Lecture 15
LTL model checking for DTMCs and MDPs

Alessandro Abate

Department of Computer Science
University of Oxford

Overview

- Limitations of PCTL, review of LTL
- Recall
  - deterministic ω-automata (DBA or DRA) and DTMCs
- LTL model checking for DTMCs
  - measurability
  - complexity
  - PCTL* model checking for DTMCs
- LTL model checking for MDPs

Limitations of PCTL

- PCTL, although useful in practice, has limited expressivity
  - essentially: probability of reaching states in X, passing only through states in Y (and within k time steps)

- Alternative logics can be used, for example:
  - LTL [Pnu77] – non-probabilistic linear-time temporal logic
  - PCTL* [ASB+95,BdA95] – subsumes both PCTL and LTL

- In PCTL, temporal operators always appear inside $P_{\sim p}$ [...]
  - (in CTL, they always appear inside A or E)
  - in LTL (and PCTL*), temporal operators can be combined

Review – CTL, PCTL and LTL

- CTL
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid A\,\psi \mid E\,\psi$
  - $\psi ::= X\,\phi \mid \phi\, U\, \phi$

- PCTL
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p}\,[\,\psi\,]$
  - $\psi ::= X\,\phi \mid \phi\, U^{\leq k}\, \phi \mid \phi\, U\, \phi$

- LTL
  - path formulae only
  - $\psi ::= \text{true} \mid a \mid \psi \wedge \psi \mid \neg\psi \mid X\,\psi \mid \psi\, U\, \psi$

LTL + probabilities

- Same idea as PCTL: probabilities over sets of paths satisfying (path) formulae
  - for a state s of a DTMC and an LTL formula ψ:
  - Prob(s, ψ) = Pr_s { ω ∈ Path(s) | ω ⊨ ψ }
  - all such path sets are measurable (see later)

- For MDPs, we can again consider lower/upper bounds
  - $p_{min}(s, \psi) = \inf_{\sigma \in Adv} Prob^{\sigma}(s, \psi)$
  - $p_{max}(s, \psi) = \sup_{\sigma \in Adv} Prob^{\sigma}(s, \psi)$
  - (over LTL formula ψ)

- For DTMCs and MDPs, an LTL specification often comprises an LTL (path) formula and a probability bound
  - e.g. $P_{>0.99}$ [F (req ∧ X ack)]

Recall – DBA and DRA

- Deterministic Büchi automata (DBA)
  - $(Q, \Sigma, \delta, q_0, F)$
  - accepting run must visit some state in F infinitely often
  - less expressive than nondeterministic Büchi automata (NBA)

- Deterministic Rabin automata (DRA)
  - $(Q, \Sigma, \delta, q_0, Acc)$
  - $Acc = \{ (L_i, K_i) \mid 1 \leq i \leq k \}$
  - for some pair (L_i, K_i), the states in L_i must be visited finitely often and (some of) the states in K_i visited infinitely often
  - equally expressive as NBA
  - expresses all ω-regular properties; and hence all LTL formulae

Product DTMC for a DBA

- For DTMC D and DBA A

$$\mathrm{Prob}^{D}(s, A) = \mathrm{Prob}^{D\otimes A}((s,q_s), \mathrm{GF}\ accept)$$

  - where $q_s = \delta(q_0, L(s))$

- Hence:

$$\mathrm{Prob}^{D}(s, A) = \mathrm{Prob}^{D\otimes A}((s,q_s), \mathrm{F}\ T_{GFaccept})$$

  - where $T_{GFaccept}$ is the union of all BSCCs T in $D\otimes A$ with $T \cap Sat(accept) \neq \emptyset$

- Reduces to computing BSCCs and reachability probabilities

Product DTMC for a DRA

- For DTMC D and DRA A

$$\boxed{\mathrm{Prob}^{D}(s, A) = \mathrm{Prob}^{D \otimes A}((s,q_s), \vee_{1 \leq i \leq k} (FG \neg l_i \wedge GF\ k_i))}$$

  - where $q_s = \delta(q_0, L(s))$
- Hence:

$$\boxed{\mathrm{Prob}^{D}(s, A) = \mathrm{Prob}^{D \otimes A}((s,q_s), F\ T_{Acc})}$$

  - where T_{Acc} is the union of all accepting BSCCs in $D \otimes A$
  - an accepting BSCC T of $D \otimes A$ is such that, for some $1 \leq i \leq k$:
    - $q \models \neg l_i$ for all $(s,q) \in T$ and $q \models k_i$ for some $(s,q) \in T$
    - i.e. $T \cap (S \times L_i) = \varnothing$ and $T \cap (S \times K_i) \neq \varnothing$
- Reduces to computing BSCCs and reachability probabilities

LTL model checking for DTMCs

- Model check LTL specification $P_{\sim p} [\ \psi\ ]$ against DTMC D

- 1. Generate a deterministic Rabin automaton (DRA) for ψ
  - build nondeterministic Büchi automaton (NBA) for ψ [VW94]
  - convert the NBA to a DRA [Saf88]
- 2. Construct product DTMC $D \otimes A$
- 3. Identify accepting BSCCs of $D \otimes A$
- 4. Compute probability of reaching accepting BSCCs
  - from all states of the $D \otimes A$
- 5. Compare probability for (s, q_s) against p for each s

- Qualitative LTL model checking – no probabilities needed

Measurability of ω-regular properties

- For any ω-regular property ψ
  - the set of ψ-satisfying paths in any DTMC D is measurable
- Hence, the same applies to
  - any LTL formula
  - any regular safety property

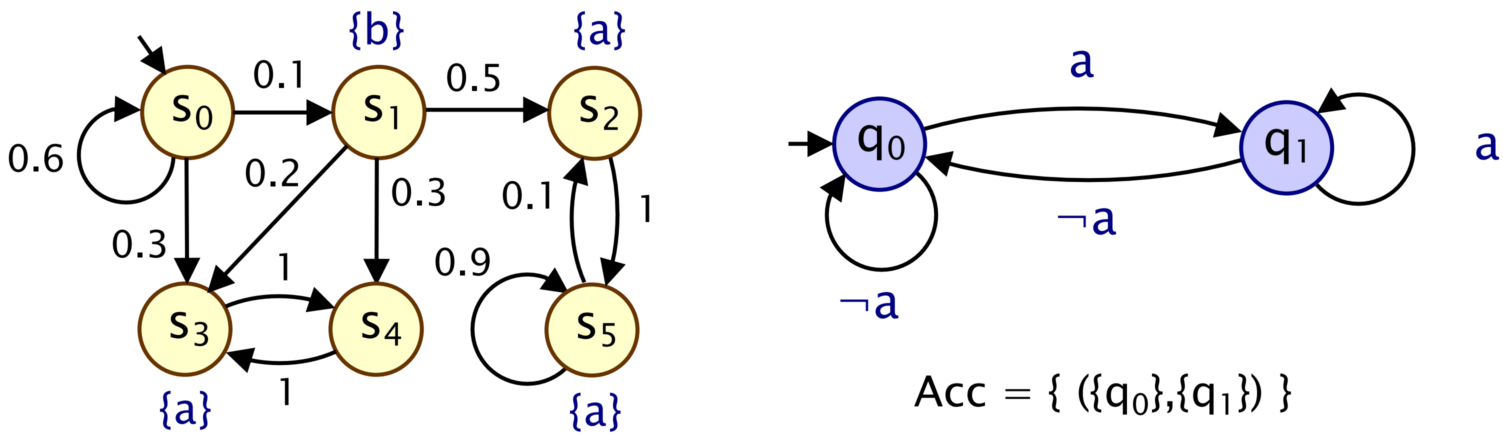
- Proof sketch
  - any ω-regular property can be represented by a DRA A
  - we can construct $D \otimes A$, in which there is a direct mapping from any path ω in D to a path ω' in $D \otimes A$
  - $\omega \vDash \psi$ iff $\omega' \vDash \bigvee_{1 \leq i \leq k} (FG \neg l_i \wedge GF\, k_i)$
  - GF Φ and FG Φ are measurable (see lecture 3)
  - $\wedge$ and $\vee$ = intersection/union (which preserve measurability)

Complexity

- Complexity of model checking LTL formula ψ on DTMC D
  - is doubly exponential in |ψ| and polynomial in |D|
  - (for the algorithm presented in these lectures)
- Converting LTL formula ψ to DRA A
  - for some LTL formulae of size n, size of smallest DRA is 2^{2^n}
- BSCC computation
  - Tarjan algorithm – linear in model size (states/transitions)
- Probabilistic reachability
  - linear equations – cubic in (product) model size
- In total: O(poly(|D|,|A|))
- In practice: |ψ| is small and |D| is large
- Complexity can be reduced to single exponential in |ψ|
  - see e.g. [CY88,CY95]

Example 3 (Lec 15) revisited

- Model check $P_{>0.2}$ [FG a]



Acc = { ({q_0},{q_1}) }

- Result:
  - Prob(FG a) = [0.125, 0.5, 1, 0, 0, 1]
  - Sat($P_{>0.2}$ [FG a]) = { s_1, s_2, s_5 }

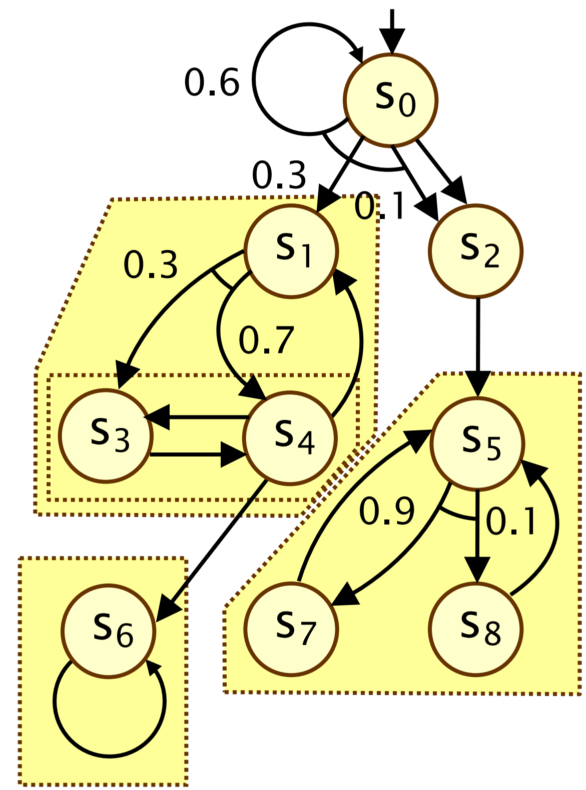
PCTL* model checking

- PCTL* syntax:
  - $\phi ::= \text{true} \mid a \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p} [\ \psi\ ]$
  - $\psi ::= \phi \mid \psi \wedge \psi \mid \neg\psi \mid X\ \psi \mid \psi\ U\ \psi$
- Example:
  - $P_{>p} [\ GF\ (\ send \rightarrow P_{>0} [\ F\ ack\ ]\ )\ ]$

- PCTL* model checking algorithm
  - bottom-up traversal of parse tree for formula (like PCTL)
  - to model check $P_{\sim p} [\ \psi\ ]$:
    - replace maximal state subformulae with atomic propositions
    - (state subformulae already model checked recursively)
    - path specification ψ is now an LTL formula
    - which can be model checked as for LTL

On to MDPs: Recall End Components

- End components of MDPs are the analogue of BSCCs in DTMCs
- An end component is a strongly connected sub-MDP
- A sub-MDP comprises a subset of states and a subset of the actions/distributions available in those states, which is closed under probabilistic branching



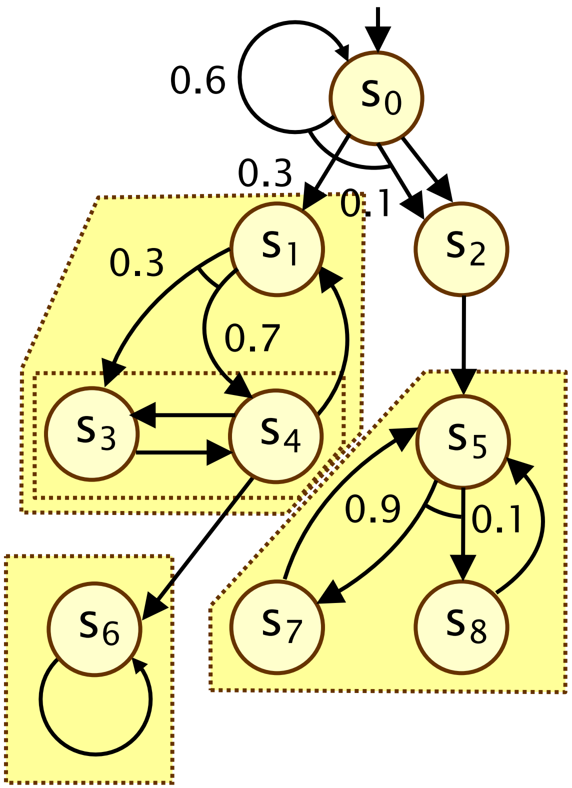
Note:
- action labels omitted
- probabilities omitted where =1

On to MDPs: Recall End Components

- End components of MDPs are the analogue of BSCCs in DTMCs

- 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

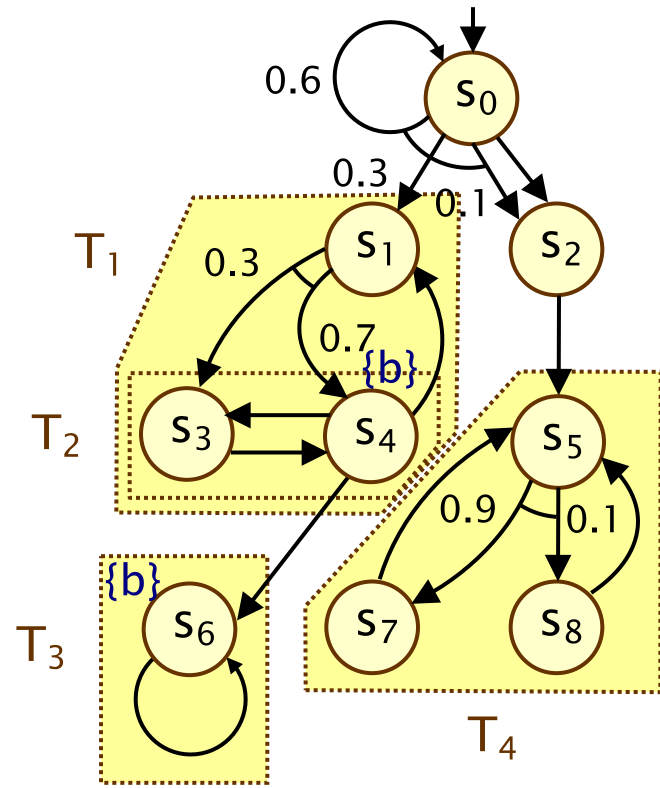


Repeated reachability and Persistence

- Maximum probabilities
  - $p_{max}(s, GF\ a) = p_{max}(s, F\ T_{GFa})$
    - where T_{GFa} is the union of sets T for all end components (T,**Steps**) with $T \cap Sat(a) \neq \emptyset$
  - $p_{max}(s, FG\ a) = p_{max}(s, F\ T_{FGa})$
    - where T_{FGa} is the union of sets T for all end components (T,**Steps**) with $T \subseteq Sat(a)$
- Minimum probabilities
  - need to compute from maximum probabilities…
  - $p_{min}(s, GF\ a) = 1 - p_{max}(s, FG\neg a)$
  - $p_{min}(s, FG\ a) = 1 - p_{max}(s, GF\neg a)$

Example

- Check: $P_{<0.8}$ [GF b] for s_0
- Compute p_{max}(GF b)
  - $p_{max}(GF\ b) = p_{max}(s, F\ T_{GFb})$
  - T_{GFb} is the union of sets T for all end components with $T \cap Sat(b) \neq \varnothing$
  - $Sat(b) = \{ s_4, s_6 \}$
  - $T_{GFb} = T_1 \cup T_2 \cup T_3 = \{ s_1, s_3, s_4, s_6 \}$
  - $p_{max}(s, F\ T_{GFb}) = 0.75$
  - $p_{max}(GF\ b) = 0.75$
- Result: $s_0 \vDash P_{<0.8}$ [GF b]



Automata-based properties for MDPs

- For an MDP M and automaton A over alphabet 2^{AP}
  - consider probability of "satisfying" language $L(A) \subseteq (2^{AP})^\omega$
  - $Prob^{M,\sigma}(s, A) = Pr_s^{M,\sigma}\{ \omega \in Path^{M,\sigma}(s) \mid trace(\omega) \in L(A) \}$
  - $p_{max}^M(s, A) = \sup_{\sigma \in Adv} Prob^{M,\sigma}(s, A)$
  - $p_{min}^M(s, A) = \inf_{\sigma \in Adv} Prob^{M,\sigma}(s, A)$

- Might need minimum or maximum probabilities
  - e.g. $s \models P_{\geq 0.99} [\, \psi_{good} \,] \Leftrightarrow p_{min}^M (s, \psi_{good}) \geq 0.99$
  - e.g. $s \models P_{\leq 0.05} [\, \psi_{bad} \,] \Leftrightarrow p_{max}^M (s, \psi_{bad}) \leq 0.05$
- But, ω-regular properties are closed under negation
  - as are automata (under complementation) representing them
  - so can always consider (e.g.,) maximum probabilities…
  - $p_{max}^M(s, \psi_{bad})$ or $1 - p_{min}^M(s, \psi_{good})$

LTL model checking for MDPs

- Model check LTL specification $P_{\sim p}$ [ψ] against MDP M
- 1. Convert problem to one needing maximum probabilities
  - e.g. convert $P_{>p}$ [ψ] to $P_{<1-p}$ [$\neg\psi$]
- 2. Generate a DRA for ψ (or $\neg\psi$)
  - build nondeterministic Büchi automaton (NBA) for ψ [VW94]
  - convert the NBA to a DRA [Saf88]
- 3. Construct product MDP $M \otimes A$
- 4. Identify accepting end components (ECs) of $M \otimes A$
- 5. Compute max probability of reaching accepting ECs
  - from all states of the $D \otimes A$
- 6. Compare probability for (s, q_s) against p, for each s

Product MDP for a DRA

- For an MDP $M = (S, s_{init}, \mathbf{Steps}, L)$
- and a (total) DRA $A = (Q, \Sigma, \delta, q_0, Acc)$
  - where $Acc = \{ (L_i, K_i) \mid 1 \leq i \leq k \}$

- The product MDP $M \otimes A$ is:
  - the MDP $(S \times Q, (s_{init}, q_{init}), \mathbf{Steps}', L')$ where:

$$q_{init} = \delta(q_0, L(s_{init}))$$

$$\mathbf{Steps}'((s,q)) = \{ \mu^q \mid \mu \in \mathrm{Step}(s) \}$$

$$\mu^q(s', q') = \begin{cases} \mu(s') & \text{if } q' = \delta(q, L(s)) \\ 0 & \text{otherwise} \end{cases}$$

$l_i \in L'((s,q))$ if $q \in L_i$ and $k_i \in L'((s,q))$ if $q \in K_i$

(i.e. state sets of acceptance condition used as labels)

Product MDP for a DRA

- For MDP M and DRA A

$$\boxed{p_{max}{}^{M}(s, A) = p_{max}{}^{M\otimes A}((s,q_s), \vee_{1\leq i\leq k} (FG\ \neg l_i \wedge GF\ k_i))}$$

  – where $q_s = \delta(q_0, L(s))$

- Hence:

$$\boxed{p_{max}{}^{M}(s, A) = p_{max}{}^{M\otimes A}((s,q_s), F\ T_{Acc})}$$

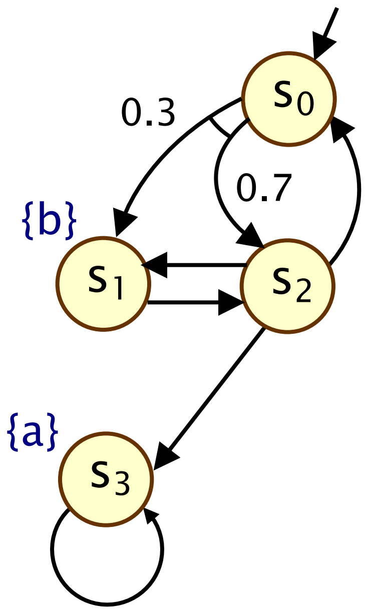
  – where T_{Acc} is the union of all sets T for accepting end components (T,**Steps'**) in D⊗A

  – an accepting end component is such that, for some $1 \leq i \leq k$:

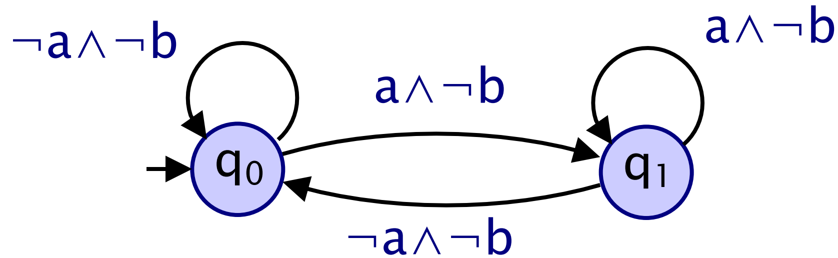
    - $(s,q) \vDash \neg l_i$ for all $(s,q) \in T$ and $(s,q) \vDash k_i$ for some $(s,q) \in T$
    - i.e. $T \cap (S \times L_i) = \emptyset$ and $T \cap (S \times K_i) \neq \emptyset$

MDPs - Example 1

- Model check $P_{<0.8}$ [G ¬b ∧ GF a]



DRA (in fact DBA):



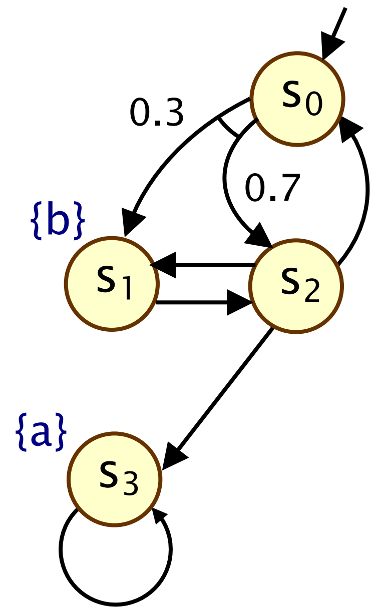
Acc = { (∅, {q_1}) }

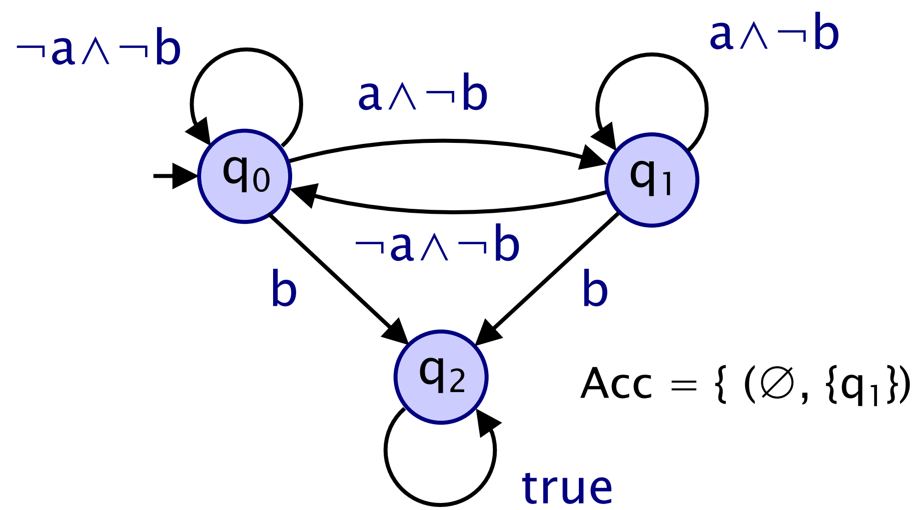
- Result:
  - $\underline{p}_{max}$(G ¬b ∧ GF a) = [0.7, 0, 1, 1]
  - Sat($P_{<0.8}$ [G ¬b ∧ GF a]) = { s_0, s_1 }

MDPs – Example 1

- Model check $P_{<0.8}$ [G ¬b ∧ GF a]

Total DRA (in fact DBA):

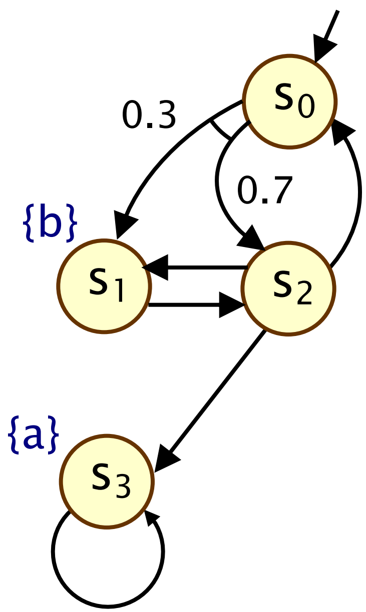




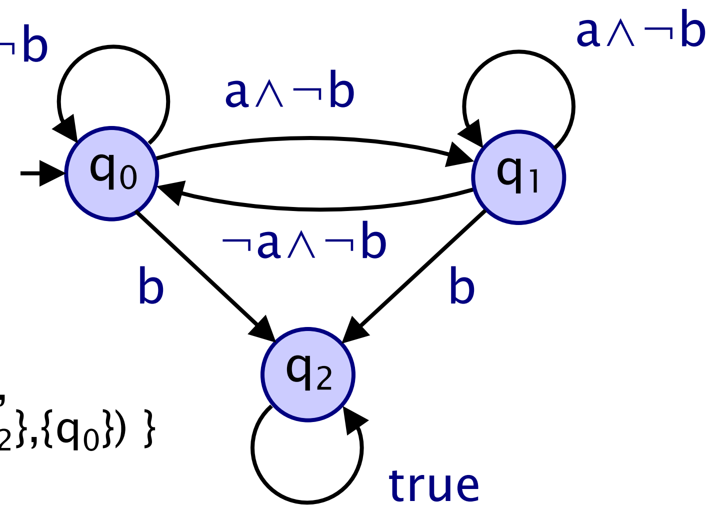
- Result:
  - $\underline{p}_{max}$(G ¬b ∧ GF a) = [0.7, 0, 1, 1]
  - Sat($P_{<0.8}$ [G ¬b ∧ GF a]) = { s_0, s_1 }

MDPs – Example 2

- Model check $P_{>0}$ [G ¬b ∧ GF a]
  - p_{min}(s, G ¬b ∧ GF a) = 1 – p_{max}(s, ¬(G ¬b ∧ GF a))
    = 1 – p_{max}(s, F b ∨ FG ¬a))



DRA:



Acc = { (∅,{q_2}), ({q_1,q_2},{q_0}) }

- Result: $\underline{p}_{min}$(G ¬b ∧ GF a) = [0, 0, 0, 1]
  - Sat($P_{>0}$ [G ¬b ∧ GF a]) = {s_3}

LTL model checking for MDPs

- Maximal end components
  - can optimise LTL model checking using maximal end components (there may be exponentially many ECs)
- Qualitative LTL model checking
  - no numerical computation: use Prob1E, Prob0A algorithms
- Complexity of model checking LTL formula ψ on MDP M
  - is doubly exponential in |ψ| and polynomial in |M|
  - unlike DTMCs, this cannot be improved upon
- PCTL* model checking
  - LTL model checking can be adapted to PCTL*, as for DTMCs
- Optimal adversaries for LTL formulae
  - memoryless adversary always exists for p_{max}(s, GF a) and for p_{max}(s, FG a), <u>but not for arbitrary LTL formulae</u>

Summing up…

- Deterministic ω-automata (DBA or DRA) and DTMCs
  - probability of language acceptance reduces to probabilistic reachability of set of accepting BSCCs in product DTMC
- LTL model checking for DTMCs
  - via construction of DRA for LTL formula
  - complexity: (doubly) exponential in the size of the LTL formula and polynomial in the size of the DTMC
  - measurability of any ω-regular property on a DTMC
- PCTL* model checking for DTMCs
  - combination of PCTL and LTL model checking algorithm
- LTL model checking for MDPs
  - max. probabilities of reaching accepting end components
  - min. probabilities through negation of max. probabilities