Probabilistic Systems

Part 2: Markov decision processes

Dr. Gethin Norman

School of Computing Science
University of Glasgow

gethin.norman@glasgow.ac.uk

Markov Decision Processes

Markov Decision Processes (MDPs)

Paths, strategies and probabilities for MDPs

Probabilistic reachability for MDPs

- qualitative probabilistic reachability
- optimality equations
- computing reachability probabilities

Nondeterminism

Some aspects of a system may not be probabilistic and therefore 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}

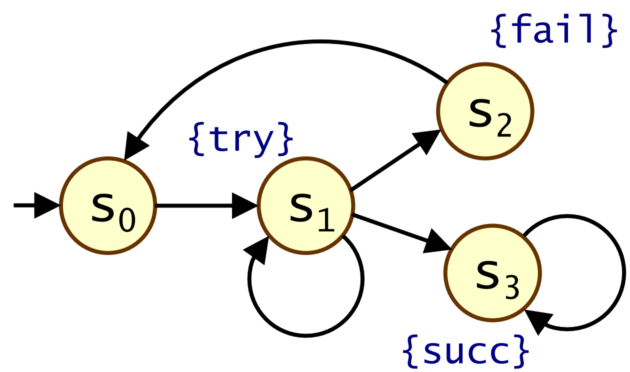
Abstraction

- e.g. partition DTMC into similar (but not identical) states

Probability vs. nondeterminism

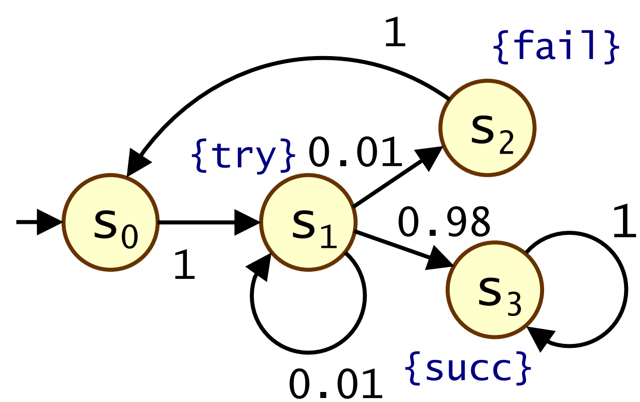
Labelled transition system

- (S, s_0, T, L) where $T \subseteq S \times S$
- 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 choices

Like DTMCs:

- discrete set of states representing possible configurations of the system being modelled
- transitions between states occur in discrete time-steps

Probabilistic and nondeterministic behaviour in each state:

- a nondeterministic choice between available actions
- once an action is chosen the successor state is chosen probabilistically based on the action and the current state

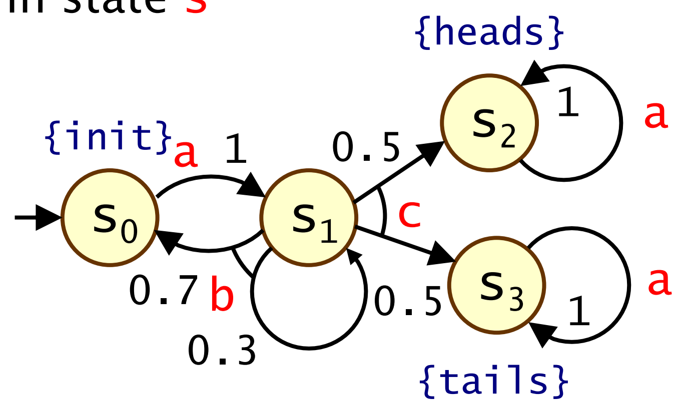
Markov decision processes

Formally, an MDP M is a tuple (S,s_0,P,L) where:

- S is a finite set of states ("state space")
- $s_0 \in S$ is a initial state
- $L:S \rightarrow 2^{AP}$ is a labelling function
- $P:S \times A \rightarrow Dist(S)$ is a (partial) transition probability function

  where A is a set of actions and $Dist(S)$ is the set of discrete probability distributions over the set of states S

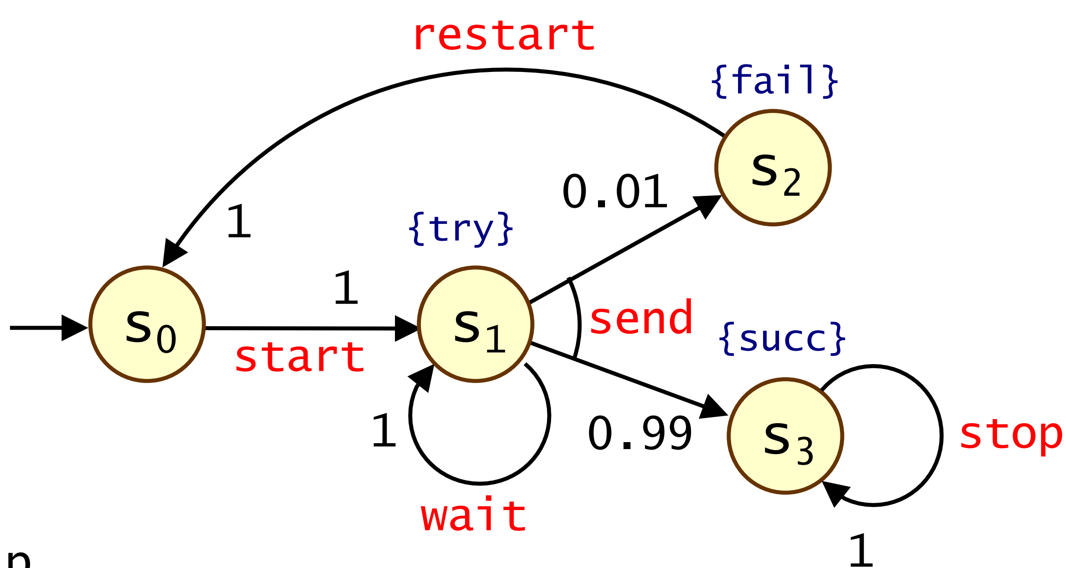
  - in state s, action a is available (can be performed) if $P(s,a)$ is defined
  - we denote by $A(s)$ the available actions in state s



Simple MDP example

Modification of the simple DTMC communication protocol

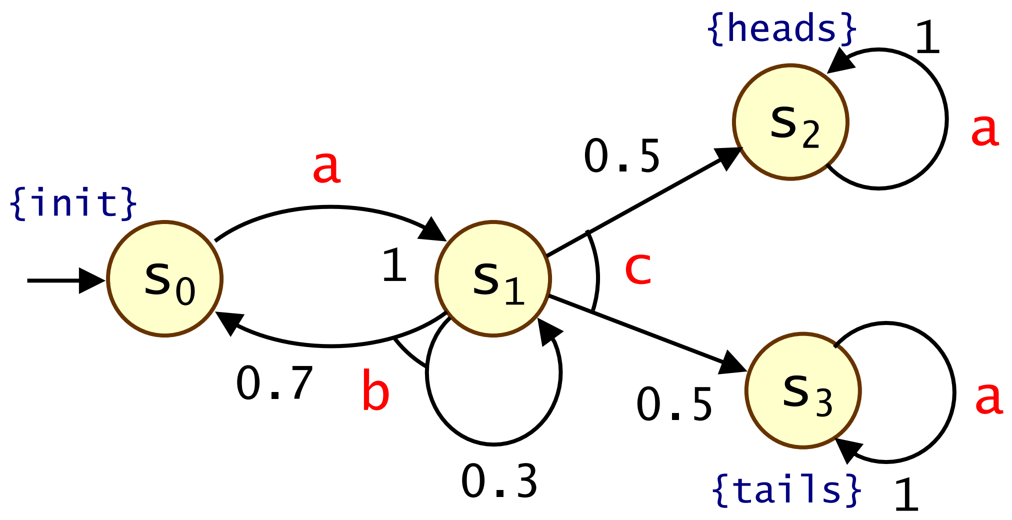
- after one step, process starts trying to send a message
- then, a nondeterministic choice between: (a) waiting a step because the channel is busy; (b) sending the message
- if the latter, with probability 0.99 send successfully and stops and with probability 0.01, message sending fails, and protocol restarts



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 50-50 random choice between heads/tails



Simple MDP example 2

$M = (S, s_0, \mathbf{P}, L)$

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

$AP = \{init, heads, tails\}$

$L(s_0)=\{init\}$

$L(s_1)=\emptyset$

$L(s_2)=\{heads\}$

$L(s_3)=\{tails\}$

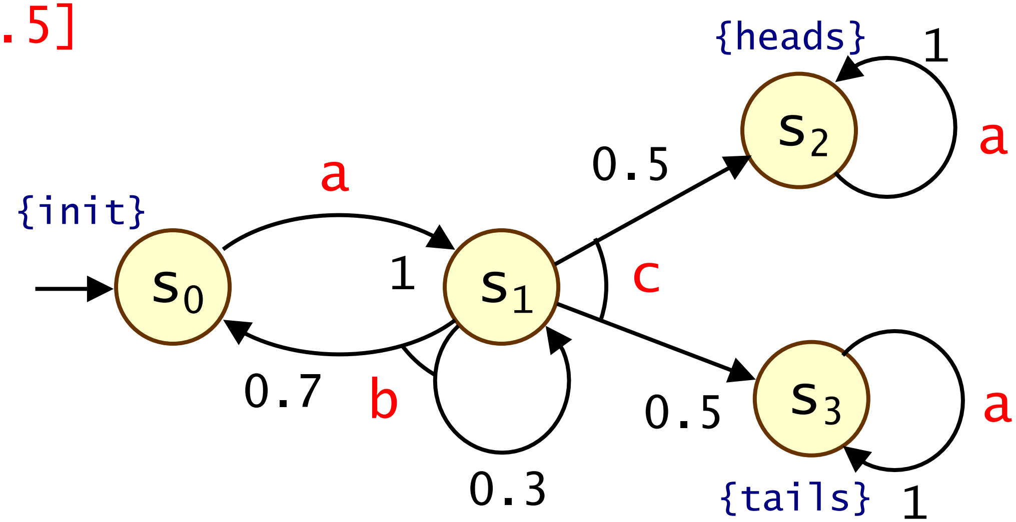
$\mathbf{P}(s_0,a) = [s_1 \mapsto 1]$

$\mathbf{P}(s_1,b) = [s_0 \mapsto 0.7, s_1 \mapsto 0.3]$

$\mathbf{P}(s_1,c) = [s_2 \mapsto 0.5, s_3 \mapsto 0.5]$

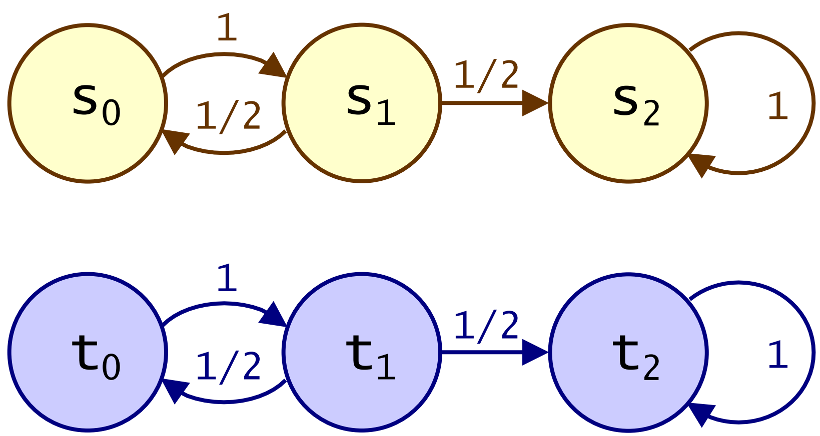
$\mathbf{P}(s_2,a) = [s_2 \mapsto 1]$

$\mathbf{P}(s_3,a) = [s_3 \mapsto 1]$



Example – Parallel composition

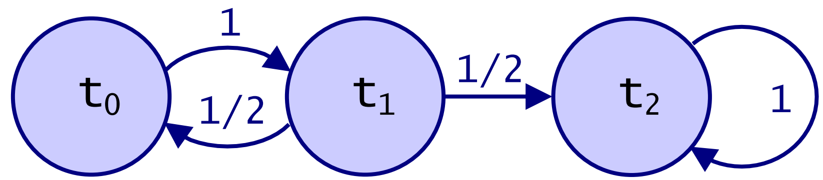
Asynchronous parallel composition of two 3-state DTMCs

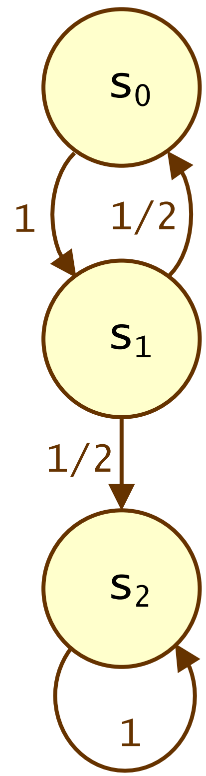


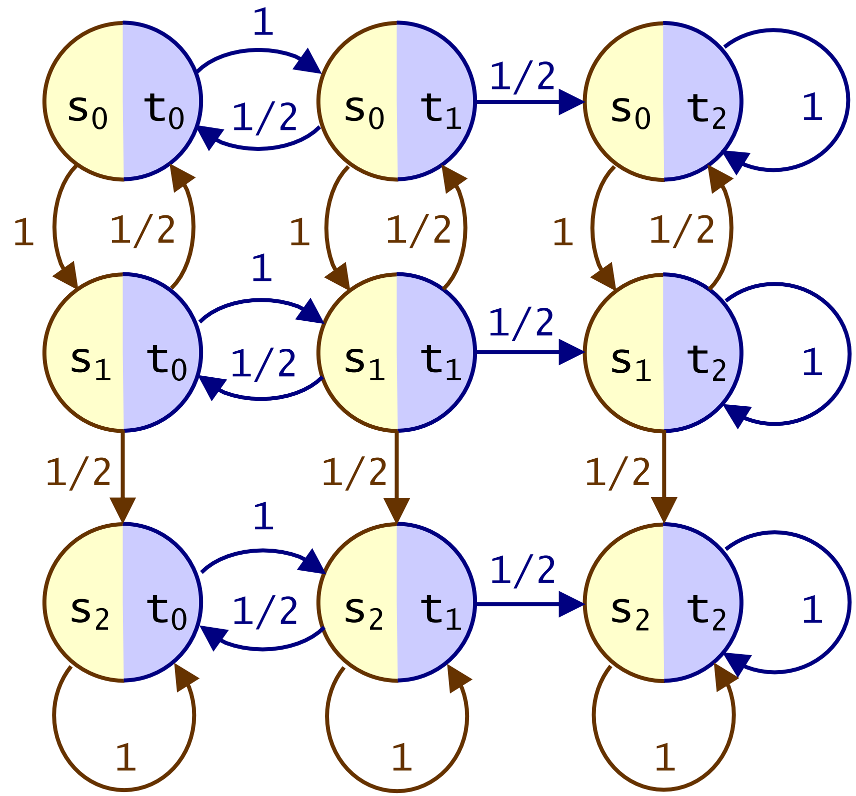
Example – Parallel composition

Asynchronous parallel composition of two 3-state DTMCs

Action labels omitted here







Markov Decision Processes

Markov Decision Processes (MDPs)

Paths, strategies and probabilities for MDPs

Probabilistic reachability for MDPs

- qualitative probabilistic reachability
- optimality equations
- computing reachability probabilities

Paths and probabilities

A (finite or infinite) path through an MDP

- is a sequence $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \ldots$
- such that $P(s,a_i)(s_{i+1})>0$ for all $i \geq 0$
- represents an execution (i.e. one possible behaviour) of the system which the MDP is modelling

Path(s) is the set of all infinite paths of MDP starting from state s

- $Path_{fin}(s)$ is the set of all finite paths starting from state s

Paths resolve both nondeterministic and probabilistic choices

- how to reason about probabilities?

Strategies

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

A strategy resolves nondeterministic choice in an MDP

- also known as a “scheduler”, “policy” or “adversary”

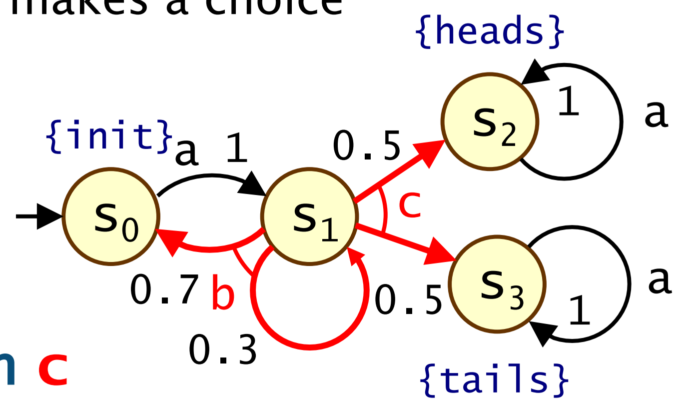
Formally:

- a strategy σ of an MDP is a function mapping every finite path $\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} s_n$ to an available action of s_n
  - i.e. resolves nondeterminism based on execution history
  - given what has happened (the history) what action to perform next

Strategies – Examples

Consider the previous example MDP

- note s_1 is the only state for which there is more than one available action
  - i.e. s_1 is the only state for which a strategy makes a choice



Strategy σ_1 picks action c the first time

- $\sigma_1(s_0 s_1)=c$

Strategy σ_2 picks action b the first time, then c

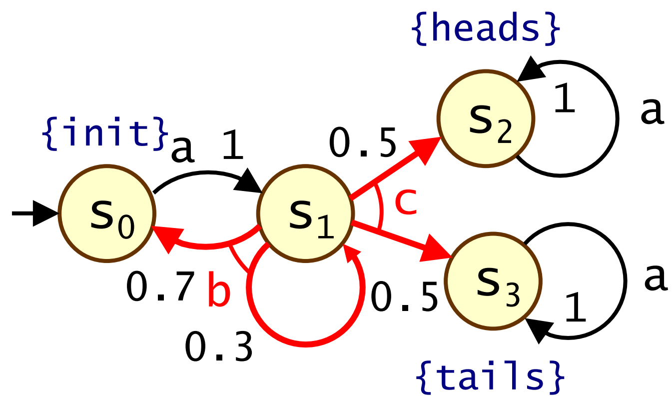
- $\sigma_2(s_0 s_1)=b$
- $\sigma_2(s_0 s_1 s_1)=c$
- $\sigma_2(s_0 s_1 s_0 s_1)=c$

Note: actions omitted from paths for clarity

Strategies and paths

$Path^{\sigma}(s) \subseteq Path(s)$

- (infinite) paths from s where nondeterminism resolved by σ
- i.e. paths are of the form $\pi \xrightarrow{a} s$ and $\sigma(\pi) = (a)$



Strategy σ_1 picks action c the first time

- $Path^{\sigma_1}(s_0) = \{ s_0 s_1 s_2^{\omega},\ s_0 s_1 s_3^{\omega} \}$

Strategy σ_2 picks action b the first time, then c

- $Path^{\sigma_2}(s_0) = \{ s_0 s_1 s_0 s_1 s_2^{\omega},\ s_0 s_1 s_0 s_1 s_3^{\omega},\ s_0 s_1 s_1 s_2^{\omega},\ s_0 s_1 s_1 s_3^{\omega} \}$

Strategies – Induced DTMCs

For a given starting state s, a strategy σ of an MDP induces an infinite-state DTMC $D(s,\sigma)$

$D(s,\sigma) = (Path^{\sigma}_{fin}(s), s, P^{\sigma}, L)$ where:

- states of the DTMC are the finite paths of the MDP starting in state s
- initial state is s (the path starting in s of length 0)
- $P^{\sigma}(\pi, \pi') = P(last(\pi), a)(s')$ if $\pi' = \pi \xrightarrow{a} s'$ and $\sigma(\pi) = a$
- $P^{\sigma}(\pi, \pi') = 0$ otherwise
- labelling of a path just given by the labelling of the last state of the path

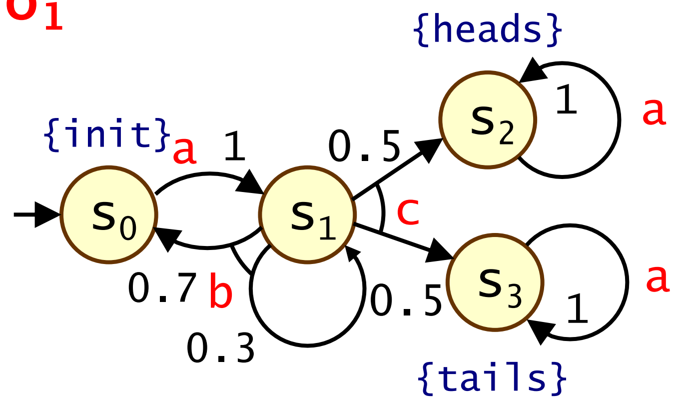
1-to-1 correspondence between $Path^{\sigma}(s)$ and paths of $D(s,\sigma)$

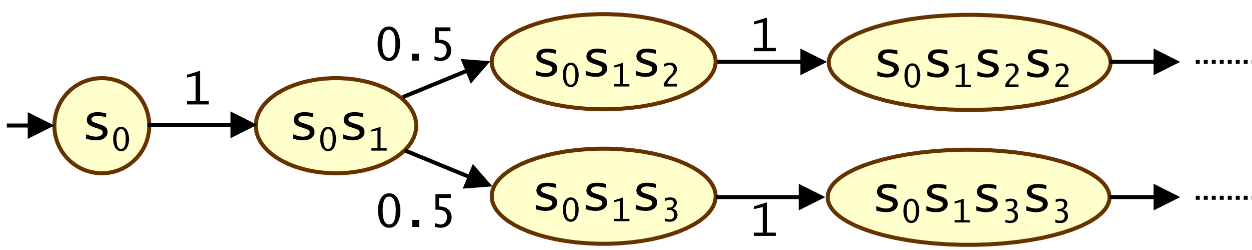
This therefore gives us a probability measure over $Path^{\sigma}(s)$

- by using probability measure over the paths of $D(s,\sigma)$

Strategies – Examples

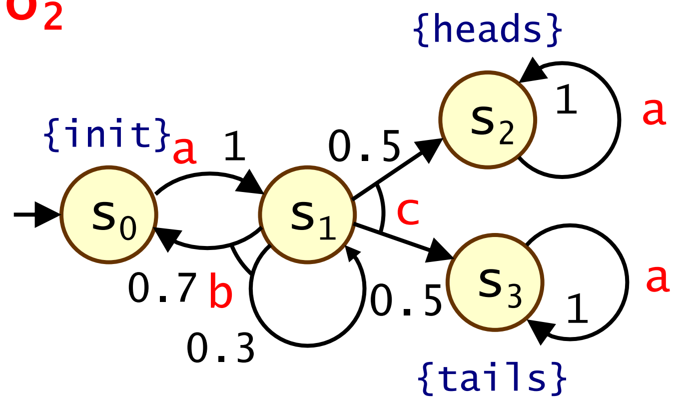
- **Fragment of induced DTMC for strategy σ_1**
  - σ_1 picks action c the first time

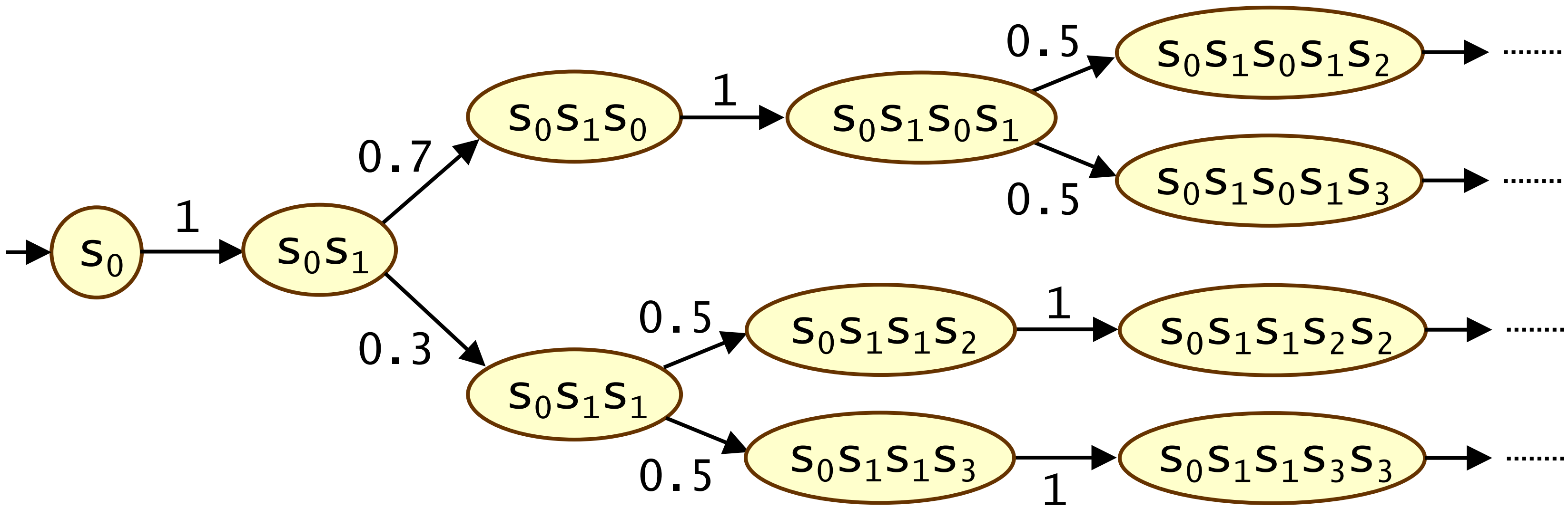




Strategies – Examples

- Fragment of induced DTMC for strategy σ_2
  - σ_2 picks action b the first time, and then c





Markov Decision Processes

Markov Decision Processes (MDPs)

Paths, strategies and probabilities for MDPs

Probabilistic reachability for MDPs

- qualitative probabilistic reachability
- optimality equations
- computing reachability probabilities

Probabilistic Reachability

Probabilistic reachability

- fundamental concept in quantitative verification
- concerns probability of reaching a target set T
  - $P^{\sigma}(s,T)$ probability of reaching T under the strategy σ from state s
  - as for DTMCs

MDP provides best-/worst-case analysis

- based on lower/upper bounds on probabilities over all strategies
- $P^{min}(s,T) = \inf_{\sigma} P^{\sigma}(s,T)$
  - the minimum probability of reaching T over all strategies
- $P^{max}(s,T) = \sup_{\sigma} P^{\sigma}(s,T)$
  - the maximum probability of reaching T over all strategies
- vectors: $P^{min}(T)$ and $P^{max}(T)$ values for all states of an MDP

Examples – target T equals {tails}

Consider strategy σ_i that first selects b the first i-1 times in state s_1 and then c

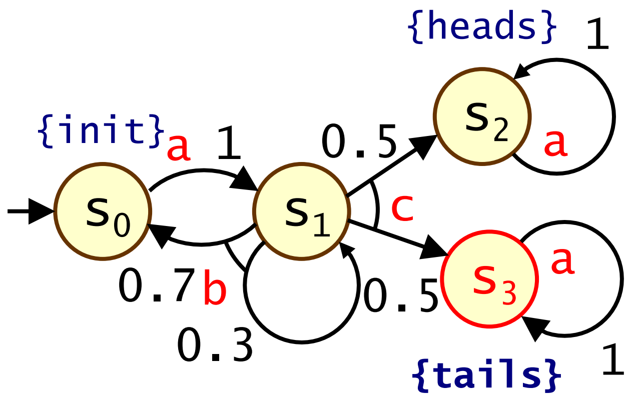
$P^{\sigma 1}(s_0,T) = 0.5$

$P^{\sigma 2}(s_0,T) = 0.5$

…

$P^{min}(s_0,T) = 0.0$

$P^{max}(s_0,T) = 0.5$



Examples – target T equals {tails}

Consider strategy σ_i that first selects b the first i-1 times in state s_1 and then c

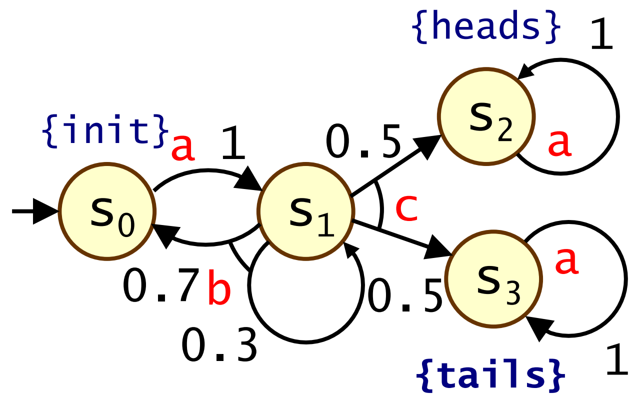
$P^{\sigma 1}(s_0,T) = 0.5$

$P^{\sigma 2}(s_0,T) = 0.5$

…

$P^{min}(s_0,T) = 0.0$

$P^{max}(s_0,T) = 0.5$



$P^{\sigma 1}(s_0,T) = 0.5$

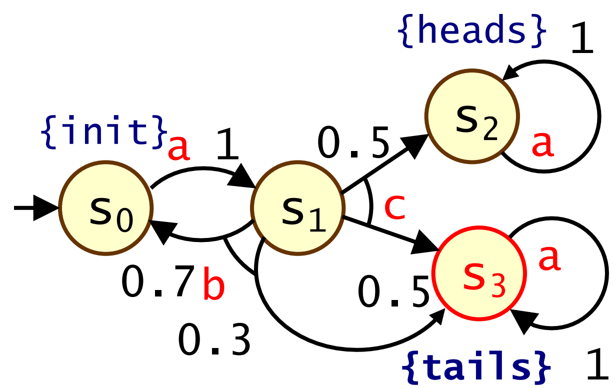
$P^{\sigma 2}(s_0,T) = 0.3+0.7\cdot 0.5 = 0.65$

$P^{\sigma 3}(s_0,T) = 0.3+0.7\cdot 0.3+0.7\cdot 0.7\cdot 0.5 = 0.755$

…

$P^{min}(s_0,T) = 0.5$

$P^{max}(s_0,T) = 1.0$



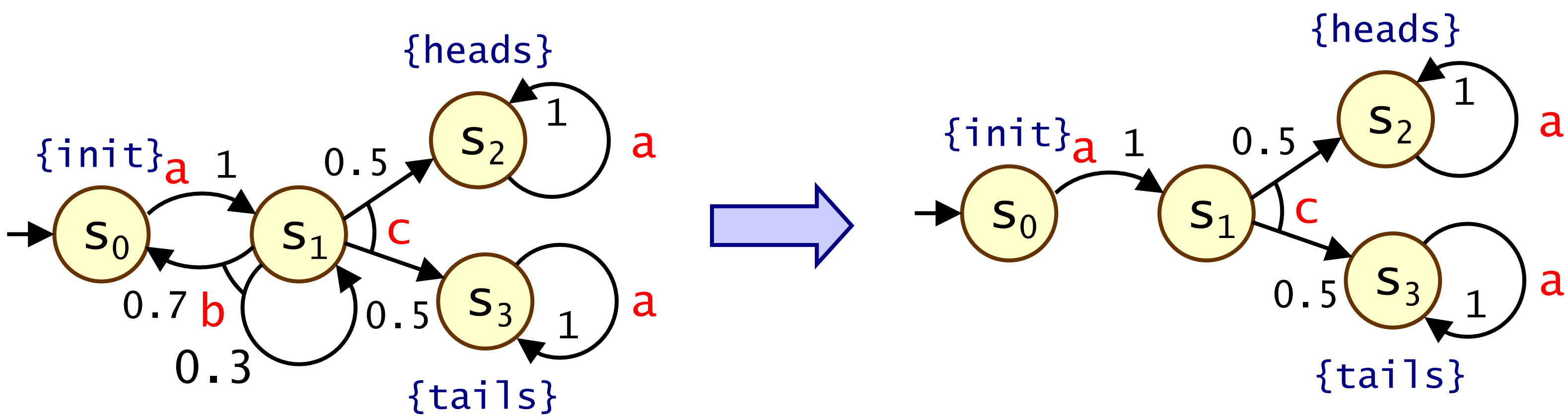
Memoryless strategies

Memoryless strategies always pick same choice in a state

- also known as: positional, Markov, simple
- can write as a mapping from states to available actions
- induced DTMC can be mapped to a |S|-state DTMC

From previous example:

- strategy σ_1 (picks c in s_1) is memoryless; σ_2 is not



Other classes of strategies

Finite-memory strategies

- 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 strategies

- maps finite paths to a probability distribution over available actions
- generalises deterministic schedulers
- still induces a (possibly infinite state) DTMC

Fair strategies

- fairness assumptions on resolution of nondeterminism

Markov Decision Processes

Markov Decision Processes (MDPs)

Paths, strategies and probabilities for MDPs

Probabilistic reachability for MDPs

- qualitative probabilistic reachability
- optimality equations
- computing reachability probabilities

Qualitative probabilistic reachability

Consider the problem of determining states s for which $P^{min}(s,T)$ or $P^{max}(s,T)$ is non-zero (or zero)

- max case: $S^{max>0} = \{ s \in S \mid P^{max}(s,T)>0 \}$
- this is just (non-probabilistic) reachability

```
R := T
done := false
while (done = false)
    R' = R ∪ { s∈S | ∃a∈A . ∃s'∈R .P(s,a)(s')>0}
    if (R'=R) then done := true
    R := R'
endwhile
return R
```

Qualitative probabilistic reachability

Consider the problem of determining states s for which $P^{min}(s,T)$ or $P^{max}(s,T)$ is non-zero (or zero)

– min case: $S^{min>0} = \{ s \in S \mid P^{min}(s,T)>0 \}$

note: quantification over all choices

```
R := T
done := false
while (done = false)
    R' = R ∪ { s∈S | ∀a∈A . ∃s'∈R .P(s,a)(s')>0}
    if (R'=R) then done := true
    R := R'
endwhile
return R
```

Probabilistic Reachability – Optimality equations

The values $P^{min}(s,T)$ are the unique solution of the equations:

$$x_s = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \in S \setminus S^{min>0} \\ \min_{a \in A(s)} \{ \sum_{s' \in S} P(s,a)(s') \cdot x_{s'} \} & \text{otherwise} \end{cases}$$

optimal solution for state s uses optimal solution for successors s'

case when $P^{min}(s,T)=0$

This is an instance of the Bellman equation, the basis of dynamic programming techniques

Probabilistic Reachability – Optimality equations

The values $P^{max}(s,T)$ are the unique solution of the equations:

$$x_s = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \in S \backslash S^{max>0} \\ \max_{a \in A(s)} \{ \sum_{s' \in S} P(s,a)(s') \cdot x_{s'} \} & \text{otherwise} \end{cases}$$

case when $P^{max}(s,T)=0$

Memoryless strategies

Recall memoryless strategies always pick same choice in a state

Memoryless strategies suffice for probabilistic reachability

- i.e. there exist memoryless strategies σ_{min} and σ_{max} such that:
- $P^{\sigma_{min}}(s,T) = P^{min}(s,T)$ for all states $s \in S$
- $P^{\sigma_{max}}(s,T) = P^{max}(s,T)$ for all states $s \in S$

Can construct memoryless strategies from optimal solution:

- $\sigma_{min}(s) = \text{argmin} \{ \Sigma_{s \in S} P(s,a)(s') \cdot P^{min}(s,T) \mid a \in A(s) \}$
- $\sigma_{max}(s) = \text{argmax} \{ \Sigma_{s \in S} P(s,a)(s') \cdot P^{max}(s,T) \mid a \in A(s) \}$

Memoryless strategies

Memoryless strategies not always sufficient

- although they are sufficient for reachability in turn-based games

Finite-memory strategies are required for

- bounded properties
- LTL and automata-based properties

Randomized strategies are required for concurrent games

Finite-memory strategies and randomised strategies are required for multi-objective properties

Markov Decision Processes

Markov Decision Processes (MDPs)

Paths, strategies and probabilities for MDPs

Probabilistic reachability for MDPs

- qualitative probabilistic reachability
- optimality equations
- computing reachability probabilities

Computing reachability probabilities

Several approaches…

Value iteration

- approximate with iterative solution method
- corresponds to fixed point computation

Preferable in practice, e.g. in PRISM

Reduction to a linear programming (LP) problem

- solve with linear optimisation techniques
- exact solution using well-known methods

better complexity; good for small examples

Policy iteration

- iteration over strategies

Method 1 – Value iteration (min)

- For **minimum** probabilities $\mathbf{P^{min}(s,T)}$ it can be shown that:
  - $P^{min}(s,T) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \in S \setminus S^{min>0} \\ 0 & \text{if } n=0 \\ \min_{a \in A(s)} \{ \sum_{s' \in S} P(s,a)(s') \cdot x_{s'}^{(n-1)} \} & \text{otherwise} \end{cases}$$

- Approximate iterative solution technique
  - iterations terminated when solution converges sufficiently

Method 1 – Value iteration (max)

- For **maximum** probabilities $P^{max}(s,T)$ it can be shown that:
  - $P^{max}(s,T) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \in S \backslash S^{max>0} \\ 0 & \text{if } n=0 \\ \max_{a\in A(s)} \{ \sum_{s'\in S} P(s,a)(s') \cdot x_{s'}^{(n-1)} \} & \text{otherwise} \end{cases}$$

- Approximate iterative solution technique
  - iterations terminated when solution converges sufficiently

Value iteration as a fixed point

Can view as a **fixed point** computation over vectors $\mathbf{y} \in [0,1]^S$

- for example, for minimum consider the function $F : [0,1]^S \rightarrow [0,1]^S$

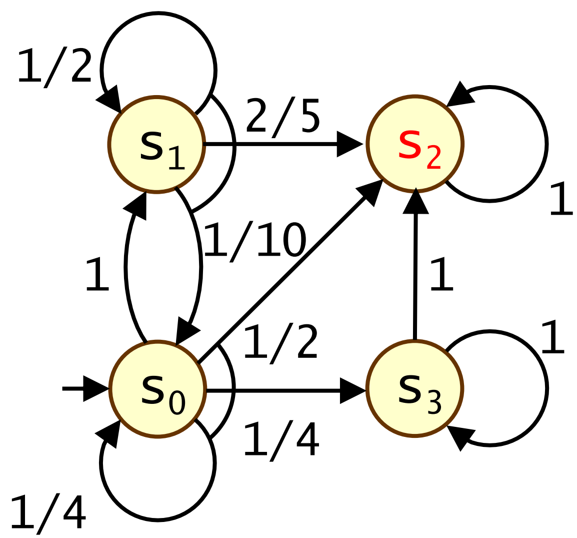
$$F(y)(s) = \begin{cases} 1 & \text{if } s \in T \\ 0 & \text{if } s \in S \setminus S^{min>0} \\ \min_{a \in A(s)} \{ \sum_{s' \in S} P(s,a)(s') \cdot y(s') \} & \text{otherwise} \end{cases}$$

If we let $\mathbf{x^{(0)}=0}$ and $\mathbf{x^{(n+1)}=F(x^{(n)})}$ then we have that

- $x^{(0)} \leq x^{(1)} \leq x^{(2)} \leq x^{(3)} \leq \ldots$
- $P^{min}(T) = \lim_{n\rightarrow\infty} x^{(n)}$
- $F(P^{min}(T)) = P^{min}(T)$ and it is the unique fixed point

Example

Minimum/maximum probability of reaching $T=\{s_2\}$



Example – Value iteration (min)

Compute: $P^{min}(s_i, T)$ where $T=\{s_2\}$

$S^{min>0} = \{s_0,\ s_1,\ s_2\}$

$[\ x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}\ ]$

n=0: [0, 0, 1, 0]

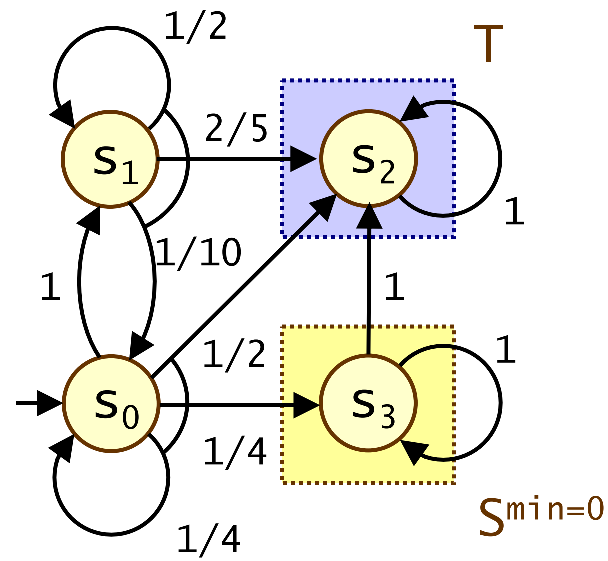
n=1: [min(1·0,0.25·0+0.25·0+0.5·1),0.01·0+0.5·0+0.4·1, 1, 0]

=[0, 0.4, 1, 0]

n=2: [min(1·0.4,0.25·0+0.25·0+0.5·1),0.01·0+0.5·0.4+0.4·1, 1, 0]

=[0.4, 0.6, 1, 0]

n=3: …

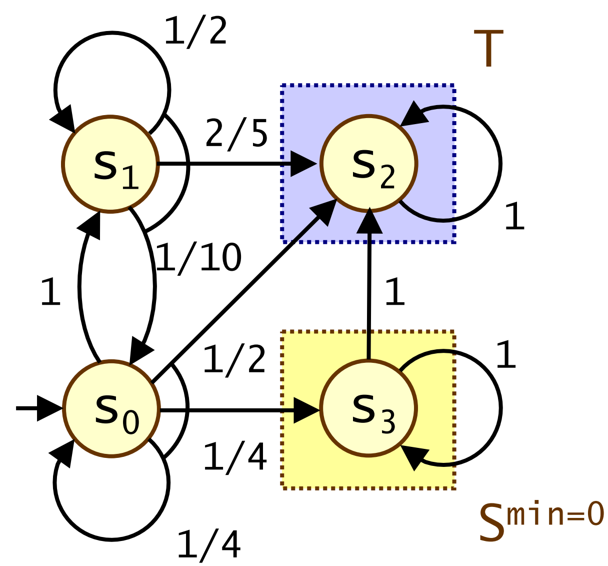


Example – Value iteration (min)

[$x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}$]

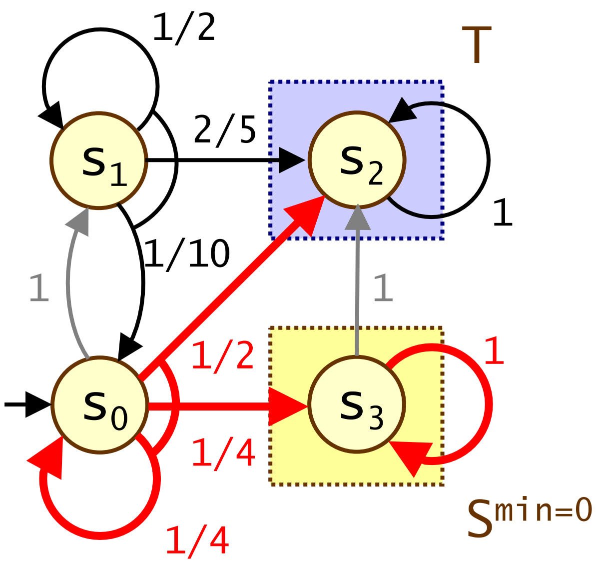
```
n=0:  [ 0.000000, 0.000000, 1, 0 ]
n=1:  [ 0.000000, 0.400000, 1, 0 ]
n=2:  [ 0.400000, 0.600000, 1, 0 ]
n=3:  [ 0.600000, 0.740000, 1, 0 ]
n=4:  [ 0.650000, 0.830000, 1, 0 ]
n=5:  [ 0.662500, 0.880000, 1, 0 ]
n=6:  [ 0.665625, 0.906250, 1, 0 ]
n=7:  [ 0.666406, 0.919688, 1, 0 ]
n=8:  [ 0.666602, 0.926484, 1, 0 ]
…
n=20: [ 0.666667, 0.933332, 1, 0 ]
n=21: [ 0.666667, 0.933332, 1, 0 ]
    ≈ [ 2/3,      14/15,    1, 0 ]
```

$\underline{P}^{min}(T)$=[2/3, 14/15, 1, 0]



Generating an optimal strategy

Minimum strategy σ_{min}



$[\ x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}\ ]$

…

n=20: [0.666667, 0.933332, 1, 0]

n=21: [0.666667, 0.933332, 1, 0]

$\approx$ [2/3, 14/15, 1, 0]

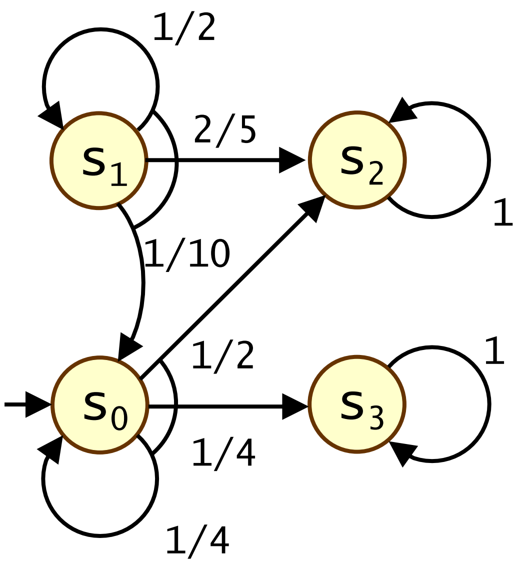
s_0 : min(1·14/15, 1/2·1+1/4·0+1/4·2/3)

= min(14/15, 2/3)

$s_3 \in S \backslash S^{min>0}$

Generating an optimal strategy

- DTMC $D(s_0,\sigma_{min})$



$[\ x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}\ ]$

…

n=20: [0.666667, 0.933332, 1, 0]

n=21: [0.666667, 0.933332, 1, 0]

≈ [2/3, 14/15, 1, 0]

s_0 : min(1·14/15, 1/2·1+1/4·0+1/4·2/3)

= min(14/15, 2/3)

Linear programming

Linear programming

- optimisation of a linear objective function
- subject to linear (in)equality constraints

Many standard solution techniques exist, e.g. Simplex, ellipsoid method, interior point method

General form:

- n variables: $x_1, x_2, \ldots, x_n$
- maximise (or minimise): $c_1x_1 + c_2x_2 + \cdots + c_nx_n$
- subject to constraints
  - $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2$

    $\vdots \qquad \vdots$
  - $a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m$

In matrix/vector form:
Maximise (or minimise)
$\mathbf{c} \cdot \mathbf{x}$
subject to
$\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$

Method 2 – Linear programming problem

Minimum probabilities $P^{min}(s,T)$ can be computed as follows:

- $P^{min}(s,T)=1$ if $s \in T$
- $P^{min}(s,T)=0$ if $s \in S \setminus S^{min>0}$
- values for remaining states $S^?$ can be obtained as the unique solution of the following linear programming problem:

> maximize $\Sigma_{s \in S^?} x_s$ subject to the constraints:
>
> $$x_s \leq \Sigma_{s' \in S^?} P(s,a)(s') \cdot x_{s'} + \Sigma_{s' \in T} P(s,a)(s')$$
>
> for all $s \in S^?$ and $a \in A(s)$

Method 2 – Linear programming problem

Maximum probabilities $P^{max}(s,T)$ can be computed as follows:

- $P^{max}(s,T)=1$ if $s \in T$
- $P^{max}(s,T)=0$ if $s \in S \backslash S^{max>0}$
- values for remaining states $S^?$ can be obtained as the unique solution of the following linear programming problem:

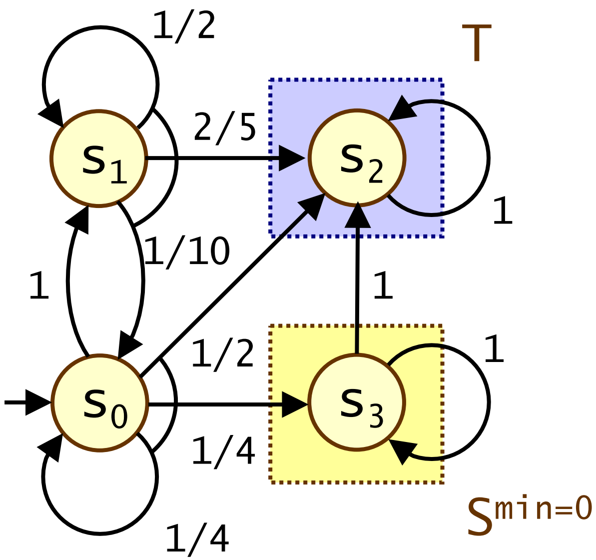
minimize $\Sigma_{s \in S^?} x_s$ subject to the constraints:

$$x_s \geq \Sigma_{s' \in S^?} P(s,a)(s') \cdot x_{s'} + \Sigma_{s' \in T} P(s,a)(s')$$

for all $s \in S^?$ and $a \in A(s)$

differences from min case

Example – Linear programming (min)



Let $x_i = P^{min}(s_i, T)$

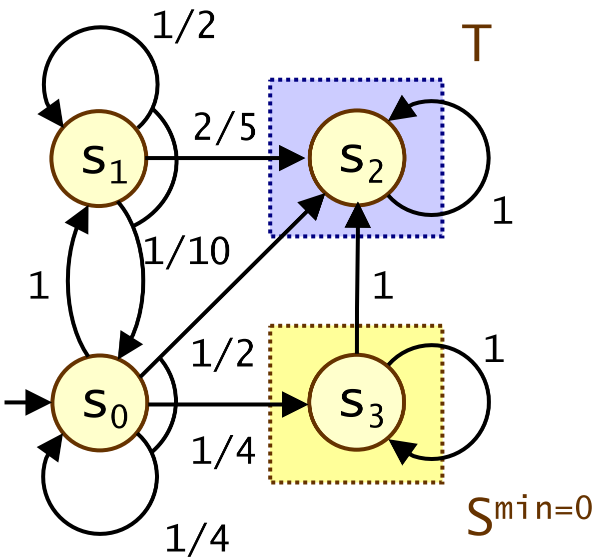
T: $x_2=1$, $S^{min=0}$: $x_3=0$

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

maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $x_0 \leq 1/4 \cdot x_0 + 1/2$
- $x_1 \leq 1/10 \cdot x_0 + 1/2 \cdot x_1 + 2/5$

Example – Linear programming (min)



Let $x_i = P^{min}(s_i, T)$

T: $x_2=1$, $S^{min=0}$: $x_3=0$

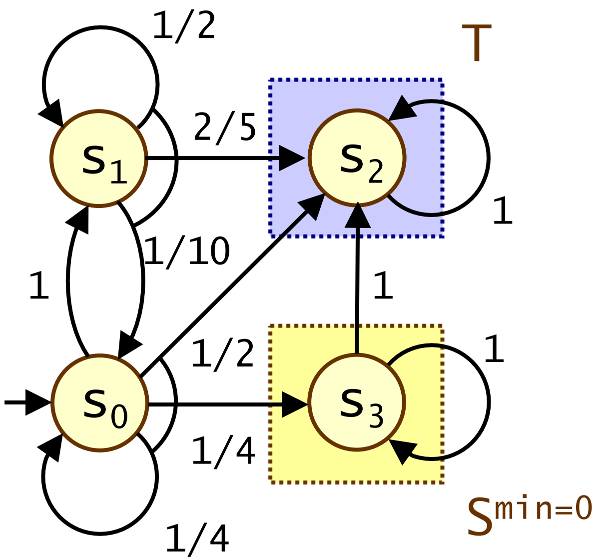
For $S^? = \{s_0, s_1\}$:

maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $3/4 \cdot x_0 \leq 1/2$
- $1/2 \cdot x_1 \leq 1/10 \cdot x_0 + 2/5$

rearranging

Example – Linear programming (min)



Let $x_i = P^{min}(s_i, T)$

T: $x_2=1$, $S^{min=0}$: $x_3=0$

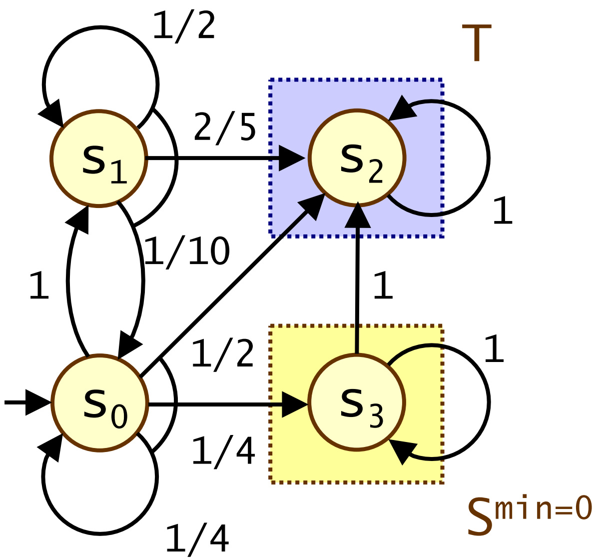
For $S^? = \{s_0, s_1\}$:

maximise x_0+x_1 subject to constraints:

- $x_0 \le x_1$
- $x_0 \le 2/3$
- $x_1 \le 1/5 \cdot x_0 + 4/5$

rearranging

Example – Linear programming (min)



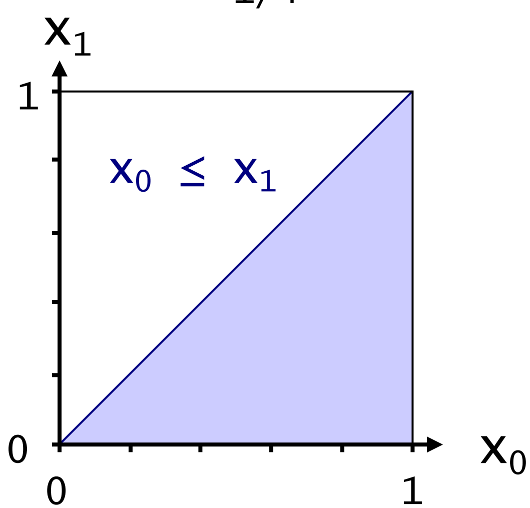
Let $x_i = P^{min}(s_i, T)$

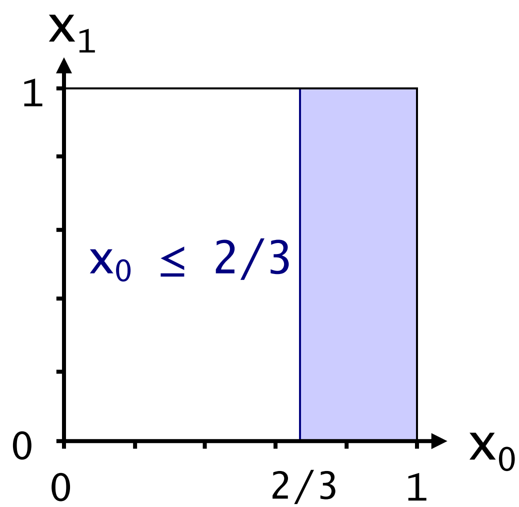
T: $x_2=1$, $S^{min=0}$: $x_3=0$

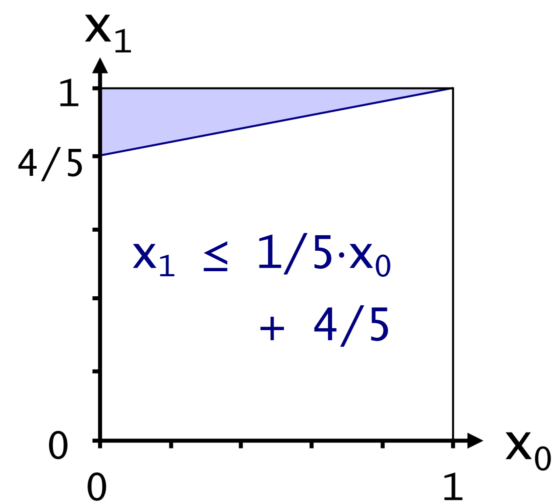
For $S^? = \{s_0, s_1\}$:

maximise x_0+x_1 subject to constraints:

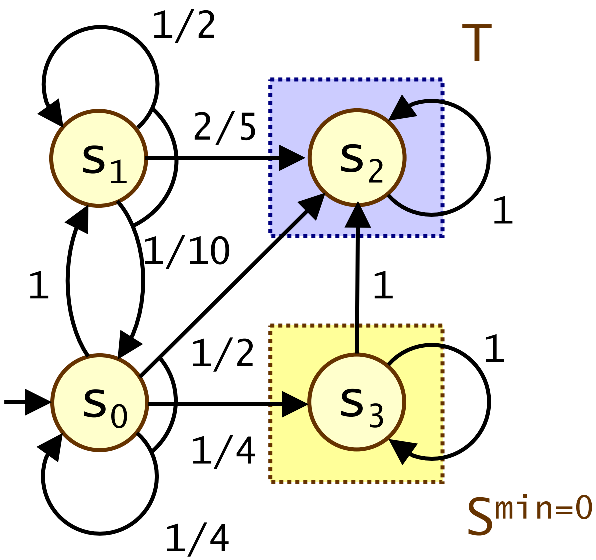
- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 1/5 \cdot x_0 + 4/5$







Example – Linear programming (min)



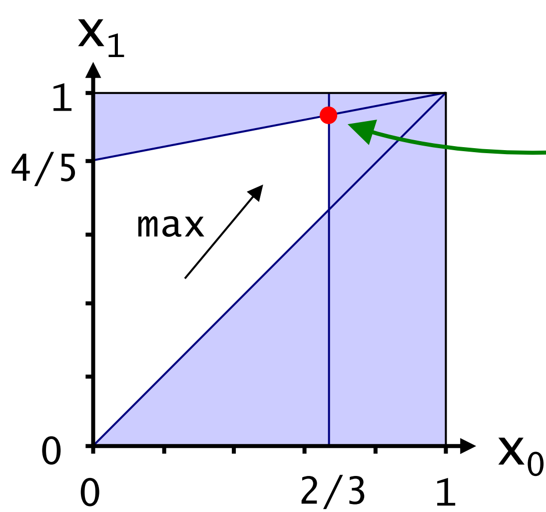
Let $x_i = P^{min}(s_i, T)$

T: $x_2=1$, $S^{min=0}$: $x_3=0$

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

maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 1/5 \cdot x_0 + 4/5$

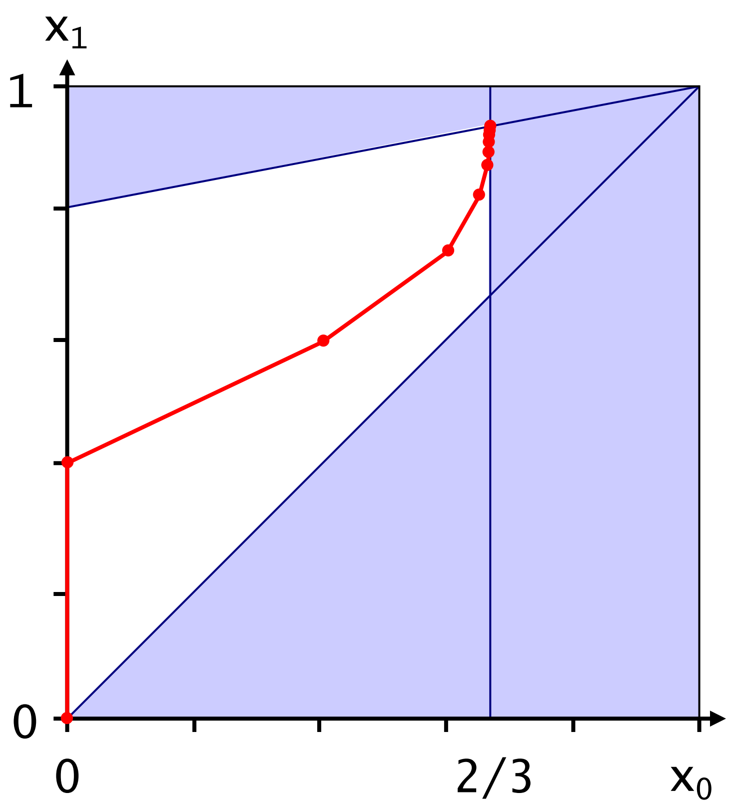


Solution: $(x_0, x_1) = (2/3, 14/15)$

$P^{min}(T)$

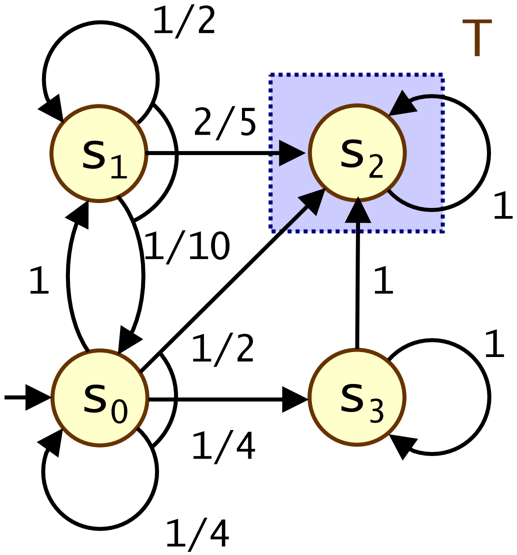
$= [\ 2/3,\ 14/15,\ 1,\ 0\ ]$

Example – Value iteration + LP



	[$x_0^{(n)}$,	$x_1^{(n)}$,	$x_2^{(n)}$,	$x_3^{(n)}$]
n=0:	[0.000000,	0.000000,	1,	0]
n=1:	[0.000000,	0.400000,	1,	0]
n=2:	[0.400000,	0.600000,	1,	0]
n=3:	[0.600000,	0.740000,	1,	0]
n=4:	[0.650000,	0.830000,	1,	0]
n=5:	[0.662500,	0.880000,	1,	0]
n=6:	[0.665625,	0.906250,	1,	0]
n=7:	[0.666406,	0.919688,	1,	0]
n=8:	[0.666602,	0.926484,	1,	0]
…				
n=20:	[0.666667,	0.933332,	1,	0]
n=21:	[0.666667,	0.933332,	1,	0]
≈	[2/3,	14/15,	1,	0]

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

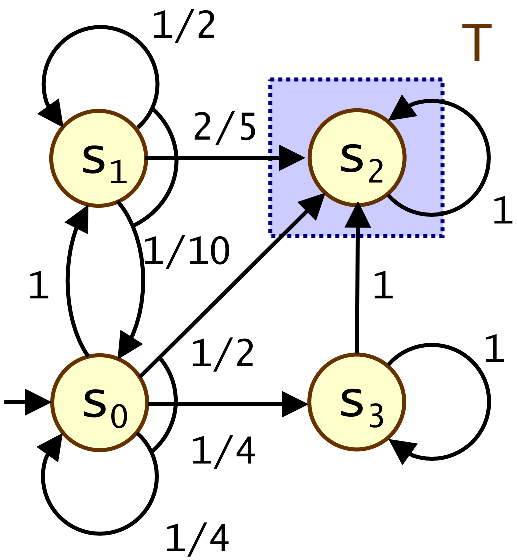
T: $x_2=1$, $S^{max=0}$: $\emptyset$

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

T: $x_2 = 1$, $S^{max=0}$: $\emptyset$

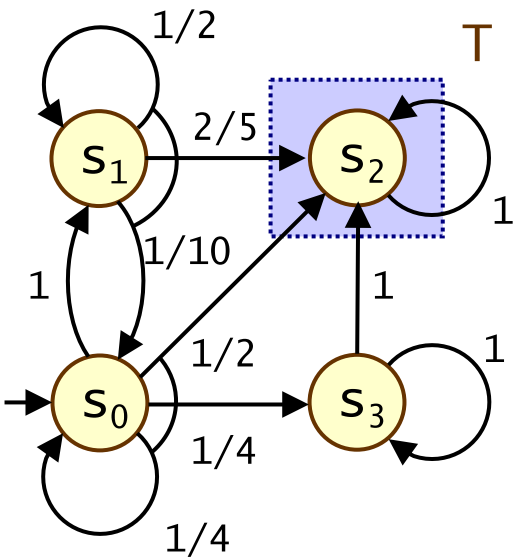
For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0 + x_1 + x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

rearranging

Example – Linear programming (max)



Let $x_i = P^{max}(s_i,T)$

T: $x_2=1$, $S^{max=0}$: $\emptyset$

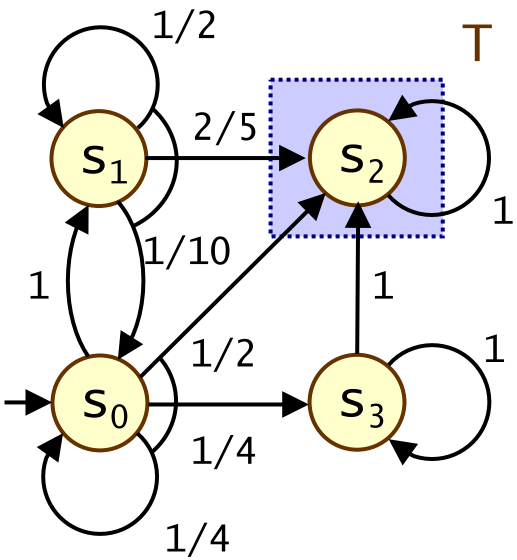
For $S^? = \{s_0,s_1,s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

rearranging

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

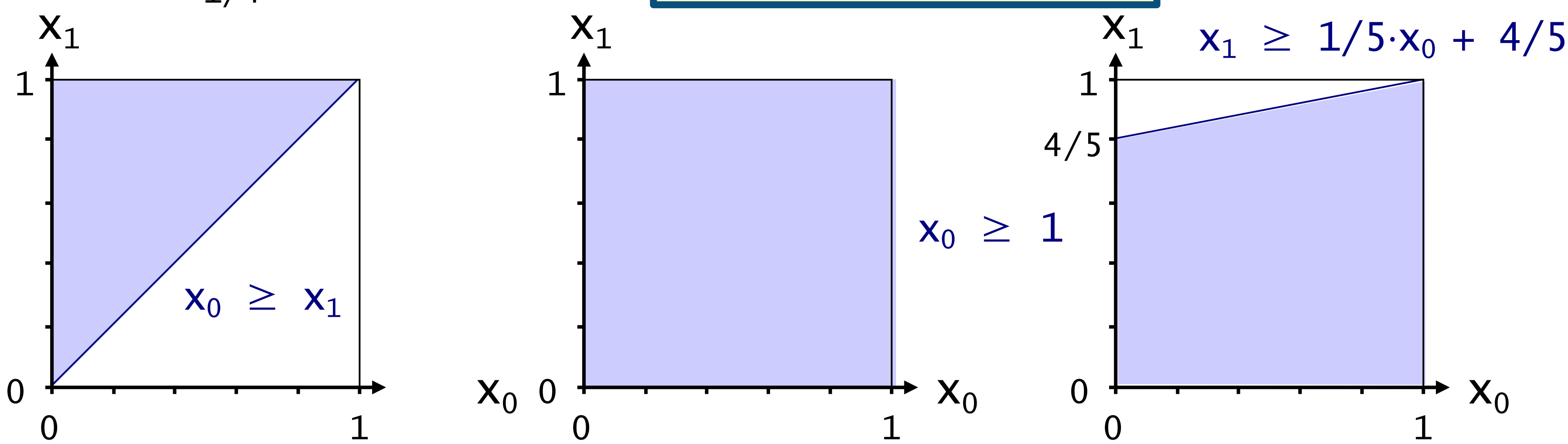
T: $x_2 = 1$, $S^{max=0}$: $\emptyset$

For $S^? = \{s_0, s_1, s_3\}$:

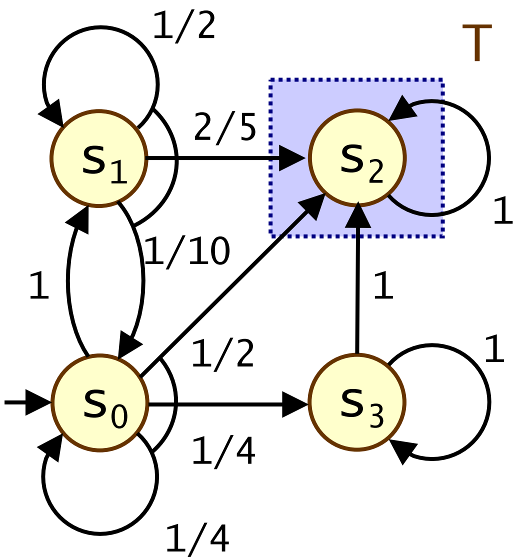
minimise $x_0 + x_1 + x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$

- $x_3 \geq 1$
- $x_3 \geq x_3$



Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

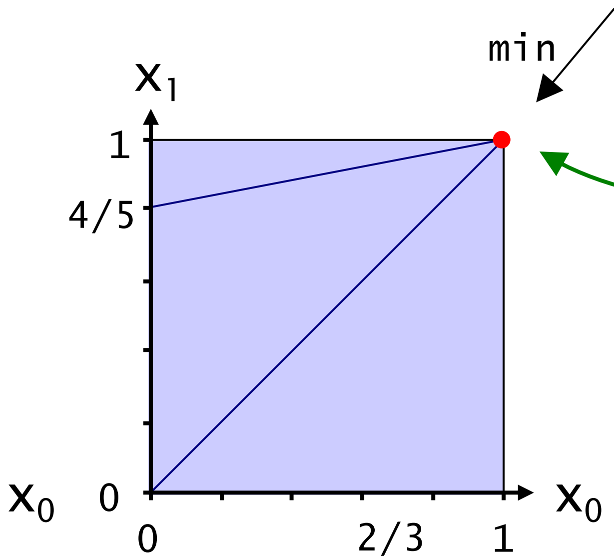
T: $x_2=1$, $S^{max=0}$: $\emptyset$

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$

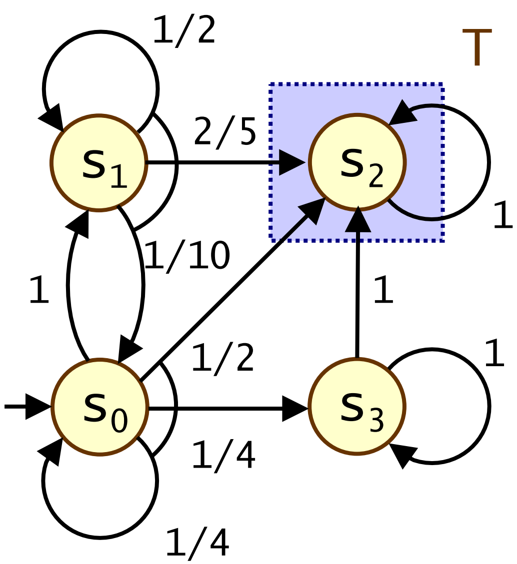
- $x_3 \geq 1$
- $x_3 \geq x_3$



(only feasible) solution:

$(x_0, x_1) = (1, 1)$

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

T: $x_2=1$, $S^{max=0}$: ∅

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

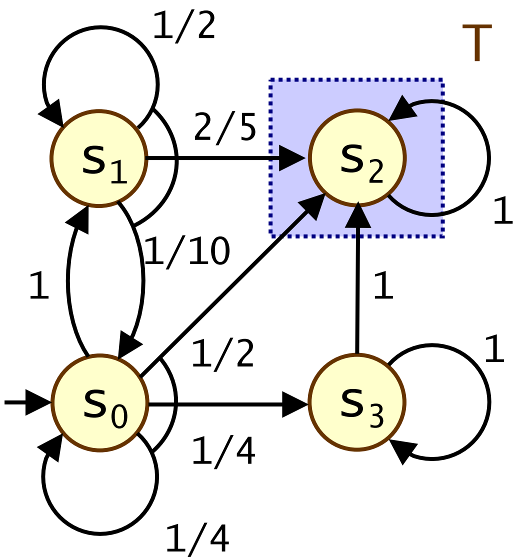
Solution:

- $(x_0, x_1, x_2, x_3) = (1, 1, ?, ?)$

(only feasible) solution:

$(x_0, x_1) = (1, 1)$

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

T: $x_2=1$, $S^{max=0}$: ∅

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

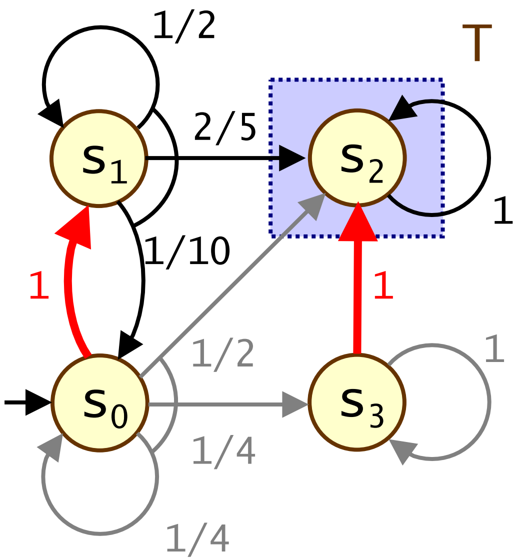
Solution:

- $(x_0, x_1, x_2, x_3) = (1, 1, 1, 1)$

(only feasible) solution:

$(x_0, x_1) = (1, 1)$

Example – Linear programming (max)



Let $x_i = P^{max}(s_i, T)$

T: $x_2=1$, $S^{max=0}$: $\emptyset$

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

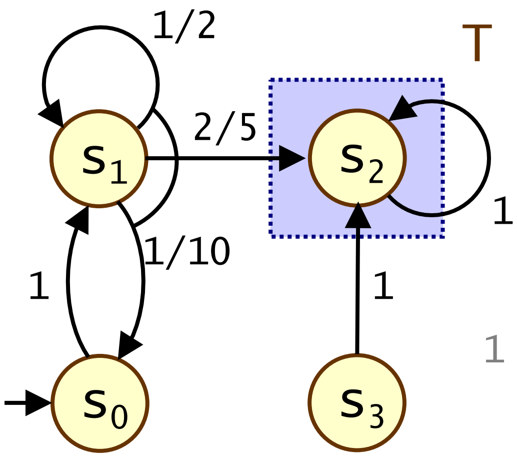
Solution:

- $(x_0, x_1, x_2, x_3) = (1, 1, 1, 1)$

Maximum memoryless adversary σ_{min}

(only feasible) solution:
$(x_0, x_1) = (1, 1)$

Example – Linear programming (max)



DTMC $D(s_0,\sigma_{max})$

Let $x_i = P^{max}(s_i, T)$

T: $x_2=1$, $S^{max=0}$: $\emptyset$

For $S^? = \{s_0, s_1, s_3\}$:

minimise $x_0+x_1+x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 1$
- $x_1 \geq 1/5 \cdot x_0 + 4/5$
- $x_3 \geq 1$
- $x_3 \geq x_3$

Solution:

- $(x_0, x_1, x_2, x_3) = (1, 1, 1, 1)$

(only feasible) solution:

$(x_0, x_1) = (1, 1)$

Method 3 – Policy iteration

Value iteration:

- iterates over (vectors of) probabilities

Policy iteration:

- iterates over adversaries ("policies")

1. start with an arbitrary (memoryless) adversary σ
2. compute the reachability probabilities $P^{\sigma}(s,T)$ for σ
3. improve the adversary in each state
4. repeat steps 2 and 3 until no change in adversary

Termination:

- finite number of memoryless adversaries
- improvement (in min/max probabilities) each time

More general probabilistic properties

For example, once can compute the minimum and maximum probability an LTL formula ψ is true

1. convert problem to one needing maximum probabilities
   - e.g. to find a minimum probability $P_{min=?}[\psi] = 1 - P_{max=?}[\neg\psi]$
2. Generate a deterministic Rabin automaton (DRA) for ψ (or $\neg\psi$)
3. Construct product MDP $M \otimes A$
4. Identify accepting end components (ECs) of $M \otimes A$
   - an EC is a set of states such that there is an strategy under which one remains in the set, and visits all states infinitely often with probability 1
5. Compute maximum probability of reaching accepting ECs
   - from all states of the $M \otimes A$

One last thing – Complexity and Rewards

When using linear programming

- main task solution of linear optimization problem of size |S|
  - can be solved with ellipsoid method (polynomial in |S|)
- and qualitative algorithms (max |S| steps)

Reward Structures for MDPs

- reward accumulated in a state
- reward accumulated when performing a specific action in a state

Can then compute the minimum and maximum expected accumulated rewards before reaching a target

- solution methods as for probabilistic reachability