Lecture 12
Markov Decision Processes

Dr. Dave Parker

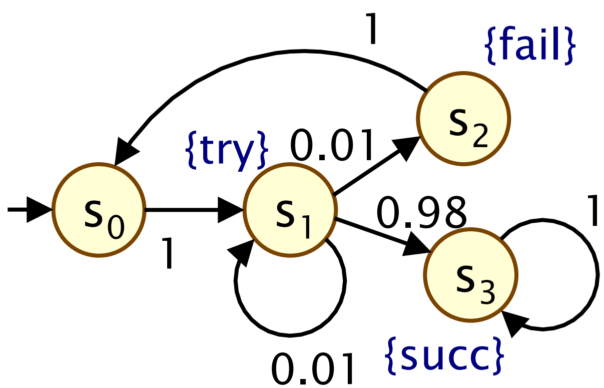
Department of Computer Science
University of Oxford

Overview

- Nondeterminism
- Markov decision processes (MDPs)
- Paths, probabilities and adversaries
- End components

Recap: DTMCs

- Discrete-time Markov chains (DTMCs)
  - discrete state space, transitions are discrete time-steps
  - from each state, choice of successor state (i.e. which transition) is determined by a discrete probability distribution



- DTMCs are fully probabilistic
  - well suited to modelling, for example, simple random algorithms or synchronous probabilistic systems where components move in lock-step

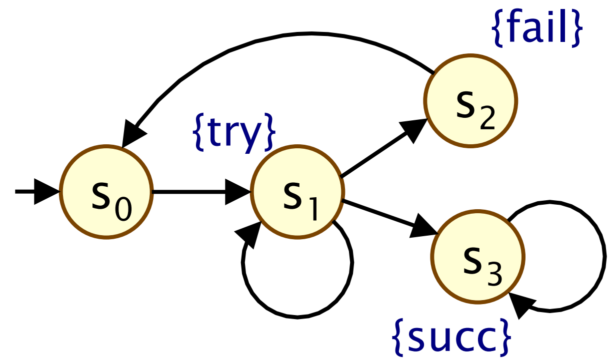
Nondeterminism

- But, some aspects of a system may not be probabilistic and should not be modelled probabilistically; for example:

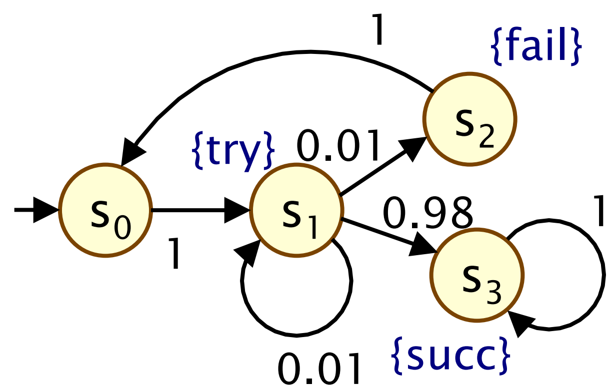
- Concurrency – scheduling of parallel components
  - e.g. randomised distributed algorithms – multiple probabilistic processes operating asynchronously
- Unknown environments
  - e.g. probabilistic security protocols – unknown adversary
- Underspecification – unknown model parameters
  - e.g. a probabilistic communication protocol designed for message propagation delays of between d_{min} and d_{max}
- Abstraction
  - e.g. partition DTMC into similar (but not identical) states

Probability vs. nondeterminism

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



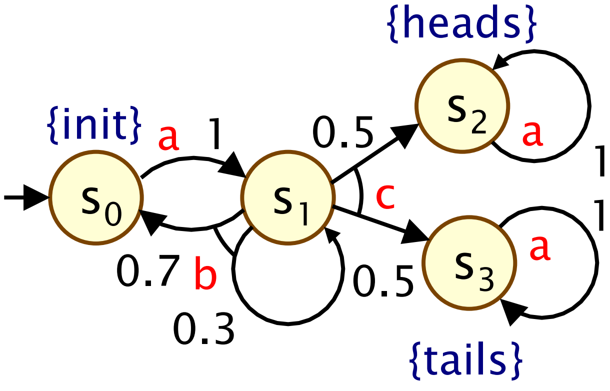
- Discrete-time Markov chain
  - (S,s_0,P,L) where $P : S \times S \rightarrow [0,1]$
  - choice is probabilistic



- 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

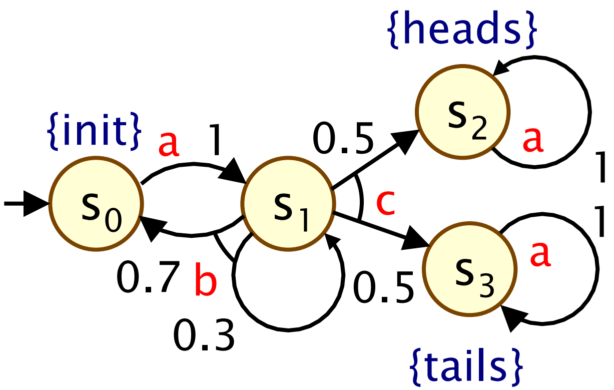


Markov decision processes

- 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

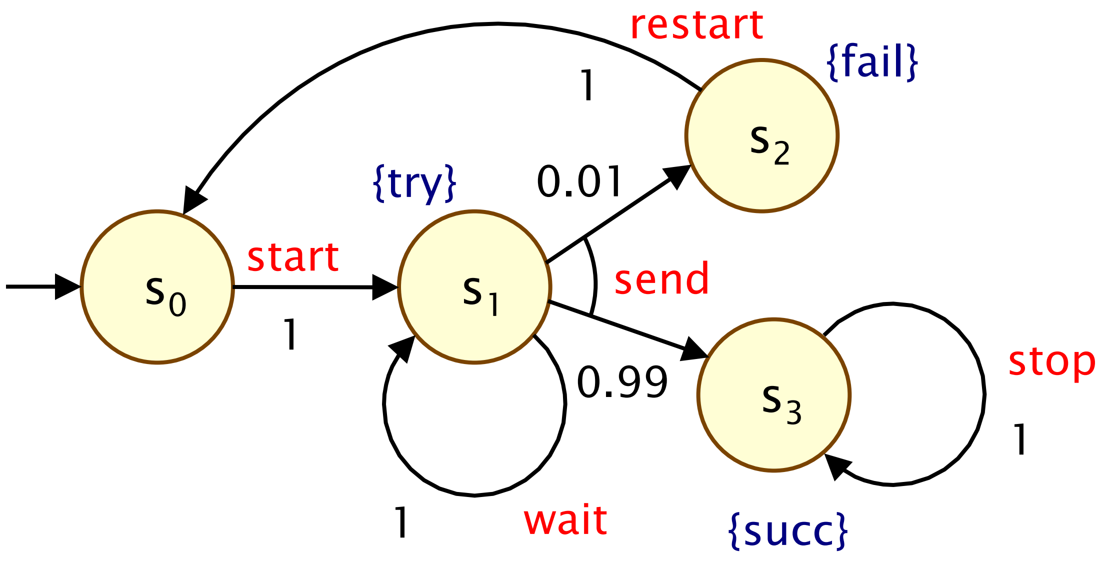
    where Act is a set of actions and Dist(S) is the set of discrete probability distributions over the set S
  - $L : S \rightarrow 2^{AP}$ is a labelling with atomic propositions

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



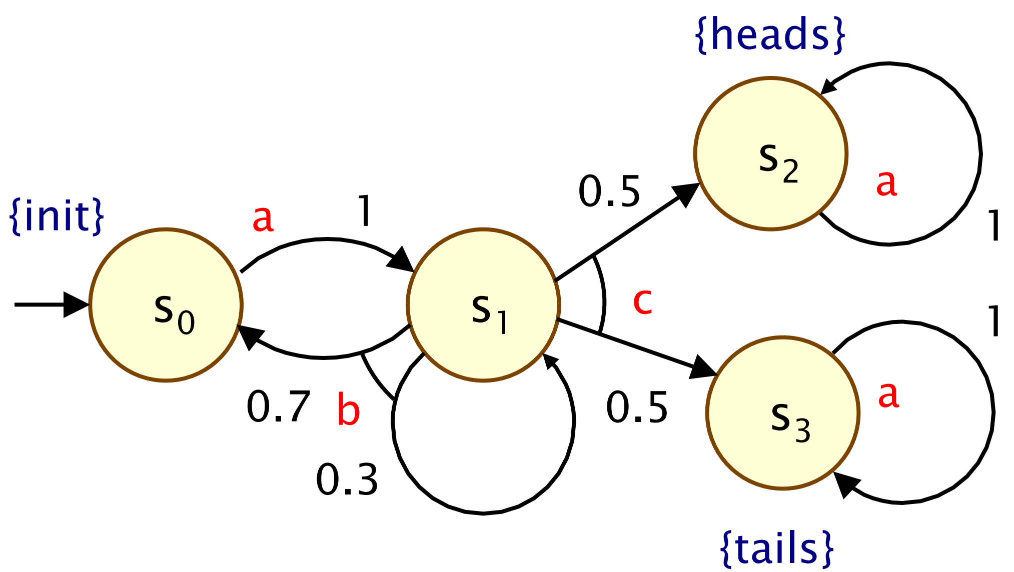
Simple MDP example

- 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



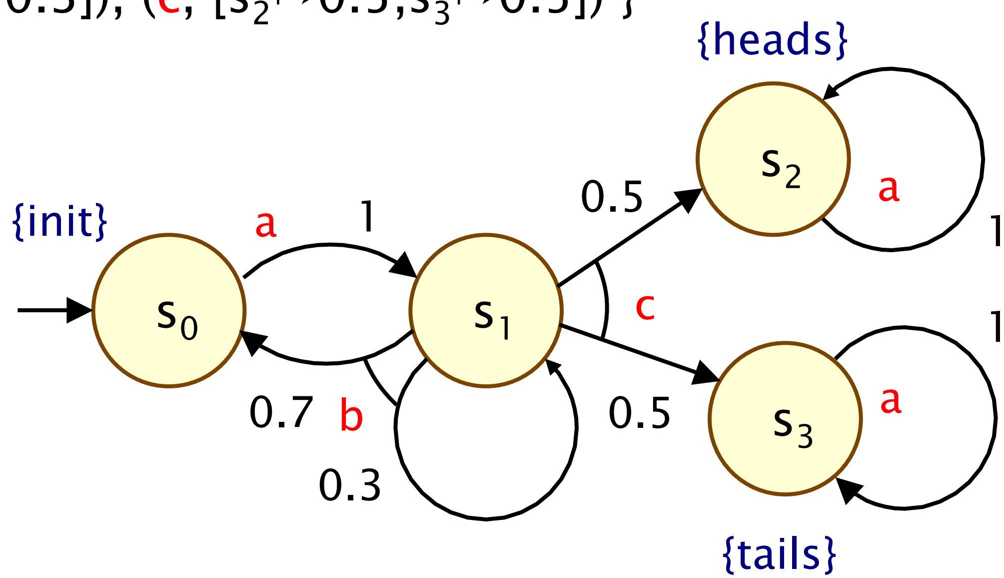
Simple MDP example 2

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

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

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

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



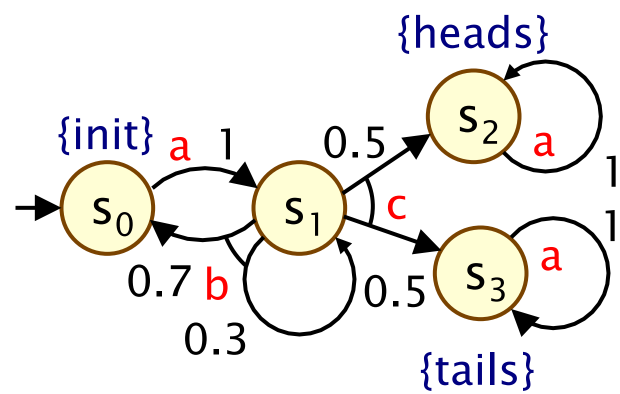
The transition probability function

- 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, $s_3 \mapsto 1$) }

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



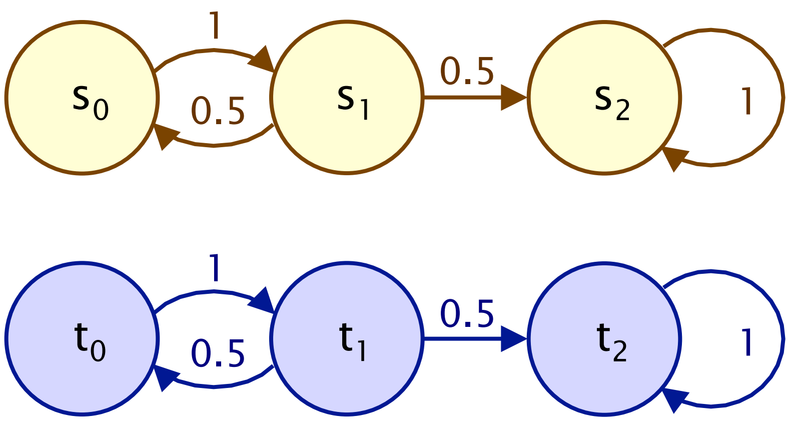
Example – Parallel composition

Asynchronous parallel composition of two 3-state DTMCs

PRISM code:

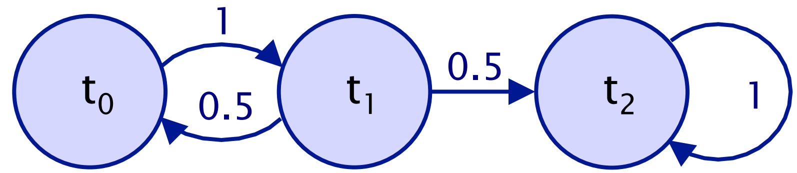
```
module M1
    s : [0..2] init 0;
    [] s=0 -> (s'=1);
    [] s=1 -> 0.5:(s'=0) + 0.5:(s'=2);
    [] s=2 -> (s'=2);
endmodule
```

```
module M2 = M1 [ s=t ] endmodule
```

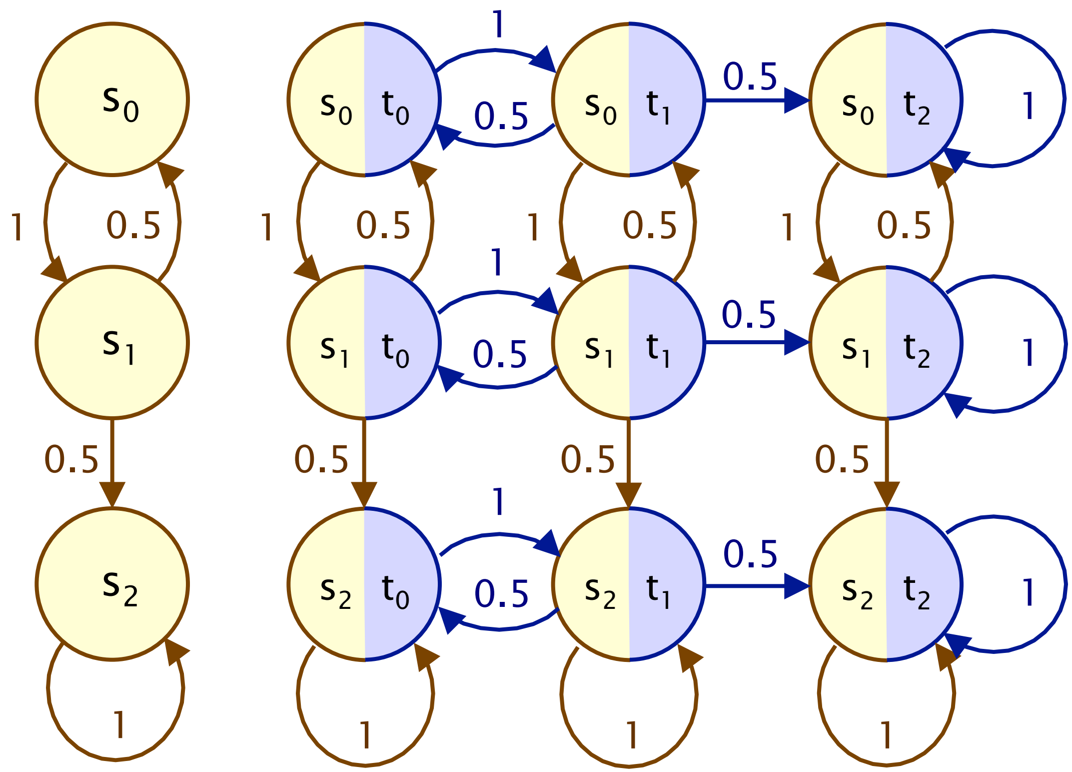


Example – Parallel composition

Asynchronous parallel composition of two 3-state DTMCs



Action labels omitted here

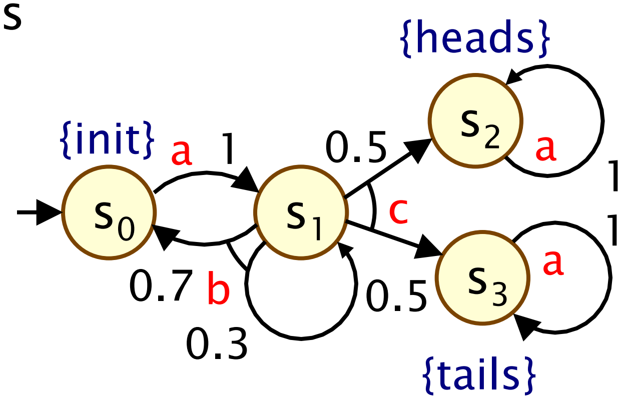


Paths and probabilities

- 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
  - $Path_{fin}(s)$ = set of all finite paths from s

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



Adversaries

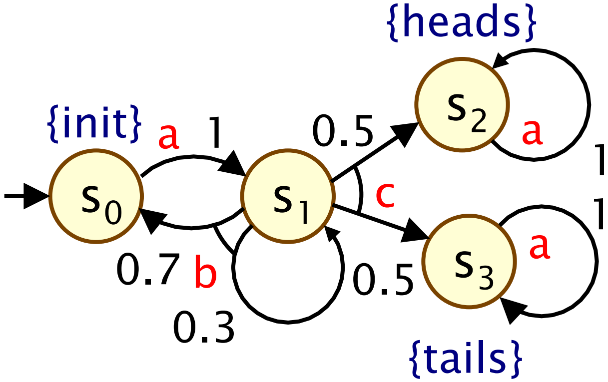
- To consider the probability of some behaviour of the MDP
  - first need to resolve the nondeterministic choices
  - …which results in a DTMC
  - …for which we can define a probability measure over paths

- 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 $\omega = s_0(a_0,\mu_0)s_1 \ldots s_n$ 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

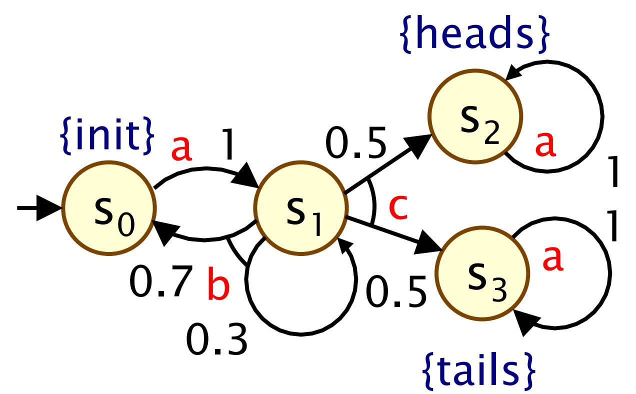


- Adversary σ_1
  - picks action c the first time
  - $\sigma_1(s_0s_1)=(c,\mu_c)$
- Adversary σ_2
  - picks action b the first time, then c
  - $\sigma_2(s_0s_1)=(b,\mu_b)$, $\sigma_2(s_0s_1s_1)=(c,\mu_c)$, $\sigma_2(s_0s_1s_0s_1)=(c,\mu_c)$

(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 $\sigma(s_0(a_0,\mu_0)s_1\ldots s_n)) = (a_n,\mu_n)$



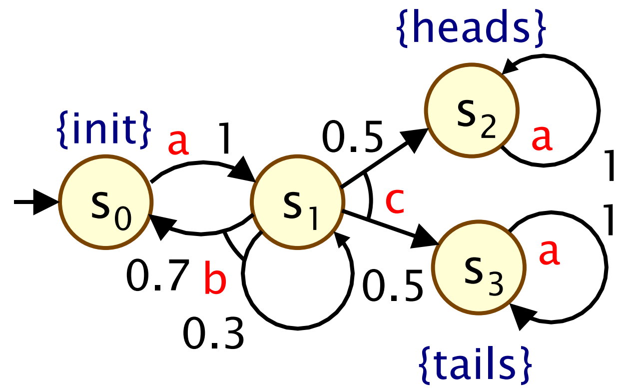
- Adversary σ_1
  - (picks action c the first time)
  - $Path^{\sigma_1}(s_0) = \{ s_0s_1s_2^{\omega}, s_0s_1s_3^{\omega} \}$
- Adversary σ_2
  - (picks action b the first time, then c)
  - $Path^{\sigma_2}(s_0) = \{ s_0s_1s_0s_1s_2^{\omega}, s_0s_1s_0s_1s_3^{\omega}, s_0s_1s_1s_2^{\omega}, s_0s_1s_1s_3^{\omega} \}$

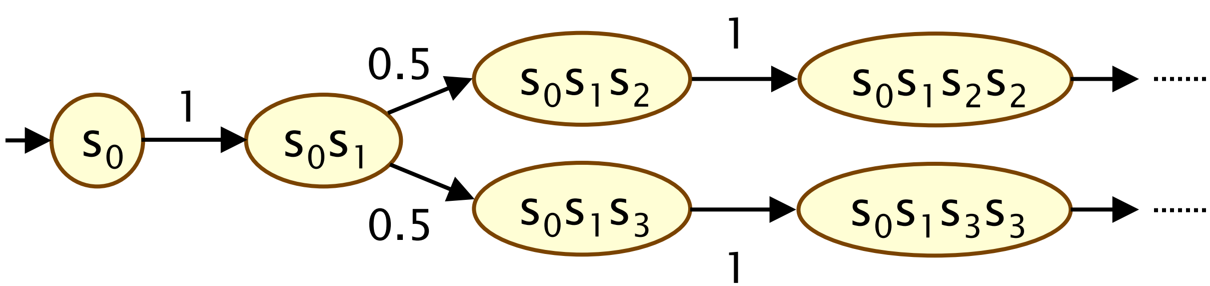
Induced DTMCs

- Adversary σ for MDP induces an infinite-state DTMC D^σ
- $D^\sigma = (Path^\sigma_{fin}(s), s, \mathbf{P}^\sigma_s)$ where:
  - states of the DTMC are the finite paths of σ starting in state s
  - initial state is s (the path starting in s of length 0)
  - $\mathbf{P}^\sigma_s(\omega,\omega') = \mu(s')$ if $\omega' = \omega(a, \mu)s'$ and $\sigma(\omega) = (a,\mu)$
  - $\mathbf{P}^\sigma_s(\omega,\omega') = 0$ otherwise
- 1-to-1 correspondence between $Path^\sigma(s)$ and paths of D^σ
- This gives us a probability measure Pr^σ_s over $Path^\sigma(s)$
  - from probability measure over paths of D^σ

Adversaries – Examples

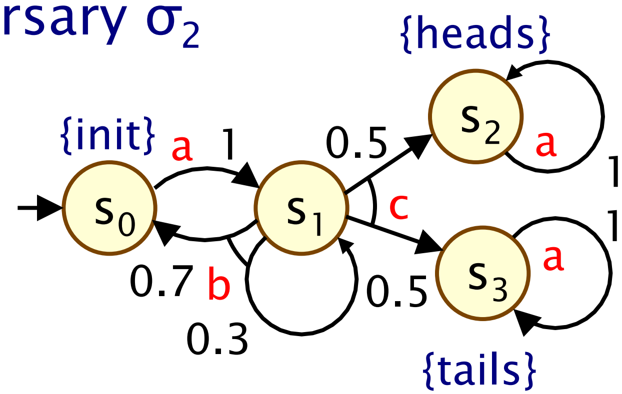
- Fragment of induced DTMC for adversary σ_1
  - σ_1 picks action c the first time

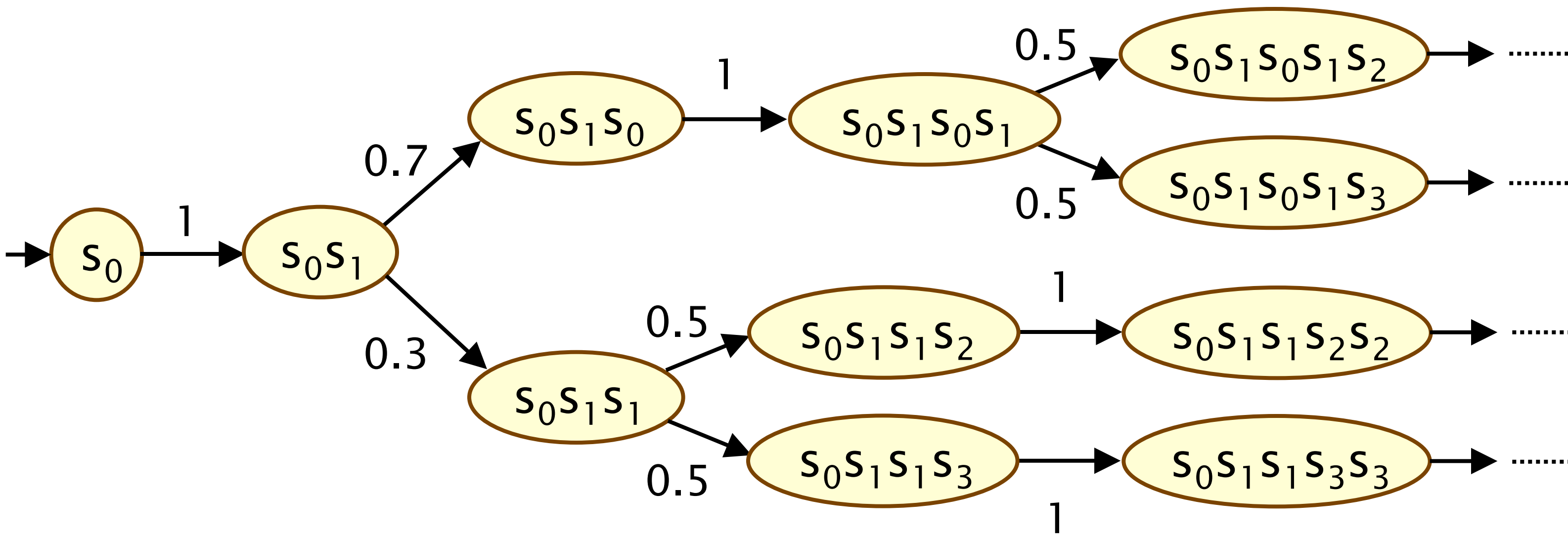




Adversaries – Examples

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



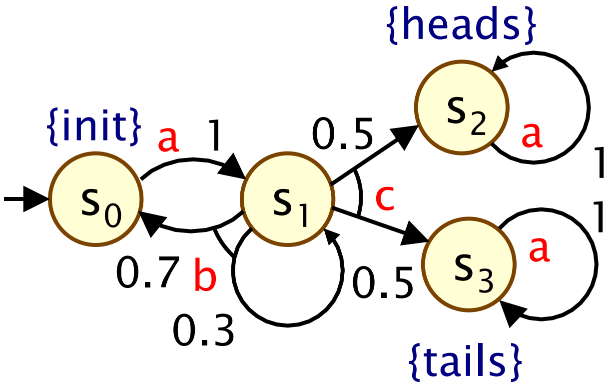


MDPs and probabilities

- $Prob^{\sigma}(s, \psi) = Pr^{\sigma}_{s} \{ \omega \in Path^{\sigma}(s) \mid \omega \vDash \psi \}$
  - for some path formula ψ
  - e.g. $Prob^{\sigma}(s, F\ tails)$
- MDP provides best-/worst-case analysis
  - based on lower/upper bounds on probabilities
  - over all possible adversaries

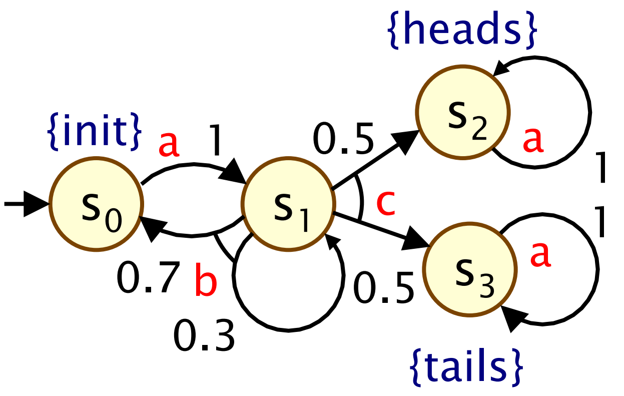
$$p_{min}(s,\psi) = \inf_{\sigma \in Adv} Prob^{\sigma}(s,\psi)$$

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

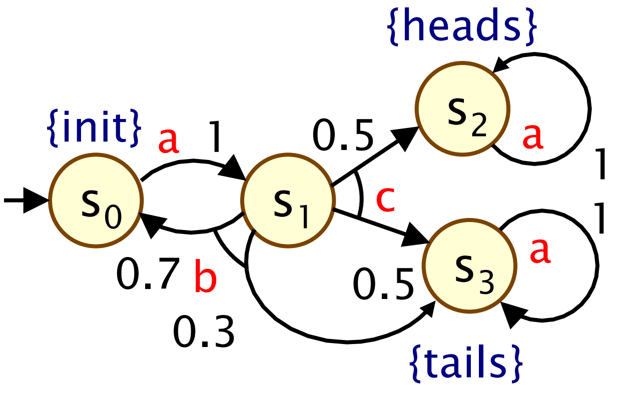


Examples

- $Prob^{\sigma 1}(s_0, F\ tails) = 0.5$
- $Prob^{\sigma 2}(s_0, F\ tails) = 0.5$
  - (where σ_i picks b i−1 times then c)
- …
- $p_{max}(s_0, F\ tails) = 0.5$
- $p_{min}(s_0, F\ tails) = 0$

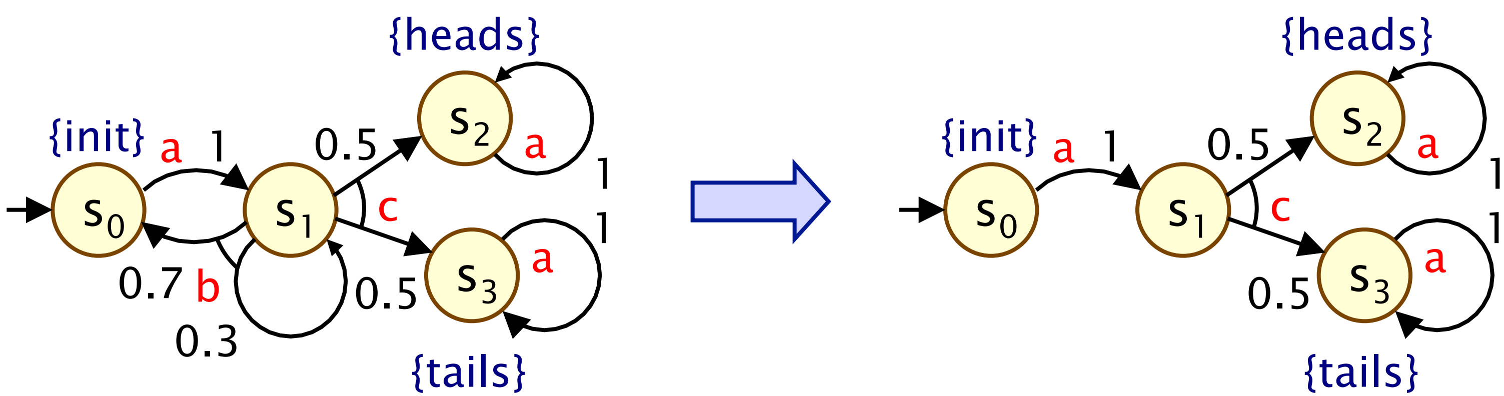


- $Prob^{\sigma 1}(s_0, F\ tails) = 0.5$
- $Prob^{\sigma 2}(s_0, F\ tails)$
  $= 0.3+0.7\cdot 0.5 = 0.65$
- $Prob^{\sigma 3}(s_0, F\ tails)$
  $= 0.3+0.7\cdot 0.3+0.7\cdot 0.7\cdot 0.5 = 0.755$
- …
- $p_{max}(s_0, F\ tails) = 1$
- $p_{min}(s_0, F\ tails) = 0.5$



Memoryless adversaries

- Memoryless adversaries always pick same choice in a state
  - also known as: positional, Markov, simple
  - formally, $\sigma(s_0(a_0,\mu_0)s_1...s_n)$ depends only on s_n
  - can write as a mapping from states, i.e. $\sigma(s)$ for each $s \in S$
  - induced DTMC can be mapped to a $|S|$-state DTMC
- From previous example:
  - adversary σ_1 (picks c in s_1) is memoryless; σ_2 is not



Other classes of adversary

- Finite-memory adversary
  - finite number of modes, which can govern choices made
  - formally defined by a deterministic finite automaton
  - induced DTMC (for finite MDP) again mapped to finite DTMC

- Randomised adversary
  - maps finite paths $s_0(a_1,\mu_1)s_1...s_n$ in MDP to a probability distribution over element of **Steps**(s_n)
  - generalises deterministic schedulers
  - still induces a (possibly infinite state) DTMC

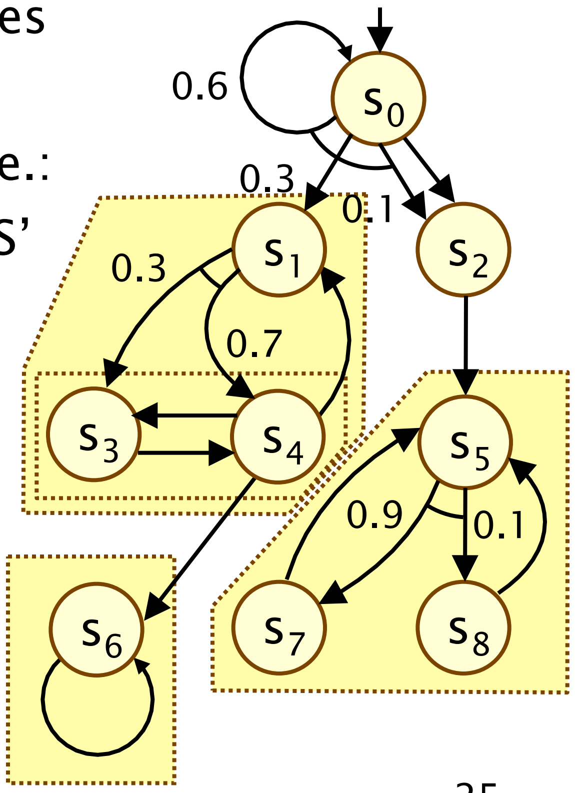
- Fair adversary
  - fairness assumptions on resolution of nondeterminism

End components

- Consider an MDP M = (S,s_{init},**Steps**,L)

- A sub-MDP of M is a pair (S',**Steps'**) where:
  - S' ⊆ S is a (non-empty) subset of M's states
  - **Steps'**(s) ⊆ **Steps**(s) for each s ∈ S'
  - is closed under probabilistic branching, i.e.:
  - { s' | μ(s')>0 for some (a,μ)∈**Steps'**(s) } ⊆ S'

- An end component of M is a strongly connected sub-MDP



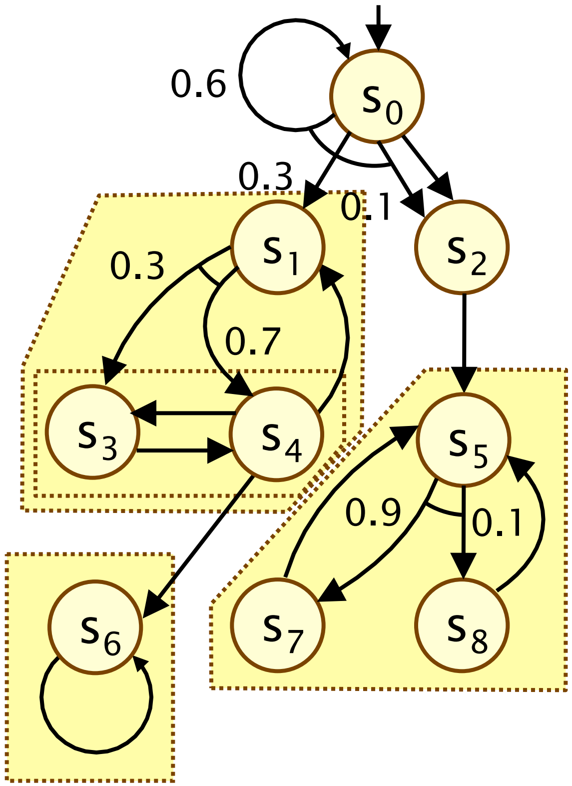
End components

- For finite MDPs…

- For every end component, there is an adversary which, with probability 1, forces the MDP to remain in the end component and visit all its states infinitely often

- Under every adversary σ, with probability 1 an end component will be reached and all of its states visited infinitely often

  – (analogue of fundamental property of finite DTMCs)



Summing up…

- Nondeterminism
  - concurrency, unknown environments/parameters, abstraction
- Markov decision processes (MDPs)
  - discrete-time + probability and nondeterminism
  - nondeterministic choice between multiple distributions
- Adversaries
  - resolution of nondeterminism only
  - induced set of paths and (infinite state DTMC)
  - induces DTMC yields probability measure for adversary
  - best-/worst-case analysis: minimum/maximum probabilities
  - memoryless adversaries
- End components
  - long-run behaviour: analogue of BSCCs for DTMCs