An Overview of Probabilistic Model Checking

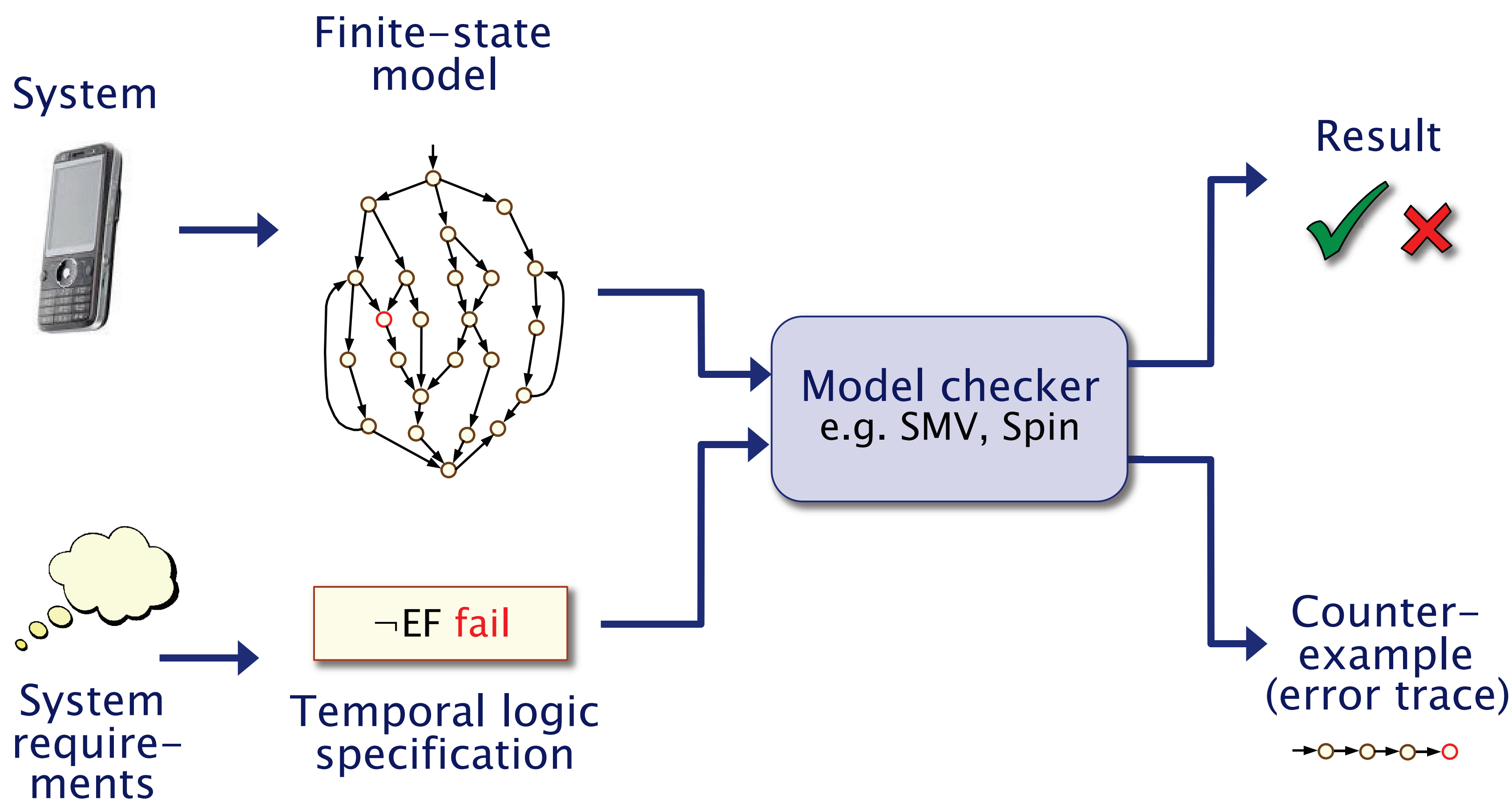
Presented by
Mahsa Varshosaz

FMV 92

Probabilistic model checking

- Probabilistic model checking…
  - is a formal verification technique for modelling and analysing systems that exhibit probabilistic behaviour
- Formal verification…
  - is the application of rigorous, mathematics-based techniques to establish the correctness of computerised systems

Verification via model checking

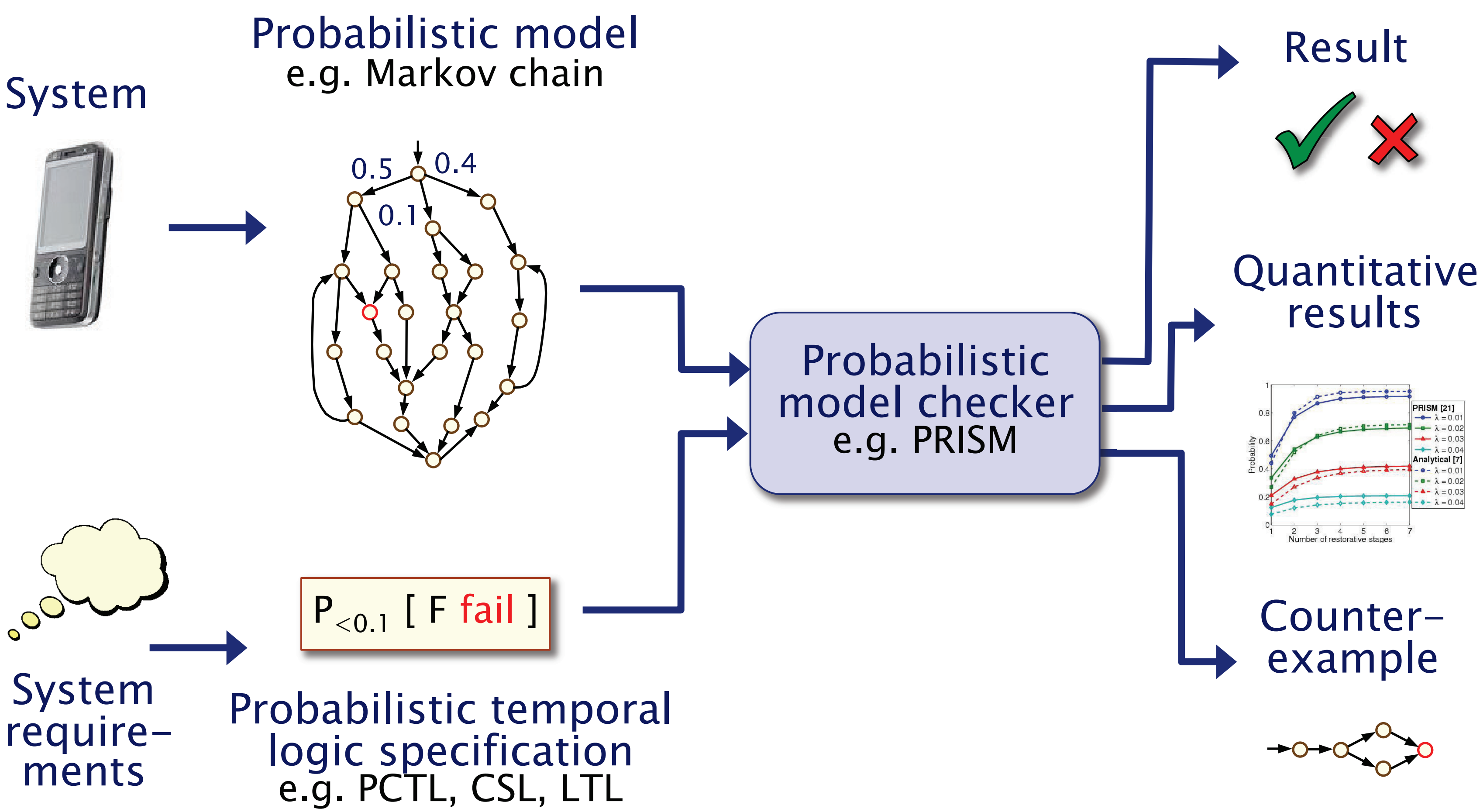


New challenges for verification

- Many properties other than correctness are important
- Need to guarantee…
  - safety, reliability, performance, dependability
  - resource usage, e.g. battery life
  - security, privacy, trust, anonymity, fairness
  - and much more…

- Quantitative, as well as qualitative requirements:
  - "how reliable is my car's Bluetooth network?"
  - "how efficient is my phone's power management policy?"
  - "is my bank's web-service secure?"

Probabilistic model checking



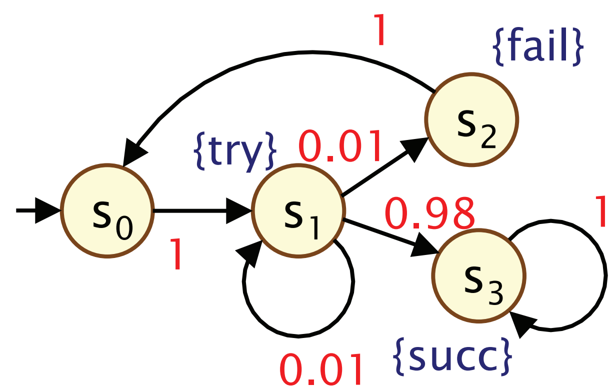
Probabilistic model checking inputs

- Models: variants of Markov chains
  - discrete-time Markov chains (DTMCs)
    - discrete time, discrete probabilistic behaviours only
  - continuous-time Markov chains (CTMCs)
    - continuous time, continuous probabilistic behaviours
  - Markov decision processes (MDPs)
    - DTMCs, plus nondeterminism
- Specifications
  - informally:
    - "probability of delivery within time deadline is …"
    - "expected time until message delivery is …"
    - "expected power consumption is …"
  - formally:
    - probabilistic temporal logics (PCTL, CSL, LTL, PCTL*, …)
    - e.g. $P_{<0.05}$ [F err/total>0.1], $P_{=?}$ [$F^{\leq t}$ reply_count=k]

Discrete-time Markov chains

- State-transition systems augmented with probabilities

- States
  - set of states representing possible configurations of the system being modelled
- Transitions
  - transitions between states model evolution of system's state; occur in discrete time-steps
- Probabilities
  - probabilities of making transitions between states are given by discrete probability distributions

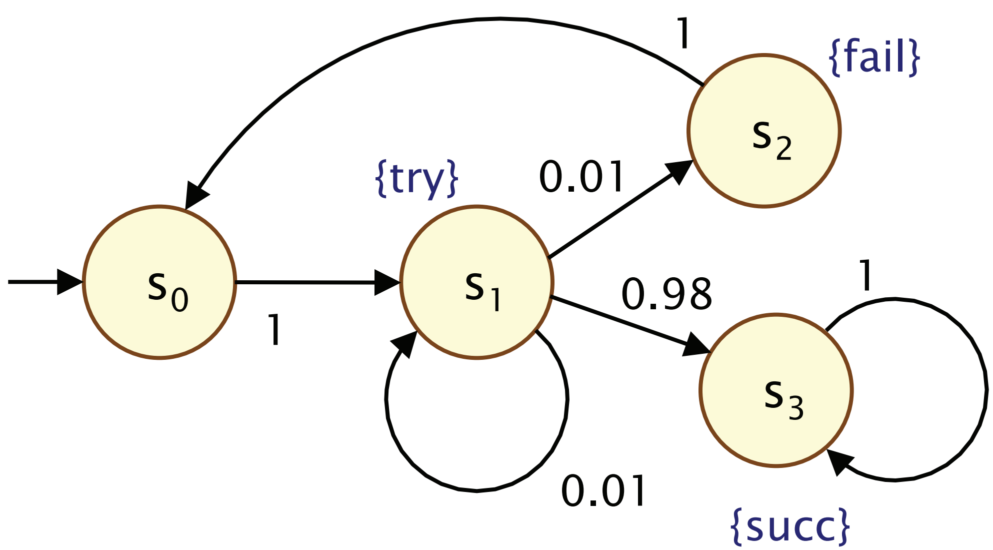


Markov property

- If the current state is known, then the future states of the system are independent of its past states
- i.e. the current state of the model contains all information that can influence the future evolution of the system
- also known as “memorylessness”

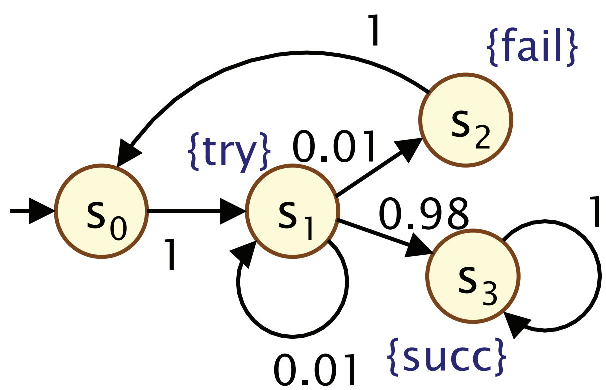
Simple DTMC example

- Modelling a very simple communication protocol
  - after one step, process starts trying to send a message
  - with probability 0.01, channel unready so wait a step
  - with probability 0.98, send message successfully and stop
  - with probability 0.01, message sending fails, restart



Discrete-time Markov chains

- Formally, a DTMC D is a tuple $(S, s_{init}, \mathbf{P}, L)$ where:
  - S is a set of states (“state space”)
  - $s_{init} \in S$ is the initial state
  - $\mathbf{P} : S \times S \rightarrow [0,1]$ is the transition probability matrix
    where $\Sigma_{s' \in S}\, \mathbf{P}(s,s') = 1$ for all $s \in S$
  - $L : S \rightarrow 2^{AP}$ is function labelling states with atomic propositions (taken from a set AP)



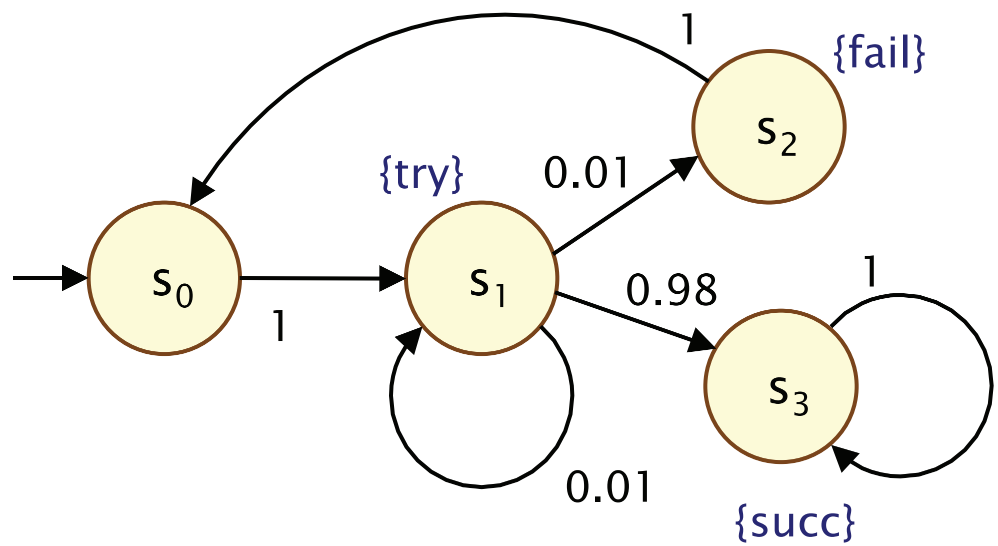
Simple DTMC example

$D = (S, s_{init}, \mathbf{P}, L)$

$S = \{s_0, s_1, s_2, s_3\}$
$s_{init} = s_0$

$AP = \{try, fail, succ\}$
$L(s_0) = \emptyset$,
$L(s_1) = \{try\}$,
$L(s_2) = \{fail\}$,
$L(s_3) = \{succ\}$

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



PCTL

- Temporal logic for describing properties of DTMCs
  - PCTL = Probabilistic Computation Tree Logic [HJ94]
  - essentially the same as the logic pCTL of [ASB+95]

- Extension of (non-probabilistic) temporal logic CTL
  - key addition is probabilistic operator P
  - quantitative extension of CTL's A and E operators

- Example
  - $\mathrm{send} \rightarrow P_{\geq 0.95}\ [\ F^{\leq 10}\ \mathrm{deliver}\ ]$
  - "if a message is sent, then the probability of it being delivered within 10 steps is at least 0.95"

PCTL syntax

- PCTL syntax:

  ψ is true with probability ~p

  – ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | $P_{\sim p}$ [ψ] (state formulae)

  – ψ ::= $X\ \phi$ | $\phi\ U^{\leq k}\ \phi$ | $\phi\ U\ \phi$ (path formulae)

  "next" "bounded until" "until"

  – where a is an atomic proposition, $p \in [0,1]$ is a probability bound, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

- A PCTL formula is always a state formula
  – path formulae only occur inside the P operator

PCTL semantics for DTMCs

- Semantics for non-probabilistic operators same as for CTL:
  - $s \vDash \phi$ denotes "s satisfies ϕ" or "ϕ is true in s"
  - $\omega \vDash \psi$ denotes "ω satisfies ψ" or "ψ is true along ω"

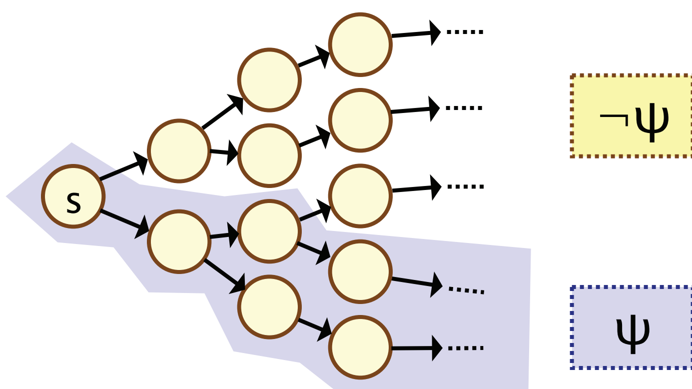
- For a state s of a DTMC $(S, s_{init}, \mathbf{P}, L)$:
  - $s \vDash \text{true}$ always
  - $s \vDash a \Leftrightarrow a \in L(s)$
  - $s \vDash \phi_1 \wedge \phi_2 \Leftrightarrow s \vDash \phi_1$ and $s \vDash \phi_2$
  - $s \vDash \neg\phi \Leftrightarrow s \nvDash \phi$

$U^{\leq k}$ not in CTL (but could easily be added)

- For a path ω of a DTMC $(S, s_{init}, \mathbf{P}, L)$:
  - $\omega \vDash X\,\phi \Leftrightarrow \omega(1) \vDash \phi$
  - $\omega \vDash \phi_1\ U^{\leq k}\ \phi_2 \Leftrightarrow \exists i \leq k$ such that $\omega(i) \vDash \phi_2$ and $\forall j<i,\ \omega(j) \vDash \phi_1$
  - $\omega \vDash \phi_1\ U\ \phi_2 \Leftrightarrow \exists k \geq 0$ s.t. $\omega(k) \vDash \phi_2$ and $\forall i<k\ \omega(i) \vDash \phi_1$

PCTL semantics for DTMCs

- Semantics of the probabilistic operator P
  - informal definition: $s \vDash P_{\sim p} [\ \psi\ ]$ means that "the probability, from state s, that ψ is true for an outgoing path satisfies ~p"
  - example: $s \vDash P_{<0.25} [\ X\ fail\ ] \Leftrightarrow$ "the probability of atomic proposition fail being true in the next state of outgoing paths from s is less than 0.25"
  - formally: $s \vDash P_{\sim p} [\psi] \Leftrightarrow \mathrm{Prob}(s, \psi) \sim p$
  - where: $\mathrm{Prob}(s, \psi) = Pr_s \{\ \omega \in \mathrm{Path}(s) \mid \omega \vDash \psi\ \}$



$\mathrm{Prob}(s, \psi) \sim p$?

PCTL examples

- $P_{<0.05}$ [F err/total>0.1]
  - "with probability at most 0.05, more than 10% of the NAND gate outputs are erroneous?"
- $P_{\geq 0.8}$ [$F^{\leq k}$ reply_count=n]
  - "the probability that the sender has received n acknowledgements within k clock-ticks is at least 0.8"
- $P_{<0.4}$ [$\neg fail_A$ U $fail_B$]
  - "the probability that component B fails before component A is less than 0.4"
- $\neg$oper $\rightarrow$ $P_{\geq 1}$ [F ($P_{>0.99}$ [$G^{<=100}$ oper])]
  - "if the system is not operational, it almost surely reaches a state from which it has a greater than 0.99 chance of staying operational for 100 time units"

PCTL model checking for DTMCs

- Algorithm for PCTL model checking [CY88,HJ94,CY95]
  - inputs: DTMC $D=(S,s_{init},P,L)$, PCTL formula ϕ
  - output: $Sat(\phi) = \{ s \in S \mid s \vDash \phi \}$ = set of states satisfying ϕ

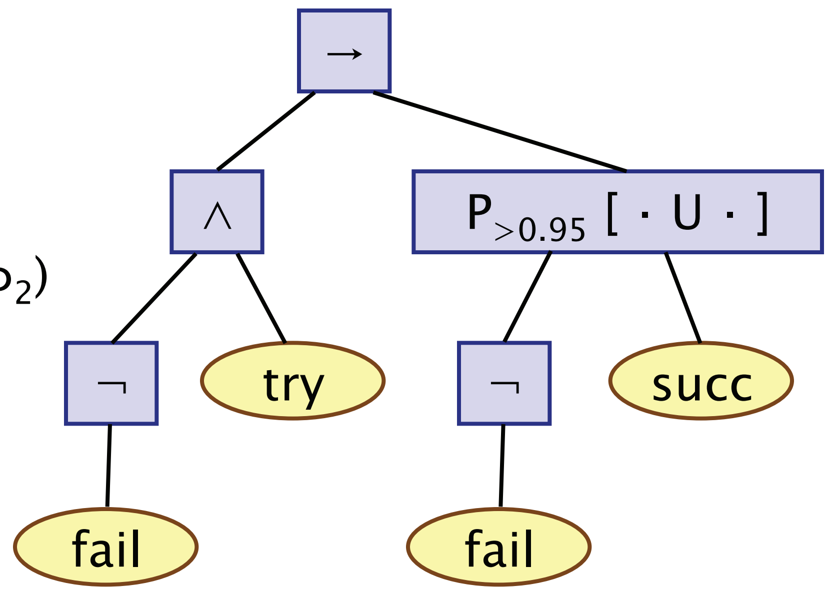
- What does it mean for a DTMC D to satisfy a formula ϕ?
  - sometimes, want to check that $s \vDash \phi\ \forall\ s \in S$, i.e. $Sat(\phi) = S$
  - sometimes, just want to know if $s_{init} \vDash \phi$, i.e. if $s_{init} \in Sat(\phi)$

- Sometimes, focus on quantitative results
  - e.g. compute result of $P_{=?}$ [F error]
  - e.g. compute result of $P_{=?}$ [$F^{\leq k}$ error] for $0 \leq k \leq 100$

PCTL model checking for DTMCs

- Basic algorithm proceeds by induction on parse tree of ϕ
  - example: $\phi = (\neg\text{fail} \wedge \text{try}) \rightarrow P_{>0.95} [\ \neg\text{fail}\ U\ \text{succ}\ ]$

- For the non-probabilistic operators:
  - $\text{Sat}(\text{true}) = S$
  - $\text{Sat}(a) = \{ s \in S \mid a \in L(s) \}$
  - $\text{Sat}(\neg\phi) = S \setminus \text{Sat}(\phi)$
  - $\text{Sat}(\phi_1 \wedge \phi_2) = \text{Sat}(\phi_1) \cap \text{Sat}(\phi_2)$



- For the $P_{\sim p} [\ \psi\ ]$ operator:
  - need to compute the probabilities $\text{Prob}(s, \psi)$ for all states $s \in S$
  - $\text{Sat}(P_{\sim p} [\ \psi\ ]) = \{ s \in S \mid \text{Prob}(s, \psi) \sim p \}$

Probability computation

- Three temporal operators to consider:

- Next: $P_{\sim p}[\ X\ \phi\ ]$

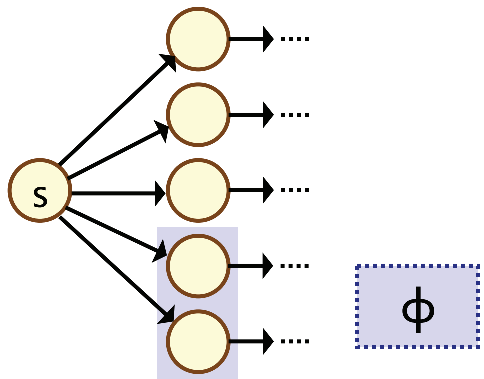
- Bounded until: $P_{\sim p}[\ \phi_1\ U^{\leq k}\ \phi_2\ ]$
  - adaptation of bounded reachability for DTMCs

- Until: $P_{\sim p}[\ \phi_1\ U\ \phi_2\ ]$
  - adaptation of reachability for DTMCs
  - graph-based "precomputation" algorithms
  - techniques for solving large linear equation systems

PCTL next for DTMCs

- Computation of probabilities for PCTL next operator
  - $Sat(P_{\sim p}[\ X\ \phi\ ]) = \{\ s \in S \mid Prob(s,\ X\ \phi) \sim p\ \}$
  - need to compute $Prob(s,\ X\ \phi)$ for all $s \in S$

- Sum outgoing probabilities for transitions to ϕ-states
  - $Prob(s,\ X\ \phi) = \Sigma_{s' \in Sat(\phi)}\ \mathbf{P}(s,s')$

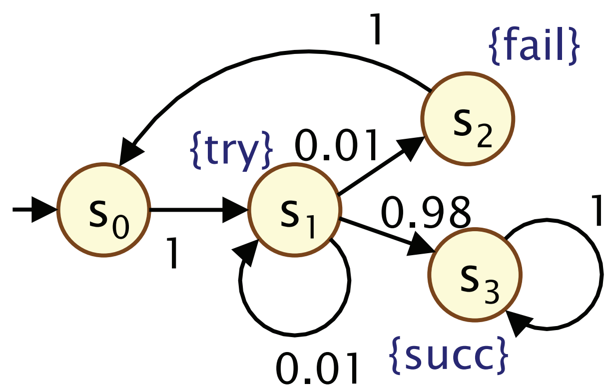


- Compute vector $\underline{Prob}(X\ \phi)$ of probabilities for all states s
  - $\underline{Prob}(X\ \phi) = \mathbf{P} \cdot \underline{\phi}$
  - where $\underline{\phi}$ is a 0-1 vector over S with $\underline{\phi}(s) = 1$ iff $s \vDash \phi$
  - computation requires a single matrix-vector multiplication

PCTL next – Example

- Model check: $P_{\geq 0.9}$ [X (¬try ∨ succ)]
  - Sat (¬try ∨ succ) = (S \ Sat(try)) ∪ Sat(succ)
    = ($\{s_0,s_1,s_2,s_3\} \setminus \{s_1\}) \cup \{s_3\} = \{s_0,s_2,s_3\}$
  - <u>Prob</u>(X (¬try ∨ succ)) = **P** · <u>(¬try ∨ succ)</u> = …

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.99 \\ 1 \\ 1 \end{bmatrix}$$

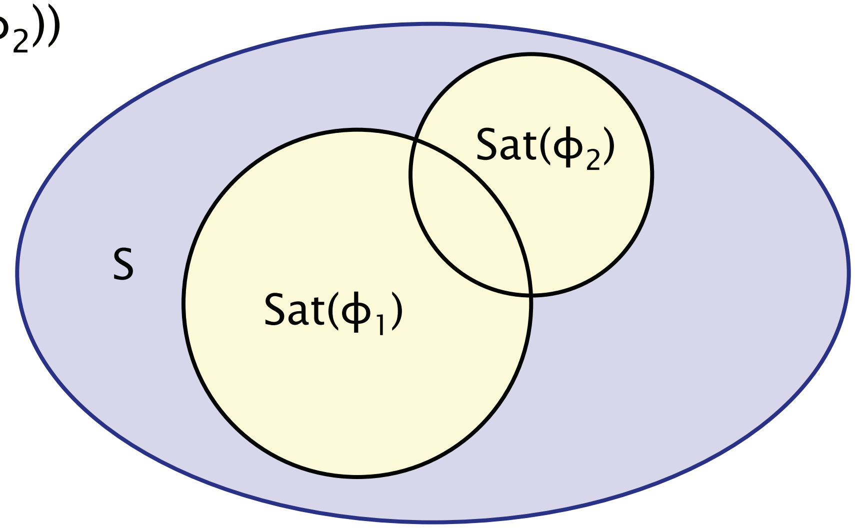


- Results:
  - <u>Prob</u>(X (¬try ∨ succ)) = [0, 0.99, 1, 1]
  - Sat($P_{\geq 0.9}$ [X (¬try ∨ succ)]) = $\{s_1, s_2, s_3\}$

PCTL bounded until for DTMCs

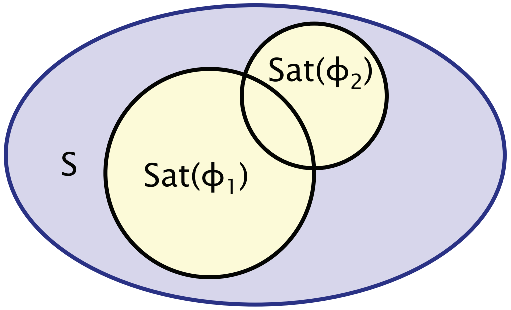
- Computation of probabilities for PCTL $U^{\leq k}$ operator
  - $Sat(P_{\sim p}[\ \phi_1\ U^{\leq k}\ \phi_2\ ]) = \{\ s \in S \mid Prob(s,\ \phi_1\ U^{\leq k}\ \phi_2) \sim p\ \}$
  - need to compute $Prob(s,\ \phi_1\ U^{\leq k}\ \phi_2)$ for all $s \in S$

- First identify (some) states where probability is trivially 1/0
  - $S^{yes} = Sat(\phi_2)$
  - $S^{no} = S \setminus (Sat(\phi_1) \cup Sat(\phi_2))$



PCTL bounded until for DTMCs

- Let:
  - $S^{yes} = Sat(\phi_2)$
  - $S^{no} = S \setminus (Sat(\phi_1) \cup Sat(\phi_2))$
- And let:
  - $S^{?} = S \setminus (S^{yes} \cup S^{no})$



- Compute solution of recursive equations:

$$
Prob(s, \phi_1\, U^{\leq k}\, \phi_2) = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ 0 & \text{if } s \in S^{?} \text{ and } k = 0 \\ \sum_{s' \in S} P(s,s') \cdot Prob(s', \phi_1\, U^{\leq k-1}\, \phi_2) & \text{if } s \in S^{?} \text{ and } k > 0 \end{cases}
$$

PCTL bounded until for DTMCs

- Simultaneous computation of vector $\underline{\text{Prob}}(\phi_1\ U^{\leq k}\ \phi_2)$
  - i.e. probabilities $\text{Prob}(s, \phi_1\ U^{\leq k}\ \phi_2)$ for all $s \in S$

- Iteratively define in terms of matrices and vectors
  - define matrix $\mathbf{P}'$ as follows: $\mathbf{P}'(s,s') = \mathbf{P}(s,s')$ if $s \in S^?$, $\mathbf{P}'(s,s') = 1$ if $s \in S^{yes}$ and $s=s'$, $\mathbf{P}'(s,s') = 0$ otherwise
  - $\underline{\text{Prob}}(\phi_1\ U^{\leq 0}\ \phi_2) = \underline{\phi}_2$
  - $\underline{\text{Prob}}(\phi_1\ U^{\leq k}\ \phi_2) = \mathbf{P}' \cdot \underline{\text{Prob}}(\phi_1\ U^{\leq k-1}\ \phi_2)$
  - requires k matrix-vector multiplications

- Note that we could express this in terms of matrix powers
  - $\underline{\text{Prob}}(\phi_1\ U^{\leq k}\ \phi_2) = (\mathbf{P}')^k \cdot \underline{\phi}_2$ and compute $(\mathbf{P}')^k$ in $\log_2 k$ steps
  - but this is actually inefficient: $(\mathbf{P}')^k$ is much less sparse than $\mathbf{P}'$

PCTL bounded until – Example

- Model check: $P_{>0.98}$ [$F^{\leq 2}$ succ] $\equiv$ $P_{>0.98}$ [true $U^{\leq 2}$ succ]
  - Sat (true) = S = $\{s_0,s_1,s_2,s_3\}$, Sat(succ) = $\{s_3\}$
  - $S^{yes} = \{s_3\}$, $S^{no} = \varnothing$, $S^? = \{s_0,s_1,s_2\}$, **P’ = P**
  - $\underline{\text{Prob}}(\text{true } U^{\leq 0} \text{ succ}) = \underline{\text{succ}} = [0, 0, 0, 1]$

$$\underline{\text{Prob}}(\text{true } U^{\leq 1} \text{ succ}) = P' \cdot \underline{\text{Prob}}(\text{true } U^{\leq 0} \text{ succ}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.98 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{\text{Prob}}(\text{true } U^{\leq 2} \text{ succ}) = P' \cdot \underline{\text{Prob}}(\text{true } U^{\leq 1} \text{ succ}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0.98 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.98 \\ 0.9898 \\ 0 \\ 1 \end{bmatrix}$$

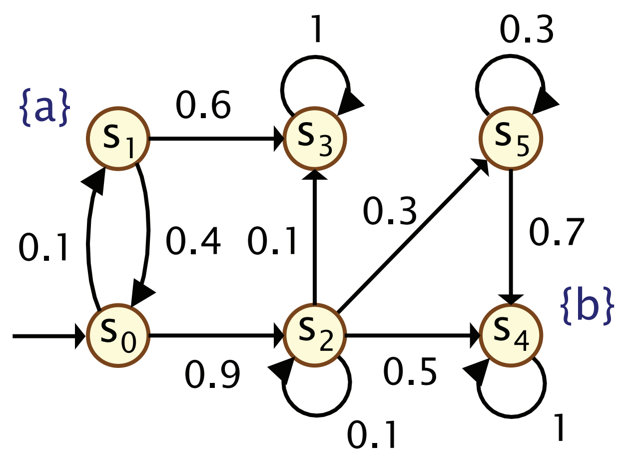
  - Sat($P_{>0.98}$ [$F^{\leq 2}$ succ]) = $\{s_1, s_3\}$

PCTL until for DTMCs

- Computation of probabilities $\mathrm{Prob}(s, \phi_1 \text{ U } \phi_2)$ for all $s \in S$
- First, identify all states where the probability is 1 or 0
  - $S^{yes} = \mathrm{Sat}(P_{\geq 1} [\ \phi_1 \text{ U } \phi_2\ ])$
  - $S^{no} = \mathrm{Sat}(P_{\leq 0} [\ \phi_1 \text{ U } \phi_2\ ])$
- Then solve linear equation system for remaining states

- Running example:

$P_{>0.8}\ [\neg a \text{ U } b\ ]$



PCTL until – linear equations

- Probabilities Prob(s, ϕ_1 U ϕ_2) can now be obtained as the unique solution of the following set of linear equations
  - essentially the same as for probabilistic reachability

$$\text{Prob}(s, \phi_1 \text{ U } \phi_2) = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ \sum_{s' \in S} P(s,s') \cdot \text{Prob}(s', \phi_1 \text{ U } \phi_2) & \text{otherwise} \end{cases}$$

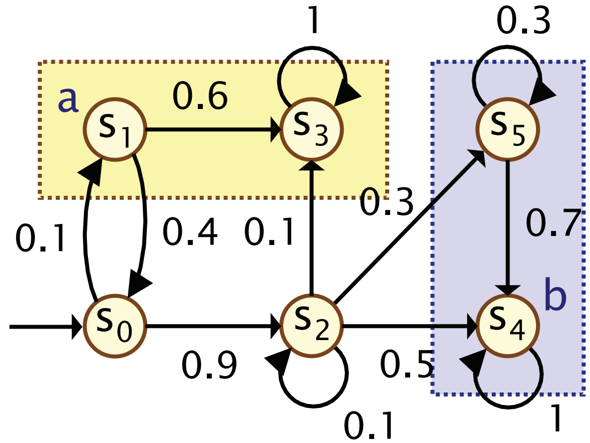
- Can also be reduced to a system in $|S^?|$ unknowns instead of $|S|$ where $S^? = S \setminus (S^{yes} \cup S^{no})$

PCTL until – linear equations

- Example: $P_{>0.8}$ [¬a U b]
- Let x_i = Prob(s_i, ¬a U b)

S^{no} =

Sat($P_{\leq 0}$ [¬a U b])



S^{yes} =

Sat($P_{\geq 1}$ [¬a U b])

$x_1 = x_3 = 0$

$x_4 = x_5 = 1$

$x_2 = 0.1x_2+0.1x_3+0.3x_5+0.5x_4 = 8/9$

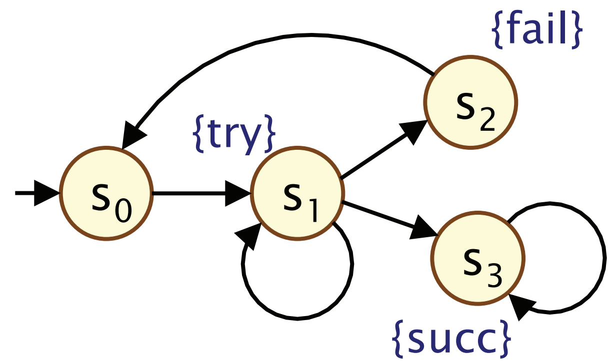
$x_0 = 0.1x_1+0.9x_2 = 0.8$

Prob(¬a U b) = $\underline{x}$ = [0.8, 0, 8/9, 0, 1, 1]

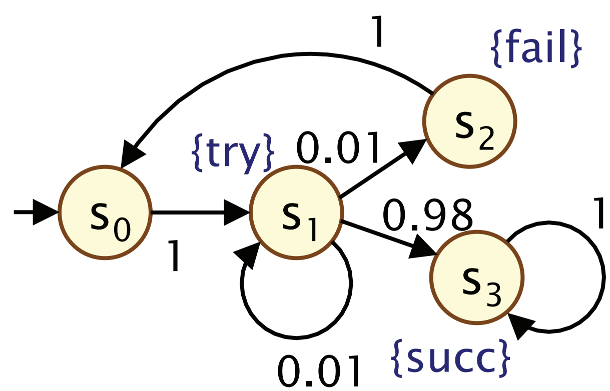
Sat($P_{>0.8}$ [¬a U b]) = { s_2, s_4, s_5 }

Probability vs. nondeterminism

- Labelled transition system
  - (S,s_0,R,L) where $R \subseteq S \times S$
  - choice is nondeterministic



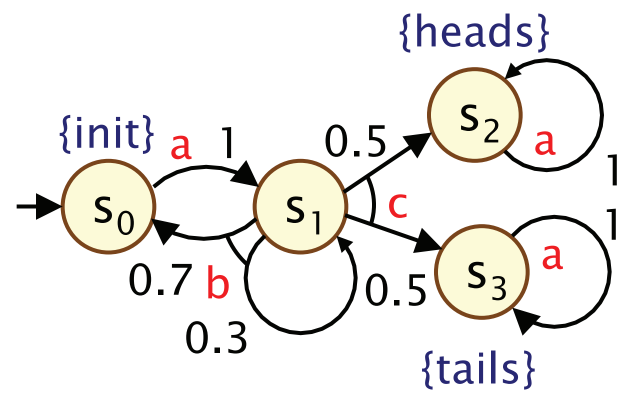
- Discrete-time Markov chain
  - (S,s_0,P,L) where $P : S \times S \rightarrow [0,1]$
  - choice is probabilistic



- How to combine?

Markov decision processes

- Markov decision processes (MDPs)
  - extension of DTMCs which allow nondeterministic choice
- Like DTMCs:
  - discrete set of states representing possible configurations of the system being modelled
  - transitions between states occur in discrete time-steps
- Probabilities and nondeterminism
  - in each state, a nondeterministic choice between several discrete probability distributions over successor states

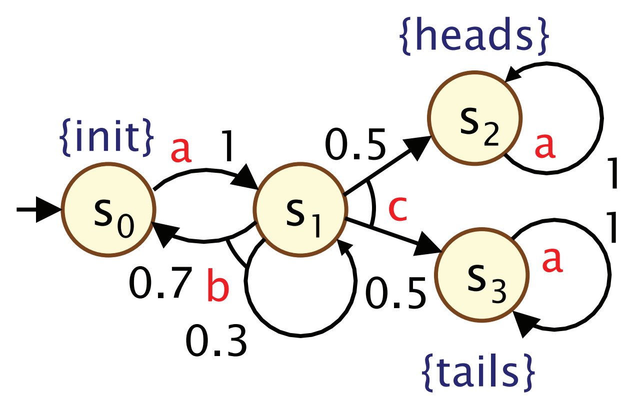


Markov decision processes

- Formally, an MDP M is a tuple (S,s_{init},**Steps**,L) where:
  - S is a finite set of states ("state space")
  - $s_{init} \in S$ is the initial state
  - **Steps** : $S \rightarrow 2^{Act \times Dist(S)}$ is the transition probability function

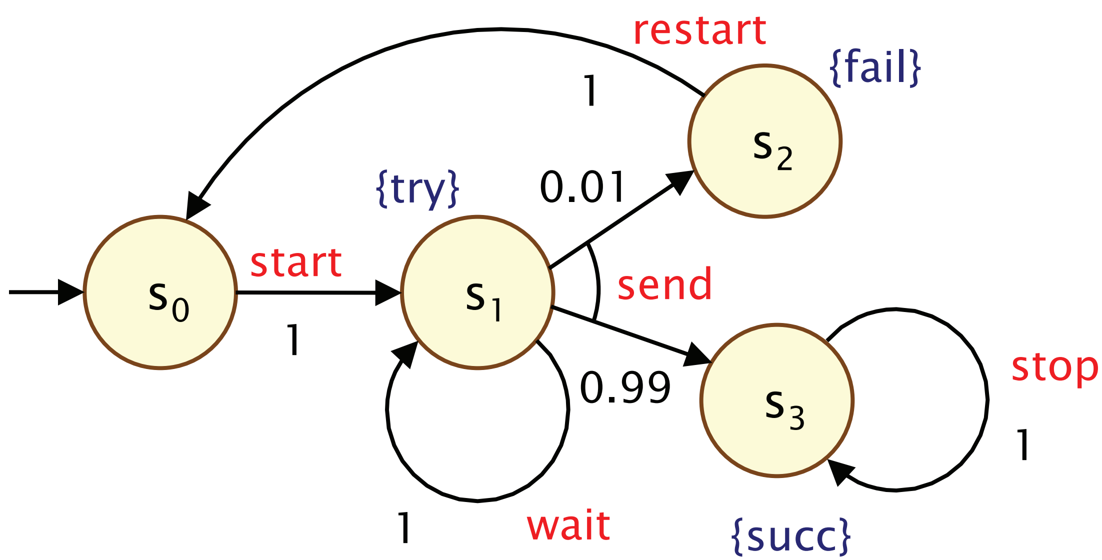
    where Act is a set of actions and Dist(S) is the set of discrete probability distributions over the set S
  - L : $S \rightarrow 2^{AP}$ is a labelling with atomic propositions

- Notes:
  - Steps(s) is always non-empty, i.e. no deadlocks
  - the use of actions to label distributions is optional



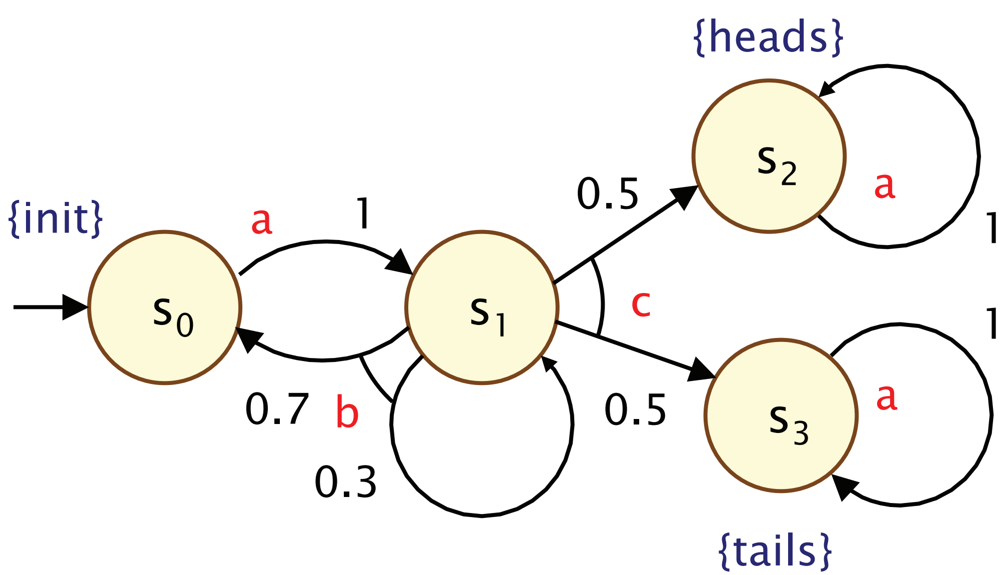
Simple MDP example

- Modification of the simple DTMC communication protocol
  - after one step, process starts trying to send a message
  - then, a nondeterministic choice between: (a) waiting a step because the channel is unready; (b) sending the message
  - if the latter, with probability 0.99 send successfully and stop
  - and with probability 0.01, message sending fails, restart



Simple MDP example 2

- Another simple MDP example with four states
  - from state s_0, move directly to s_1 (action a)
  - in state s_1, nondeterministic choice between actions b and c
  - action b gives a probabilistic choice: self-loop or return to s_0
  - action c gives a 0.5/0.5 random choice between heads/tails



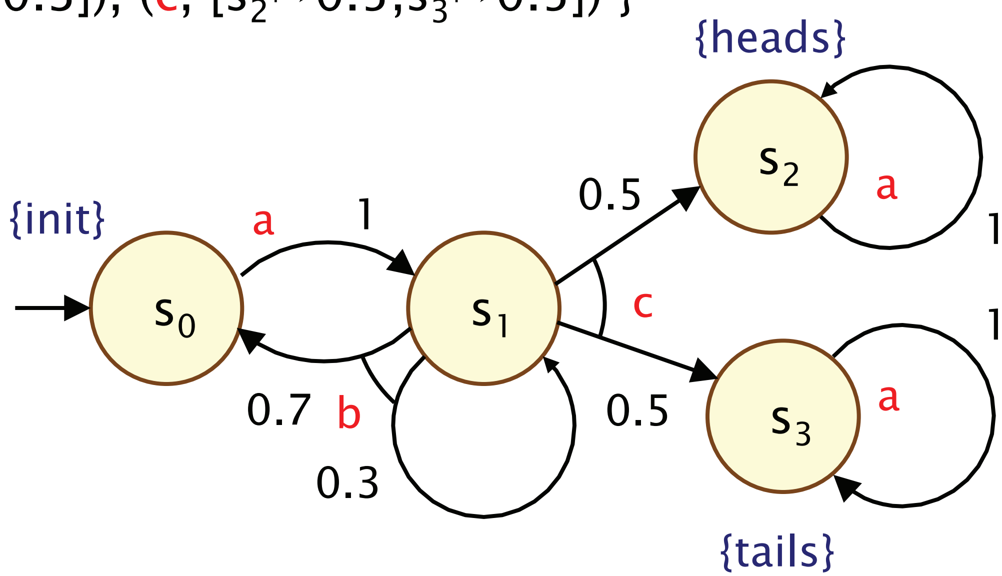
Simple MDP example 2

$M = (S, s_{init}, \mathbf{Steps}, L)$

$S = \{s_0, s_1, s_2, s_3\}$
$s_{init} = s_0$

$AP = \{init, heads, tails\}$
$L(s_0)=\{init\}$,
$L(s_1)=\varnothing$,
$L(s_2)=\{heads\}$,
$L(s_3)=\{tails\}$

$\mathbf{Steps}(s_0) = \{ (a, s_1 \mapsto 1) \}$
$\mathbf{Steps}(s_1) = \{ (b, [s_0 \mapsto 0.7, s_1 \mapsto 0.3]), (c, [s_2 \mapsto 0.5, s_3 \mapsto 0.5]) \}$
$\mathbf{Steps}(s_2) = \{ (a, s_2 \mapsto 1) \}$
$\mathbf{Steps}(s_3) = \{ (a, s_3 \mapsto 1) \}$



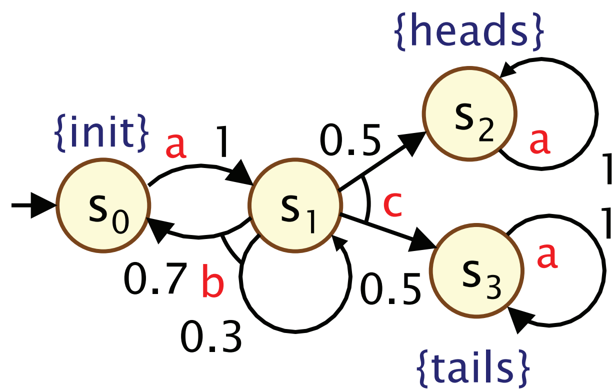
The transition probability function

- It is often useful to think of the function **Steps** as a matrix
  - non-square matrix with $|S|$ columns and $\Sigma_{s \in S}$ $|$**Steps**$(s)|$ rows

- Example (for clarity, we omit actions from the matrix)

Steps$(s_0) = \{ (a, s_1 \mapsto 1) \}$
Steps$(s_1) = \{ (b, [s_0 \mapsto 0.7, s_1 \mapsto 0.3]), (c, [s_2 \mapsto 0.5, s_3 \mapsto 0.5]) \}$
Steps$(s_2) = \{ (a, s_2 \mapsto 1) \}$
Steps$(s_3) = \{ (a, s_3 \mapsto 1) \}$

$$\text{Steps} = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ \hline 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

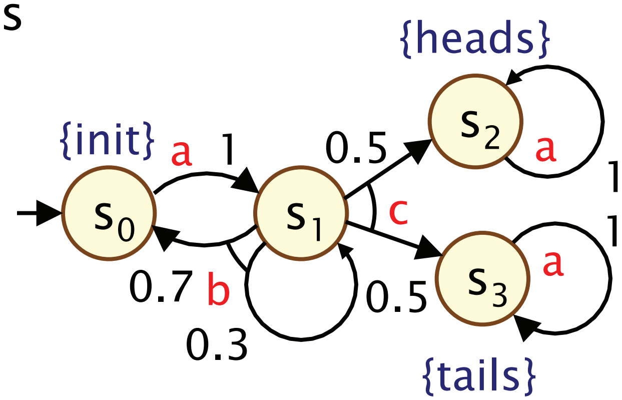


Paths and probabilities

- A (finite or infinite) path through an MDP
  - is a sequence of states and action/distribution pairs
  - e.g. $s_0(a_0,\mu_0)s_1(a_1,\mu_1)s_2\ldots$
  - such that $(a_i,\mu_i) \in$ **Steps**(s_i) and $\mu_i(s_{i+1}) > 0$ for all $i \geq 0$
  - represents an execution (i.e. one possible behaviour) of the system which the MDP is modelling

- Path(s) = set of all paths through MDP starting in state s
  - $\text{Path}_{\text{fin}}(s)$ = set of all finite paths from s

- Paths resolve both nondeterministic and probabilistic choices
  - how to reason about probabilities?



Adversaries

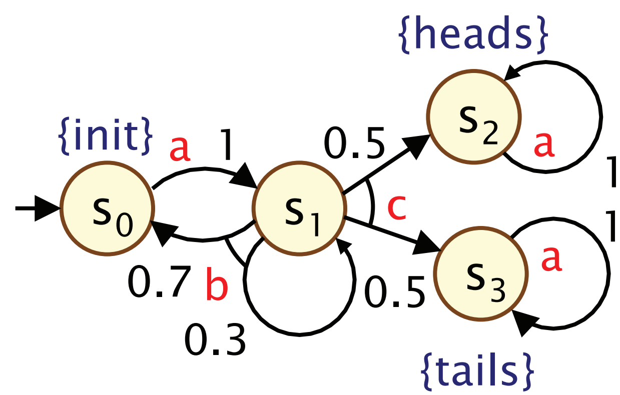
- To consider the probability of some behaviour of the MDP
  - first need to resolve the nondeterministic choices
  - …which results in a DTMC
  - …for which we can define a probability measure over paths

- An adversary resolves nondeterministic choice in an MDP
  - also known as "schedulers", "policies" or "strategies"
- Formally:
  - an adversary σ of an MDP M is a function mapping every finite path $\omega = s_0(a_0,\mu_0)s_1...s_n$ to an element $\sigma(\omega)$ of **Steps**(s_n)
  - i.e. resolves nondeterminism based on execution history

- Adv (or Adv_M) denotes the set of all adversaries

Adversaries – Examples

- Consider the previous example MDP
  - note that s_1 is the only state for which $|\mathbf{Steps}(s)| > 1$
  - i.e. s_1 is the only state for which an adversary makes a choice
  - let μ_b and μ_c denote the probability distributions associated with actions b and c in state s_1

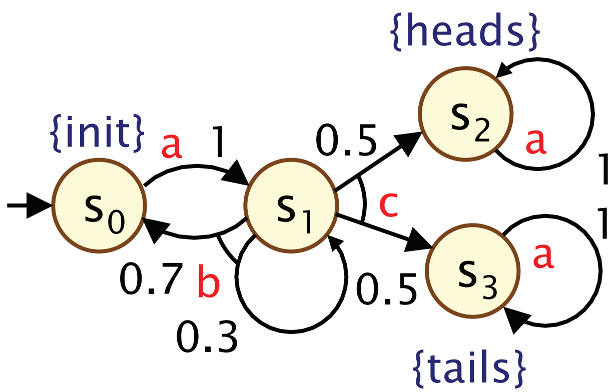


- Adversary σ_1
  - picks action c the first time
  - $\sigma_1(s_0 s_1)=(c,\mu_c)$
- Adversary σ_2
  - picks action b the first time, then c
  - $\sigma_2(s_0 s_1)=(b,\mu_b)$, $\sigma_2(s_0 s_1 s_1)=(c,\mu_c)$, $\sigma_2(s_0 s_1 s_0 s_1)=(c,\mu_c)$

(Note: actions/distributions omitted from paths for clarity)

Adversaries and paths

- $Path^{\sigma}(s) \subseteq Path(s)$
  - (infinite) paths from s where nondeterminism resolved by σ
  - i.e. paths $s_0(a_0,\mu_0)s_1(a_1,\mu_1)s_2\ldots$
  - for which $\sigma(s_0(a_0,\mu_0)s_1\ldots s_n)) = (a_n,\mu_n)$

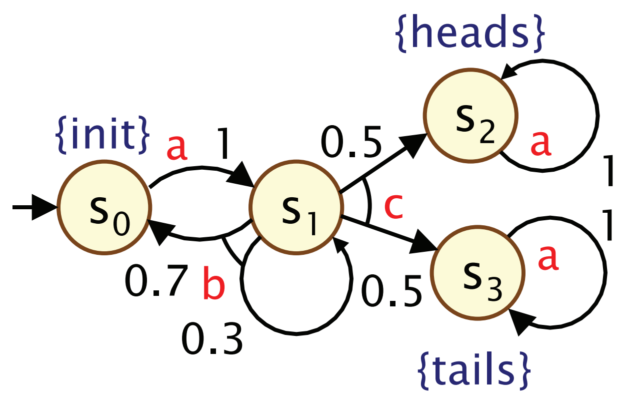


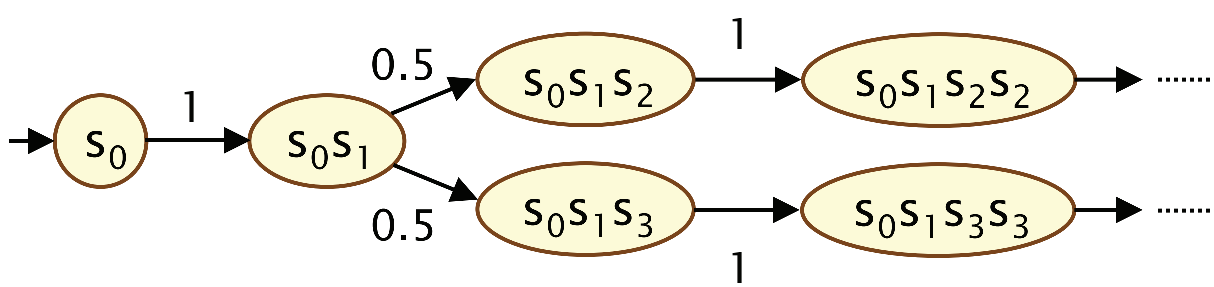
- Adversary σ_1
  - (picks action c the first time)
  - $Path^{\sigma_1}(s_0) = \{\ s_0s_1s_2^{\omega},\ s_0s_1s_3^{\omega}\ \}$

- Adversary σ_2
  - (picks action b the first time, then c)
  - $Path^{\sigma_2}(s_0) = \{\ s_0s_1s_0s_1s_2^{\omega},\ s_0s_1s_0s_1s_3^{\omega},\ s_0s_1s_1s_2^{\omega},\ s_0s_1s_1s_3^{\omega}\ \}$

Adversaries – Examples

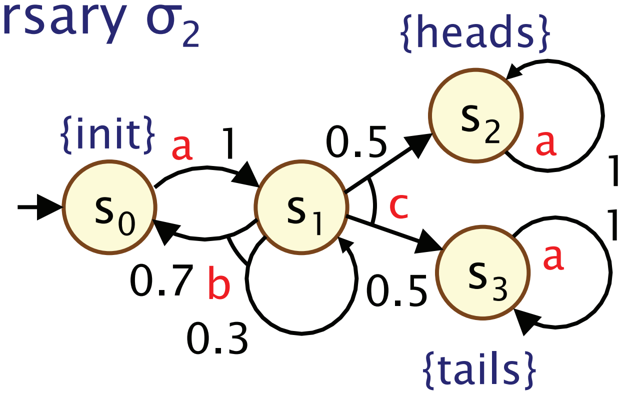
- Fragment of induced DTMC for adversary σ_1
  - σ_1 picks action c the first time

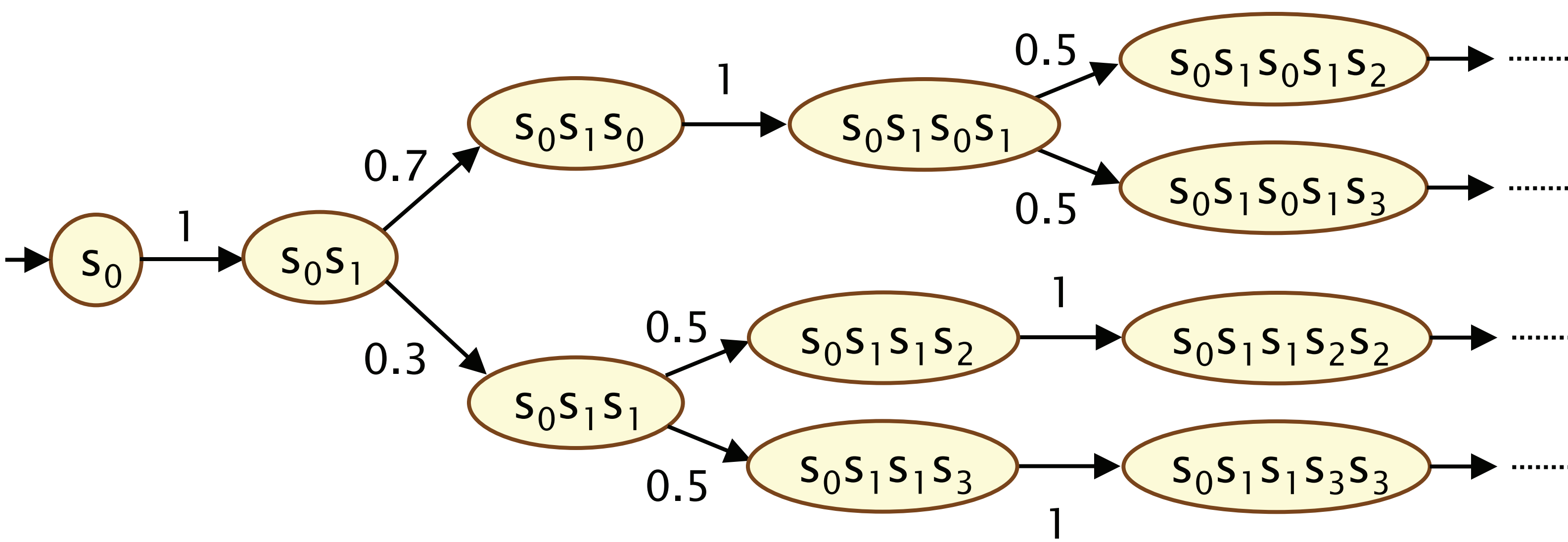




Adversaries – Examples

- Fragment of induced DTMC for adversary σ_2
  - σ_2 picks action b, then c





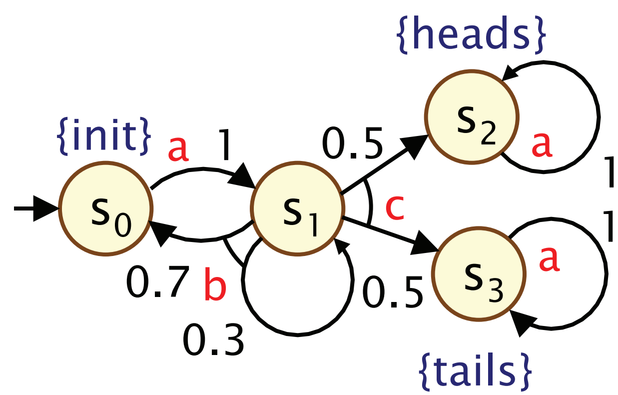
MDPs and probabilities

- $\mathrm{Prob}^{\sigma}(s, \psi) = \mathrm{Pr}^{\sigma}_{s} \{ \omega \in \mathrm{Path}^{\sigma}(s) \mid \omega \vDash \psi \}$
  - for some path formula ψ
  - e.g. $\mathrm{Prob}^{\sigma}(s, \mathrm{F}\ \mathrm{tails})$

- MDP provides best-/worst-case analysis
  - based on lower/upper bounds on probabilities
  - over all possible adversaries

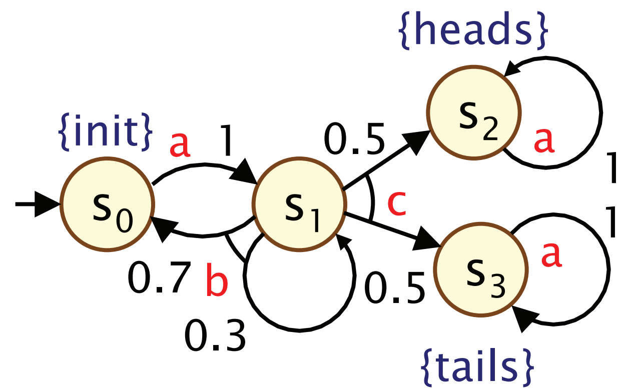
$$p_{min}(s,\psi) = \inf_{\sigma \in Adv} \mathrm{Prob}^{\sigma}(s,\psi)$$

$$p_{max}(s,\psi) = \sup_{\sigma \in Adv} \mathrm{Prob}^{\sigma}(s,\psi)$$

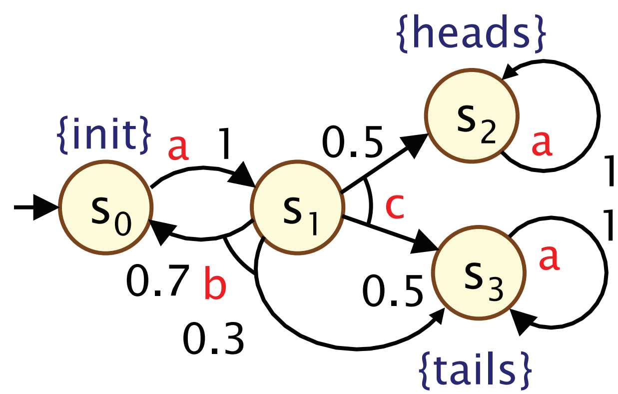


Examples

- $Prob^{\sigma 1}(s_0, F\ tails) = 0.5$
- $Prob^{\sigma 2}(s_0, F\ tails) = 0.5$
  - (where σ_i picks b i−1 times then c)
- …
- $p_{max}(s_0, F\ tails) = 0.5$



- $Prob^{\sigma 1}(s_0, F\ tails) = 0.5$
- $Prob^{\sigma 2}(s_0, F\ tails)$
  $= 0.3+0.7\cdot0.5 = 0.65$
- $Prob^{\sigma 3}(s_0, F\ tails)$
  $= 0.3+0.7\cdot0.3+0.7\cdot0.7\cdot0.5 = 0.755$
- …
- $p_{max}(s_0, F\ tails) = 1$



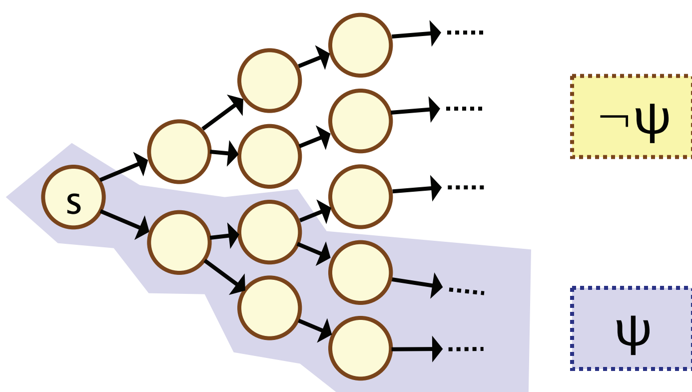
PCTL semantics for MDPs

- PCTL formulas interpreted over states of an MDP
  - $s \models \phi$ denotes ϕ is "true in state s" or "satisfied in state s"

- Semantics of (non-probabilistic) state formulas and of path formulas are identical to those for DTMCs:
- For a state s of the MDP (S, s_{init}, **Steps**, L):
  - $s \models a \quad \Leftrightarrow \quad a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \quad \Leftrightarrow \quad s \models \phi_1$ and $s \models \phi_2$
  - $s \models \neg\phi \quad \Leftrightarrow \quad s \models \phi$ is false
- For a path $\omega = s_0(a_1,\mu_1)s_1(a_2,\mu_2)s_2\ldots$ in the MDP:
  - $\omega \models X\,\phi \quad \Leftrightarrow \quad s_1 \models \phi$
  - $\omega \models \phi_1\ U^{\leq k}\ \phi_2 \quad \Leftrightarrow \quad \exists i \leq k$ such that $s_i \models \phi_2$ and $\forall j<i,\ s_j \models \phi_1$
  - $\omega \models \phi_1\ U\ \phi_2 \quad \Leftrightarrow \quad \exists k \geq 0$ such that $\omega \models \phi_1\ U^{\leq k}\ \phi_2$

PCTL semantics for MDPs

- Semantics of the probabilistic operator P
  - can only define probabilities for a specific adversary σ
  - $s \models P_{\sim p} [\psi]$ means "the probability, from state s, that ψ is true for an outgoing path satisfies ~p for all adversaries σ"
  - formally $s \models P_{\sim p} [\psi] \Leftrightarrow Prob^{\sigma}(s, \psi) \sim p$ for all adversaries σ
  - where $Prob^{\sigma}(s, \psi) = Pr^{\sigma}_{s} \{ \omega \in Path^{\sigma}(s) \mid \omega \models \psi \}$



$Prob^{\sigma}(s, \psi) \sim p$

Minimum and maximum probabilities

- Letting:
  - $p_{max}(s, \psi) = \sup_{\sigma \in Adv} Prob^{\sigma}(s, \psi)$
  - $p_{min}(s, \psi) = \inf_{\sigma \in Adv} Prob^{\sigma}(s, \psi)$

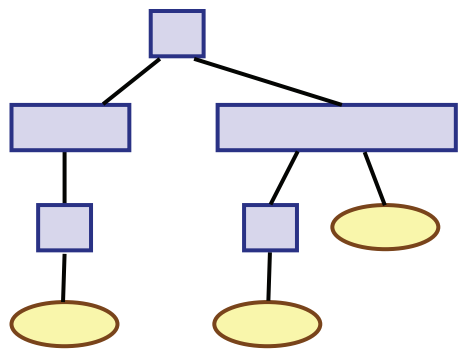
- We have:
  - if $\sim \in \{\geq,>\}$, then $s \vDash P_{\sim p} [\, \psi \,] \;\Leftrightarrow\; p_{min}(s, \psi) \sim p$
  - if $\sim \in \{<,\leq\}$, then $s \vDash P_{\sim p} [\, \psi \,] \;\Leftrightarrow\; p_{max}(s, \psi) \sim p$

- Model checking $P_{\sim p}[\, \psi \,]$ reduces to the computation over all adversaries of either:
  - the minimum probability of ψ holding
  - the maximum probability of ψ holding

PCTL model checking for MDPs

- Algorithm for PCTL model checking [BdA95]
  - inputs: MDP $M=(S,s_{init},\mathbf{Steps},L)$, PCTL formula ϕ
  - output: $Sat(\phi) = \{ s \in S \mid s \models \phi \}$ = set of states satisfying ϕ
- Often, also consider quantitative results
  - e.g. compute result of $P_{min=?}$ [$F^{\leq t}$ stable] for $0 \leq t \leq 100$

- Basic algorithm same as PCTL for DTMCs
  - proceeds by induction on parse tree of ϕ
- For the non-probabilistic operators:
  - $Sat(true) = S$
  - $Sat(a) = \{ s \in S \mid a \in L(s) \}$
  - $Sat(\neg\phi) = S \setminus Sat(\phi)$
  - $Sat(\phi_1 \wedge \phi_2) = Sat(\phi_1) \cap Sat(\phi_2)$



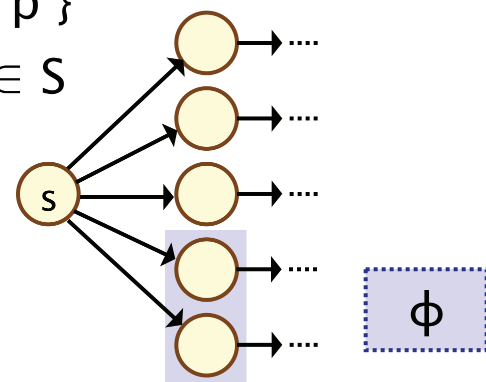
PCTL model checking for MDPs

- Main task: model checking $P_{\sim p}$ [ψ] formulae
  - reduces to computation of min/max probabilities
  - i.e. $p_{min}(s, \psi)$ or $p_{max}(s, \psi)$ for all $s \in S$
  - dependent on whether $\sim \in \{\geq,>\}$ or $\sim \in \{<,\leq\}$

- Three cases:
  - next ($X\ \phi$)
  - bounded until ($\phi_1\ U^{\leq k}\ \phi_2$)
  - unbounded until ($\phi_1\ U\ \phi_2$)

PCTL next for MDPs

- Computation of probabilities for PCTL next operator
- Consider case of minimum probabilities…
  - $Sat(P_{\sim p}[\ X\ \phi\ ]) = \{\ s \in S \mid p_{min}(s,\ X\ \phi) \sim p\ \}$
  - need to compute $p_{min}(s,\ X\ \phi)$ for all $s \in S$
- Recall in the DTMC case
  - sum outgoing probabilities for transitions to ϕ-states
  - $Prob(s,\ X\ \phi) = \Sigma_{s' \in Sat(\phi)}\ \mathbf{P}(s,s')$



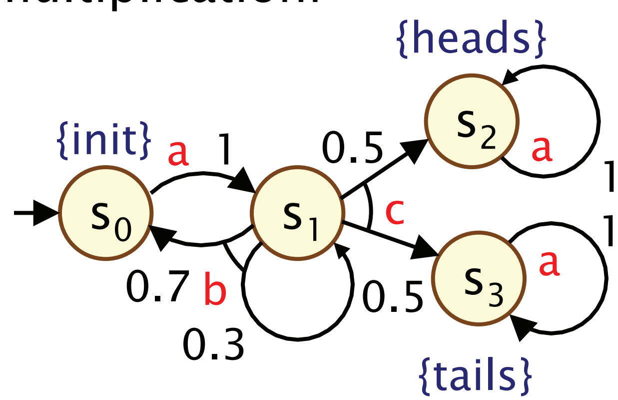
- For MDPs, perform computation for each distribution available in s and then take minimum:
  - $p_{min}(s,\ X\ \phi) = \min\{\ \Sigma_{s' \in Sat(\phi)}\ \mu(s') \mid (a,\mu) \in Steps(s)\ \}$

- Maximum probabilities case is analogous

PCTL next – Example

- Model check: $P_{\geq 0.5}$ [X heads]
  - lower probability bound so minimum probabilities required
  - Sat (heads)= $\{s_2\}$
  - e.g. $p_{min}(s_1, \text{X heads}) = \min(0, 0.5) = 0$
  - can do all at once with matrix-vector multiplication:

$$\text{Steps} \cdot \underline{\text{heads}} = \left[\begin{array}{cccc} 0 & 1 & 0 & 0 \\ \hline 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \left[\begin{array}{c} 0 \\ \hline 0 \\ 0.5 \\ \hline 1 \\ \hline 0 \end{array}\right]$$



- Extracting the minimum for each state yields
  - $\underline{p}_{min}(\text{X heads}) = [0, 0, 1, 0]$
  - $Sat(P_{\geq 0.5} [\text{ X heads }]) = \{s_2\}$

PCTL bounded until for MDPs

- Computation of probabilities for PCTL $U^{\leq k}$ operator
- Consider case of minimum probabilities…
  - $Sat(P_{\sim p}[\ \phi_1\ U^{\leq k}\ \phi_2\ ]) = \{\ s \in S \mid p_{min}(s, \phi_1\ U^{\leq k}\ \phi_2) \sim p\ \}$
  - need to compute $p_{min}(s, \phi_1\ U^{\leq k}\ \phi_2)$ for all $s \in S$
- First identify (some) states where probability is 1 or 0
  - $S^{yes} = Sat(\phi_2)$ and $S^{no} = S \setminus (Sat(\phi_1) \cup Sat(\phi_2))$
- Then solve the recursive equations:

$$p_{min}(s, \phi_1\ U^{\leq k}\ \phi_2) = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ 0 & \text{if } s \in S^{?} \text{ and } k = 0 \\ \min\left\{ \sum_{s' \in S} \mu(s') \cdot p_{min}(s', \phi_1\ U^{\leq k-1}\ \phi_2) \mid (a,\mu) \in Steps(s) \right\} & \text{if } s \in S^{?} \text{ and } k > 0 \end{cases}$$

- Maximum probabilities case is analogous

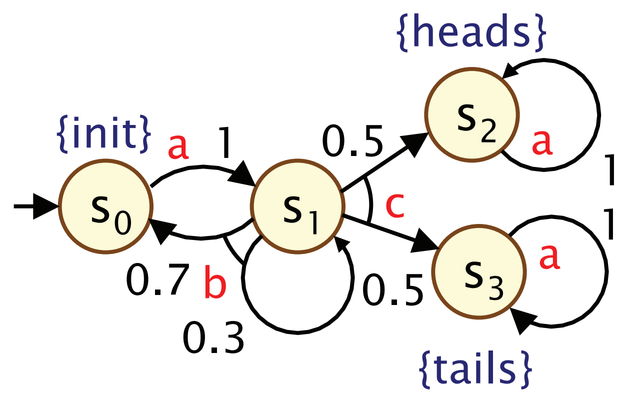
PCTL bounded until for MDPs

- Simultaneous computation of vector $\underline{p}_{min}(\phi_1\ U^{\leq k}\ \phi_2)$
  - i.e. probabilities $p_{min}(s, \phi_1\ U^{\leq k}\ \phi_2)$ for all $s \in S$

- Recursive definition in terms of matrices and vectors
  - similar to DTMC case
  - requires k matrix-vector multiplications
  - in addition requires k minimum operations

PCTL bounded until – Example

- Model check: $P_{<0.95}$ [$F^{\leq 3}$ init] $\equiv$ $P_{<0.95}$ [true $U^{\leq 3}$ init]
  - upper probability bound so maximum probabilities required
  - Sat (true) = S and Sat (init) = $\{s_0\}$
  - $S^{yes} = \{s_0\}$ and $S^{no} = \varnothing$
  - $S^{?} = \{s_1, s_2, s_3\}$
- The vector of probabilities is computed successively as:
  - $\underline{p}_{max}(\text{true } U^{\leq 0} \text{ init }) = [\ 1, 0, 0, 0\ ]$
  - $\underline{p}_{max}(\text{true } U^{\leq 1} \text{ init }) = [\ 1, 0.7, 0, 0\ ]$
  - $\underline{p}_{max}(\text{true } U^{\leq 2} \text{ init }) = [\ 1, 0.91, 0, 0\ ]$
  - $\underline{p}_{max}(\text{true } U^{\leq 3} \text{ init }) = [\ 1, 0.973, 0, 0\ ]$
- Hence, the result is:
  - Sat($P_{<0.95}$ [$F^{\leq 3}$ init]) = $\{ s_2, s_3 \}$



PCTL until for MDPs

- Computation of probabilities for all $s \in S$:
  - $p_{min}(s, \phi_1 \text{ U } \phi_2)$ or $p_{max}(s, \phi_1 \text{ U } \phi_2)$
- Essentially the same as computation of reachability probabilities (see previous lecture)
  - just need to consider additional ϕ_1 constraint
- Overview:
  - precomputation:
    - identify states where the probability is 0 (or 1)
  - several options to compute remaining values:
    - value iteration
    - reduction to linear programming

Method 1 – Value iteration (min)

- Minimum probabilities satisfy:
  - $p_{min}(s, \phi_1 \text{ U } \phi_2) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \min\left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'}^{(n-1)} \mid (a,\mu) \in Steps(s) \right\} & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

- Approximate iterative solution:
  - compute vector $\underline{x}^{(n)}$ for "sufficiently large" n
  - in practice: terminate iterations when some pre-determined convergence criteria satisfied
  - e.g. $\max_s | \underline{x}^{(n)}(s) - \underline{x}^{(n-1)}(s)) | < \varepsilon$ for some tolerance ε

Method 1 – Value iteration (max)

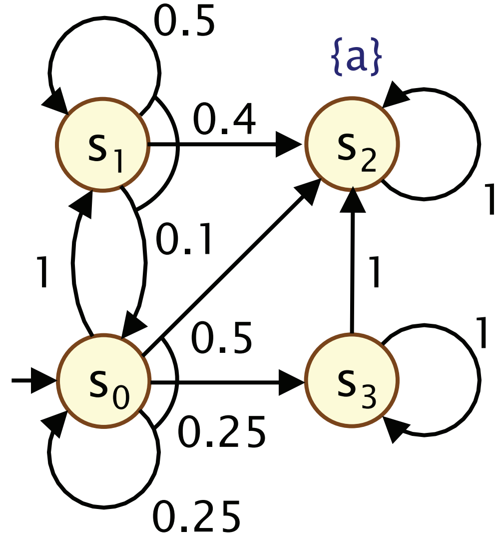
- Similarly, maximum probabilities satisfy:
  - $p_{max}(s, \phi_1 \text{ U } \phi_2) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in S^{yes} \\ 0 & \text{if } s \in S^{no} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \max \left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'}^{(n-1)} \mid (a,\mu) \in Steps(s) \right\} & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

- …and can be approximated iteratively

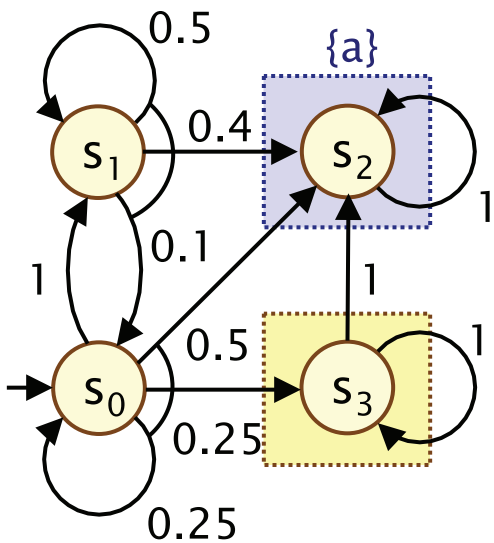
PCTL until – Example

- Model check: $P_{>0.5}$ [F a] ≡ $P_{>0.5}$ [true U a]
  - lower probability bound so minimum probabilities required



PCTL until – Example

- Model check: $P_{>0.5}$ [F a] $\equiv$ $P_{>0.5}$ [true U a]
  - lower probability bound so minimum probabilities required



S^{yes} = Sat(a)

Prob0E

S^{no} = { $s \in S$ | $p_{min}(s, F\ a) = 0$ }

Method 2 – Linear optimisation problem

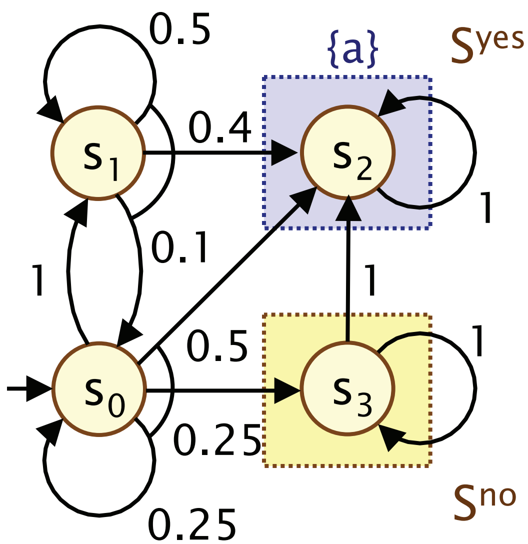
- Probabilities for states in $S^? = S \setminus (S^{yes} \cup S^{no})$ can also be obtained from a linear optimisation problem
- Minimum probabilities:

$$\boxed{\begin{array}{c} \text{maximize } \sum_{s \in S^?} x_s \text{ subject to the constraints:} \\ x_s \leq \sum_{s' \in S^?} \mu(s') \cdot x_{s'} + \sum_{s' \in S^{yes}} \mu(s') \\ \text{for all } s \in S^? \text{ and for all } (a, \mu) \in \textbf{Steps}(s) \end{array}}$$

- Maximum probabilities:

$$\boxed{\begin{array}{c} \text{minimize } \sum_{s \in S^?} x_s \text{ subject to the constraints:} \\ x_s \geq \sum_{s' \in S^?} \mu(s') \cdot x_{s'} + \sum_{s' \in S^{yes}} \mu(s') \\ \text{for all } s \in S^? \text{ and for all } (a, \mu) \in \textbf{Steps}(s) \end{array}}$$

PCTL until – Example



Let $x_i = p_{min}(s_i, F\ a)$

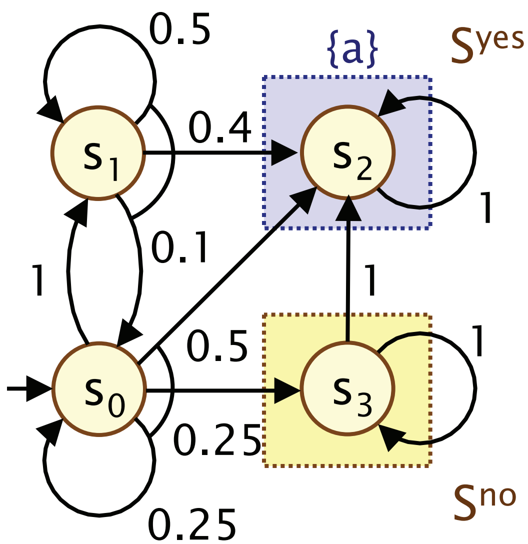
S^{yes}: $x_2=1$, S^{no}: $x_3=0$

For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $x_0 \leq 0.25 \cdot x_0 + 0.5$
- $x_1 \leq 0.1 \cdot x_0 + 0.5 \cdot x_1 + 0.4$

PCTL until – Example



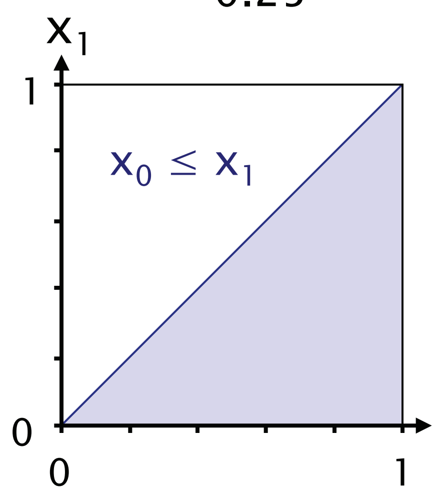
Let $x_i = p_{min}(s_i, F\ a)$

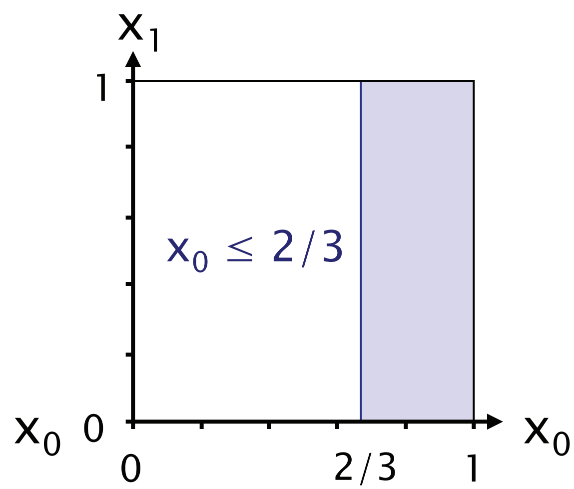
S^{yes}: $x_2=1$, S^{no}: $x_3=0$

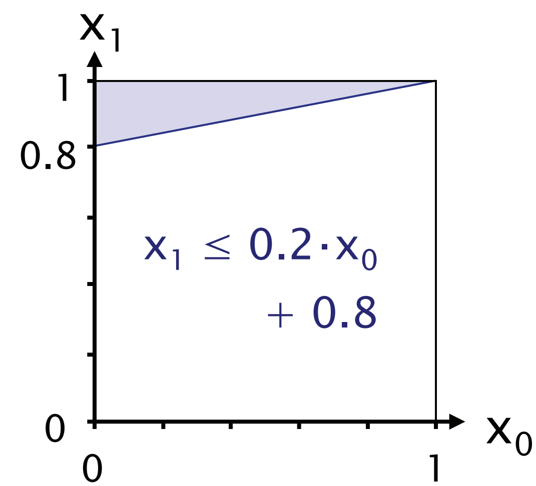
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

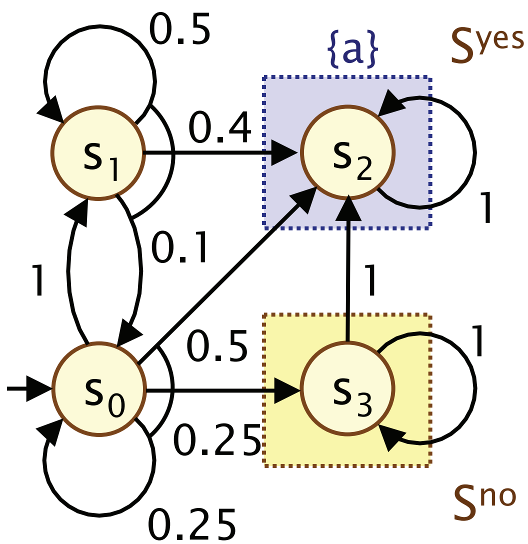
- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 0.2 \cdot x_0 + 0.8$







PCTL until – Example



Let $x_i = p_{min}(s_i, F\ a)$

S^{yes}: $x_2=1$, S^{no}: $x_3=0$

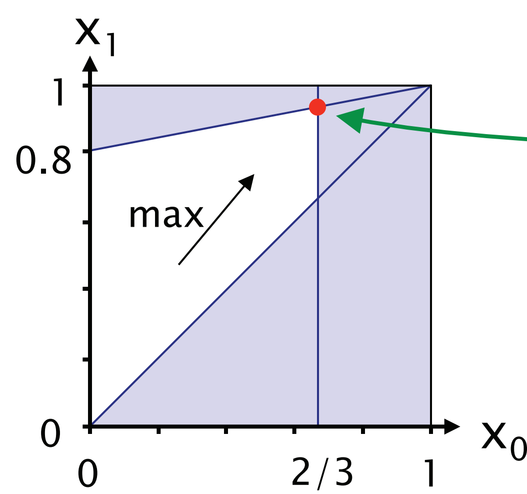
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 0.2 \cdot x_0 + 0.8$

$\underline{p}_{min}(F\ a) =$
$[\ 2/3,\ 14/15,\ 1,\ 0\ ]$

$Sat(P_{>0.5}\ [F\ a]) =$
$\{\ s_0,\ s_1,\ s_2\ \}$



Solution:

(x_0, x_1)

$=$

$(2/3, 14/15)$