Lecture 11
Reachability in MDPs

Alessandro Abate

Department of Computer Science
University of Oxford

Recall – MDPs

- Markov decision process: $M = (S, s_{init}, \mathbf{Steps}, L)$
- Adversary $\sigma \in Adv$ resolves nondeterminism
- σ induces set of paths $Path^\sigma(s)$ and DTMC D^σ
- D^σ yields probability space Pr^σ_s over $Path^\sigma(s)$
- $Prob^\sigma(s, \psi) = Pr^\sigma_s \{ \omega \in Path^\sigma(s) \mid \omega \vDash \psi \}$
- MDP yields minimum/maximum probabilities:

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

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

Probabilistic reachability

- Minimum and maximum probability of reaching target set
  - target set = all states labelled with atomic proposition a

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

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

- Vectors: $\underline{p}_{min}(F\ a)$ and $\underline{p}_{max}(F\ a)$
  - minimum/maximum probabilities for all states of the MDP

Overview

- Qualitative probabilistic reachability
  - case where $p_{min}>0$ or $p_{max}>0$
- Optimality equation
- Memoryless adversaries suffice
  - finitely many adversaries to consider
- Computing reachability probabilities
  - value iteration (fixed point computation)
  - linear programming problem
  - policy iteration

Qualitative probabilistic reachability

- Consider the problem of determining states for which $p_{min}(s, F\ a)$ or $p_{max}(s, F\ a)$ is zero (or non-zero)
  - max case: $S^{max=0} = \{ s \in S \mid p_{max}(s, F\ a) = 0 \}$
  - this is just (non-probabilistic) reachability analysis

```
R := Sat(a)
done := false
while (done = false)
    R' = R ∪ { s ∈ S | ∃(a,μ)∈Steps(s) . ∃s'∈R . μ(s')>0}
    if (R'=R) then done := true
    R := R'
endwhile
return S\R
```

Qualitative probabilistic reachability

- Min case: $S^{min=0} = \{ s \in S \mid p_{min}(s, F\ a) = 0 \}$

```
R := Sat(a)
done := false
while (done = false)
    R' = R ∪ { s ∈ S | ∀(a,μ)∈Steps(s) . ∃s'∈R . μ(s')>0}
    if (R'=R) then done := true
    R := R'
endwhile
return S\R
```

note: quantification over all choices

Optimality (min)

- The values $p_{min}(s, F\ a)$ are the unique solution of the following equations:

$$x_s = \begin{cases} 1 & \text{if } s \in Sat(a) \\ 0 & \text{if } s \in S^{min=0} \\ \min\left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'} \mid (a,\mu) \in \textbf{Steps}(s) \right\} & \text{otherwise} \end{cases}$$

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

$S^{min=0} = \{ s \mid p_{min}(s, F\ a)=0 \}$

- This is an instance of the Bellman equation
  - (basis of dynamic programming techniques)

Optimality (max)

- Likewise, the values $p_{max}(s, F\ a)$ are the unique solution of the following equations:

$$x_s = \begin{cases} 1 & \text{if } s \in Sat(a) \\ 0 & \text{if } s \in S^{max=0} \\ \max\left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'} \mid (a, \mu) \in \textbf{Steps}(s) \right\} & \text{otherwise} \end{cases}$$

$S^{max=0} = \{ s \mid p_{max}(s, F\ a) = 0 \}$

Memoryless adversaries

- Memoryless adversaries suffice for probabilistic reachability
  - i.e. there exist memoryless adversaries σ_{min} & σ_{max} such that:
  - $Prob^{\sigma_{min}}(s, F\ a) = p_{min}(s, F\ a)$ for all states $s \in S$
  - $Prob^{\sigma_{max}}(s, F\ a) = p_{max}(s, F\ a)$ for all states $s \in S$

- Construct adversaries from optimal solution:

$$\sigma_{min}(s) = \text{argmin}\left\{\sum_{s' \in S} \mu(s') \cdot p_{min}(s', F\ a) \mid (a,\mu) \in \mathbf{Steps}(s)\right\}$$

$$\sigma_{max}(s) = \text{argmax}\left\{\sum_{s' \in S} \mu(s') \cdot p_{max}(s', F\ a) \mid (a,\mu) \in \mathbf{Steps}(s)\right\}$$

  - locally optimising steps
  - moreover, for max, must ensure that a is eventually reached (e.g. avoid self-loops - cf. later)

Computing reachability probabilities

- Several approaches…

- 1. Value iteration
  - approximate with iterative solution method
  - corresponds to fixed-point computation

Preferable in practice, e.g. in PRISM

- 2. Reduction to a linear programming (LP) problem
  - solve with linear optimisation techniques
  - exact solution using well-known methods

better complexity; good for small examples

- 3. Policy iteration
  - iteration over adversaries

Method 1 – Value iteration (min)

- For minimum probabilities $p_{min}(s, F\ a)$ it can be shown that:
  - $p_{min}(s, F\ a) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in Sat(a) \\ 0 & \text{if } s \in S^{min=0} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \min\left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'}^{(n-1)} \mid (a,\mu) \in \mathbf{Steps}(s) \right\} & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

  - where: $S^{?} = S \setminus (Sat(a) \cup S^{min=0})$

- Approximate iterative solution
  - iterations terminated when solution converges sufficiently

Method 1 – Value iteration (max)

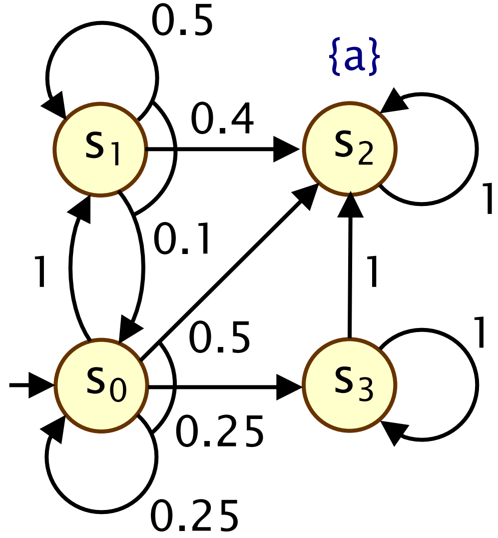
- Value iteration applies to maximum probabilities in the same way…
  - $p_{max}(s, F\ a) = \lim_{n\to\infty} x_s^{(n)}$ where:

$$x_s^{(n)} = \begin{cases} 1 & \text{if } s \in Sat(a) \\ 0 & \text{if } s \in S^{max=0} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \max\left\{ \sum_{s' \in S} \mu(s') \cdot x_{s'}^{(n-1)} \mid (a,\mu) \in \mathbf{Steps}(s) \right\} & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

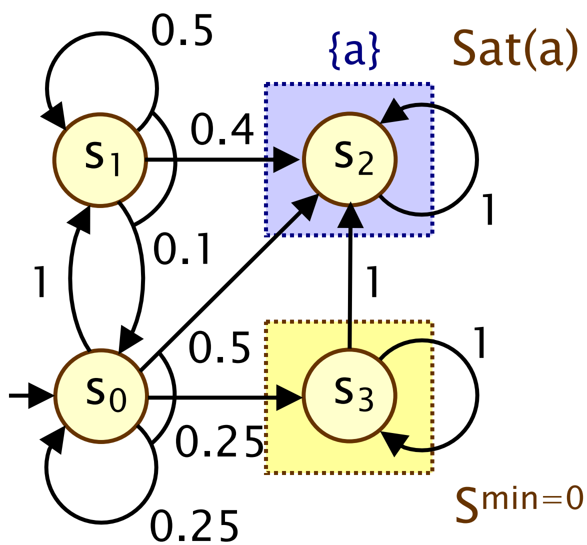
  - where: $S^{?} = S \setminus (\ Sat(a) \cup S^{max=0}\ )$

Example

- Minimum/maximum probability of reaching an a-state



Example – Value iteration (min)



Compute: $p_{min}(s_i, F\ a)$

Sat(a) = $\{s_2\}$, $S^{min=0} = \{s_3\}$, $S^? = \{s_0, s_1\}$

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

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

n=1: $[\ \min(1 \cdot 0,\ 0.25 \cdot 0 + 0.25 \cdot 0 + 0.5 \cdot 1),$

$0.1 \cdot 0 + 0.5 \cdot 0 + 0.4 \cdot 1,\ 1,\ 0\ ]$

$= [\ 0, 0.4, 1, 0\ ]$

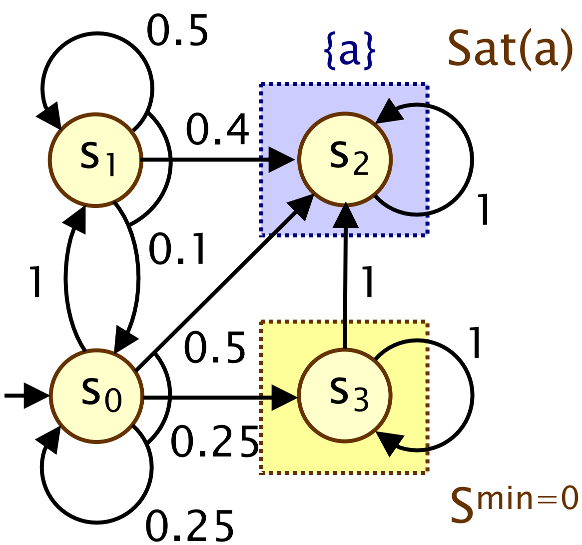
n=2: $[\ \min(1 \cdot 0.4, 0.25 \cdot 0 + 0.25 \cdot 0 + 0.5 \cdot 1),$

$0.1 \cdot 0 + 0.5 \cdot 0.4 + 0.4 \cdot 1,\ 1,\ 0\ ]$

$= [\ 0.4, 0.6, 1, 0\ ]$

n=3: …

Example – Value iteration (min)



$\underline{p}_{min}$(F a)

=

[2/3, 14/15, 1, 0]

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

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]

…

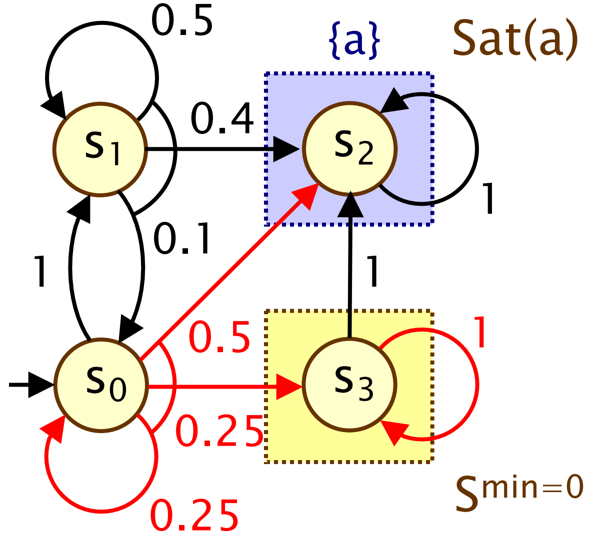
n=20: [0.666667, 0.933332, 1, 0]

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

≈ [2/3, 14/15, 1, 0]

Generating an optimal adversary

- Min adversary 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 : $\min(1 \cdot 14/15,\ 0.5 \cdot 1 + 0.25 \cdot 0 + 0.25 \cdot 2/3)$

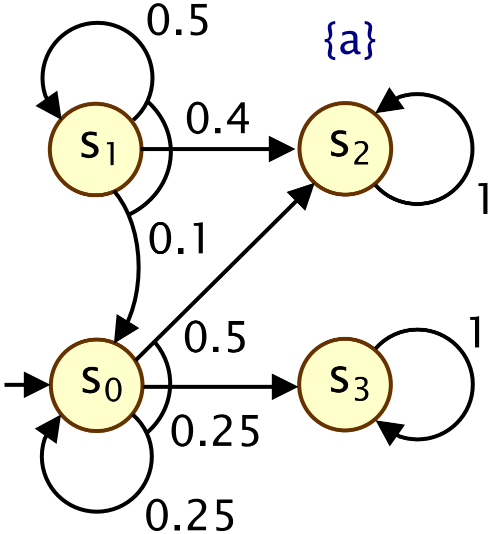
$= \min(14/15,\ 2/3)$

s_3 : $\min(1 \cdot 1,\ 1.0)$

$= \min(1,\ 0)$

Generating an optimal adversary

- DTMC $D^{\sigma_{min}}$



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

…

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

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

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

s_0 : min(1·14/15, 0.5·1+0.25·0+0.25·2/3)

=min(14/15, 2/3)

s_3 : min(1·1, 1.0)

=min(1, 0)

Value iteration as a fixed point

- Can view value iteration as a fixed point computation over vectors of probabilities $\underline{y} \in [0,1]^S$, e.g. for minimum:

$$F(\underline{y})(s) = \begin{cases} 1 & \text{if } s \in Sat(a) \\ 0 & \text{if } s \in S^{min=0} \\ \min\left\{ \sum_{s' \in S} \mu(s') \cdot \underline{y}(s') \mid (a,\mu) \in \mathbf{Steps}(s) \right\} & \text{otherwise} \end{cases}$$

- Let:
  - $\underline{x}^{(0)} = \underline{0}$ (i.e. $\underline{x}^{(0)}(s) = 0$ for all s)
  - $\underline{x}^{(n+1)} = F(\underline{x}^{(n)})$
- Then:
  - $\underline{x}^{(0)} \leq \underline{x}^{(1)} \leq \underline{x}^{(2)} \leq \underline{x}^{(3)} \leq \ldots$
  - $\underline{p}_{min}(F\ a) = \lim_{n\to\infty} \underline{x}^{(n)}$

Linear programming

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

- General form:
  - n variables: $x_1, x_2, \ldots, x_n$
  - maximise (or minimise):
    - $c_1x_1+c_2x_2+\ldots+c_nx_n$
  - subject to constraints
    - $a_{11}x_1+a_{12}x_2+\ldots a_{1n}x_n \leq b_1$
    - $a_{21}x_1+a_{22}x_2+\ldots a_{2n}x_n \leq b_2$
    - …
    - $a_{m1}x_1+a_{m2}x_2+\ldots a_{mn}x_n \leq b_m$

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

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

Method 2 – Linear programming problem

- min probabilities $p_{min}(s, F\ a)$ can be computed as follows:
  - $p_{min}(s, F\ a) = 1$ if $s \in Sat(a)$
  - $p_{min}(s, F\ a) = 0$ if $s \in S^{min=0}$
  - values for remaining states in the set $S^? = S \setminus (Sat(a) \cup S^{min=0})$ can be obtained as the unique solution of the following linear programming problem:

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

Linear programming problem (max)

- max probabilities $p_{max}(s, F\ a)$ can be computed as follows:
  - $p_{max}(s, F\ a) = 1$ if $s \in Sat(a)$
  - $p_{max}(s, F\ a) = 0$ if $s \in S^{max=0}$
  - values for remaining states in the set $S^? = S \setminus (Sat(a) \cup S^{max=0})$ can be obtained as the unique solution of the following linear programming problem:

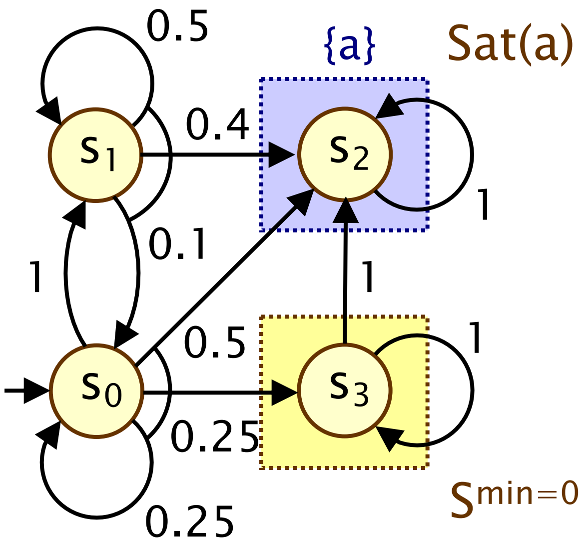
$$\text{minimize } \sum_{s \in S^?} x_s \text{ subject to the constraints :}$$

$$x_s \geq \sum_{s' \in S^?} \mu(s') \cdot x_{s'} + \sum_{s' \in Sat(a)} \mu(s')$$

$$\text{for all } s \in S^? \text{ and for all } (a, \mu) \in \textbf{Steps}(s)$$

differences from min case

Example – Linear programming (min)



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

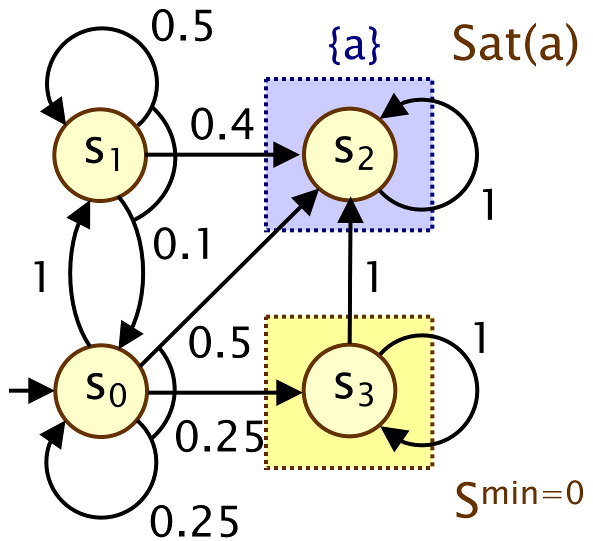
Sat(a): $x_2=1$, $S^{min=0}$: $x_3=0$

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

Maximise x_0+x_1 subject to constraints:

- $x_0 \leq x_1$
- $x_0 \leq 0.25 \cdot x_0 + 0.25 \cdot 0 + 0.5$
- $x_1 \leq 0.1 \cdot x_0 + 0.5 \cdot x_1 + 0.4$

Example – Linear programming (min)



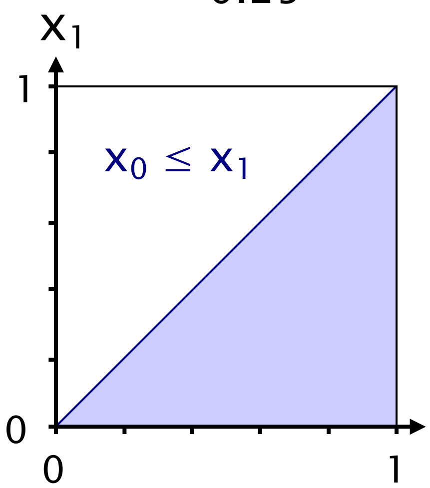
Let $x_i = p_{min}(s_i, F\ a)$

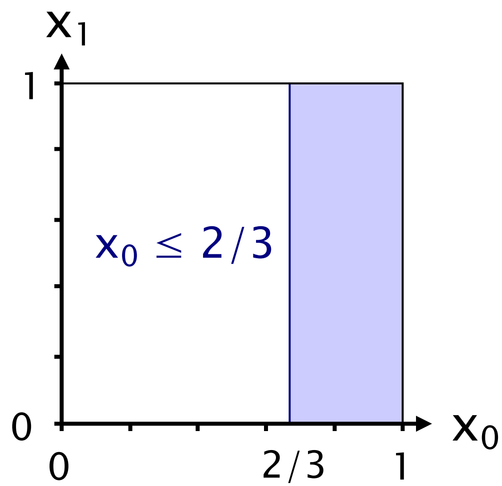
Sat(a): $x_2=1$, $S^{min=0}$: $x_3=0$

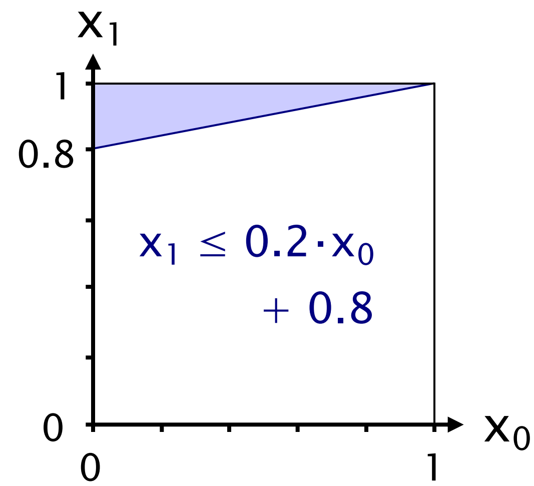
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

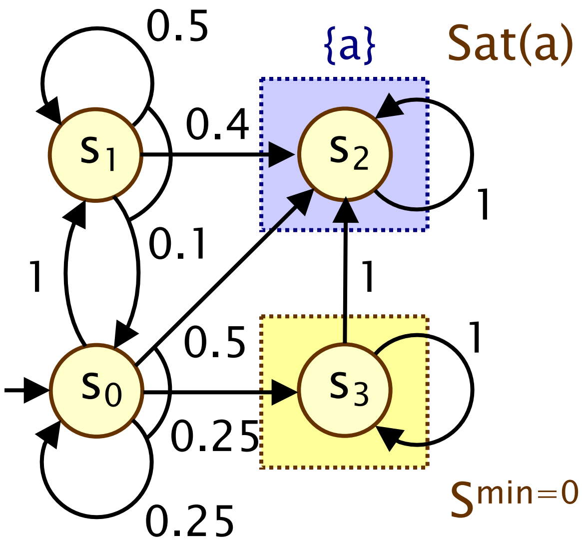
- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 0.2 \cdot x_0 + 0.8$







Example – Linear programming (min)



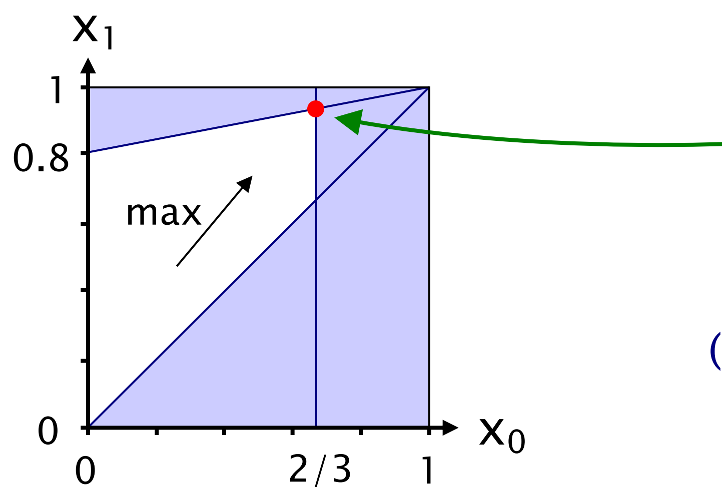
Let $x_i = p_{min}(s_i, F\ a)$

Sat(a): $x_2=1$, $S^{min=0}$: $x_3=0$

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

Maximise x_0+x_1 subject to constraints:

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



$\underline{p}_{min}(F\ a)$

$=$

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

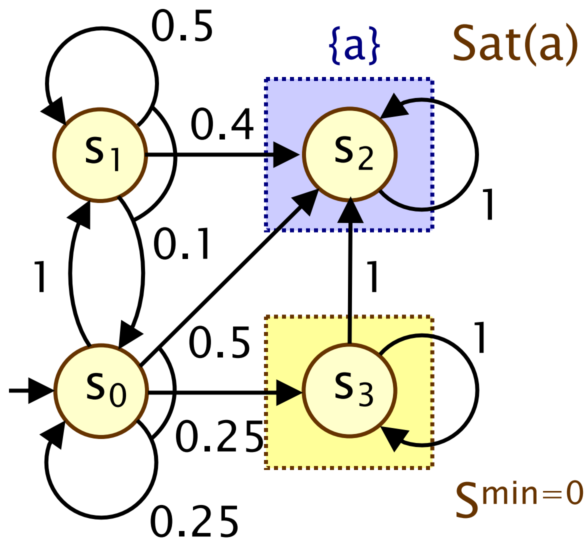
Solution:

(x_0, x_1)

$=$

$(2/3, 14/15)$

Example – Linear programming (min)



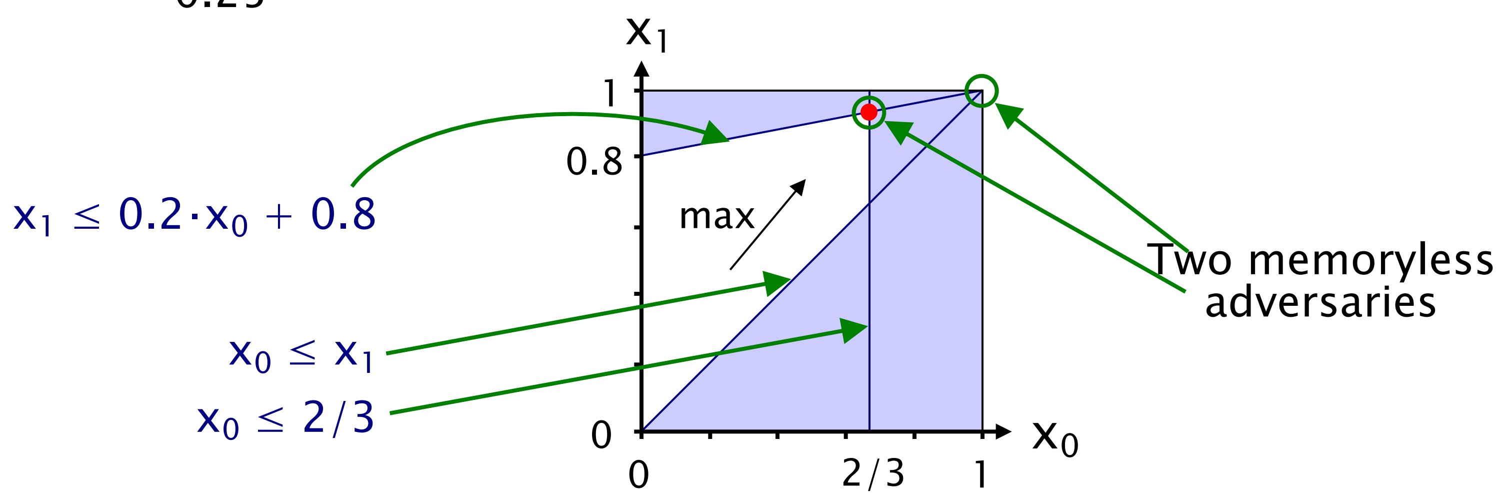
Let $x_i = p_{min}(s_i, F\ a)$

Sat(a): $x_2=1$, $S^{min=0}$: $x_3=0$

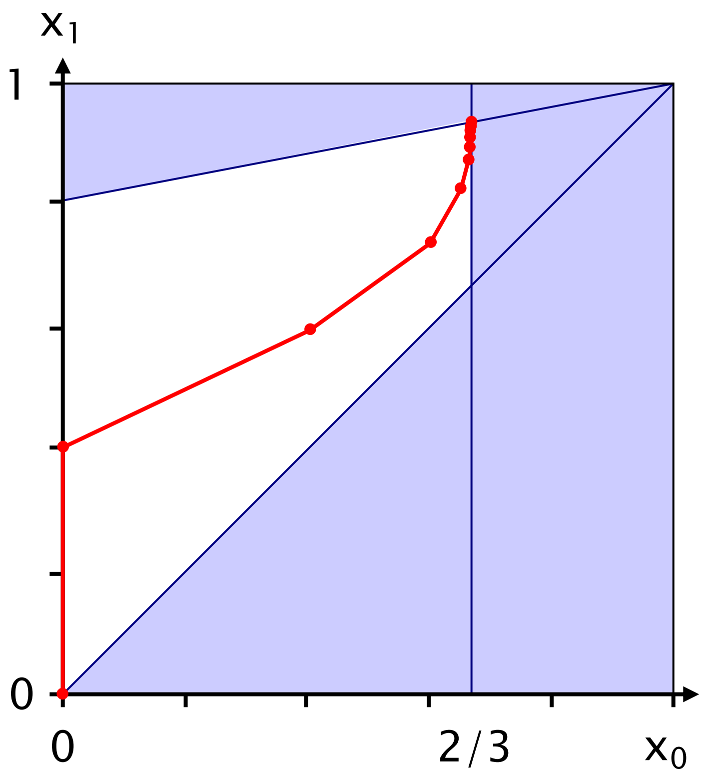
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

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



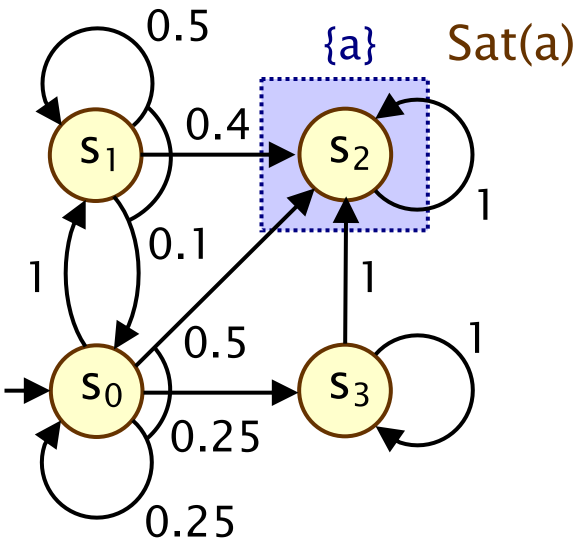
Example – Value Iteration and LP



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

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

Example – Linear programming (max)



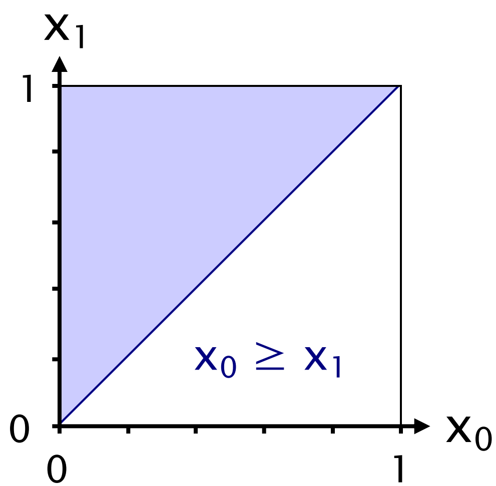
Let $x_i = p_{max}(s_i, F\ a)$

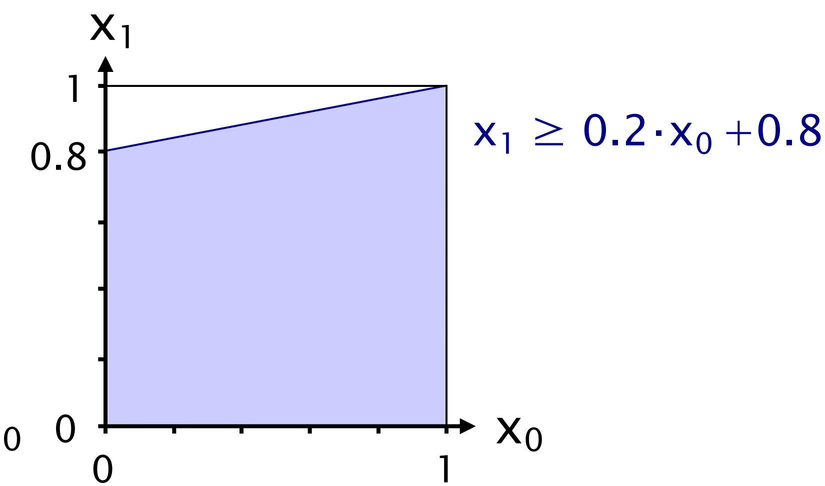
Sat(a): $x_2 = 1$, $S^{max=0} = \varnothing$

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

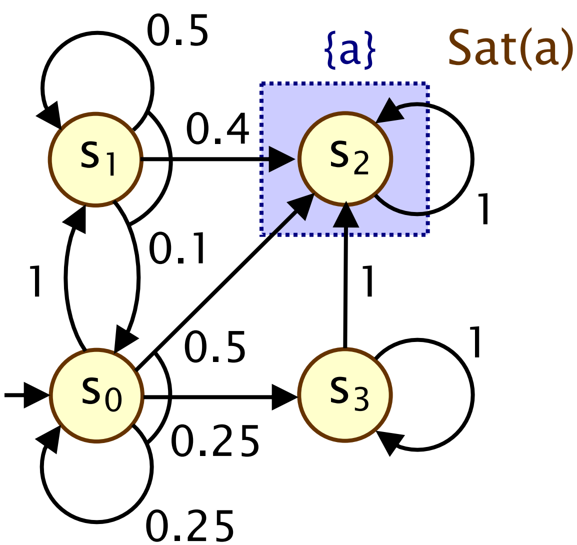
Minimise $x_0 + x_1 + x_3$ subject to constraints:

- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 0.2 \cdot x_0 + 0.8$
- $x_3 \geq 1$
- $x_3 \geq x_3$





Example – Linear programming (max)



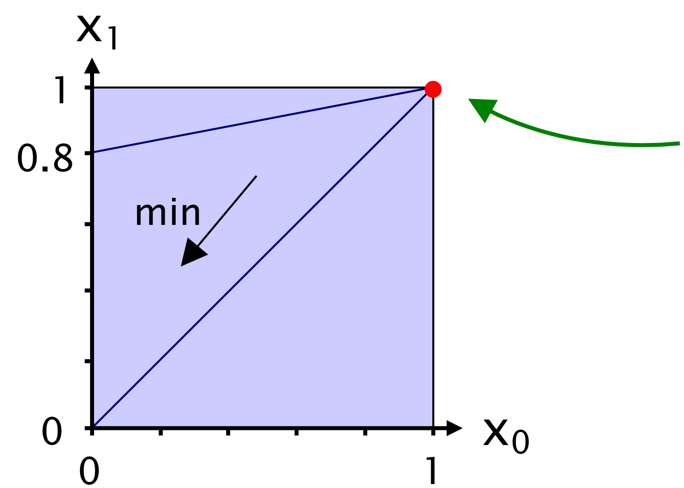
Let $x_i = p_{max}(s_i, F\ a)$

Sat(a): $x_2=1$, $S^{max=0} = \varnothing$

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

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

- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 0.2 \cdot x_0 + 0.8$
- $x_3 \geq 1$
- $x_3 \geq x_3$

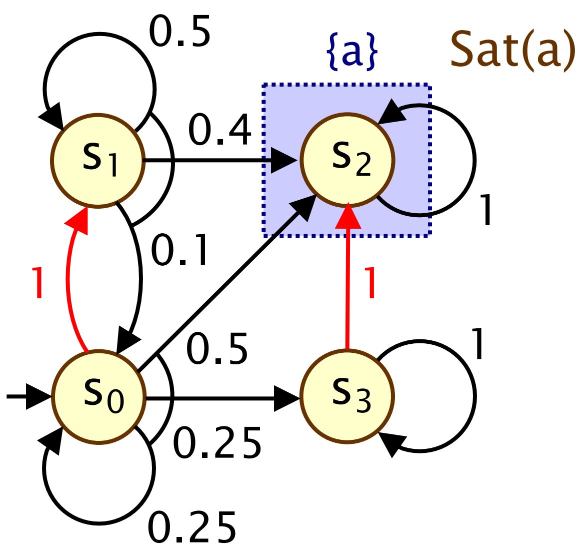


(only feasible) solution:

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

Generating an adversary

- Max adversary σ_{max}



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

Sat(a): $x_2=1$, $S^{max=0} = \varnothing$

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

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

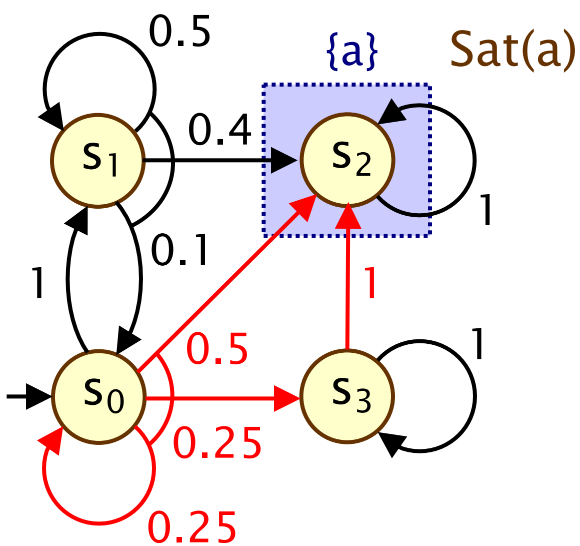
- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 0.2 \cdot x_0 + 0.8$
- $x_3 \geq 1$
- $x_3 \geq x_3$

Solution:

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

Generating an adversary

- Max adversary σ_{max}



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

Sat(a): $x_2=1$, $S^{max=0} = \varnothing$

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

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

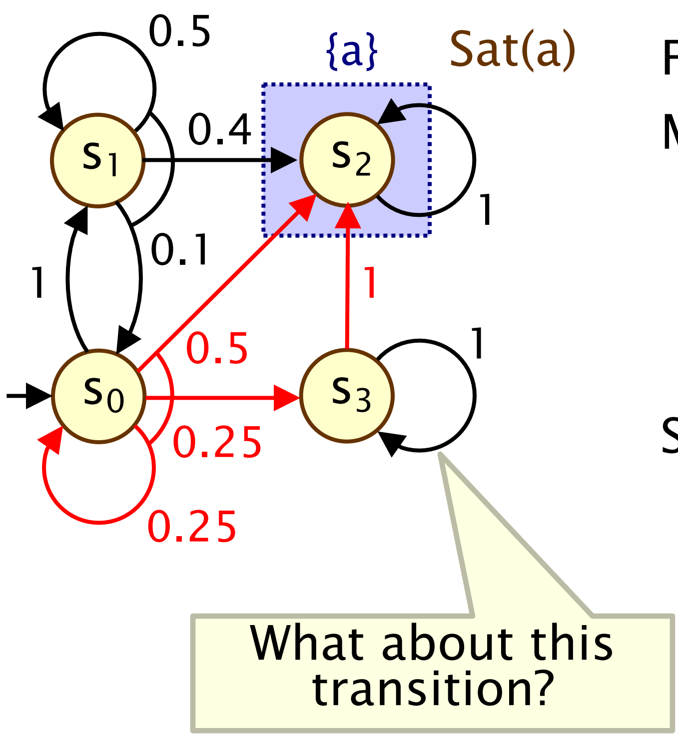
- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 0.2 \cdot x_0 + 0.8$
- $x_3 \geq 1$
- $x_3 \geq x_3$

Solution:

- $(x_0, x_1, x_3) = (1, 1, 1)$
  - $x_0 = 2/3 + 1/3 \cdot x_3$
  - $x_3 = 1$

Generating an adversary

- Max adversary σ_{max}



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

Sat(a): $x_2=1$, $S^{max=0} = \varnothing$

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

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

- $x_0 \geq x_1$
- $x_0 \geq 2/3 + 1/3 \cdot x_3$
- $x_1 \geq 0.2 \cdot x_0 + 0.8$
- $x_3 \geq 1$
- $x_3 \geq x_3$

Solution:

- $(x_0, x_1, x_3) = (1, 1, 1)$
  - $x_0 = 2/3 + 1/3 \cdot x_3$
  - $x_3 = 1$

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 $\underline{Prob}^{\sigma}(F\ a)$ for σ
- 3. Improve the adversary in each state
- 4. Repeat 2./3. until no change in adversary

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

Method 3 – Policy iteration

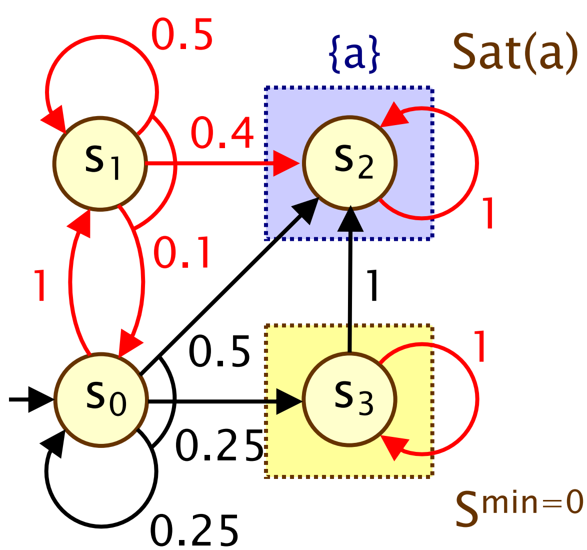
- 1. Start with an arbitrary (memoryless) adversary σ
  - pick an element of **Steps**(s) for each state $s \in S$
- 2. Compute the reachability probabilities <u>Prob</u>$^{\sigma}$(F a) for σ
  - probabilistic reachability on a DTMC
  - i.e. solve linear equation system
- 3. Improve the adversary in each state

$$\sigma'(s) = \mathrm{argmin}\left\{ \sum_{s' \in S} \mu(s') \cdot \mathrm{Prob}^{\sigma}(s', \mathrm{F}\ a) \mid (a, \mu) \in \mathbf{Steps}(s) \right\}$$

$$\sigma'(s) = \mathrm{argmax}\left\{ \sum_{s' \in S} \mu(s') \cdot \mathrm{Prob}^{\sigma}(s', \mathrm{F}\ a) \mid (a, \mu) \in \mathbf{Steps}(s) \right\}$$

- 4. Repeat 2./3. until no change in adversary

Example – Policy iteration (min)



Arbitrary adversary σ:

Compute: Prob$^{\sigma}$(F a)

Let x_i = Prob$^{\sigma}$(s_i, F a)

$x_2=1$, $x_3=0$ and:

- $x_0 = x_1$
- $x_1 = 0.1 \cdot x_0 + 0.5 \cdot x_1 + 0.4$

Solution:

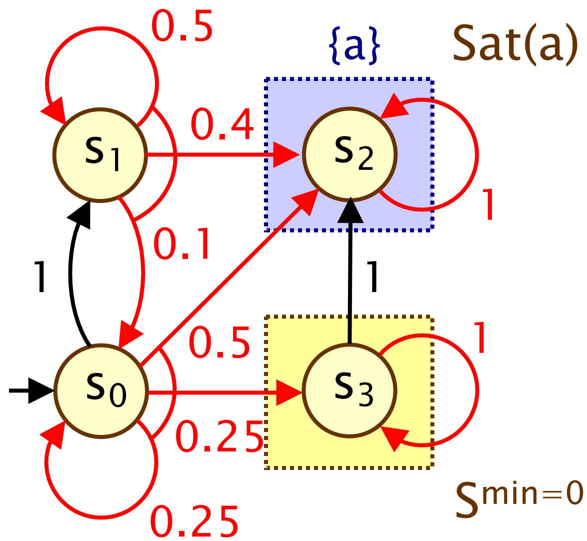
Prob$^{\sigma}$(F a) = [1, 1, 1, 0]

Refine σ in state s_0:

$\min\{1 \cdot 1, 0.5 \cdot 1 + 0.25 \cdot 0 + 0.25 \cdot 1\}$

$= \min\{1, 0.75\} = 0.75$

Example – Policy iteration (min)



Refined adversary σ':

Compute: Prob$^{\sigma'}$(F a)

Let x_i = Prob$^{\sigma'}$(s_i, F a)

$x_2=1$, $x_3=0$ and:

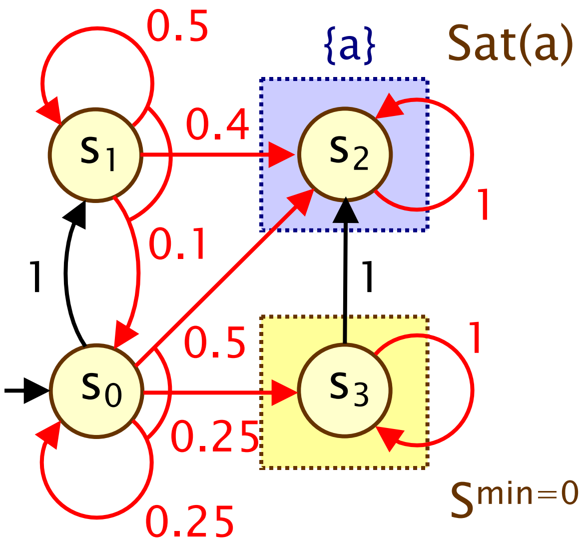
- $x_0 = 0.25 \cdot x_0 + 0.5$
- $x_1 = 0.1 \cdot x_0 + 0.5 \cdot x_1 + 0.4$

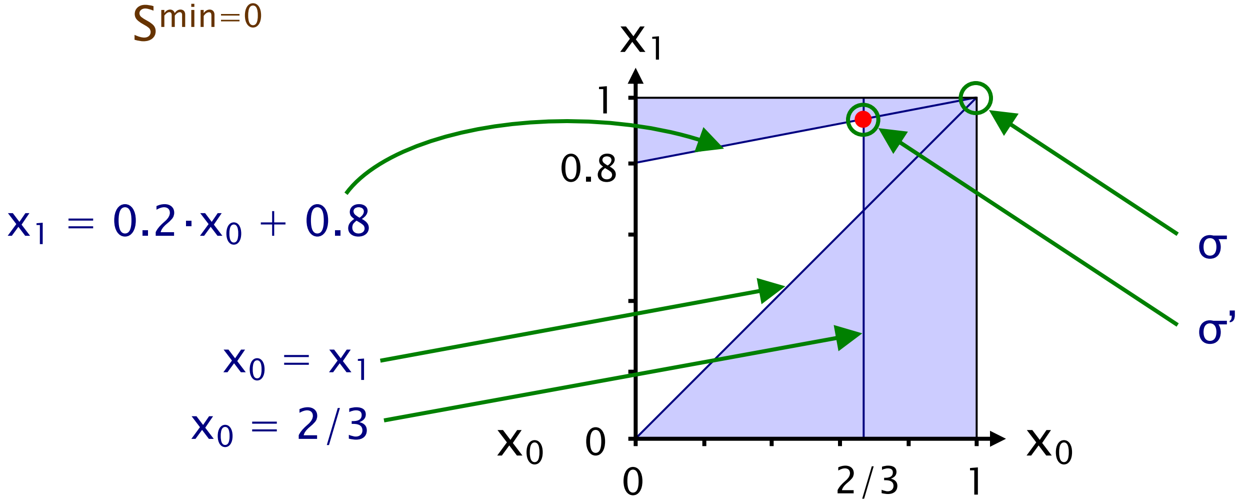
Solution:

Prob$^{\sigma'}$(F a) = [2/3, 14/15, 1, 0]

This is optimal

Example – Policy iteration (min)





Summing up…

- Probabilistic reachability in MDPs
- Qualitative case: min/max probability > 0
  - simple graph-based computation
  - need to do this first, before other computation methods
- Memoryless adversaries suffice
  - reduction to finite number of adversaries
- Computing reachability probabilities…
  (and generation of optimal adversary)
- 1. Value iteration
  - approximate; iterative; fixed point computation
- 2. Reduce to linear programming problem
  - good for small examples; doesn't scale well
- 3. Policy iteration