Lecture 12
PCTL Model Checking for MDPs

Alessandro Abate

Department of Computer Science
University of Oxford

Overview

- PCTL for MDPs
  - syntax, semantics, examples
- PCTL model checking
  - next, bounded until, until
  - precomputation algorithms
  - value iteration, linear optimisation
  - examples
- Costs and rewards

PCTL

- Temporal logic for describing properties of MDPs
  - identical syntax to the logic PCTL for DTMCs

ψ is true with probability ~p

  - ϕ ::= true | a | $\phi \wedge \phi$ | $\neg\phi$ | $P_{\sim p}$ [ψ] (state formulas)
  - ψ ::= X ϕ | ϕ $U^{\leq k}$ ϕ | ϕ U ϕ (path formulas)

"next" "bounded until" "until"

  - where a is an atomic proposition, used to identify states of interest, $p \in [0,1]$ is a probability, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

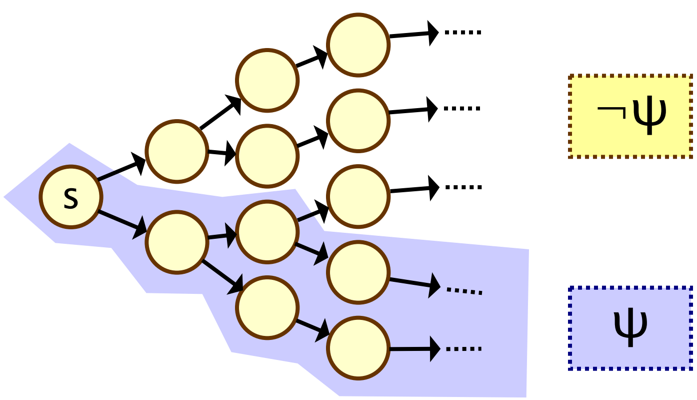
PCTL semantics for MDPs

- 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 ($S, s_{init}, \mathbf{Steps}, L$):
  - $s \models a \quad \Leftrightarrow \quad a \in L(s)$
  - $s \models \phi_1 \wedge \phi_2 \quad \Leftrightarrow \quad s \models \phi_1$ 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 inherit probabilities for a specific adversary σ from induced DTMC
  - $s \models P_{\sim p} [\psi]$ 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 $s \models P_{\sim p} [\ \psi\ ] \ \Leftrightarrow\ p_{min}(s, \psi) \sim p$
  - if $\sim \in \{<, \leq\}$, then $s \models P_{\sim p} [\ \psi\ ] \ \Leftrightarrow\ p_{max}(s, \psi) \sim p$

- 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

Other classes of adversary

- A more general semantics for PCTL over MDPs
  - parameterise by a class of adversaries Adv*
- E.g., take Adv* to be the set of all fair adversaries
  - path (strong) fairness: if a state occurs on a path infinitely often, then each non-deterministic choice occurs infinitely often [BK98]
- Only change is:
  - $s \vDash_{Adv^*} P_{\sim p} [\psi] \;\Leftrightarrow\; Prob^{\sigma}(s, \psi) \sim p$ for all adversaries $\sigma \in Adv^*$
- Original semantics obtained by taking Adv* = Adv

PCTL-derived operators

- Many of the same equivalences as for DTMCs, e.g.:
  - $F\ \phi \equiv true\ U\ \phi$ (eventually)
  - $F^{\leq k}\ \phi \equiv true\ U^{\leq k}\ \phi$
  - $G\ \phi \equiv \neg(F\ \neg\phi) \equiv \neg(true\ U\ \neg\phi)$ (always)
  - $G^{\leq k}\ \phi \equiv \neg(F^{\leq k}\ \neg\phi)$
  - etc.

- But… for example:
  - $P_{\geq p}\ [\ \psi\ ] \not\equiv \neg P_{<p}\ [\ \psi\ ]$ (negation + probability)

- Duality between min/max:
  - for any path formula ψ: $p_{min}(s, \psi) = 1 - p_{max}(s, \neg\psi)$
  - so, for example: $P_{\geq p}\ [\ G\ \phi\ ] \equiv P_{\leq 1-p}\ [\ F\ \neg\phi\ ]$

Qualitative properties

- PCTL can express qualitative properties of MDPs
  - like for DTMCs, can relate these to CTL's AF and EF operators
  - need to be careful with "there exists" and adversaries

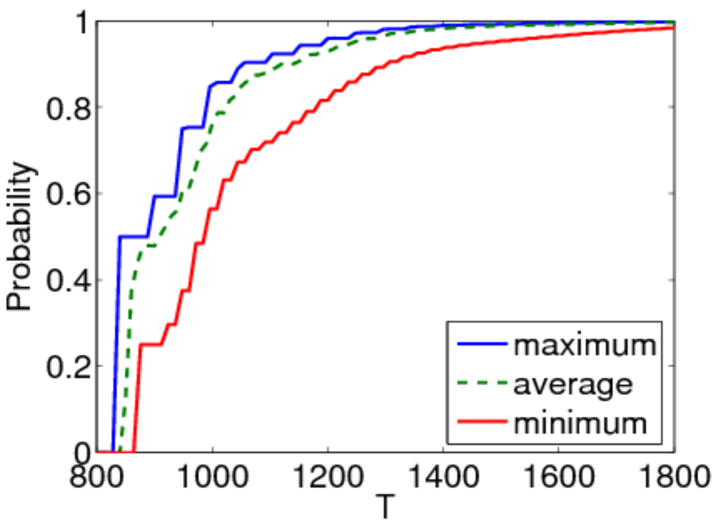
- $P_{\geq 1}$ [F ϕ] is (similar to but) weaker than AF ϕ
  - $P_{\geq 1}$ [F ϕ] $\Leftrightarrow$ $Prob^{\sigma}(s, F\ \phi) = 1$ for all adversaries σ
  - recall that "probability=1" is weaker than "for all"

- We can construct an equivalence for EF ϕ
  - EF ϕ $\not\equiv$ $P_{>0}$[F ϕ]
  - but:
  - EF ϕ $\equiv$ $\neg P_{\leq 0}$[F ϕ]

Quantitative properties

- For PCTL properties with P as the outermost operator
  - PRISM allows a quantitative form
  - for MDPs, there are two types: $P_{min=?}$ [ψ] and $P_{max=?}$ [ψ]
  - i.e. "what is the minimum/maximum probability (over all adversaries) that path formula ψ is true?"
  - model checking is no harder since it computes the values of $p_{min}(s, ψ)$ or $p_{max}(s, ψ)$ anyway
  - useful to spot patterns/trends

- Example CSMA/CD protocol
  - "min/max probability that a message is sent within the deadline"



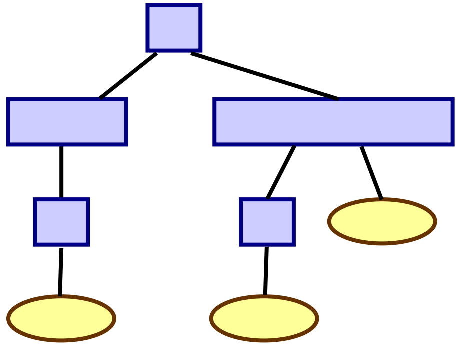
Some real PCTL examples

- Byzantine agreement protocol
  - $P_{min=?}$ [F (agreement $\wedge$ rounds$\leq$2)]
  - "what is the minimum probability that agreement is reached within two rounds?"
- CSMA/CD communication protocol
  - $P_{max=?}$ [F collisions=k]
  - "what is the maximum probability of k collisions?"
- Self-stabilisation protocols
  - $P_{min=?}$ [$F^{\leq k}$ stable]
  - "what is the minimum probability of reaching a stable state within k steps?"

PCTL model checking for MDPs

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

- 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 ∈ S | a ∈ L(s) }
  - Sat(¬ϕ) = S \ Sat(ϕ)
  - $Sat(\phi_1 \wedge \phi_2) = Sat(\phi_1) \cap Sat(\phi_2)$



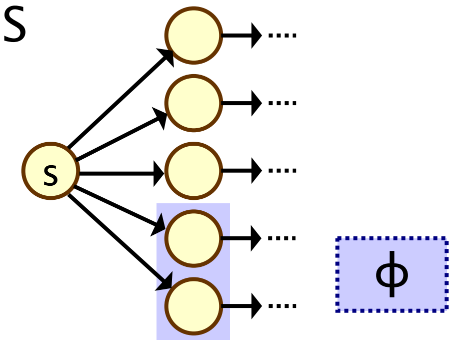
PCTL model checking for MDPs

- 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 $s \in S$
  - depending 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…
  - $Sat(P_{\sim p}[\ X\ \phi\ ]) = \{\ s \in S \mid p_{min}(s, X\ \phi) \sim p\ \}, \sim \in \{\geq,>\}$
  - 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
  - $Prob(s, X\ \phi) = \Sigma_{s' \in Sat(\phi)}\ P(s,s')$



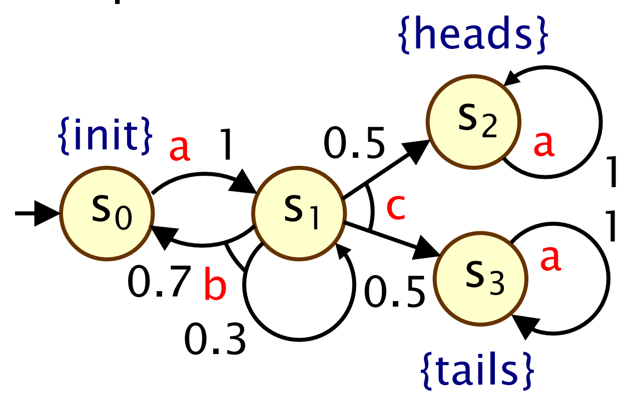
- For MDPs, perform computation for each distribution available in s and then take minimum:
  - $p_{min}(s, X\ \phi) = \min\{\ \Sigma_{s' \in Sat(\phi)}\ \mu(s') \mid (a,\mu) \in Steps(s)\ \}$

- 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. $p_{min}(s_1, X\ heads) = \min(0, 0.5) = 0$
  - can do all at once with matrix-vector multiplication:

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



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

PCTL bounded until for MDPs

- Computation of probabilities for PCTL $U^{\leq k}$ operator
- Consider case of minimum probabilities…
  - $\text{Sat}(P_{\sim p}[\ \phi_1\ U^{\leq k}\ \phi_2\ ]) = \{\ s \in S \mid p_{min}(s, \phi_1\ U^{\leq k}\ \phi_2) \sim p\ \}$, $\sim\ \in \{\geq,>\}$
  - 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
  - $S^{yes} = \text{Sat}(\phi_2)$ and $S^{no} = S \setminus (\text{Sat}(\phi_1) \cup \text{Sat}(\phi_2))$
- Then solve the recursive equations:

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

- Maximum probabilities case is analogous

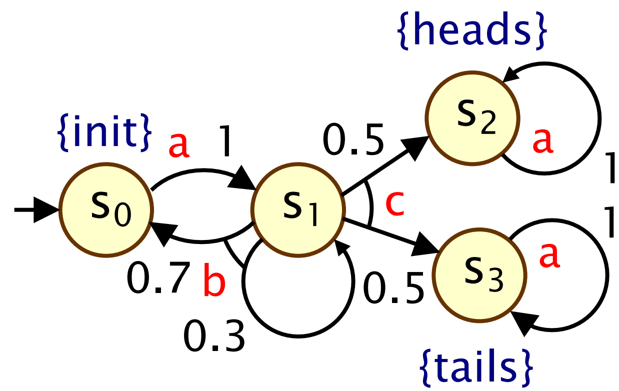
PCTL bounded until for MDPs

- Simultaneous computation of vector $\underline{p}_{\underline{min}}(\phi_1 \ U^{\leq k} \ \phi_2)$
  - 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 minimisation 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:
  - $\underline{p}_{max}$(true $U^{\leq 0}$ init) = [1, 0, 0, 0]
  - $\underline{p}_{max}$(true $U^{\leq 1}$ init) = [1, 0.7, 0, 0]
  - $\underline{p}_{max}$(true $U^{\leq 2}$ init) = [1, 0.91, 0, 0]
  - $\underline{p}_{max}$(true $U^{\leq 3}$ 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

PCTL until for MDPs – Precomputation

- Determine all states for which probability is 0
  - min case: $S^{no} = \{ s \in S \mid p_{min}(s, \phi_1 \text{ U } \phi_2)=0 \}$ – Prob0E
  - max case: $S^{no} = \{ s \in S \mid p_{max}(s, \phi_1 \text{ U } \phi_2)=0 \}$ – Prob0A
- Determine all states for which probability is 1
  - min case: $S^{yes} = \{ s \in S \mid p_{min}(s, \phi_1 \text{ U } \phi_2)=1 \}$ – ~~Prob1A~~ (not covered here)
  - max case: $S^{yes} = \{ s \in S \mid p_{max}(s, \phi_1 \text{ U } \phi_2)=1 \}$ – Prob1E
- Like for DTMCs:
  - identifying 0 states required (for uniqueness of LP problem)
  - identifying 1 states is optional (but useful optimisation)
- Advantages of precomputation
  - reduces size of numerical computation problem
  - gives exact results for the states in S^{yes} and S^{no} (no round-off)
  - suffices for model checking of qualitative properties

PCTL until for MDPs – Prob0E

- Minimum probabilities 0
  - $S^{no} = \{ s \in S \mid p_{min}(s, \phi_1 \text{ U } \phi_2) = 0 \} = Sat(\neg P_{>0} [\phi_1 \text{ U } \phi_2])$

$\text{PROB0E}(Sat(\phi_1), Sat(\phi_2))$

$R := Sat(\phi_2)$
$done :=$ **false**
while $(done =$ **false**$)$
$\quad R' := R \cup \{ s \in Sat(\phi_1) \mid \forall \mu \in Steps(s) \,.\, \exists s' \in R \,.\, \mu(s') > 0 \}$
$\quad$ **if** $(R' = R)$ **then** $done :=$ **true**
$\quad R := R'$
endwhile
return $S \backslash R$

PCTL until for MDPs – Prob0A

- Maximum probabilities 0
  - $S^{no} = \{ s \in S \mid p_{max}(s, \phi_1 \text{ U } \phi_2)=0 \}$

PROB0A($Sat(\phi_1), Sat(\phi_2)$)
1. $R := Sat(\phi_2)$
2. $done :=$ **false**
3. **while** ($done =$ **false**)
4. $\quad R' := R \cup \{s \in Sat(\phi_1) \mid \exists \mu \in Steps(s) \,.\, \exists s' \in R \,.\, \mu(s') > 0\}$
5. $\quad$ **if** ($R' = R$) **then** $done :=$ **true**
6. $\quad R := R'$
7. **endwhile**
8. **return** $S \backslash R$

PCTL until for MDPs – Prob1E

- Maximum probabilities 1
  - $S^{yes} = \{ s \in S \mid p_{max}(s, \phi_1 \text{ U } \phi_2)=1 \} = Sat(\neg P_{<1} [\phi_1 \text{ U } \phi_2])$
- Prob1E algorithm (see next slide)
  - two nested loops (double fixed point)
  - result, stored in R, will be S^{yes}; initially R is S
  - iteratively remove (some) states u with $p_{max}(u, \phi_1 \text{ U } \phi_2)<1$
    - i.e. remove (some) states for which, under no adversary σ, is $Prob^{\sigma}(s, \phi_1 \text{ U } \phi_2)=1$
  - done by inner loop which computes subset R' of R
    - R' contains ϕ_1-states with a probability distribution for which all transitions stay within R and at least one eventually reaches ϕ_2
  - note: after first iteration, R contains:
    - $\{ s \mid Prob^{\sigma}(s, \phi_1 \text{ U } \phi_2)>0 \text{ for some } \sigma \}$
    - essentially: execution of Prob0A and removal of S^{no} from R

PCTL until for MDPs – Prob1E

$\textsc{Prob1E}(Sat(\phi_1), Sat(\phi_2))$

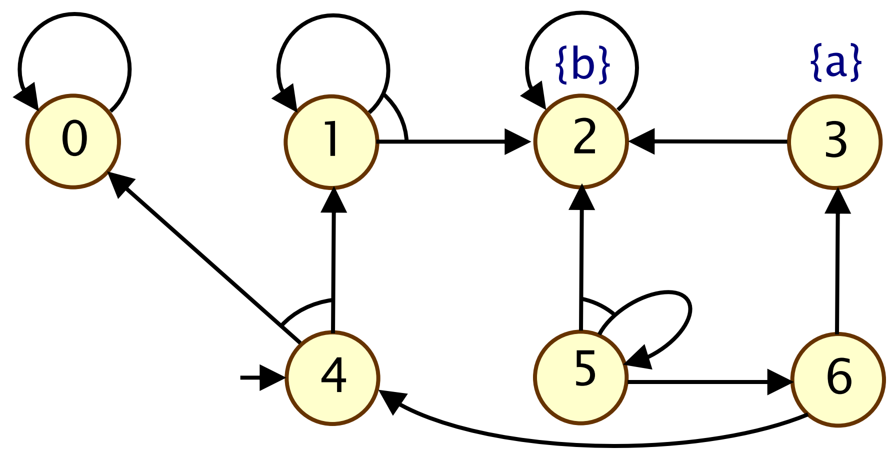
$R := S$
$done :=$ **false**
while $(done =$ **false**$)$
    $R' := Sat(\phi_2)$
    $done' :=$ **false**
    while $(done' =$ **false**$)$
        $R'' := R' \cup \{ s \in Sat(\phi_1) \mid \exists \mu \in Steps(s) \,.$
            $(\forall s' \in S \,.\, \mu(s') > 0 \rightarrow s' \in R) \wedge (\exists s' \in R' \,.\, \mu(s') > 0) \}$
        if $(R'' = R')$ **then** $done' :=$ **true**
        $R' := R''$
    endwhile
    if $(R' = R)$ **then** $done :=$ **true**
    $R := R'$
endwhile
return R

Prob1E – Example

- $S^{yes} = \{ s \in S \mid p_{max}(s, \neg a \text{ U } b)=1 \}$

- R = { 0, 1, 2, 3, 4 ,5 6 }
  - R' = {2} ; R' = {1, 2, 5} ; R' = {1, 2, 4, 5} ; R' = {1, 2, 4, 5, 6}
- R = { 1, 2, 4, 5, 6 }
  - R' = {2} ; R' = {1, 2, 5}
- R = { 1, 2, 5 }
  - R' = {2} ; R' = {1, 2, 5}
- R = { 1, 2, 5 }

- $S^{yes} = \{ 1, 2, 5 \}$



PCTL until for MDPs – Prob1A

- Minimum probabilities 1
  - $S^{yes} = \{ s \in S \mid p_{min}(s, \phi_1 \text{ U } \phi_2) = 1 \}$
- Can also be done with a graph-based algorithm
- Details omitted here
- For minimum probabilities, just take $S^{yes} = Sat(\phi_2)$
  - recall that computing states for which probability=1 is just an optimisation: it is not required for correctness

PCTL until for MDPs

- Min/max probabilities for the remaining states, i.e. $S^? = S \setminus (S^{yes} \cup S^{no})$, can be computed using either…
- 1. Value iteration
  - approximate iterative solution method
  - preferable in practice for efficiency reasons
- 2. Reduction to a linear optimisation problem
  - solve with well-known linear programming (LP) techniques
    - via simplex, ellipsoid method, interior point method
  - yields exact solution in finite number of steps
- 3. Policy iteration (not considered here)

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:

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

- 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. $\max_s \mid \underline{x}^{(n)}(s) - \underline{x}^{(n-1)}(s)) \mid < \varepsilon$ for some tolerance ε

Method 1 – Value iteration (max)

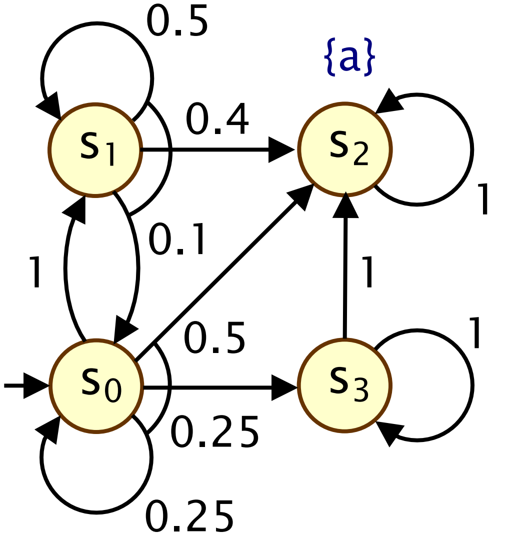
- Similarly, maximum probabilities satisfy:
  - $p_{max}(s, \phi_1 \text{ U } \phi_2) = \lim_{n\to\infty} x_s^{(n)}$ where:

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

- …and can be approximated iteratively

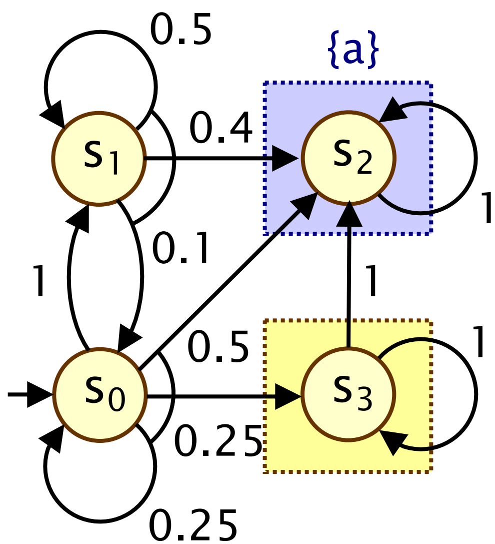
PCTL until – Example

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

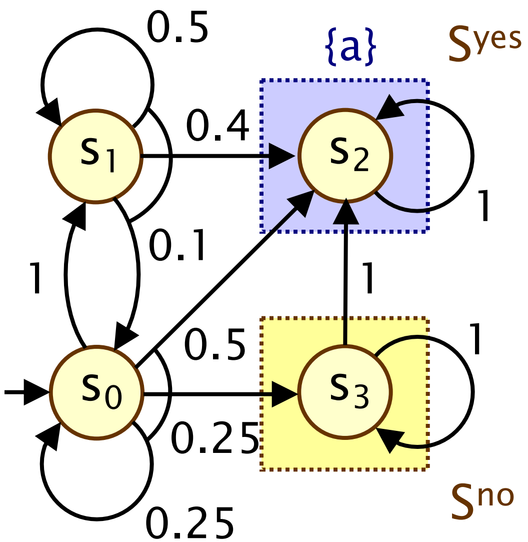


S^{yes} = Sat(a)

Prob0E

S^{no} = { s∈S | p_{min}(s, F a)=0 }

PCTL until – Example



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

$S^{yes} = \{s_2\}$, $S^{no} = \{s_3\}$, $S^? = \{s_0, s_1\}$

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

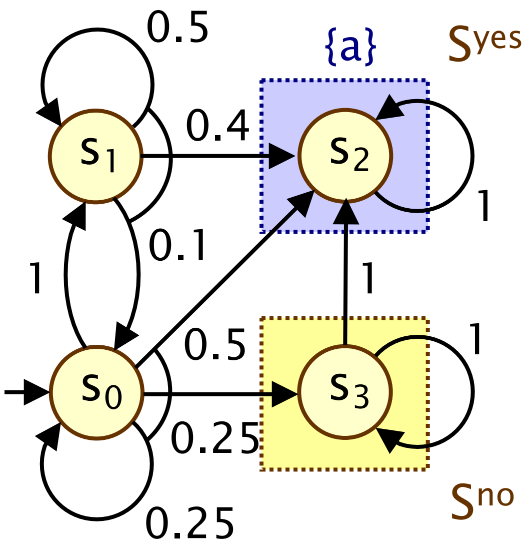
n=0: $[\ 0, 0, 1, 0\ ]$

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: …

PCTL until – Example



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

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

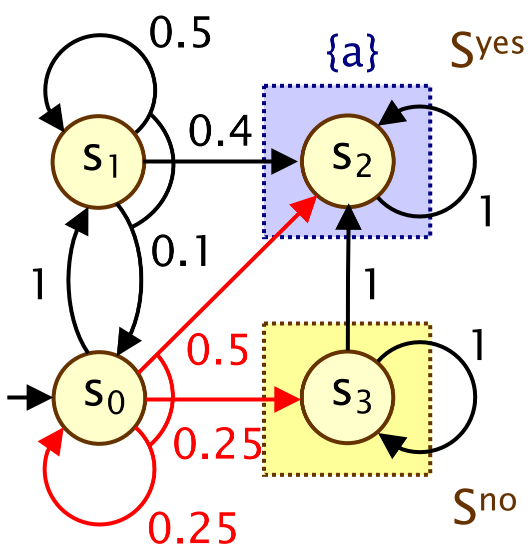
	$[\ x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}\ ]$
n=0:	[0.000000, 0.000000, 1, 0]
n=1:	[0.000000, 0.400000, 1, 0]
n=2:	[0.400000, 0.600000, 1, 0]
n=3:	[0.600000, 0.740000, 1, 0]
n=4:	[0.650000, 0.830000, 1, 0]
n=5:	[0.662500, 0.880000, 1, 0]
n=6:	[0.665625, 0.906250, 1, 0]
n=7:	[0.666406, 0.919688, 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 – Optimal adversary

- Like for reachability, can generate an optimal memoryless adversary using min/max probability values
  - and thus also a DTMC

- 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]

≈ [2/3, 14/15, 1, 0]

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

=min(14/15, 2/3)

Method 2 – Linear optimisation problem

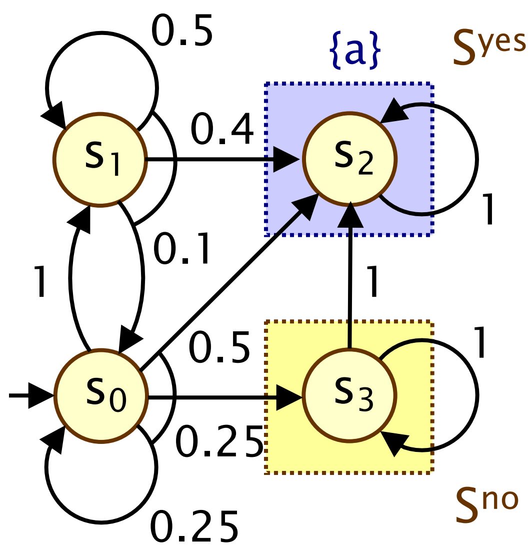
- Probabilities for states in $S^? = S \setminus (S^{yes} \cup S^{no})$ can also be obtained from a linear optimisation problem
- Minimum probabilities:

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

- Maximum probabilities:

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

PCTL until – Example



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

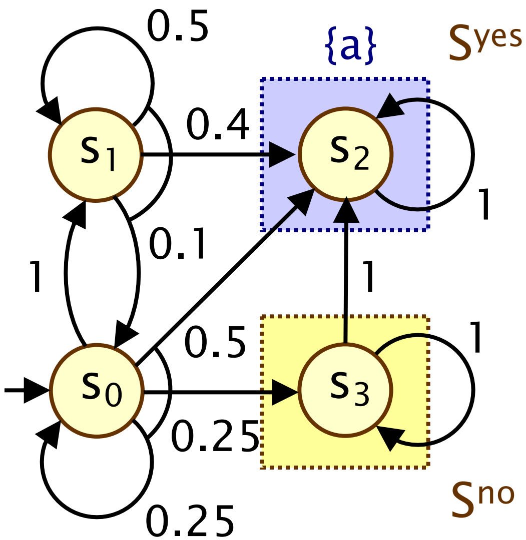
S^{yes}: $x_2=1$, S^{no}: $x_3=0$

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



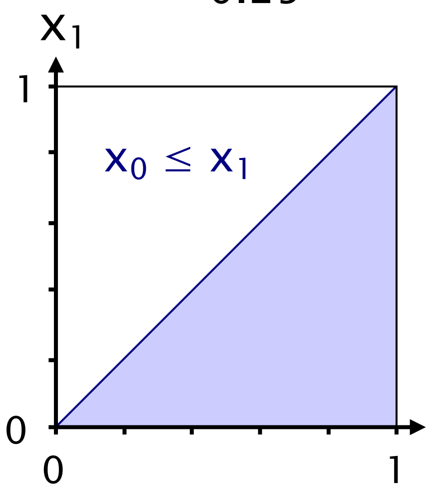
Let $x_i = p_{min}(s_i, F\ a)$

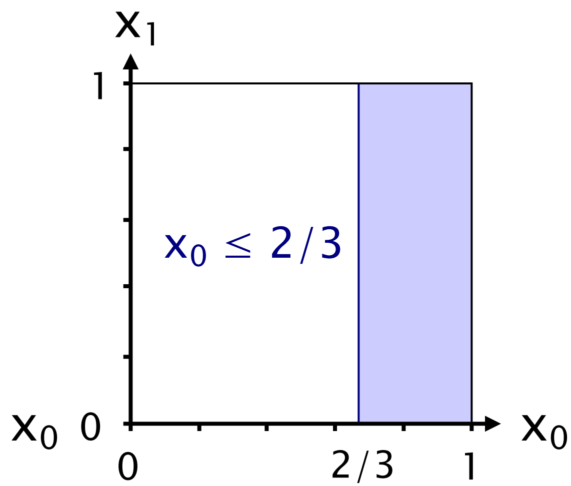
S^{yes}: $x_2=1$, S^{no}: $x_3=0$

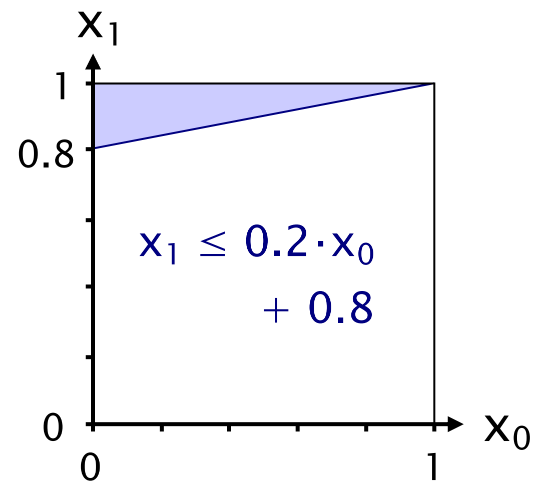
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

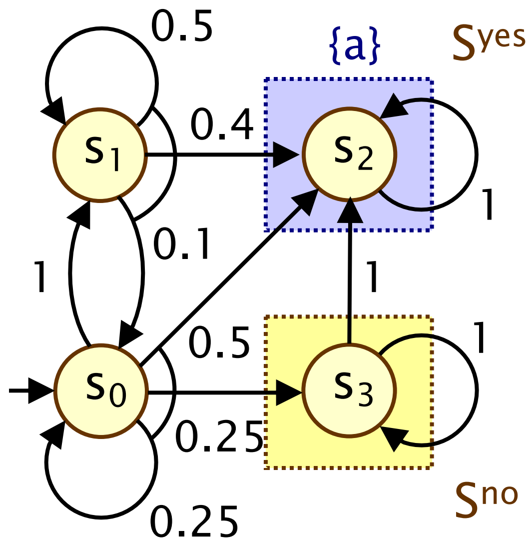
- $x_0 \leq x_1$
- $x_0 \leq 2/3$
- $x_1 \leq 0.2 \cdot x_0 + 0.8$







PCTL until – Example



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

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

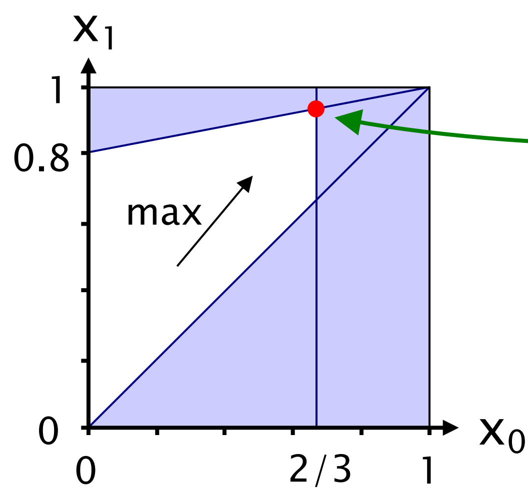
For $S^? = \{s_0, s_1\}$:

Maximise x_0+x_1 subject to constraints:

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

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

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



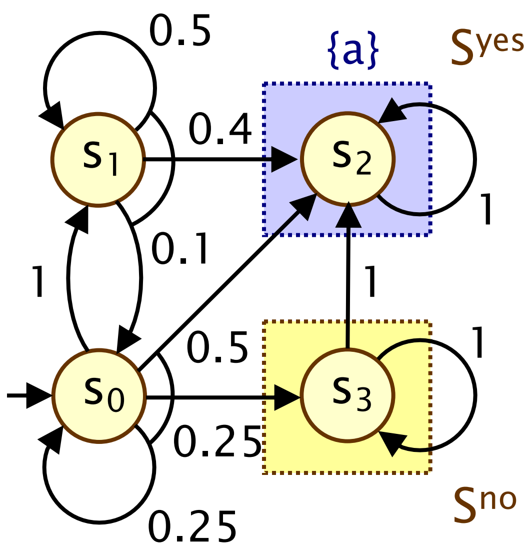
Solution:

(x_0, x_1)

$=$

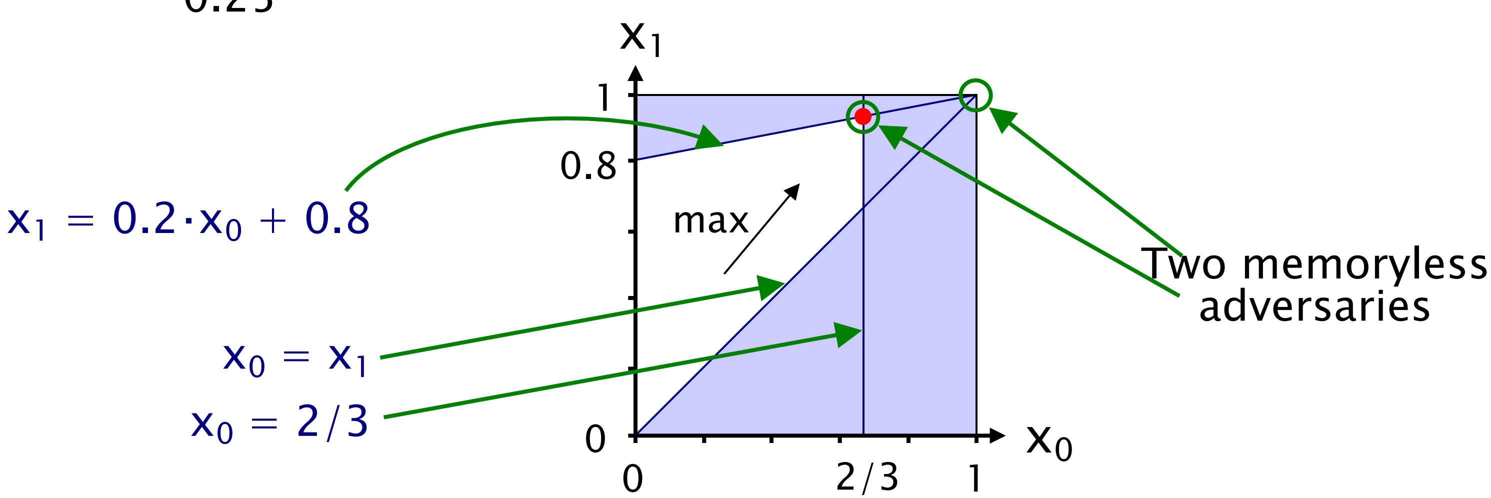
$(2/3, 14/15)$

Example – Optimal adversary



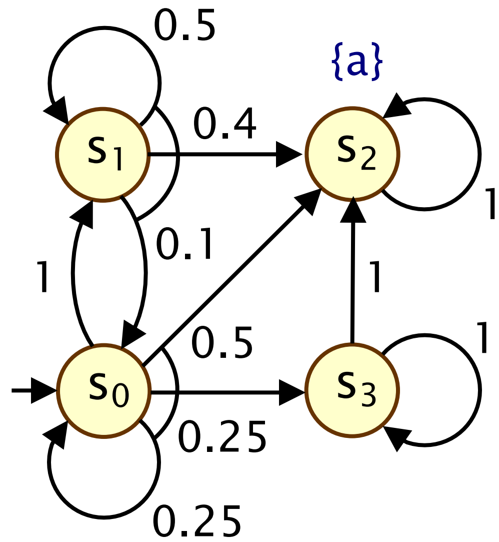
Get optimal adversary from constraints of optimisation problem that yield solution

Alternatively, use optimal probability values in value iteration function, as shown in value iteration example



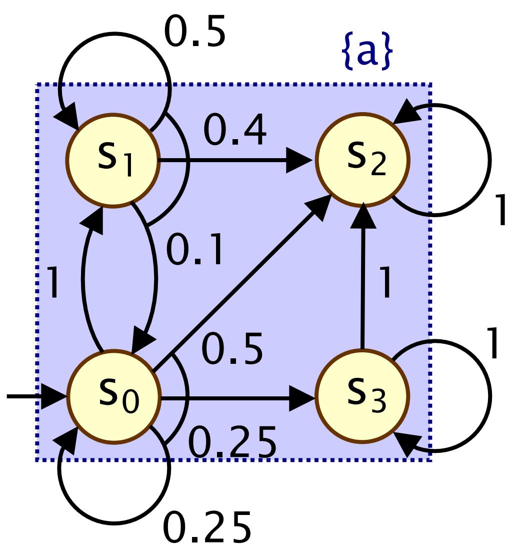
PCTL until – Example 2

- Model check: $P_{<0.1}$ [F a]
  - upper probability bound so maximum probabilities required



PCTL until – Example 2

- Model check: $P_{<0.1}$ [F a]
  - upper probability bound so maximum probabilities required



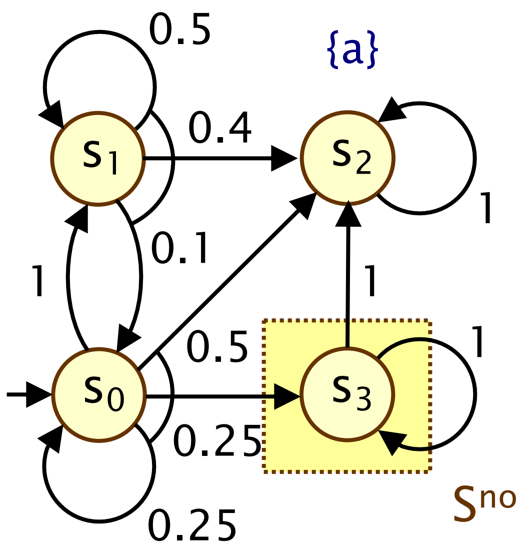
S^{yes} = { s∈S | p_{max}(s, F a)=1 } = S

S^{no} = { s∈S | p_{max}(s, F a)=0 } = ∅

- $\underline{p}_{max}$(F a) = [1, 1, 1, 1] and Sat($P_{<0.1}$ [F a]) = ∅

PCTL until – Example 3

- Model check: $P_{>0}$ [F a]
  - lower probability bound so minimum probabilities required
  - qualitative property so numerical computation can be avoided



S^{no} = { s∈S | p_{min}(s, F a)=0 }

Prob0E yields S^{no} = {s_3}

- $\underline{p}_{min}$(F a) = [?, ?, ?, 0] and Sat($P_{>0}$ [F a]) = {s_0,s_1,s_2}

Costs and rewards

- We can augment MDPs with rewards (or costs)
  - real-valued quantities assigned to states and/or actions
  - different from the DTMC case where transition rewards assigned to individual transitions

- For a MDP (S,s_{init},**Steps**,L), a reward structure is a pair ($\underline{\rho}$,ι)
  - $\underline{\rho} : S \rightarrow \mathbb{R}_{\geq 0}$ is the state reward function
  - $\iota : S \times Act \rightarrow \mathbb{R}_{\geq 0}$ is transition reward function

- As for DTMCs these can be used to compute:
  - elapsed time, power consumption, size of message queue, number of messages successfully delivered, net profit, …

PCTL and rewards

- Augment PCTL with reward-based properties
  - allow a wide range of quantitative measures of the system
  - basic notion: expected value of rewards

expected reward is ~r

$$\phi \ ::= \ \ldots \ \mid \ R_{\sim r} [\ I^{=k} \ ] \ \mid \ R_{\sim r} [\ C^{\leq k} \ ] \ \mid \ R_{\sim r} [\ F \ \phi \ ]$$

where $r \in \mathbb{R}_{\geq 0}$, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

- $R_{\sim r}$ [·] means "the expected value of · satisfies ~r for all adversaries"

PCTL and rewards

- Augment PCTL with reward-based properties
  - allow a wide range of quantitative measures of the system
  - basic notion: expected value of rewards

expected reward is ~r

$$\phi \ ::= \ \ldots \ | \ R_{\sim r} [\, I^{=k} \,] \ | \ R_{\sim r} [\, C^{\leq k} \,] \ | \ R_{\sim r} [\, F\ \phi \,]$$

"instantaneous" "cumulative" "reachability"

where $r \in \mathbb{R}_{\geq 0}$, $\sim \in \{<,>,\leq,\geq\}$, $k \in \mathbb{N}$

- $R_{\sim r} [\ \cdot\ ]$ means "the expected value of $\cdot$ satisfies $\sim r$ for all adversaries"

Types of reward formulas

- Instantaneous: $R_{\sim r}$ [$I^{=k}$]
  - the expected value of the reward at time step k is ~r for all adversaries
  - "the minimum expected queue size after exactly 90 seconds"

- Cumulative: $R_{\sim r}$ [$C^{\leq k}$]
  - the expected reward cumulated up to time step k is ~r for all adversaries
  - "the maximum expected power consumption over one hour"

- Reachability: $R_{\sim r}$ [F ϕ]
  - the expected reward cumulated before reaching a state satisfying ϕ is ~r for all adversaries
  - "the maximum expected time for the algorithm to terminate"

Reward formula semantics

- Formal semantics of the three reward operators:
  - for a state s in the MDP:
  - $s \models R_{\sim r}$ [$I^{=k}$] $\Leftrightarrow$ $Exp^{\sigma}(s, X_{I=k}) \sim r$ for all adversaries σ
  - $s \models R_{\sim r}$ [$C^{\leq k}$] $\Leftrightarrow$ $Exp^{\sigma}(s, X_{C\leq k}) \sim r$ for all adversaries σ
  - $s \models R_{\sim r}$ [F Φ] $\Leftrightarrow$ $Exp^{\sigma}(s, X_{F\Phi}) \sim r$ for all adversaries σ

$Exp^{\sigma}(s, X)$ denotes the expectation of the random variable $X : Path^{\sigma}(s) \rightarrow \mathbb{R}_{\geq 0}$ with respect to the probability measure Pr^{σ}_{s}

Reward formula semantics

- For an infinite path $\omega = s_0(a_0,\mu_0)s_1(a_1,\mu_1)s_2\ldots$

$$X_{I=k}(\omega) = \underline{\rho}(s_k)$$

$$X_{C\leq k}(\omega) = \begin{cases} 0 & \text{if } k = 0 \\ \sum_{i=0}^{k-1} \underline{\rho}(s_i) + \iota(a_i) & \text{otherwise} \end{cases}$$

$$X_{F\phi}(\omega) = \begin{cases} 0 & \text{if } s_0 \in Sat(\phi) \\ \infty & \text{if } s_i \notin Sat(\phi) \text{ for all } i \geq 0 \\ \sum_{i=0}^{k_\phi - 1} \underline{\rho}(s_i) + \iota(a_i) & \text{otherwise} \end{cases}$$

where $k_\phi = \min\{ i \mid s_i \vDash \phi \}$

(typo: iota fcn also depends on state)

Model checking reward formulas

- Instantaneous: $R_{\sim r}$ [$I^{=k}$]
  - similar to the computation of bounded until probabilities
  - solution of recursive equations
  - k matrix-vector multiplications (+ min/max)

- Cumulative: $R_{\sim r}$ [$C^{\leq k}$]
  - extension of bounded until computation
  - solution of recursive equations
  - k iterations of matrix-vector multiplication + summation

See [FKNP11] for details

- Reachability: $R_{\sim r}$ [F ϕ]
  - similar to the case for until
  - solve a linear optimization problem (or value iteration)

Model checking $R_{\sim r}$ [$I^{=k}$]

- Min/max expected instantaneous reward at step k
  - can be computed recursively, in a “backwards” fashion
  - i.e. similar to the equivalent reward operator on DTMCs

- Let: $Exp^{max}(s, X_{I=k}) = \sup_{\sigma \in Adv} Exp^{\sigma}(s, X_{I=k})$

- Then:

$$Exp^{max}(s,\ X_{I=k}) = \begin{cases} \rho(s) & \text{if } k = 0 \\ \max\left\{ \sum_{s' \in S} \mu(s') \cdot Exp^{max}(s',\ X_{I=k-1}) \mid (a,\mu) \in \mathbf{Steps}(s) \right\} & \text{if } k > 0 \end{cases}$$

- See [FKNP11] for further details

Model checking complexity

- For model checking of an MDP ($S, s_{init}, \mathbf{Steps}, L$) and PCTL formula ϕ (including reward operators)
  - complexity is linear in $|\Phi|$ and polynomial in $|S|$

- Size $|\phi|$ of ϕ is defined as number of logical connectives and temporal operators, plus sizes of temporal operators
  - model checking is performed for each operator

- Worst-case operators are $P_{\sim p}$ [ϕ_1 U ϕ_2] and $R_{\sim r}$ [F ϕ]
  - main task: solution of linear optimization problem of size $|S|$
  - can be solved with ellipsoid method (polynomial in $|S|$)
  - and also precomputation algorithm (max $|S|$ steps)

Summing up…

- PCTL for MDPs
  - same as syntax as for PCTL
  - key difference in semantics: "for all adversaries"
  - requires computation of minimum/maximum probabilities
- PCTL model checking for MDPs
  - same basic algorithm as for DTMCs
  - next: matrix-vector multiplication + min/max
  - bounded until: k matrix-vector multiplications + min/max
  - until : precomputation algorithm + numerical computation
    - precomputation: Prob0A and Prob1E for max, Prob0E for min
    - numerical computation: value iteration, linear optimisation
  - complexity linear in $|\Phi|$ and polynomial in $|S|$
- Costs and rewards