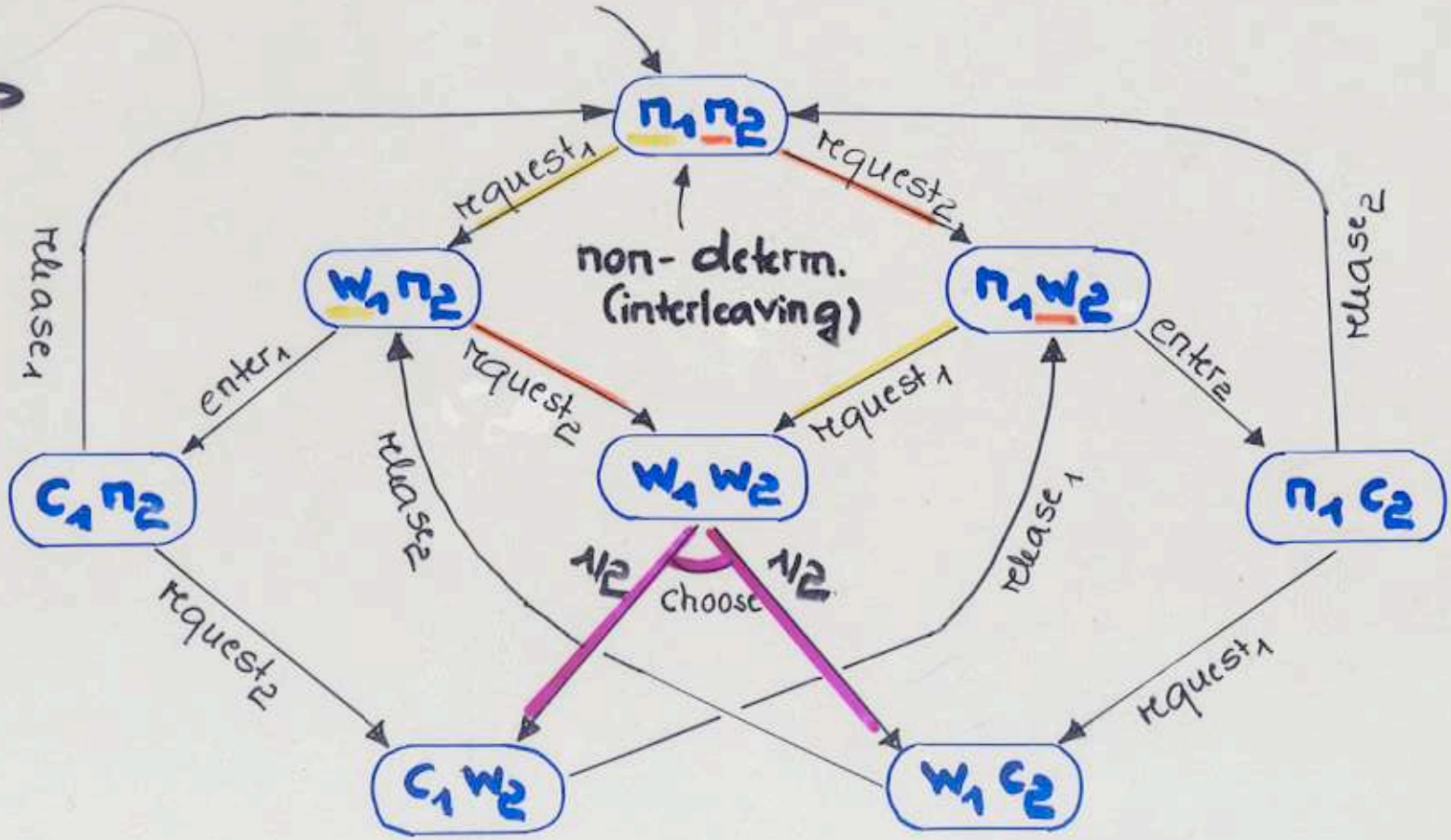
Randomized mutual exclusion protocol:

- 2 concurrent prousses with 3 phases:

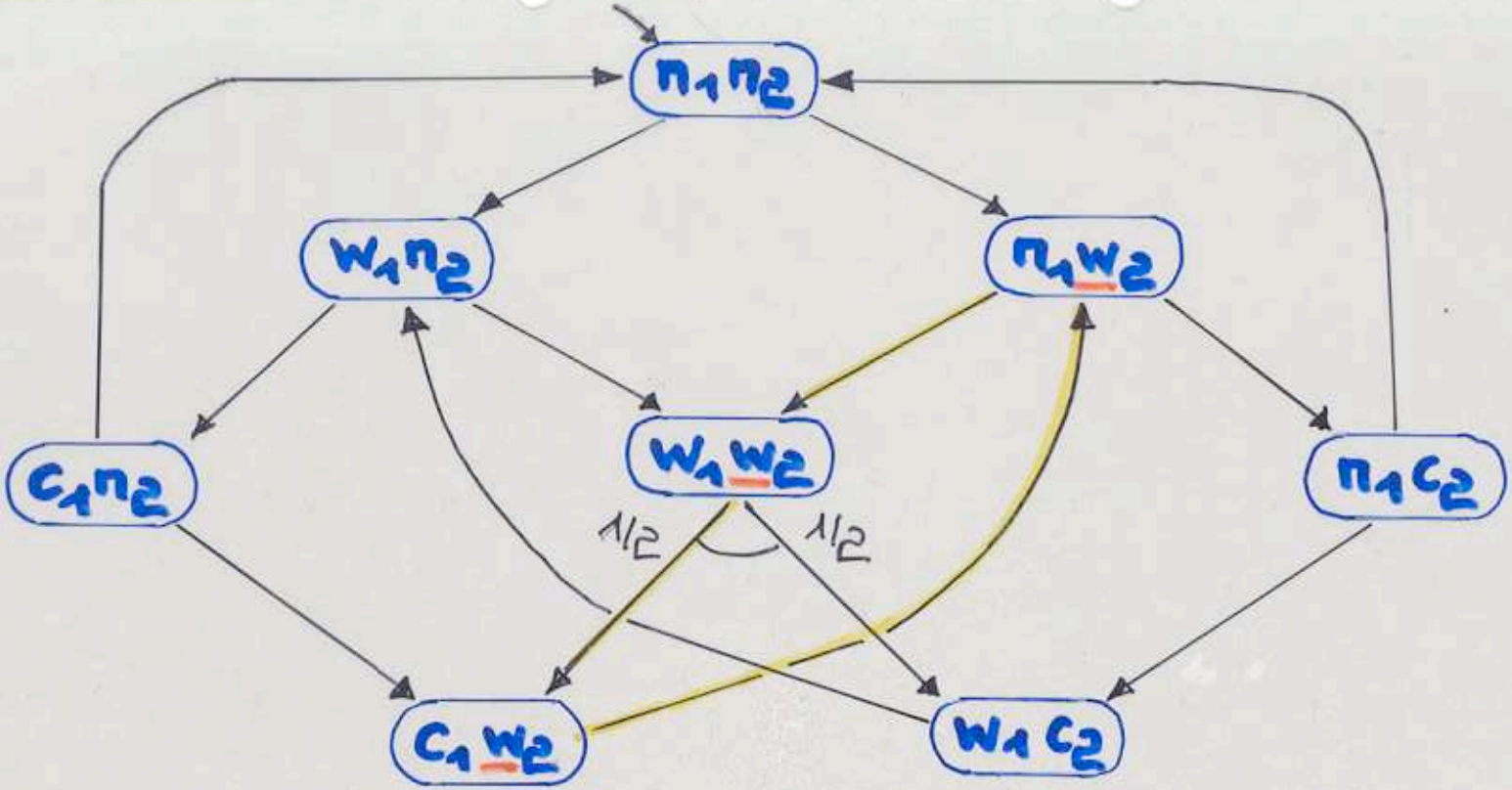
  n_i (non critical actions), w_i (wait), c_i (critical section)

- randomized arbiter: tosses a coin if both processes wait

MDP



Randomized mutual exclusion protocol:



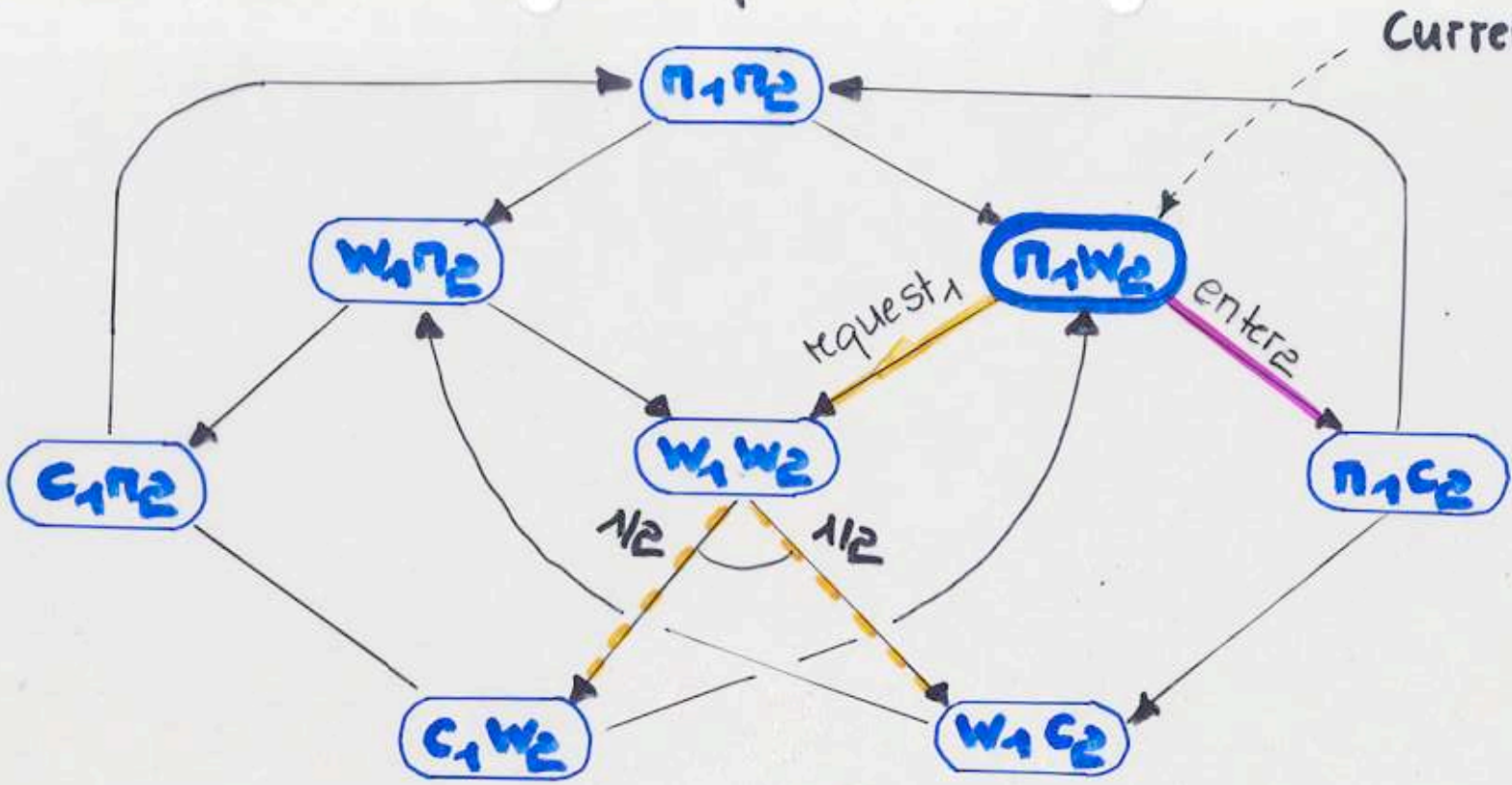
Safety: "the processes are never simultaneously in their critical sections"

Starvation freedom: "each waiting process will eventually enter its crit. sect."

does not hold on all paths, but **almost surely**

if process 2 is waiting: what is the probability that process 2 enters its critical section within the next 3 steps? ... **depends** ...

Randomized mutual exclusion protocol:



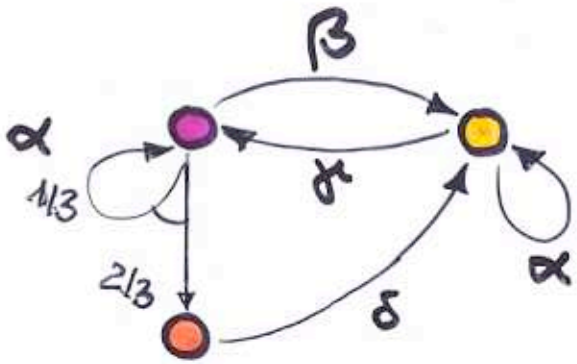
if process 2 is waiting: what is the probability that process 2 enters its critical section within the next 3 steps?

prob. 1/2 for the schedulers which choose process 1 in $\langle n_1, w_2 \rangle$

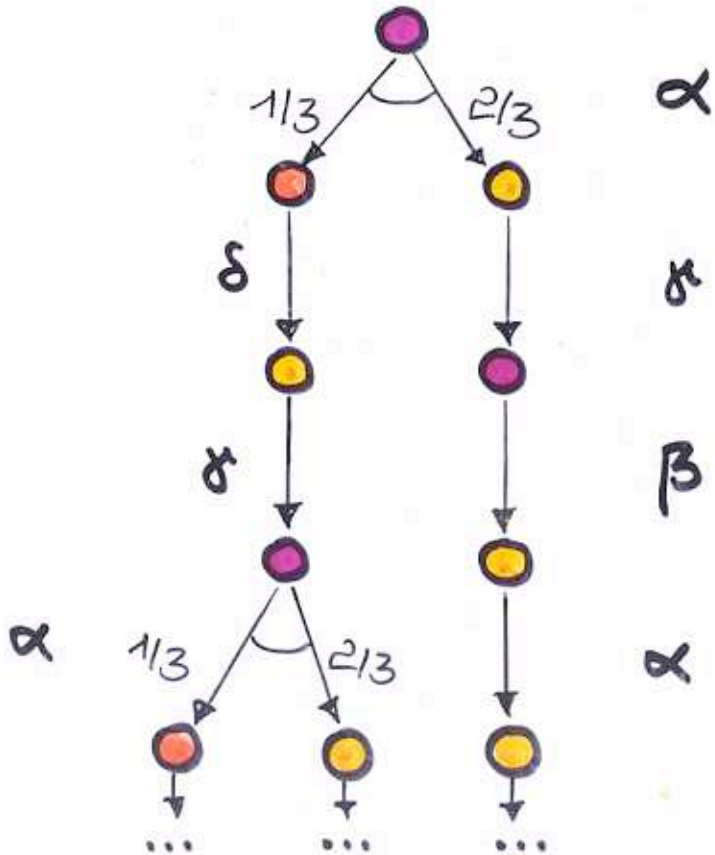
prob. 1 for the schedulers which choose process 2 in $\langle n_1, w_2 \rangle$

Probability measure in MDPs:

- requires resolving the nondeterminism by schedulers
- formally: a scheduler is a function $\mathfrak{D}: S^* \to Act$ s.t. action $\mathfrak{D}(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinit Markov chain

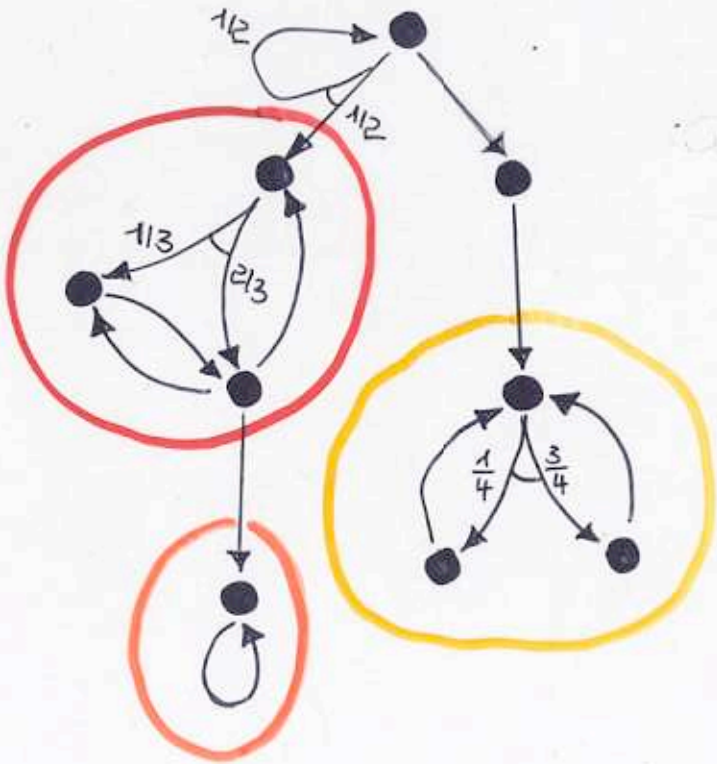


MDP



Fundamental propaty of MDPs: [Alfaro '97]

for all schedulers D, almost surely an **end component** will be reached and all its states visited infinitely often



end component:

strongly connected sub-MDP

(closed under prob. branching)

PCTL* for MDPs:

[Bianco, de Alfaro '95]

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \Pr_{\geq p}(\varphi) \mid \Pr_{\leq p}(\varphi)$$

$$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid X\varphi \mid \varphi_1 \, U \, \varphi_2$$

Interpretation of state formulae over the states of a MDP

$s \models \text{true}$

$s \models a$ iff $a \in L(s)$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models \neg \Phi$ iff $s \not\models \Phi$

$s \models \Pr_{\geq p}(\varphi)$ iff for all schedulers D (selects an action for all finite paths)

$$\Pr^D\{\sigma \in Paths(s) : \sigma \models \varphi\} \geq p$$

prob. measure in the Markov chain induced by D

PCTL* model checking for MDPs:

- recursively compute the satisfaction sets $Sat(.)$ for all sub-state formulae
- treatment of $Pr_{\leq p}(\varphi)$:
  - compute $Pr^{M}_{max}(s, \varphi) = \max_{D} Pr^{D}\{\sigma \in Paths(s) : \sigma \models \varphi\}$ for all states s
  - return $\{ s \in S : Pr^{M}_{max}(s, \varphi) \leq p \}$

Special case: $\varphi = \Diamond \Phi$ (reachability property)

compute $Pr^{M}_{max}(s, \Diamond \Phi)$ by solving a linear program

general case:

construct a DRA for φ and compute max. reachability prob. in the product

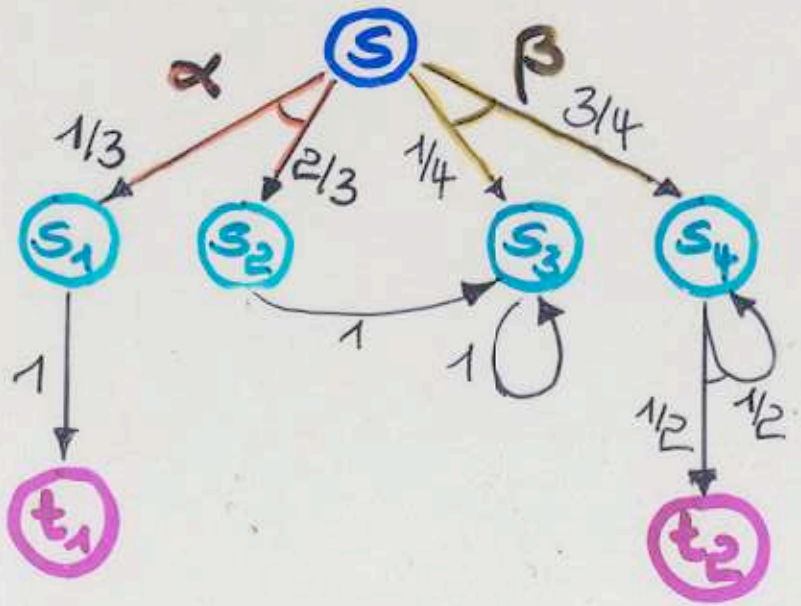
Maximal reachability probabilities

$$x_s = Pr_{max}(s, \Diamond T) = \max_{D} Pr^{D}(s, \Diamond T)$$

- the vector $(x_s)_{s \in S}$ is the least solution in $[0,1]$ of:

$$x_s = 1 \quad \text{if} \quad s \in T$$

$$x_s = \max \left\{ \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} \;:\; \alpha \text{ enabled in } s \right\} \quad \text{if} \quad s \notin T$$



$$x_s = \max \left\{ \frac{1}{3} x_{s_1} + \frac{2}{3} x_{s_2} \,,\; \frac{1}{4} x_{s_3} + \frac{3}{4} x_{s_4} \right\}$$

$$x_{s_1} = x_{t_1} = 1$$

$$\left. \begin{aligned} x_{s_2} &= x_{s_3} \\ x_{s_3} &= x_{s_3} \end{aligned} \right\} = 0$$

$$\begin{aligned} x_{s_4} &= \frac{1}{2} x_{s_4} + \frac{1}{2} x_{t_2} \\ &= \frac{1}{2} x_{s_4} + \frac{1}{2} \\ &= 1 \end{aligned}$$

Maximal reachability probabilities

$$x_s = Pr_{max}(s, \Diamond T)$$

- the vector (x_s) is the least solution in $[0,1]$ of:

$$x_s = 1 \quad \text{if } s \in T$$

$$x_s = \max\left\{ \sum_{s'} P(s,\alpha,s') \cdot x_{s'} : \alpha \text{ enabled in } s \right\} \quad \text{if } s \notin T$$

and the unique solution of

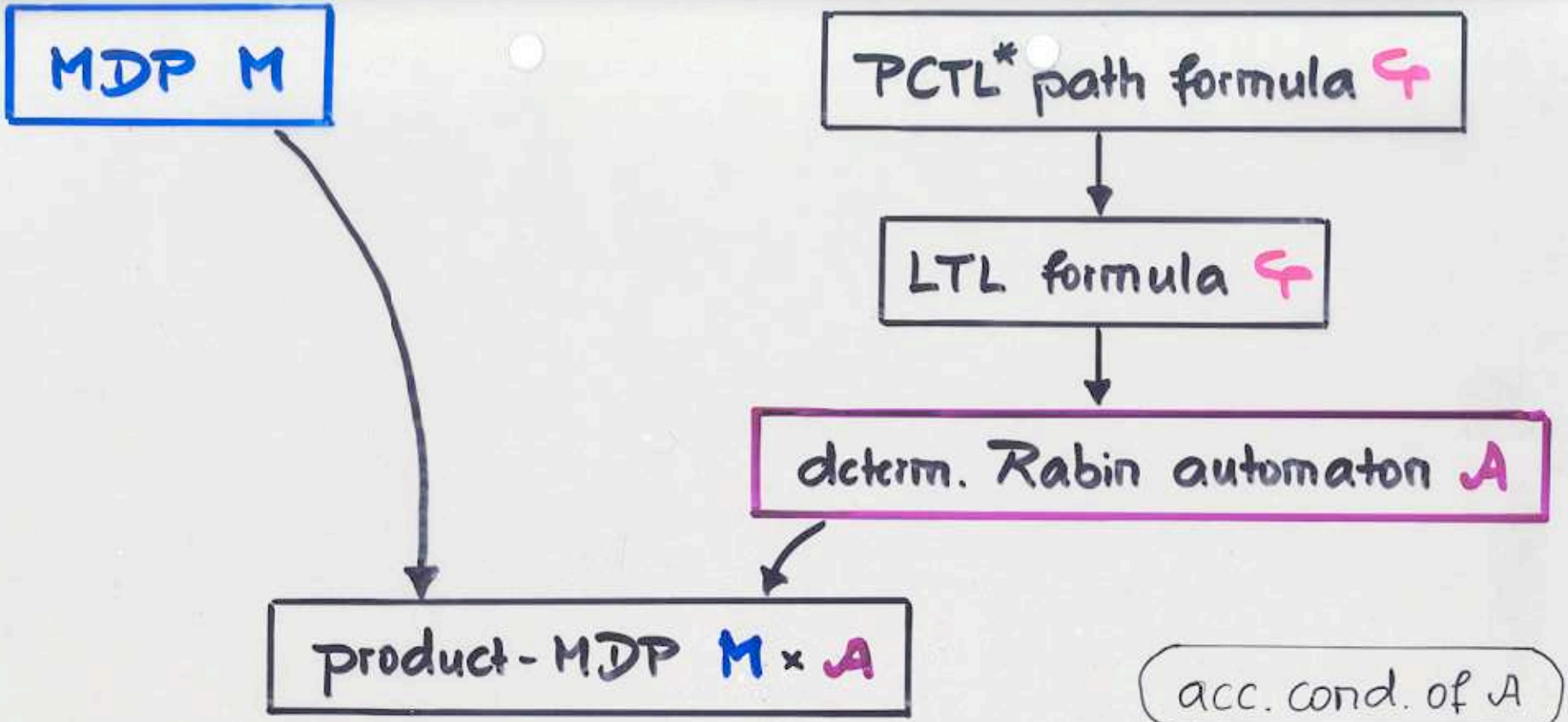
$$x_s = 1 \quad \text{if } \cancel{s \in T} \ \exists \text{ sched. } D \text{ s.t. } Pr^D(s, \Diamond T) = 1$$

$$x_s = 0 \quad \text{if } T \text{ is not reachable from } s$$

$$x_s \geqslant \sum_{s'} P(s,\alpha,s') \cdot x_{s'} \quad \text{otherwise, } \alpha \text{ enabled in } s$$

---

where $\sum_s x_s$ is minimal



acc. cond. of A

$$Pr^{M}_{max}(s, \varphi) = Pr^{M \times A}_{max}\left(\langle s, init_s \rangle, \bigvee_i (\Diamond\Box \neg L_i \wedge \Box\Diamond U_i)\right)$$

$$= Pr^{M \times A}_{max}\left(\langle s, init_s \rangle, \Diamond \text{accepting end comp.}\right)$$

end component C s.t.

$C \cap L_i = \emptyset \wedge C \cap U_i \neq \emptyset$ for some i

	PCTL model checking	PCTL* model checking
Markov chains	graph algorithms + linear equation systems PTIME	PSPACE-complete [Vardi/Wolper '86]
Markov decision process	graph algorithms + linear programs PTIME	2EXP-complete [Courcoubetis/Yannakakis '88]

Tools, e.g.:

- PRISM (Birmingham): PCTL for MC/MDPs, uses MTBDDs
- LIQUOR (Bonn): LTL for MC/MDPs, uses partial order red.
- RAPTURE (Twente): PCTL for MC/MDPs, uses abstraction/refinement

Tutorial: Formal verification of stochastic systems

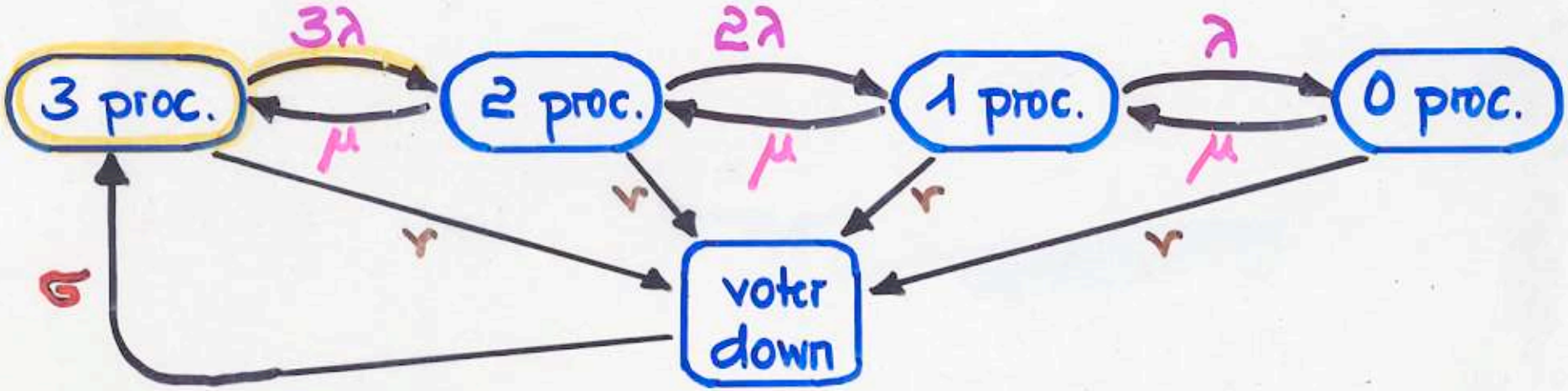
Outline:

- Markov chains and PCTL*
- Markov decision processes and PCTL*
- Continuous-time Markov chains and continuous stochastic logic

Continuous-time Markov chains

- can be viewed as transition systems with **rates** ← positive real numbers

Example : Triple modular redundant system



λ = failure rate for each processor

μ = rate for the repairing time

ν = failure rate for the voter

σ = rate for the total restart

in average : λ failures / time unit (for each processor)

⟶ 3λ failures / time unit (for 3 processors)

Continuous-time Markov chains (CTMC)

$M = (S, R, AP, L)$ + initial distribution / state

where
- S state space (here: finite)
- $R : S \times S \longrightarrow \mathbb{R}_{\geqslant 0}$ rate matrix
- AP set of atomic propositions
- $L : S \longrightarrow 2^{AP}$ labeling function

Many modelling formalisms for CTMCs:

- stochastic Petri nets [Molloy '77]
- stochastic process algebras [Herzog et al, Hillston '93]
- stochastic activity networks [Meyer & Sanders '85]
- stochastic automata networks [Plateau '85]

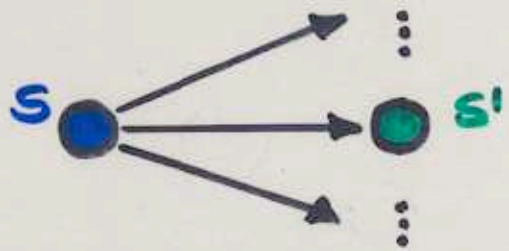
⋮

Continuous time Markov chain : $M = (S, R, AP, L)$

$R(s, s') = \lambda$ rate of the transition $s \to s'$

$$\text{Prob}\{\text{transition } s \to s' \text{ is enabled within } t \text{ time units}\} = 1 - e^{-\lambda t}$$

Race condition : competition between the outgoing transitions from s



total rate :

$$E(s) = \sum_{s'} R(s, s')$$

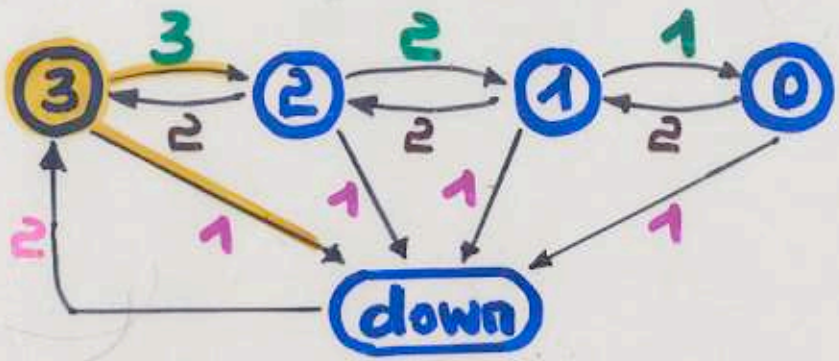
$$\text{Prob}\{\text{transition } s \to s' \text{ is taken within } t \text{ time units}\} = P(s, s') \cdot (1 - e^{-E(s) \cdot t})$$

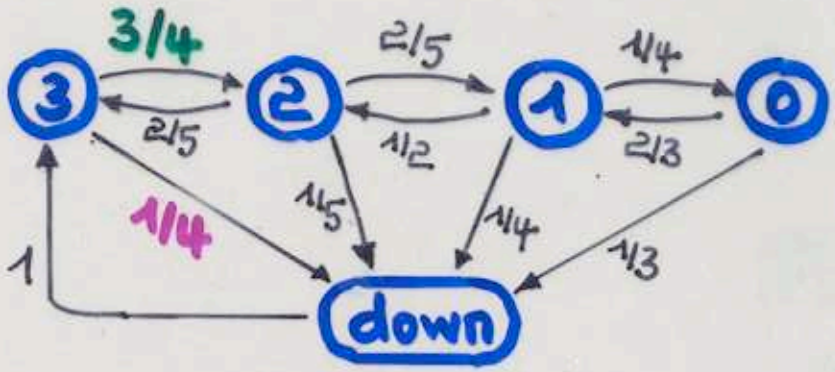
time-abstract trans. prob.

$$P(s, s') = \frac{R(s, s')}{E(s)}$$

prob. for leaving s within t time units

Continuous-time Markov chain ⇝ embedded DTMC





Rate matrix:

$$R = \begin{pmatrix} 0 & 3 & 0 & 0 & 1 \\ 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} \leftarrow 4 \\ \leftarrow 5 \\ \leftarrow 4 \\ \leftarrow 3 \\ \leftarrow 2 \end{matrix}$$

total rates

Probability matrix

$$P = \begin{pmatrix} 0 & 3/4 & 0 & 0 & 1/4 \\ 2/5 & 0 & 2/5 & 0 & 1/5 \\ 0 & 1/2 & 0 & 1/4 & 1/4 \\ 0 & 0 & 2/3 & 0 & 1/3 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \leftarrow 1 \\ \leftarrow 1 \\ \leftarrow 1 \\ \leftarrow 1 \end{matrix}$$

time-abstract transition probabilities

Paths in a CTMC:

$$s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} s_2 \xrightarrow{t_3} \dots \quad \text{where } s_i \in S,\ t_i \in \mathbb{R}_{>0}$$

- t_i = sojourn time in state s_{i-1}
- state at time t : s_i if $t_1 + \dots + t_{i-1} \leq t < t_1 + \dots + t_{i-1} + t_i$
- time divergence : $\sum_i t_i$ diverges

Probability measure for fixed initial distribution π_0

- consider the σ-algebra generated by the cylinder sets

$$C(s_0 \xrightarrow{I_1} s_1 \xrightarrow{I_2} \dots \xrightarrow{I_n} s_n) \quad \text{where } I_i \subseteq \mathbb{R}_{\geq 0} \text{ intervals}$$

$$= \text{set of } \underbrace{\text{paths}}_{\text{maximal}}\ s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} s_n \xrightarrow{t_{n+1}} s_{n+1} \xrightarrow{t_{n+2}} \dots \quad \text{with } t_i \in I_i$$

- probability : $\pi_0(s_0) \cdot \prod_{0 \leq i < n} R(s_i, s_{i+1}) \cdot \int_{I_{i+1}} e^{-E(s_i)x}\, dx$

Elementary measures for CTMCs:

- transient stat probabilities:

$$\pi(s, s', t) = Pr_s\{\sigma \in Paths : \sigma(t) = s'\}$$

Pr_s: unique starting state s; $\sigma(t)$: state at time t

- steady stat probabilities:

$$\pi(s, s') = \lim_{t \to \infty} \pi(s, s', t)$$

- limit exists for finite CTMCs
- is given by:

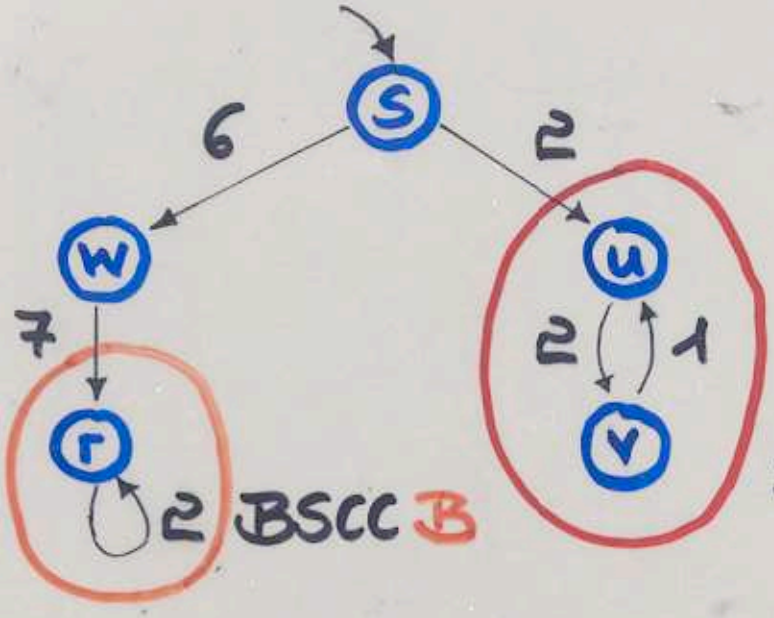
$$\pi(s, s') = Pr(s, \Diamond B) \cdot \pi_B(s') \quad \begin{cases} \text{if } s' \in B \text{ and} \\ B \text{ is a bottom SCC} \end{cases}$$

Steady state prob. inside B (independent on starting st.)

$$\pi(s, s') = 0 \quad \text{if } s' \text{ is not contained in a bottom SCC}$$

Computing steady state probabilities

... by graph analysis + linear equation system



$\pi(s, w) = 0$

$\pi(s, s) = 0$

$Pr(s, \Diamond C) = 1/4$

$\pi(s, r) = Pr(s, \Diamond B) = 3/4$

$\pi(s, v) = \frac{1}{4} \cdot \frac{2}{3}$

$\pi(s, u) = \frac{1}{4} \cdot \frac{1}{3}$

Steady state prob. inside C:

$\pi_C(v) = 2\,\pi_C(u)$

$\pi_C(v) + \pi_C(u) = 1$

$\Longrightarrow \pi_C(v) = 2/3,\ \pi_C(u) = 1/3$

Computing steady state probabilities:

- in strongly connected Markov chains (with arbitrary init. distribution/state)

> the vector $\overline{\pi} = (\pi(s))_{s \in S}$ is the unique solution of:
>
> $$\overline{\pi} \cdot Q = 0 \quad , \quad \sum_s \pi(s) = 1$$
>
> where $Q = R - \mathrm{diag}(E)$ "generator matrix"

i.e. $\sum_u \pi(u) \cdot R(u,s) = \pi(s) \cdot E(s)$ "flow balance"

- in arbitrary Markov chains:

$$\pi(s_0, s) = Pr(s_0, \Diamond B) \cdot \pi_B(s)$$

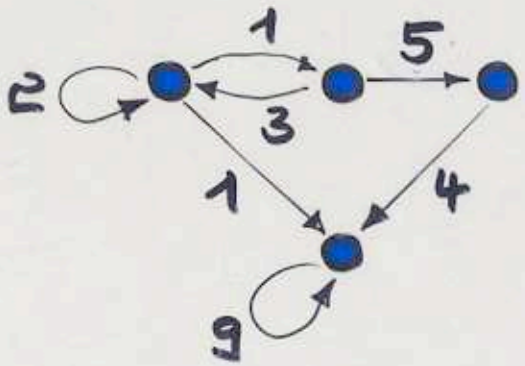
viewed as starting state

if B is a bottom SCC that contains s

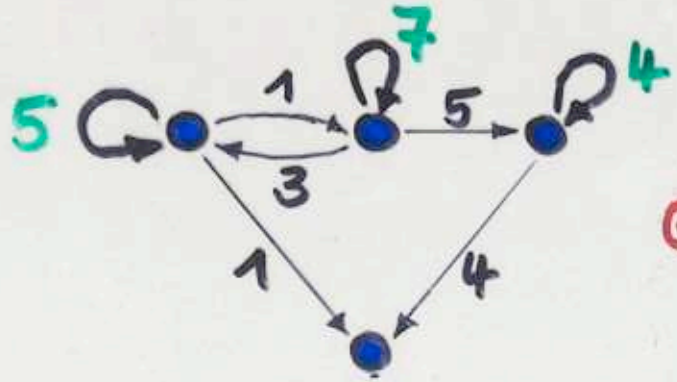
Generator matrix

Let $M = (S, R, AP, L)$ be a CTMC.

- $R : S \times S \longrightarrow \mathbb{R}_{\geq 0}$ rate matrix
- total rate vector $E : S \longrightarrow \mathbb{R}$, $E(s) = \sum_u R(s,u)$
- generator matrix $Q = R - \mathrm{diag}(E)$



$$R = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 3 & 0 & 5 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 9 \end{pmatrix} \qquad E = \begin{pmatrix} 4 \\ 8 \\ 4 \\ 9 \end{pmatrix}$$



$$Q = \begin{pmatrix} -2 & 1 & 0 & 1 \\ 3 & -8 & 5 & 0 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

does not depend on $R(s,s)$

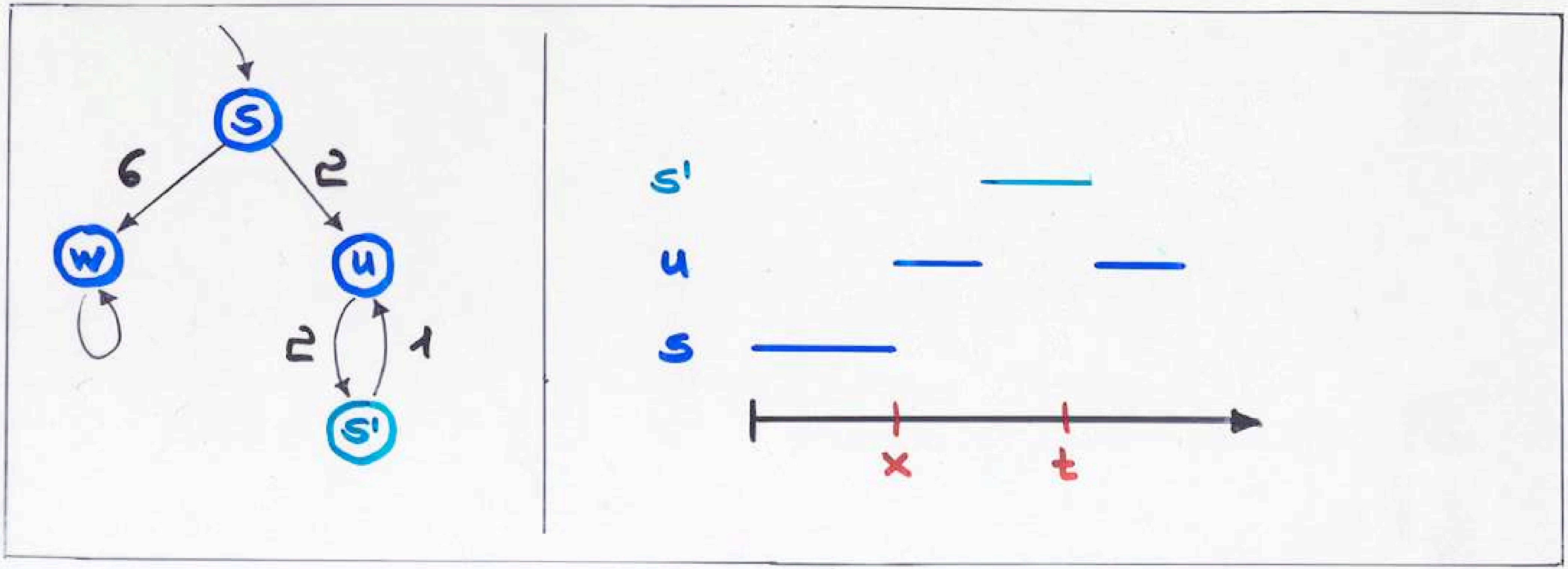
Transient state probabilities:

State at time t

$$\pi(s, s', t) = Pr_s \{ \sigma \in Paths(s) : \sigma(t) = s' \}$$

$$= \sum_u \int_0^t P(s, u, x) \cdot \pi(u, s', t-x) \, dx$$

density of the distribution for $s \longrightarrow u$



Transient state probabilities:

- $$\pi(s, s', t) = \sum_u \int_0^t P(s, u, x) \cdot \pi(u, s', t-x)\, dx$$

  Volterra integral equation

- The matrix $F(t) = \left(\pi(s, s', t) \right)_{s, s'}$ fulfills:

  $$\frac{d}{dt} F(t) = F(t) \cdot Q = Q \cdot F(t)$$

  Chapman/Kolmogoroff differential equations

  where $Q = R - \mathrm{diag}(E)$ is the generator matrix

- Thus: $$F(t) = e^{Qt} = \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!}$$

numeric computation of this infinite sum is difficult

(since the matrices Q^n become non-sparse with positive/negative entries)

Special solution for generator matrices of CTCMs: **uniformization**

Uniformization

[Jensen 1953]

for computing transient state prob. in CTMCs

- the matrix $\mathcal{F}(t) = (\pi(s, s', t))_{s,s'}$ is given by:

$$\mathcal{F}(t) = \sum_{n=0}^{\infty} \frac{(Qt)^n}{n!}$$ where Q generator matrix

- uniformization: modify the CTMC by adding self-loops and increasing the rates of self-loops s.t. all states have the same total rate λ

let P be the time-abstract prob. matrix of unif(M). Then:

$$\mathcal{F}(t) = \sum_{n=0}^{\infty} e^{-\lambda t} \cdot \frac{(\lambda t)^n}{n!} \cdot P^n \qquad \text{and } P = I + \frac{1}{\lambda} \cdot Q$$

Uniformization:

$$F(t) = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} P^n$$

where P = prob. matrix in unif(M)

Poisson probabilities:
probability for exactly n steps within t time units in unif(M)

$P^n(s, s')$ = probability for reaching s' from s with exactly n steps

vector/matrix multipl.
$P^{n-1}(s, \cdot) \cdot P$

For fixed starting state s:

$$\pi(s, \cdot, t) = \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \cdot P^n(s, \cdot)$$

For given $\varepsilon > 0$: choose k s.t. $\sum_{n=0}^{k} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \geq 1 - \varepsilon$

Then:

$$\left\| \pi(s, \cdot, t) - \sum_{n=0}^{k} e^{-\lambda t} \frac{(\lambda t)^n}{n!} P^n(s, \cdot) \right\| < \varepsilon$$

Continuous stochastic logic (CSL):

[Aziz, Sanwal, Singhal, Brayton '96]
[Baier, Haverkort, Hermanns, Katoen '00]

state formulae:

$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}_{\geq p}(\varphi) \mid \mathcal{S}_{\geq p}(\Phi)$$

path formulae:

$$\varphi ::= X^I \Phi \mid \Phi_1 U^I \Phi_2$$

also other prob. bounds, e.g., $\leq p$, $< p$

$\mathcal{P}_{\ldots}(\varphi)$: probability operator as in PCTL

$\mathcal{S}_{\ldots}(\Phi)$: steady state operator

asserts bounds for the prob. for Φ on the "long run"

Continuous stochastic logic (CSL):

state formulae:

$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathbb{P}_{\geq p}(\varphi) \mid \mathbb{S}_{\geq p}(\varphi) \mid \dots$$

path formulae:

$$\varphi ::= X^I \Phi \mid \Phi_1 \, U^I \, \Phi_2 \qquad \text{where } I \subseteq \mathbb{R}_{\geq 0} \text{ time-interval}$$

time-bounded next step

time-bounded until

derived operators:

- $\Diamond^I \Phi := \text{true} \; U^I \Phi$ — time-bounded eventually
- $\mathbb{P}_{\geq p}(\Box^I \Phi) := \mathbb{P}_{\leq 1-p}(\Diamond^I \neg\Phi)$ — time-bounded always
- transient-state operator: $\mathbb{P}_{\dots}(\Diamond^{=t} \Phi)$

Examples:

$S_{>0.8}(\text{qos})$

"the prob. for an acceptable level of quality of service achieved in the long run is >0.8"

$Pr_{<0.3}(\Diamond^{\leq 2}\, \text{full})$

"the prob. for the queue becoming full within 2 hours is <0.3"

$Pr_{>0.9}(\Box^{\leq 11}\, \neg \text{repair})$

"with prob. >0.9 no repairs will be needed in the next 11 hours"

$Pr_{\geq 0.9}(\neg \text{down}\ U^{[5,10]}\ S_{>0.8}(up_2 \vee up_3))$

"with prob. ≥ 0.9 a state will be reached between 5 and 10 time units (without going down), which guarantees the system to be operational with 2 or 3 processors when the system is in equilibrium"

$Pr_{\geq 0.5}(\Diamond^{=3}\, \text{down})$

"transient-state prob. for the system being down at time point 3 is ≥ 0.5"

Semantics of CSL: Let s be a state of a CTMC M.

$s \models true$

$s \models a$ iff $a \in L(s)$

$s \models \neg \Phi$ iff $s \not\models \Phi$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models Pr_{\geq p}(\varphi)$ iff $Pr^M(s, \varphi) \geq p$

↑ probability measure of the set of paths σ with $\sigma \models \varphi$ when s is viewed as the unique starting state

$s \models S_{\geq p}(\Phi)$ iff $\sum_{s' \models \Phi} \pi(s, s') \geq p$

↑ steady state prob. for Φ and starting state s

Satisfaction set: $Sat(\Phi) = \{ s : s \models \Phi \}$

Semantics of CSL (path formulae):

Let $\sigma = s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} s_2 \xrightarrow{t_3} \dots$ be a path in CTMC M.

$\sigma \models X^I \Phi$ iff $t_1 \in I$ and $s_1 \models \Phi$

$\sigma \models \Phi_1 \, U^I \Phi_2$ iff there ex. $t \in I$ such that:

- $\sigma(t) \models \Phi_2$
- $\sigma(t') \models \Phi_1$ for all $t' < t$

$\sigma \models \Diamond^I \Phi$ iff there ex. $t \in I$ s.t. $\sigma(t) \models \Phi$

$\sigma \models \Box^I \Phi$ iff $\sigma(t) \models \Phi$ for all $t \in I$

Recall: $\sigma(t)$ = state at time t in σ

CSL model checking:

given: CTMC M, state s, CSL state formula Φ

question: Does $s \models \Phi$ hold ?

algorithm:

- recursively compute $Sat(\cdot)$ for all subformulae of Φ
- steady state operator $S_{\geq p}(\Phi)$:
  - compute the steady state probabilities $\pi(s, s')$
  - for all states s, compute

$$\pi(s, \Phi) = \sum_{s' \models \Phi} \pi(s, s')$$

  - return $\{ s : \pi(s, \Phi) \geq p \}$

- probability operator : ...

CSL model checking: probability operator $\mathbb{P}_{\sim p}(\varphi)$

- qualitative properties: graph analysis as in CTL / PCTL, e.g.,

$$s \models \mathbb{P}_{>0}(X^I \Phi) \quad \text{iff} \quad s \models \exists X \Phi \qquad (\text{if } I \neq \emptyset)$$

$$s \models \mathbb{P}_{>0}(\Phi_1 \, U^I \, \Phi_2) \quad \text{iff} \quad s \models \exists \, \Phi_1 U \Phi_2$$

- quantitative properties: prob. bound $\sim p$ where $p \in]0,1[$

  - observation: $\mathbb{P}_{\sim p}(X^I \Phi) \equiv \mathbb{P}_{\sim p}(X^{cl(I)} \Phi)$

    (and analogous equivalence for U^I)

  - next step: $\varphi = X^{[a,b]} \Phi$

$$\mathbb{P}(s, \varphi) = \sum_{s' \models \Phi} P(s,s') \cdot \left(e^{-E(s)\,a} - e^{-E(s)\,b} \right)$$

return $\{ s : \mathbb{P}(s, \varphi) \sim p \}$

CSL model checking: probabilistic operator $P_{\sim p}(\Phi_1 \, U^I \, \Phi_2)$

- the question whether $s \models P_{>p}(\Phi_1 \, U^I \, \Phi_2)$ is decidable [Aziz et al '96]
- here: numerical computation of $P_r(s, \Phi_1 \, U^I \, \Phi_2)$ up to some tolerance ε [Baier et al '00]

Case 1: $I = [0, t]$

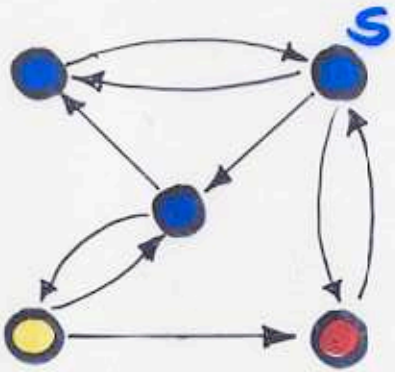
- modify M as follows:
  - make all Φ_2-states absorbing
  - make all states s with $s \not\models \exists \, \Phi_1 \, U \, \Phi_2$ absorbing
- compute $P_r^{M'}(s, \Diamond^{\leq t} \Phi_2)$

$$= \sum_{s' \models \Phi} \pi^{M'}(s, s', t)$$

by a transient analysis in M' (e.g. uniformization)

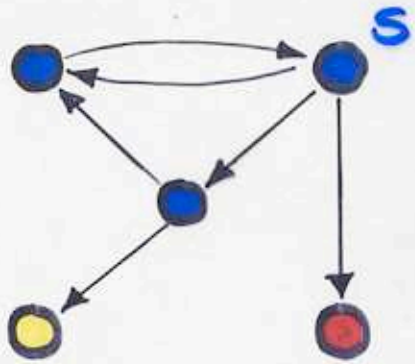
Example: $Pr_{\sim p}(\text{blue } U^{\leq 3} \text{ red})$

M:



transformation ⇝

M':



$$Pr^{M}(s, \text{blue } U^{\leq 3} \text{ red}) = Pr^{M'}(s, \Diamond^{\leq 3} \text{red})$$

$$= Pr^{M'}(s, \Diamond^{=3} \text{red})$$

transient-state probability

CSL model checking: computation of $Pr^M(s, \Phi_1 \, U^I \, \Phi_2)$

case 1: $I = [0, t]$ ...

case 2: $I = [t, t']$ where $0 < t < t'$

$$Pr^M(s, \Phi_1 \, U^{[t,t']} \, \Phi_2)$$

$$= \sum_{s' \models \Phi_1} \underbrace{Pr^M(s, \Phi_1 \, U^{=t} \, s')}_{} \cdot \underbrace{Pr^M(s', \Phi_1 \, U^{\leq t'-t} \, \Phi_2)}_{\text{case 1}}$$

= transient-state prob. $\pi^{M'}(s, s', t)$ in the CTMC that results from M by making all states u with $u \not\models \Phi_1$ absorbing

CSL model checking: computation of $Pr^M(s, \Phi_1 \, U^I \, \Phi_2)$

case 1: $I = [0, t]$

case 2: $I = [t, t']$ where $0 < t < t'$

case 3: $I = [t, t]$ where $t > 0$

$$Pr^M(s, \Phi_1 \, U^{=t} \, \Phi_2) = \sum_{s' \models \Phi_1 \wedge \Phi_2} \pi^{M'}(s, s', t)$$

where M' results from M by making all states u with $u \not\models \Phi_1$ absorbing

case 4: $I = [t, \infty[$ where $t > 0$

$$Pr^M(s, \Phi_1 \, U^{\geq t} \, \Phi_2) = \sum_{s' \models \Phi_1} \pi^{M'}(s, s', t) \cdot \underbrace{Pr^M(s, \Phi_1 \, U \, \Phi_2)}_{\text{as for PCTL}}$$

Summary: CSL model checking

- propositional logic : as in CTL
- steady state operator:

  graph analysis + linear equation systems
- time-bounded next step: matrix-vector multiplication
- time-bounded until:

  model transformation + transient analysis (uniformization)
- worst-case complexity:

$$\mathcal{O}\left(|\Phi| \cdot \left(|R| \cdot \text{unif.rate} \cdot t_{max} + |S|^3 \right) \right)$$

Tools: c.g. PRISM (Birmingham)

ETMCC (Erlangen)

Markov reward models :

- CTMC with a reward structure that assigns to each state s reward $r(s) \geq 0$
- when staying t time units in state s then reward $r(s) \cdot t$ is earned

CSRL

[Baier, Haverkort, Hermanns, Katoen '00]

- extends CSL by path formulae with constraints on the accumulated reward

$$\Phi_1 \; U^{I}_{J} \; \Phi_2 \quad \text{where} \quad I \text{ time-interval}, \; J \text{ reward interval}$$

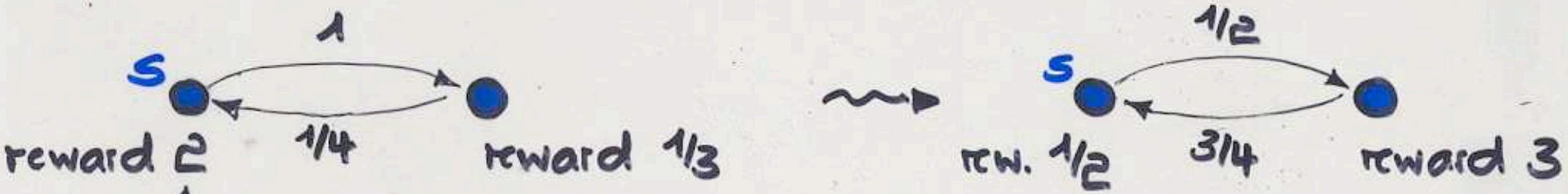
and other operators for reasoning average rewards

Duality of time and rewards:

given: CTMC $M = (S, R, AP, L, r)$ with reward structure

dual CTMC: $M^{-1} = (S, R', AP, L, r')$ where

$$R'(s,u) = \frac{R(s,u)}{r(s)} \qquad r'(s) = \frac{1}{r(s)}$$



average sojourn time: 1

average reward earned when visiting s: $2 \cdot 1 = 2$

average sojourn time: 2

average reward earned when visiting s: $2 \cdot 1/2 = 1$

Duality of time and rewards:

- given: CTMC $M = (S, R, AP, L, r)$ with reward structure

  dual CTMC: $M^{-1} = (S, R', AP, L, r')$ where

$$R'(s,u) = \frac{R(s,u)}{r(s)}, \qquad r'(s) = \frac{1}{r(s)}$$

- given: CSRL formula Φ

  dual CSRL formula Φ^{-1}: obtained by swapping time and reward constraints, i.e. $U^{I}_{J} \rightsquigarrow U^{J}_{I}$

> $s \models \Phi$ in M iff $s \models \Phi^{-1}$ in M^{-1}

... yields a technique for model checking formulae $Pr_{\sim p}(\Phi_1 U^{J}_{I} \Phi_2)$

CSRL model checking

- main procedure as for CSL
- treatment of formulae $\mathbb{P}_{\sim p}(\Phi_1 \, U_J \, \Phi_2)$:

  CSL model checking of the dual CTMC
- treatment of formulae $\mathbb{P}_{\sim p}(\Phi_1 \, U^I_J \, \Phi_2)$:

  ... difficult ...

  various methods: [Cloth '05]
  - approximation with Erlang distributions
  - discretization
  - uniformization

  ⋮

Conclusion:

- model checking of Markov chains and MDPs relies on combinations of
  - graph algorithms
  - automata based approaches for linear time properties
  - linear equation systems, linear programs
  - numerical algorithms for transient analysis
- various tools are available, e.g.,
  PRISM, ETMCC, LIQUOR, APNN Toolbox, VESTA, ...
- active research field

Recent developments:

- probabilistic timed automata
- continuous-time Markov decision processes
- MDPs with reward and discounting
- stochastic games
- abstraction techniques for stochastic models
- continuous-space stochastic models
- integration into the software design cycle
  (e.g. stochastic extensions of UML notations)
- model checking infinite stochastic models
  (e.g. push down automata, lossy channel systems)

  ⋮