PAC Statistical Model Checking for Markov Decision Processes and Stochastic Games

Pranav Ashok, Jan Křetínský, Maximilian Weininger

Technical University of Munich

July 16, 2019

Prelude

Given biased coin with $P(H) = p$

How does one reasonably estimate p?

Prelude

Given biased coin with $P(H) = p$

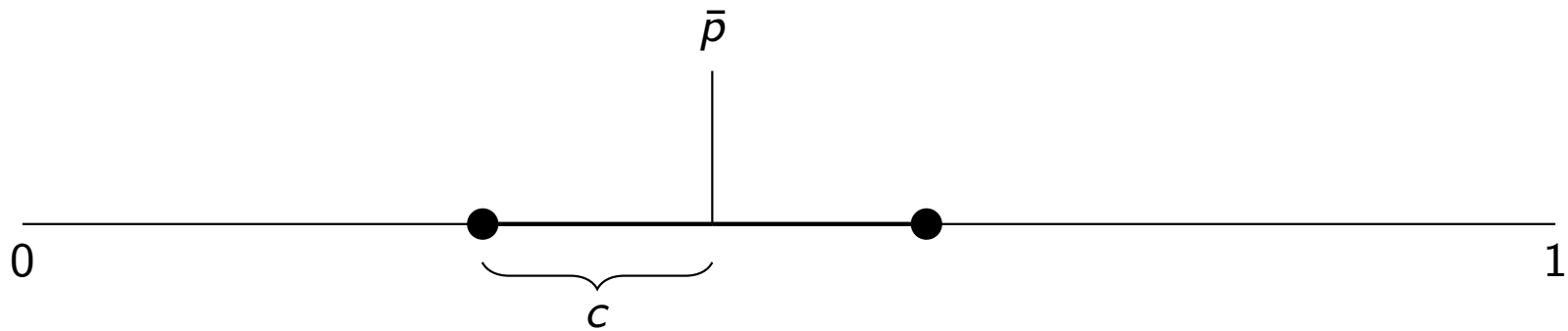
$$\bar{p} = \frac{\#H}{\#H + \#T}$$

Prelude

Given biased coin with $P(H) = p$

Hoeffding Inequality

$$P(\bar{p} - p \geq c) \leq e^{-2c^2 N}$$



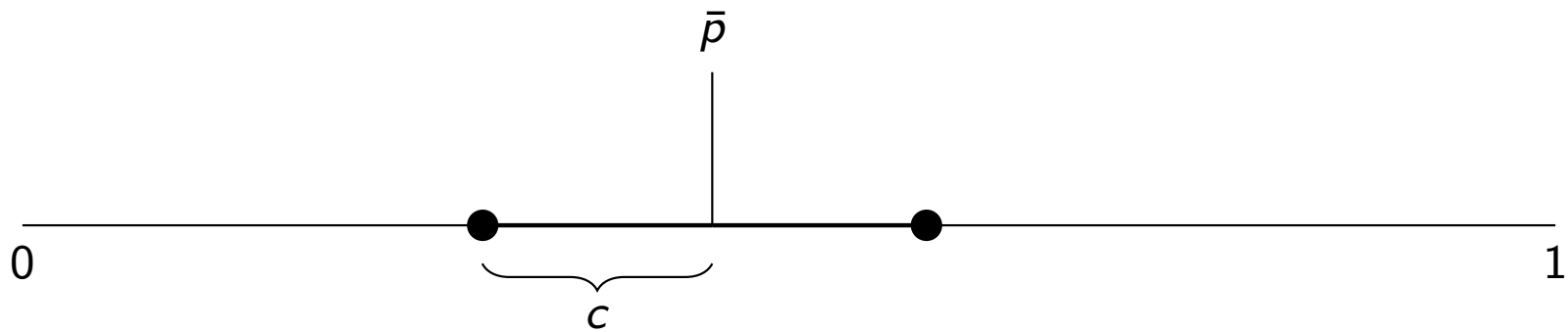
Prelude

Given biased coin with $P(H) = p$

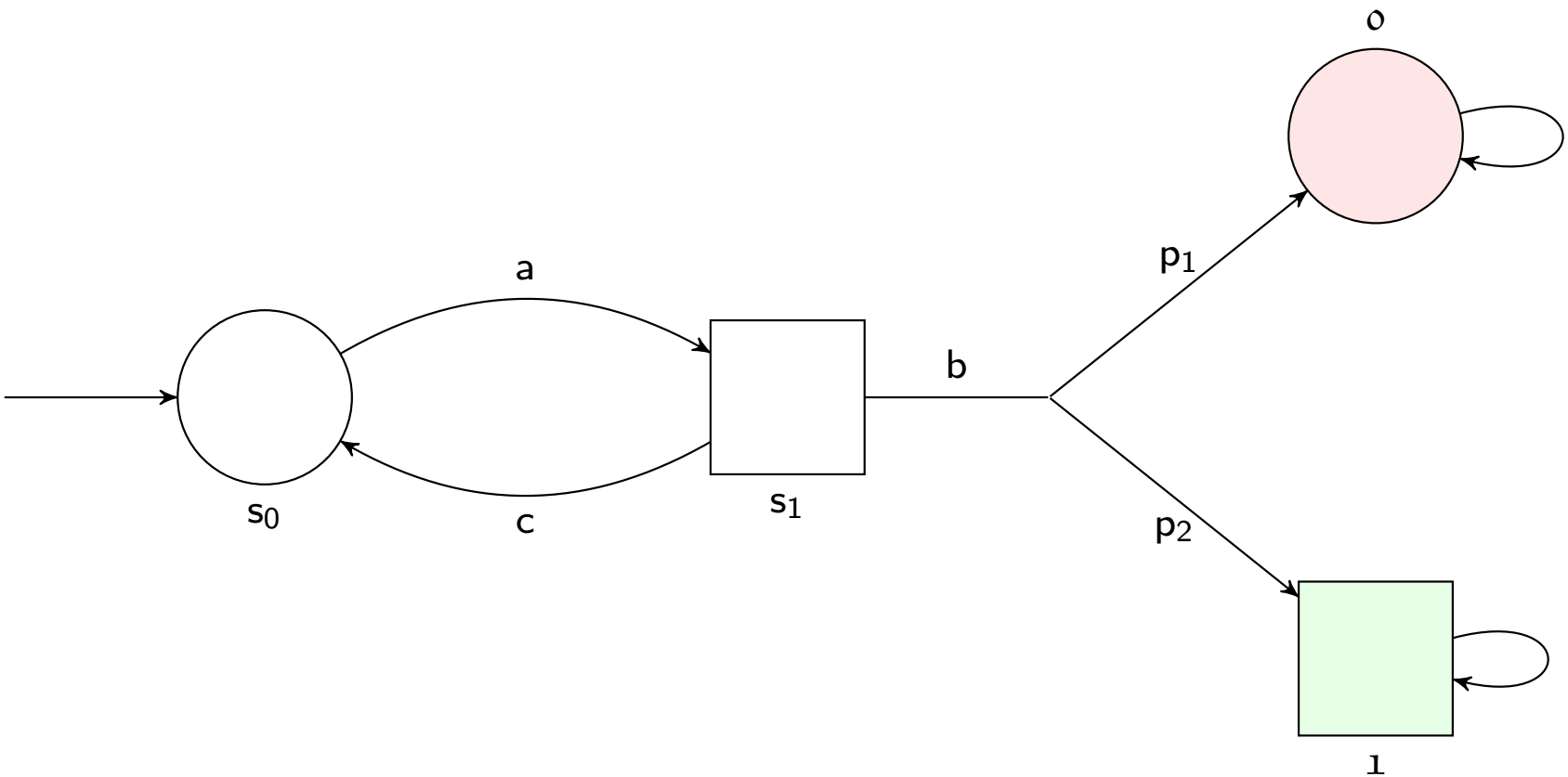
Hoeffding Inequality

$$P(\bar{p} - p \geq c) \leq e^{-2c^2 N} \leq \delta$$

error tolerance



Example: Stochastic Game



Objective
□ player: maximize P(◊1)
○ player: minimize P(◊1)

Problem Statement

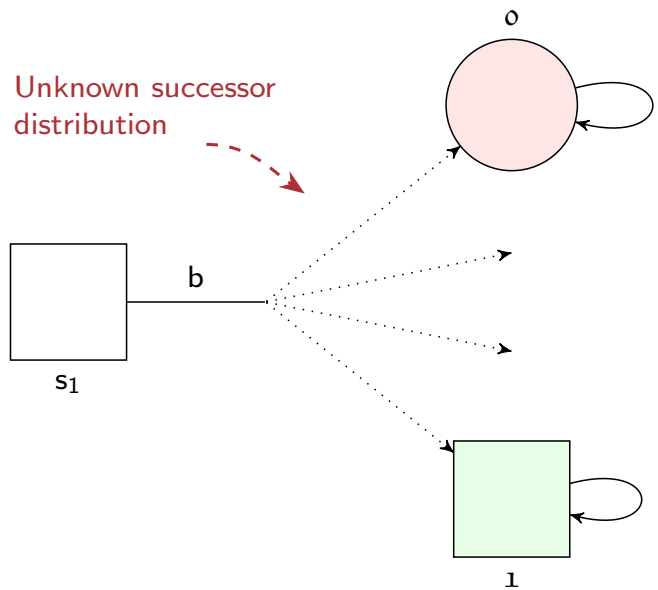
Given: 2 player limited information stochastic game (SG)

Find: max prob of reaching target over infinite horizon

Problem Statement

Given: 2 player limited information stochastic game (SG)

Find: max prob of reaching target over infinite horizon

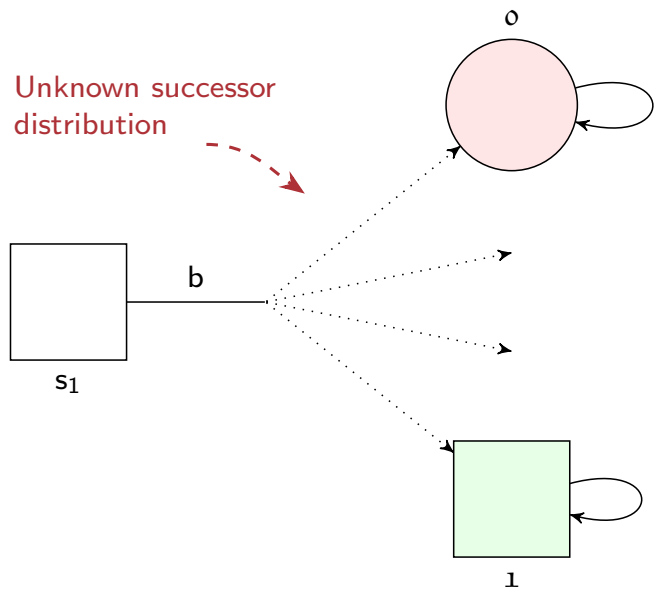


Problem Statement

Given: 2 player limited information stochastic game (SG)

Find: max prob of reaching target over infinite horizon

$$V(s_{init}) = \max_{\sigma} \min_{\tau} P^{\sigma,\tau}_{s_{init}}(\Diamond 1) = \min_{\tau} \max_{\sigma} P^{\sigma,\tau}_{s_{init}}(\Diamond 1)$$

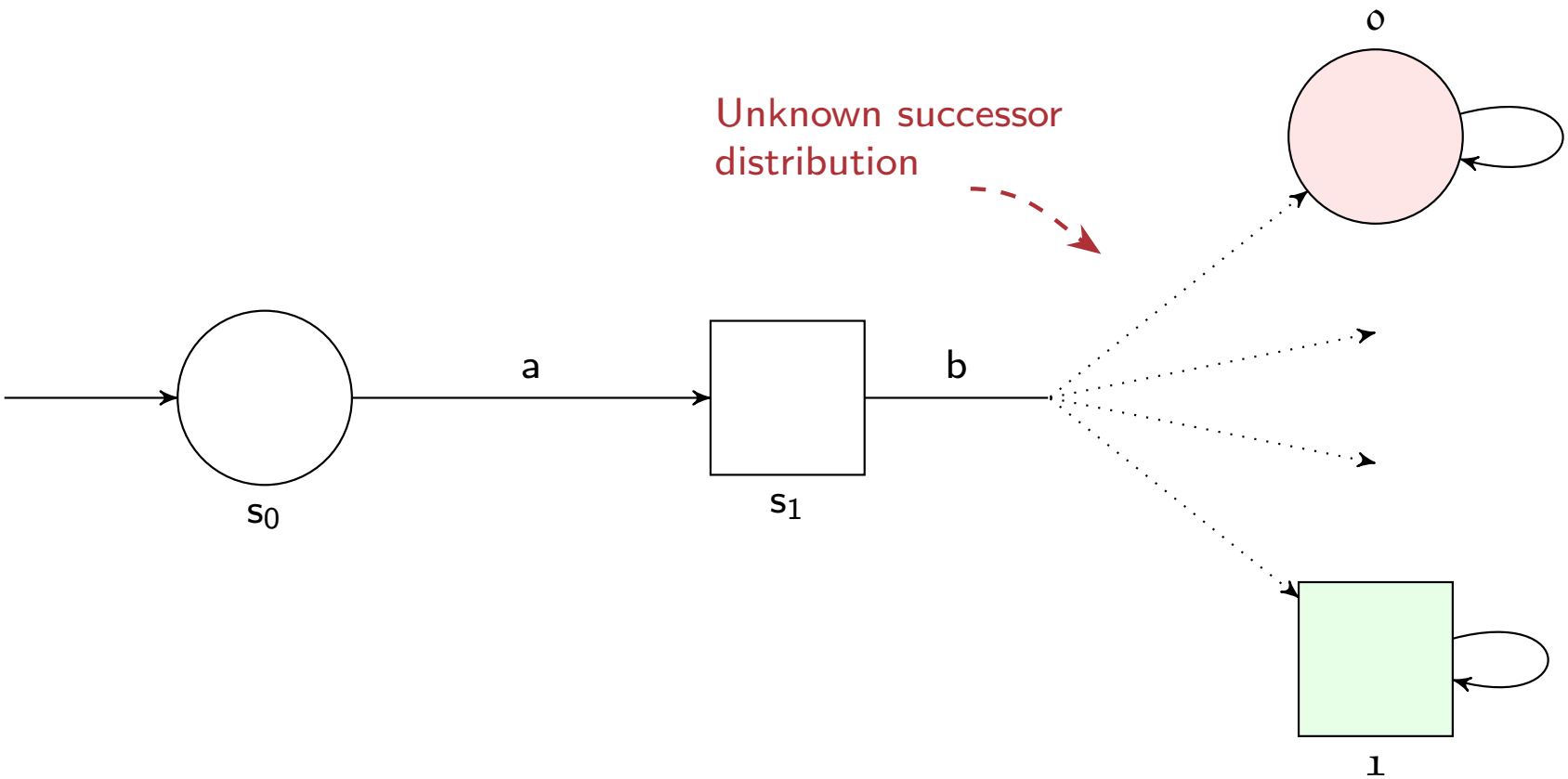


Solution Methods

Full information setting

- Quadratic programming
- Strategy iteration
- Value iteration

Challenge: Black-box (limited information setting)



Problem statement
Compute $V(s_0)$ for unbounded case with guarantees

Related work

- Statistical Model Checking for limited information settings
- Successful in models without non-determinism
- Some work on MDPs, but not much in Stochastic Games

Related work

- Statistical Model Checking for limited information settings
- Successful in models without non-determinism
- Some work on MDPs, but not much in Stochastic Games
  - 2011: Bogdoll et al., **spurious non-det.**, unbounded
  - 2012: Henriques et al., **bounded** LTL
  - 2013: Legay et al., strategy sampling, **bounded**
  - 2014: Fu et al., PAC guarantee, **require mixing time**
  - 2014: Brazdil et al., PAC, **theoretical**

Idea

Progression

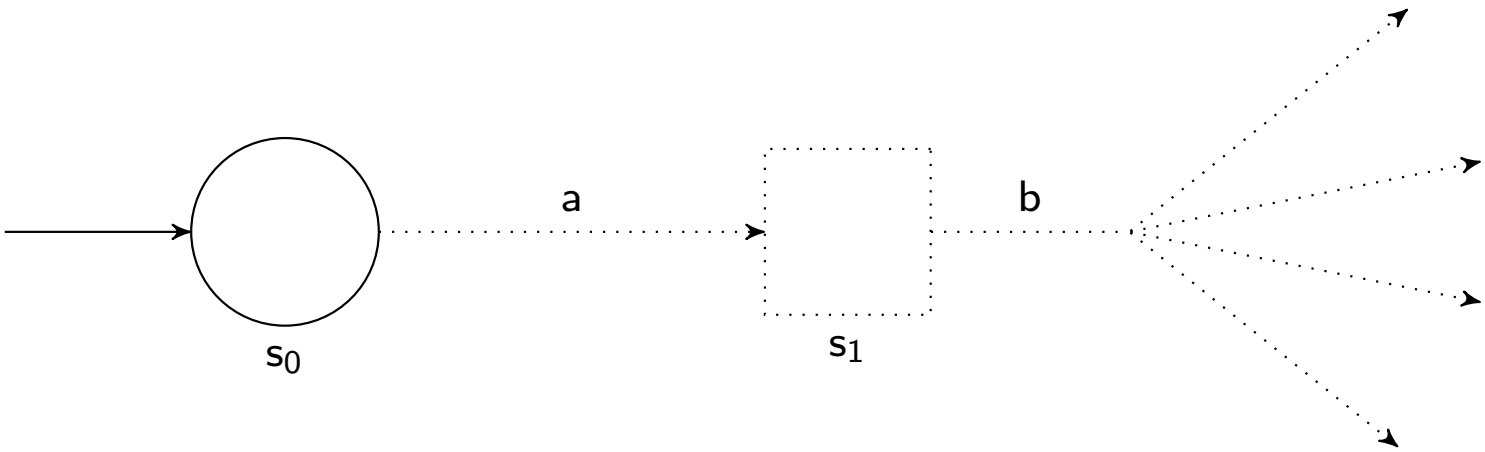
1. ATVA14: MDPs, full and partial information, unbounded
2. CAV18: SGs, full information, unbounded
3. This work: SGs, partial information, unbounded

The Algorithm

while $U - L$ is large

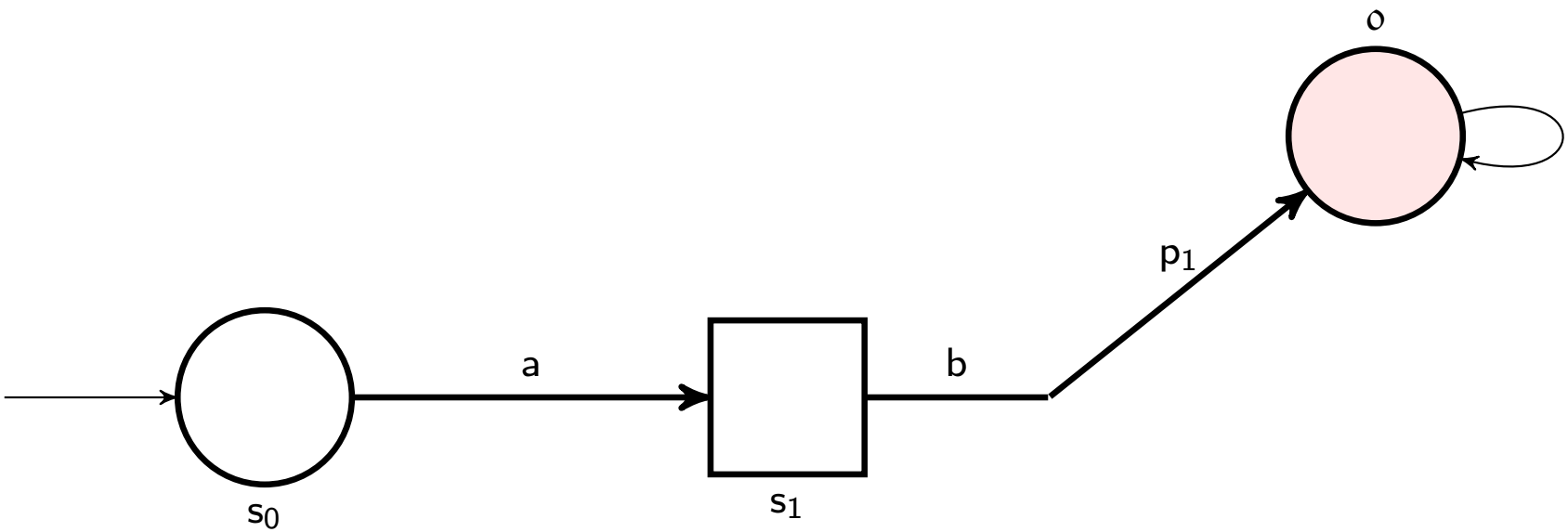
1. Simulate and estimate
2. Back-propagate

Algorithm: simulate



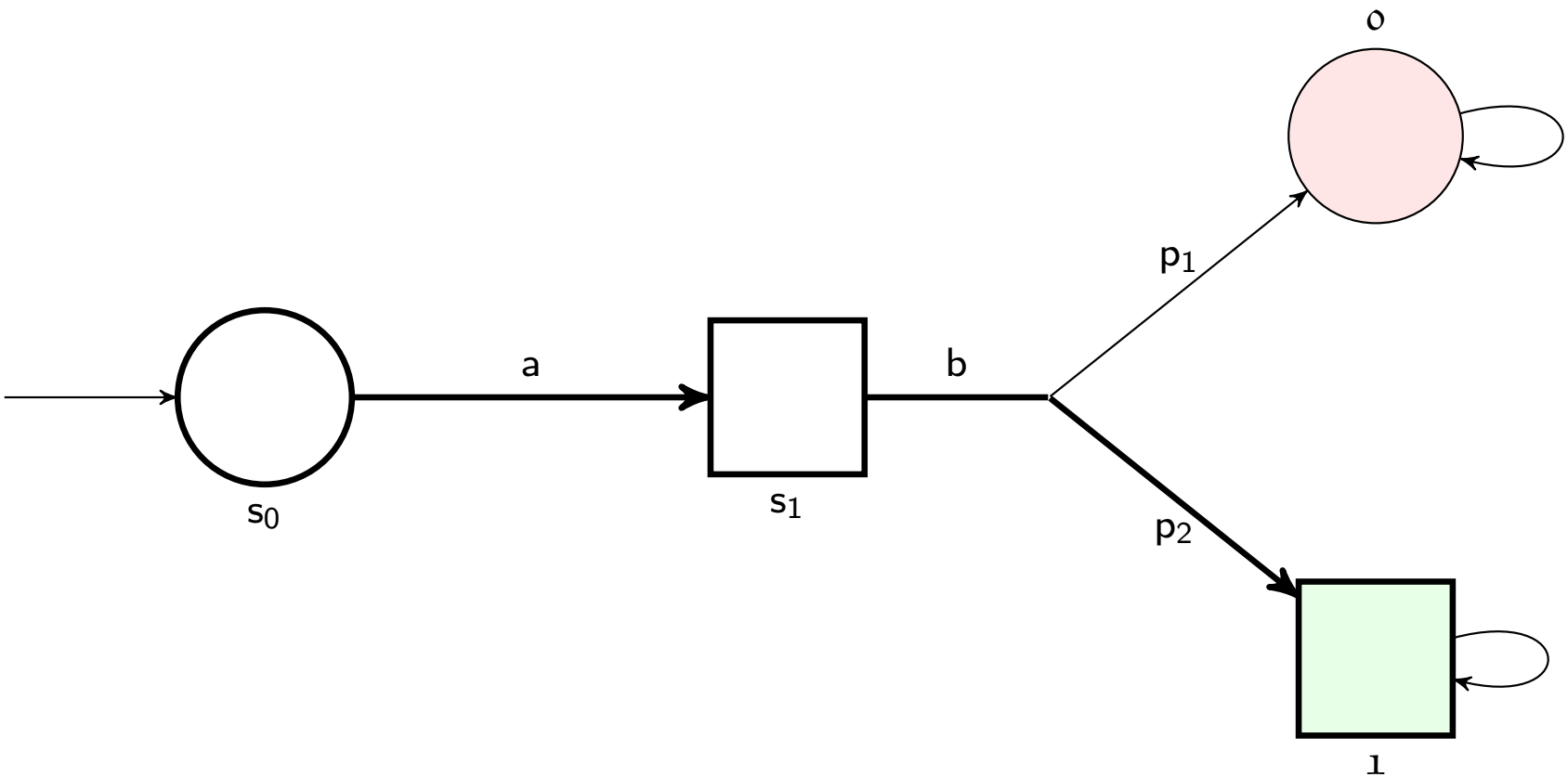
Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Algorithm: simulate



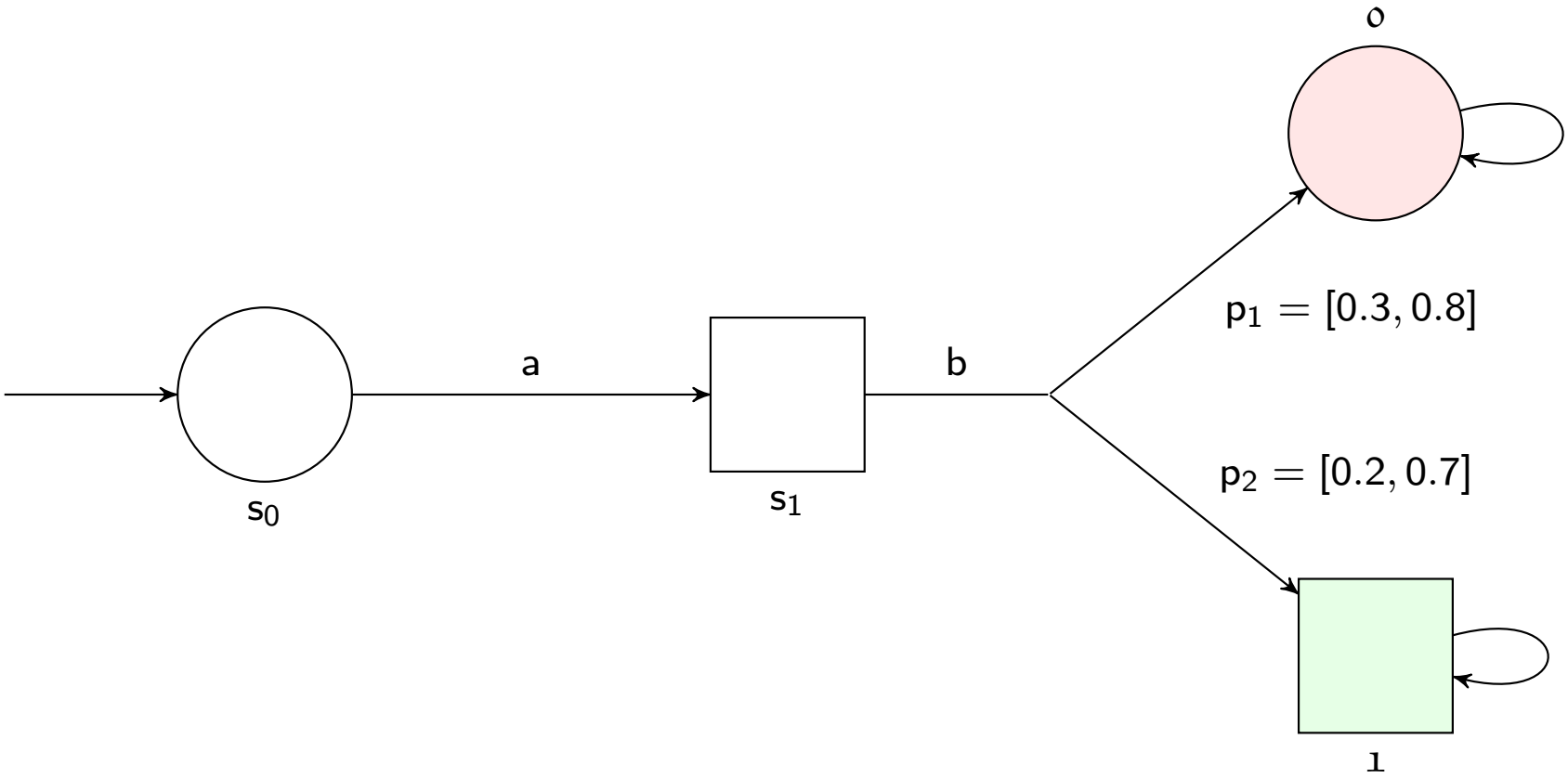
Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Algorithm: simulate



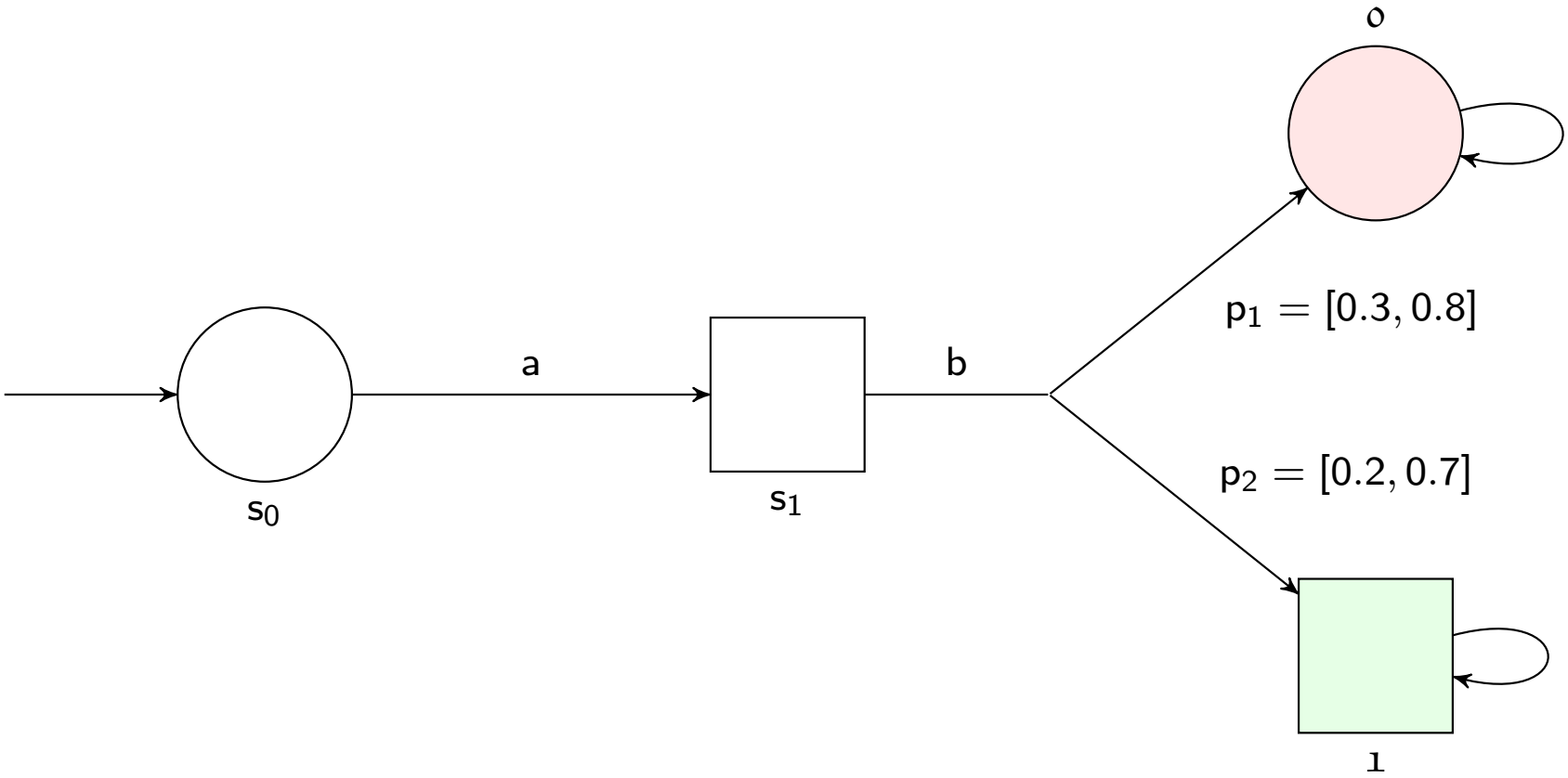
Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Algorithm: simulate



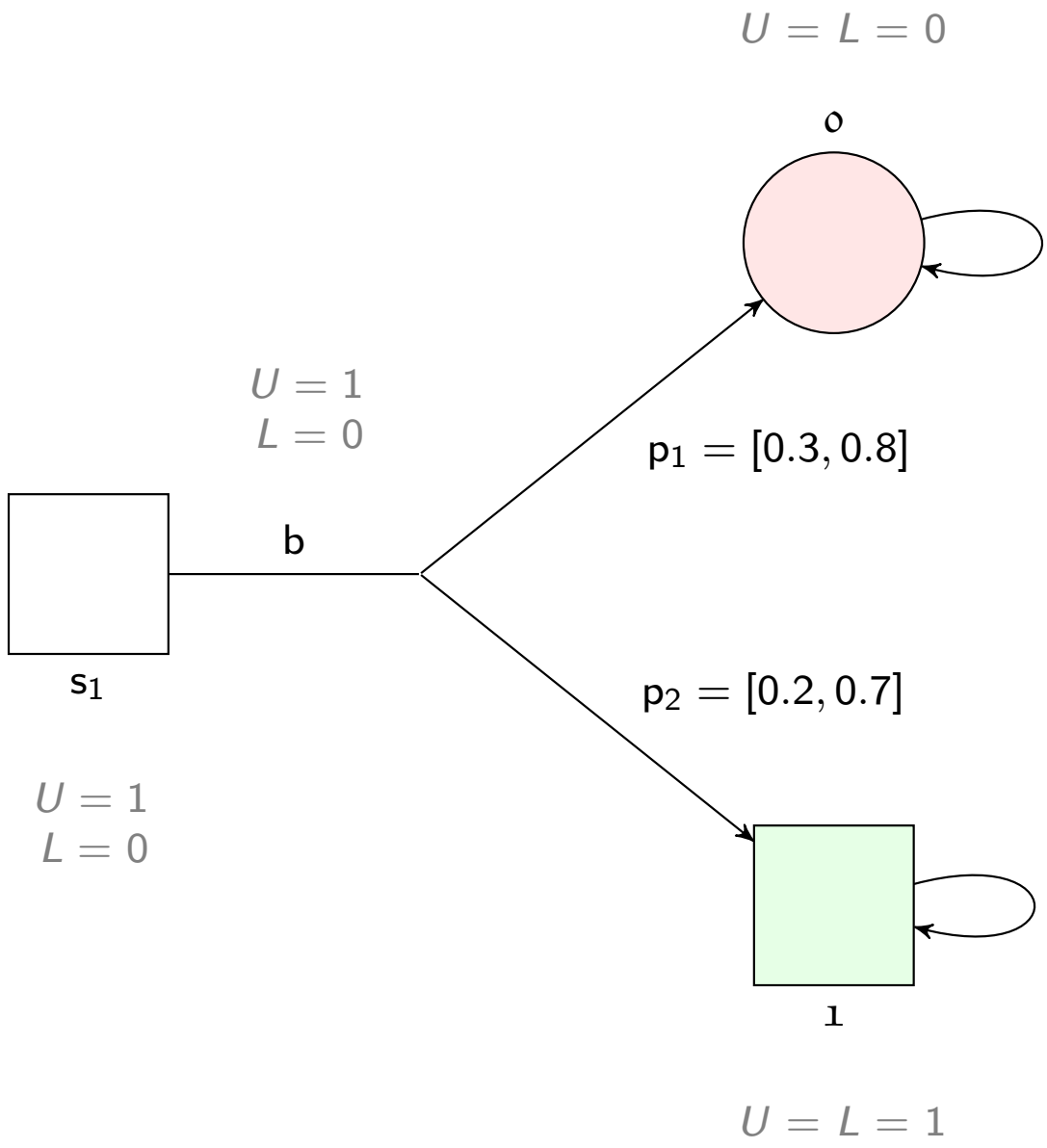
Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Algorithm: simulate



Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

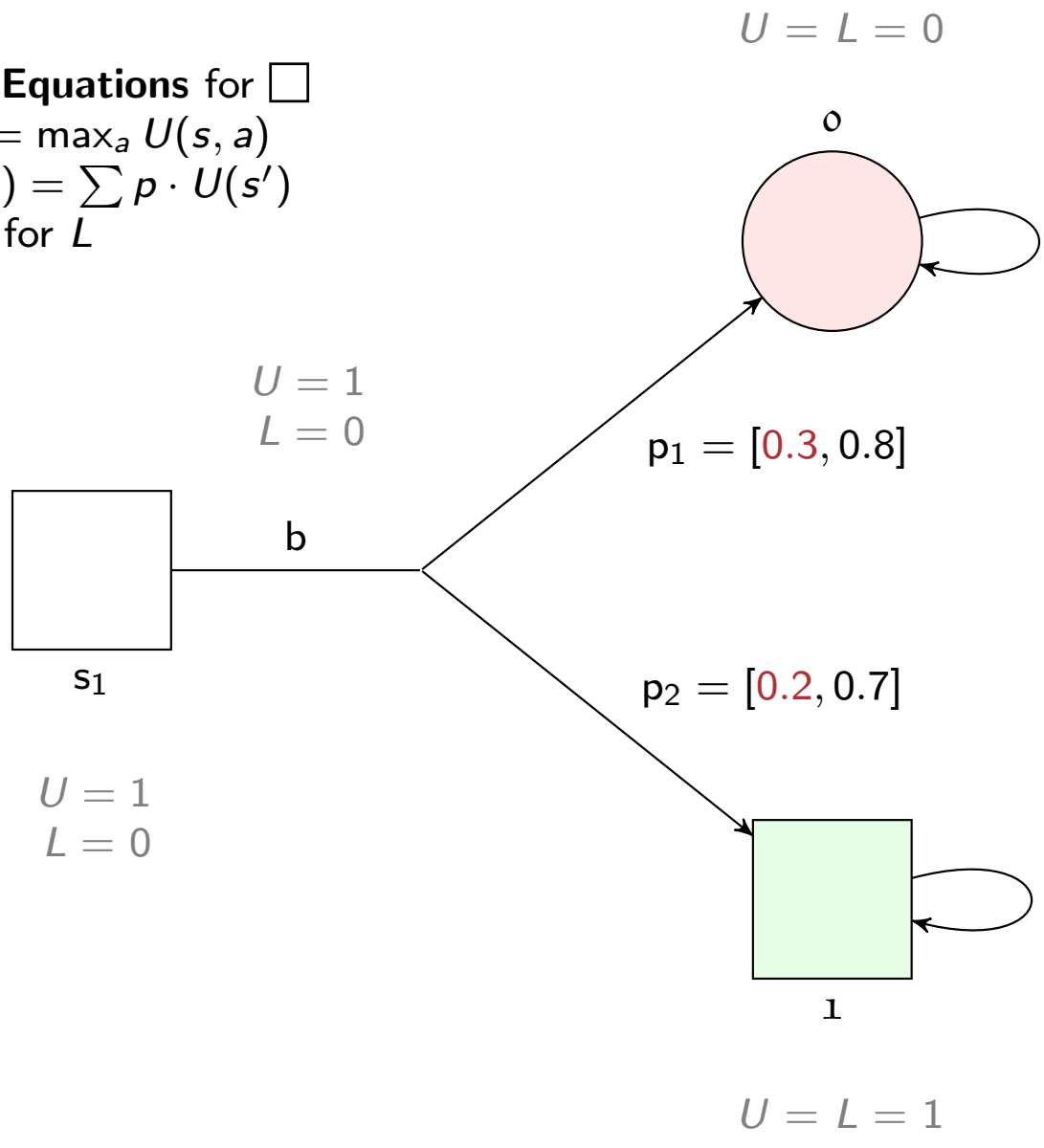
Algorithm: back-propagate



Algorithm: back-propagate

Bellman Equations for $\square$
1. $U(s) = \max_a U(s, a)$
2. $U(s, a) = \sum p \cdot U(s')$

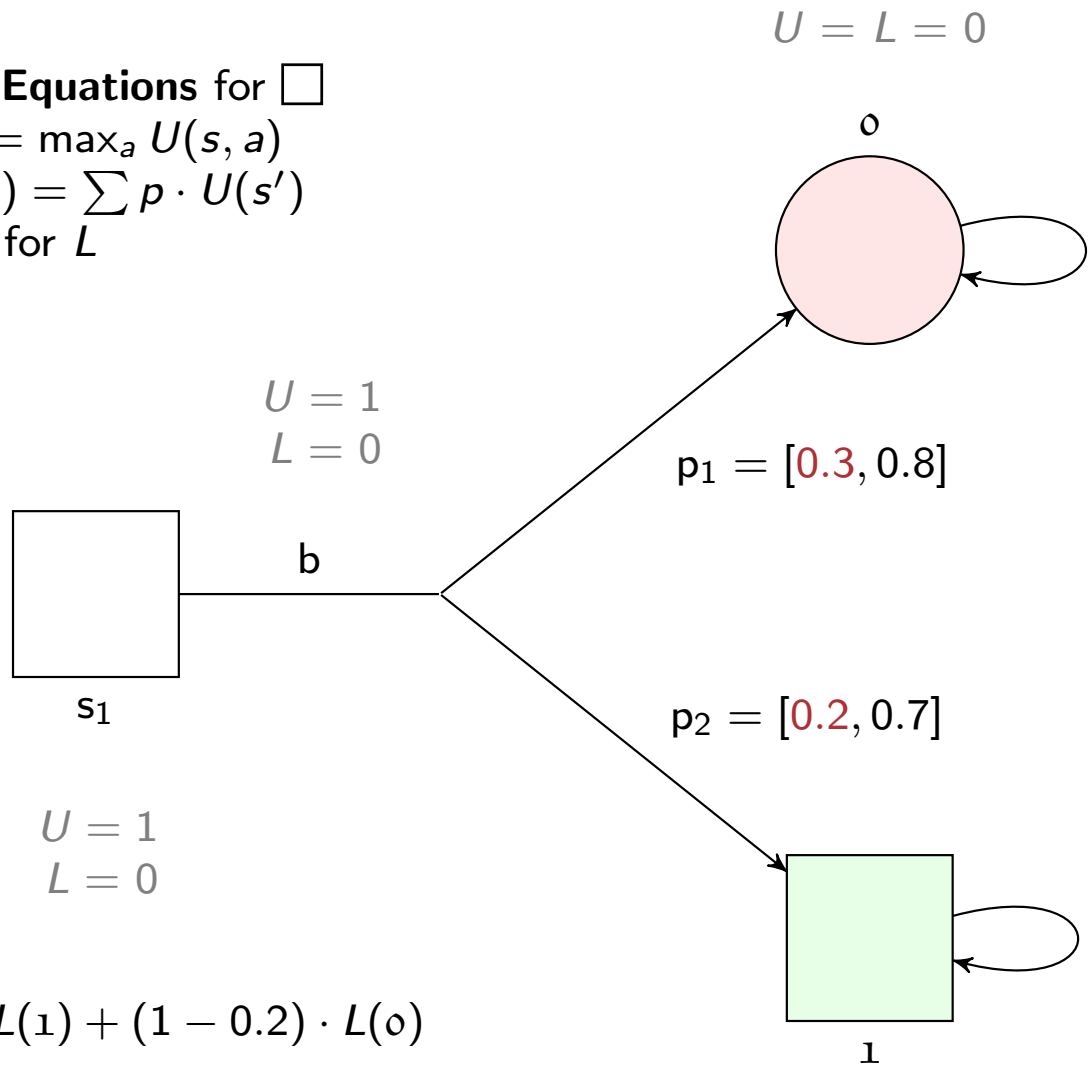
Similarly for L



Algorithm: back-propagate

Bellman Equations for $\square$
1. $U(s) = \max_a U(s, a)$
2. $U(s, a) = \sum p \cdot U(s')$

Similarly for L

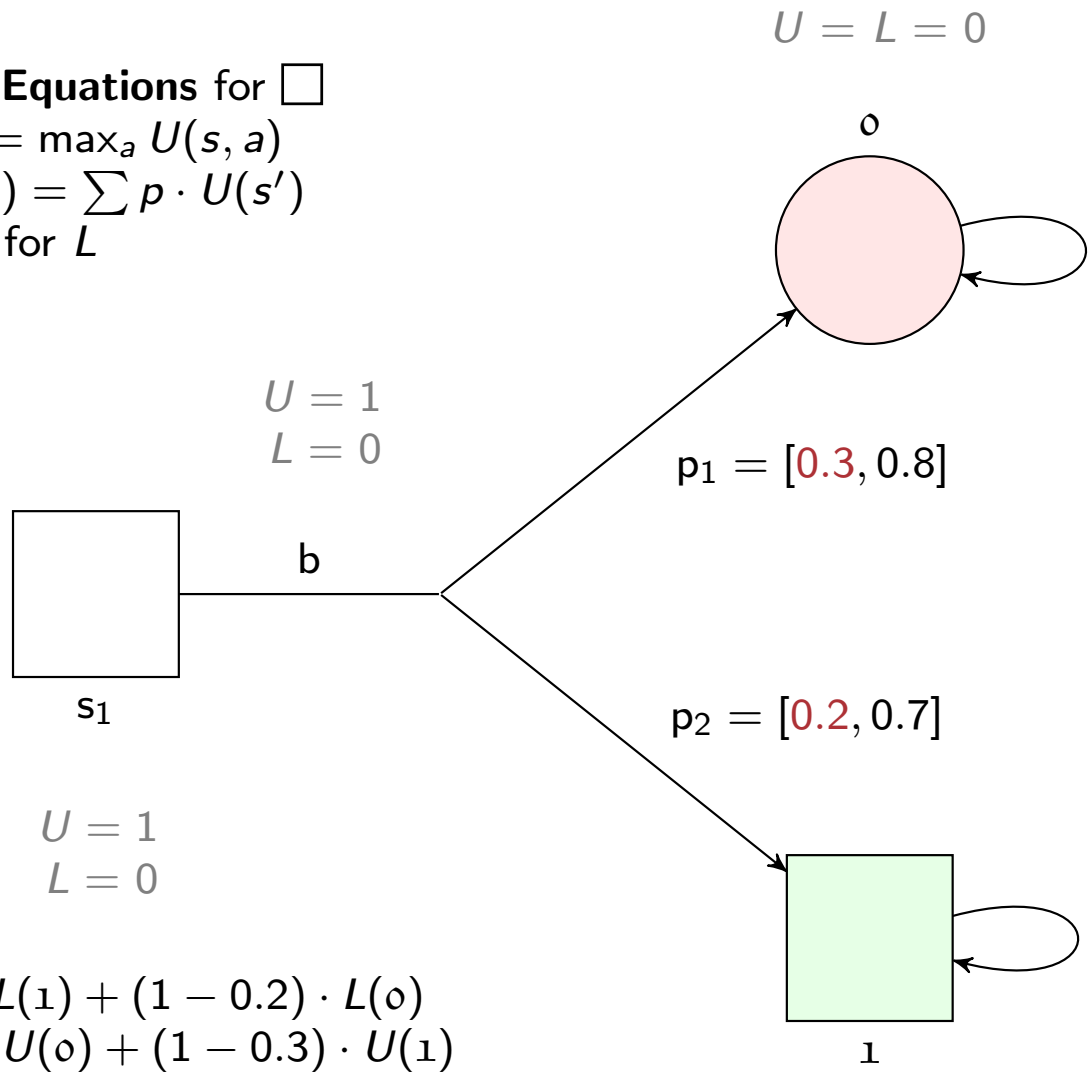


$$L(s, b) = 0.2 \cdot L(1) + (1 - 0.2) \cdot L(0)$$

Algorithm: back-propagate

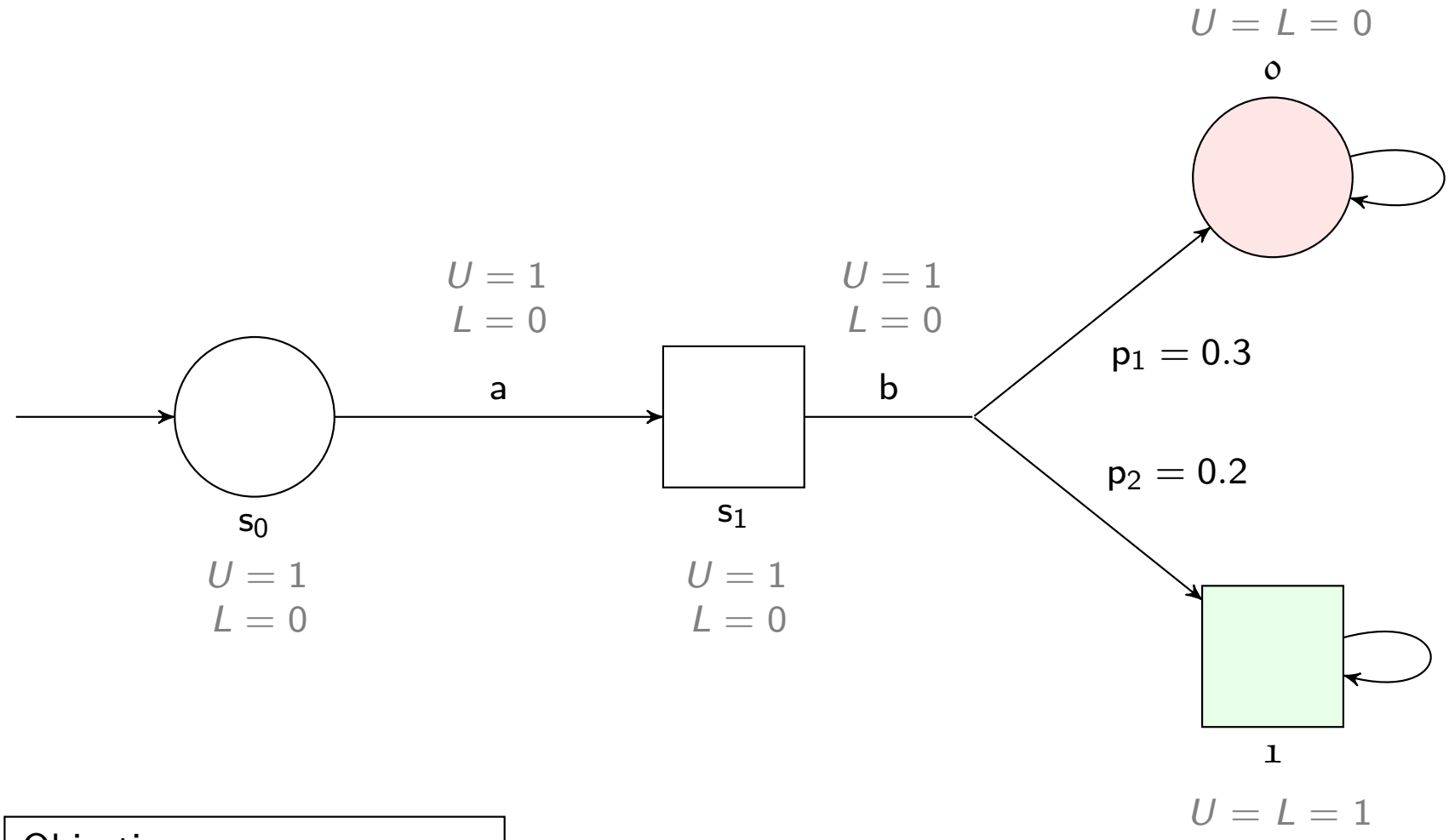
Bellman Equations for $\square$
1. $U(s) = \max_a U(s, a)$
2. $U(s, a) = \sum p \cdot U(s')$

Similarly for L



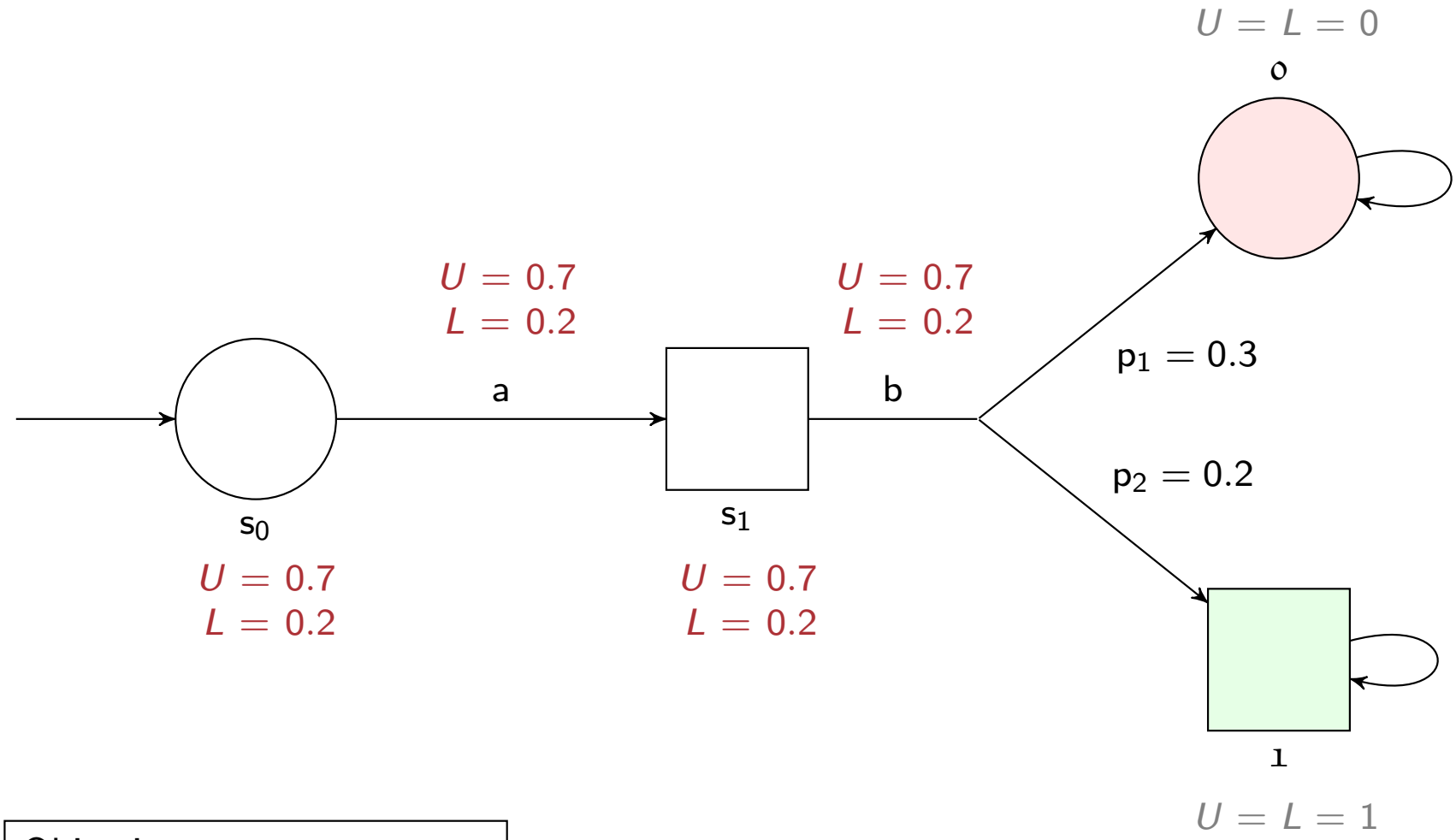
$$L(s, b) = 0.2 \cdot L(1) + (1 - 0.2) \cdot L(0)$$
$$U(s, b) = 0.3 \cdot U(0) + (1 - 0.3) \cdot U(1)$$

Algorithm: back-propagate



Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Algorithm: back-propagate



Objective
□ player: maximize P(F 1)
○ player: minimize P(F 1)

Infinite horizon and end-components

Sink states - end-components which cannot reach target

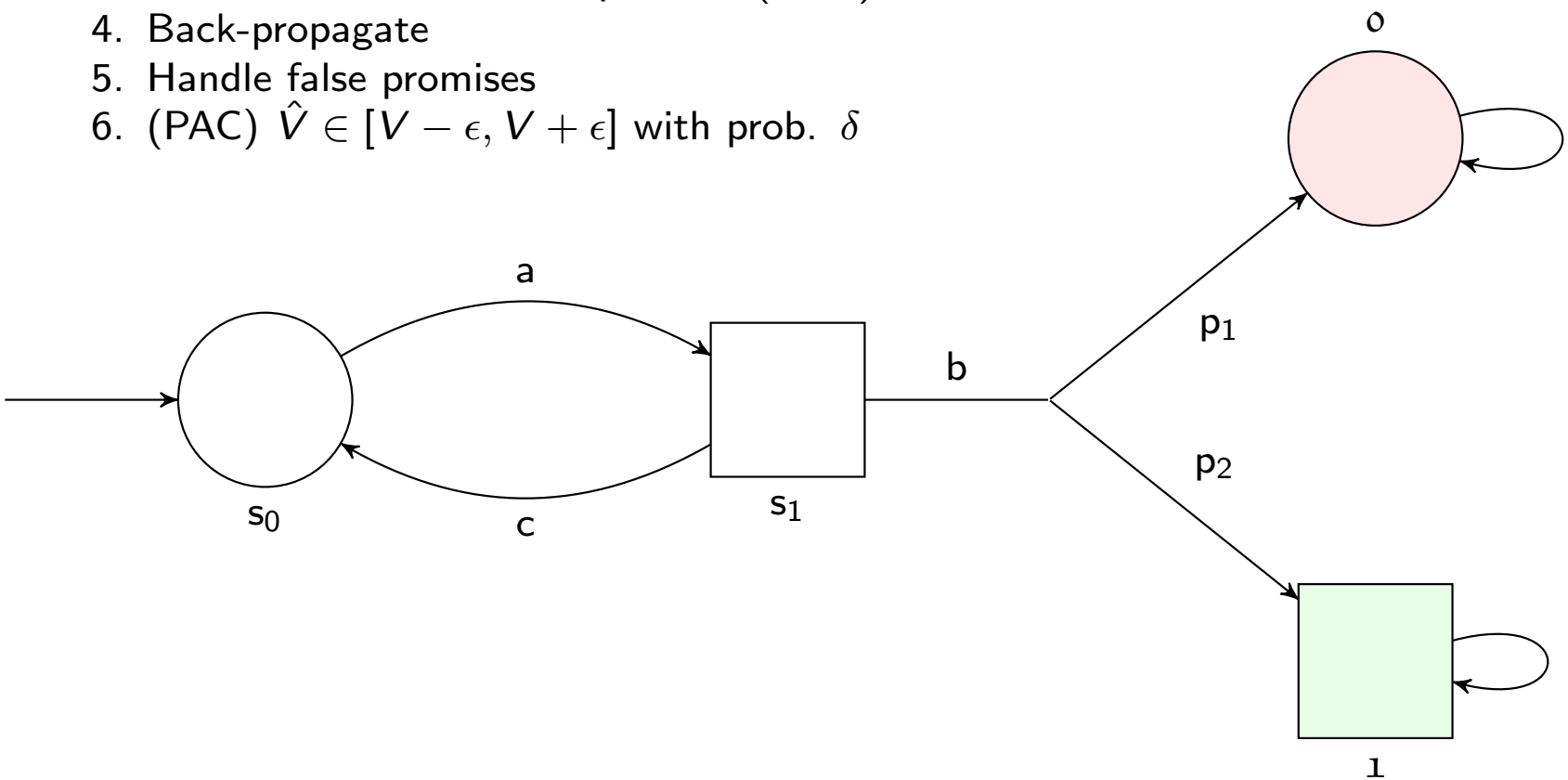
- Lift idea of Daca et. al. 2016 from SCC to EC

End-components promising false numbers to each other

- Adapt ‘deflate’ idea of Kelemendi et. al. 2018

Algorithm Summary

1. Simulate (action with best U)
2. Estimate prob (coin toss idea)
3. Realize stuck in end-components (sinks)
4. Back-propagate
5. Handle false promises
6. (PAC) $\hat{V} \in [V - \epsilon, V + \epsilon]$ with prob. δ



Experiments Summary

- Closest competitor (ATVA14): $\geq$ age of universe for 10 states

[1] only actions oracle

[2] number of successors known

Experiments Summary

- Closest competitor (ATVA14): $\geq$ age of universe for 10 states
- 16 model-property pairs (11 MDP, 5 SG)
- $\delta = 10^{-8}$, timeout = 30 minutes

[1] only actions oracle

[2] number of successors known

Experiments Summary

- Closest competitor (ATVA14): $\geq$ age of universe for 10 states
- 16 model-property pairs (11 MDP, 5 SG)
- $\delta = 10^{-8}$, timeout = 30 minutes
- Limited-information (black-box)[1]: $\varepsilon \leq 0.1$ on 6 benchmarks
- 5 benchmarks were 'hard' (no result)

[1] only actions oracle

[2] number of successors known

Experiments Summary

- Closest competitor (ATVA14): $\geq$ age of universe for 10 states
- 16 model-property pairs (11 MDP, 5 SG)
- $\delta = 10^{-8}$, timeout = 30 minutes
- Limited-information (black-box)[1]: $\varepsilon \leq 0.1$ on 6 benchmarks
- 5 benchmarks were 'hard' (no result)
- Grey-box[2]: $\varepsilon \leq 0.1$ on 14 benchmarks within 6 mins

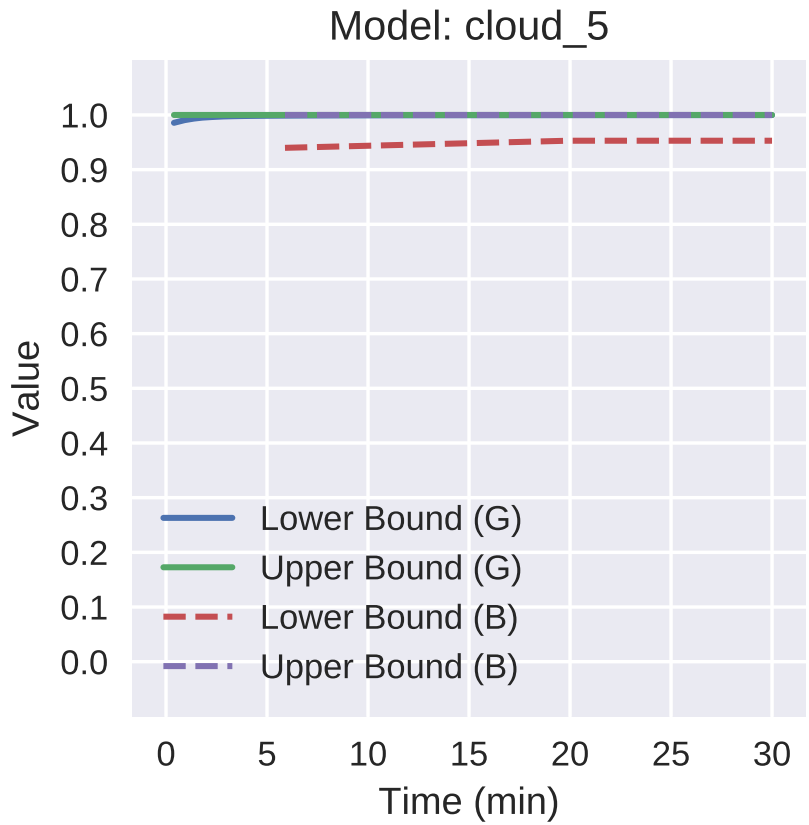
[1] only actions oracle

[2] number of successors known

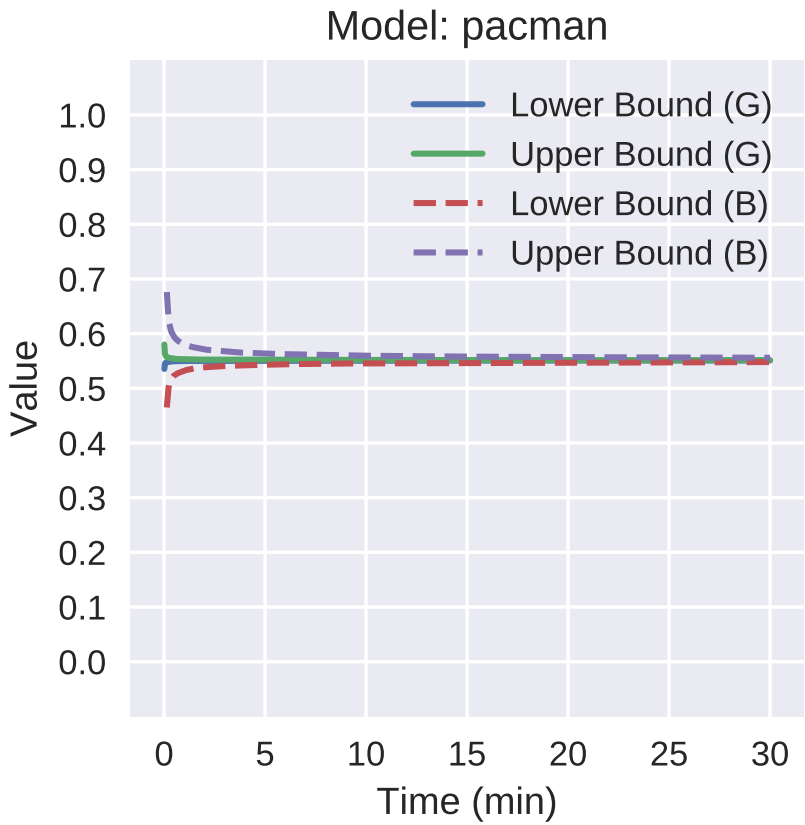
Conclusion

- PAC algorithm for reachability in limited information MDP/SG
- First algorithm to do so for SG
- First practical algorithm for MDPs

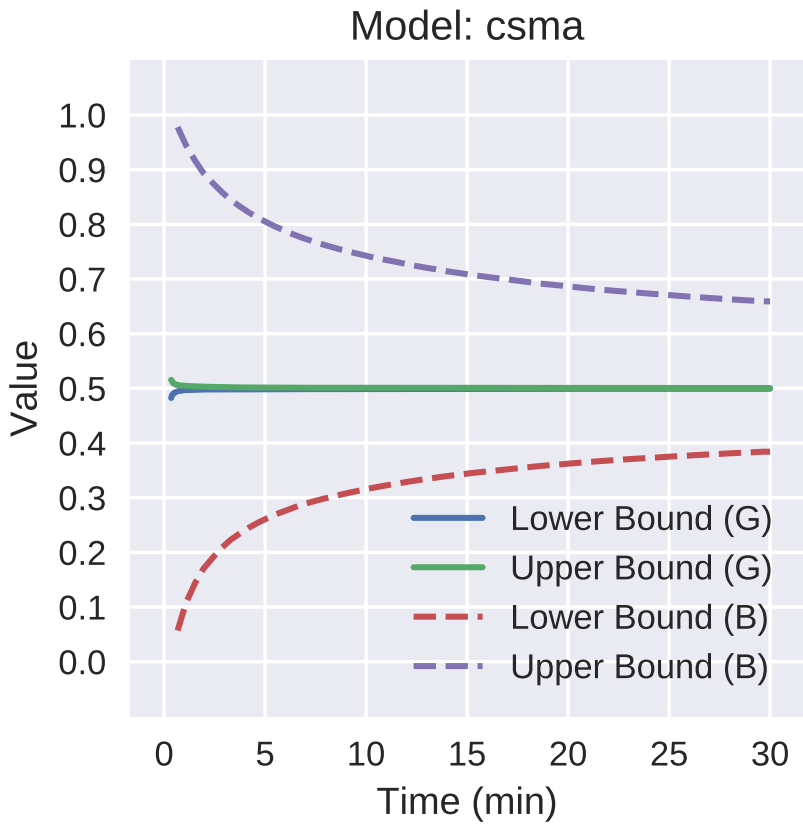
Experiments



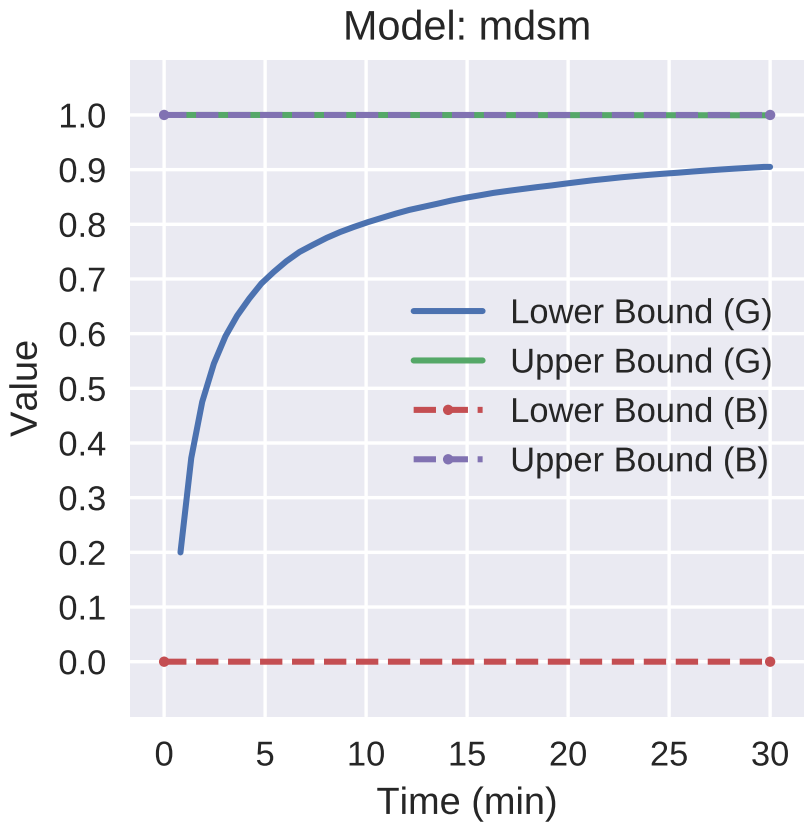
Experiments



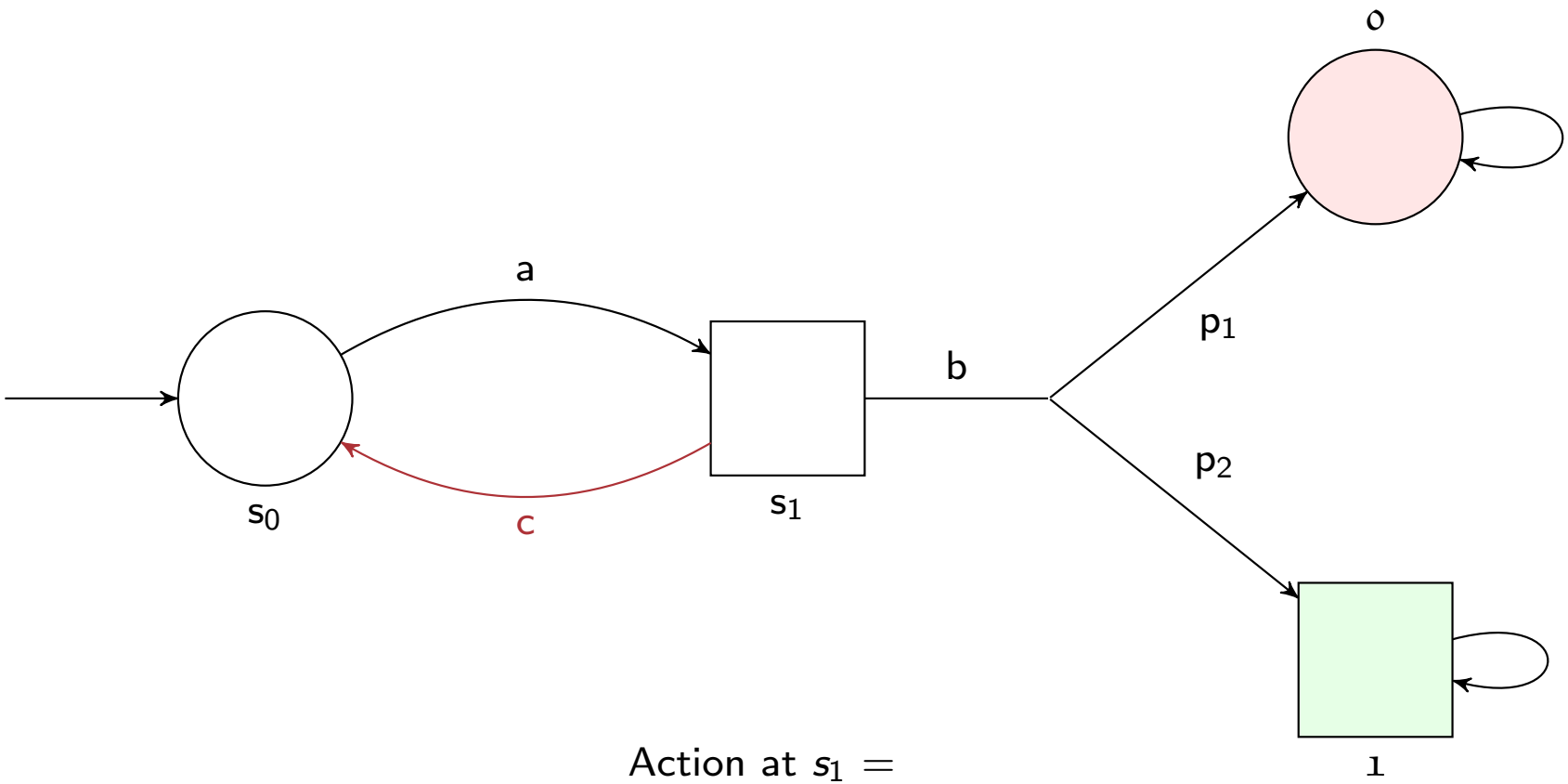
Experiments



Experiments

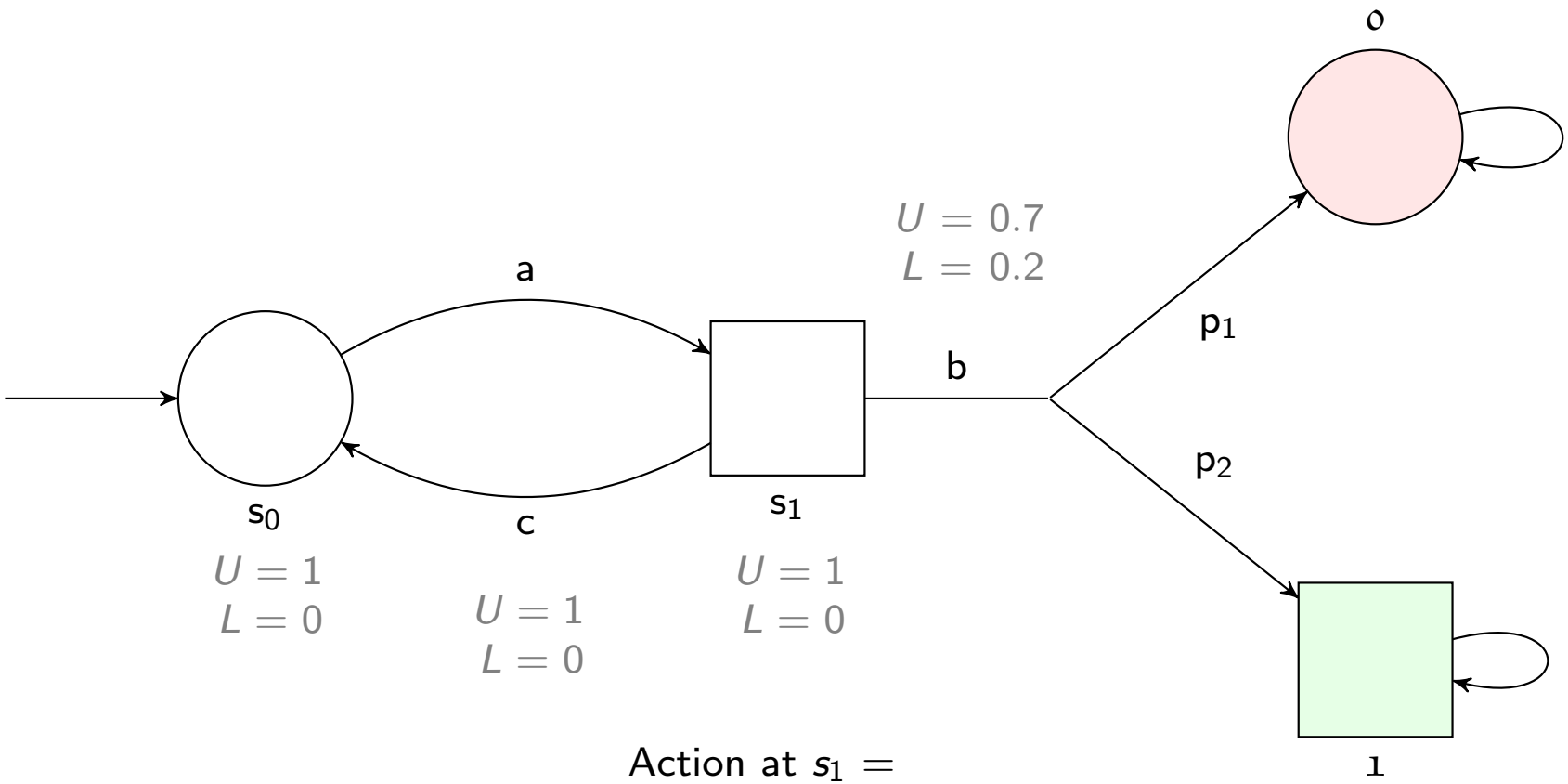


Handling non-determinism



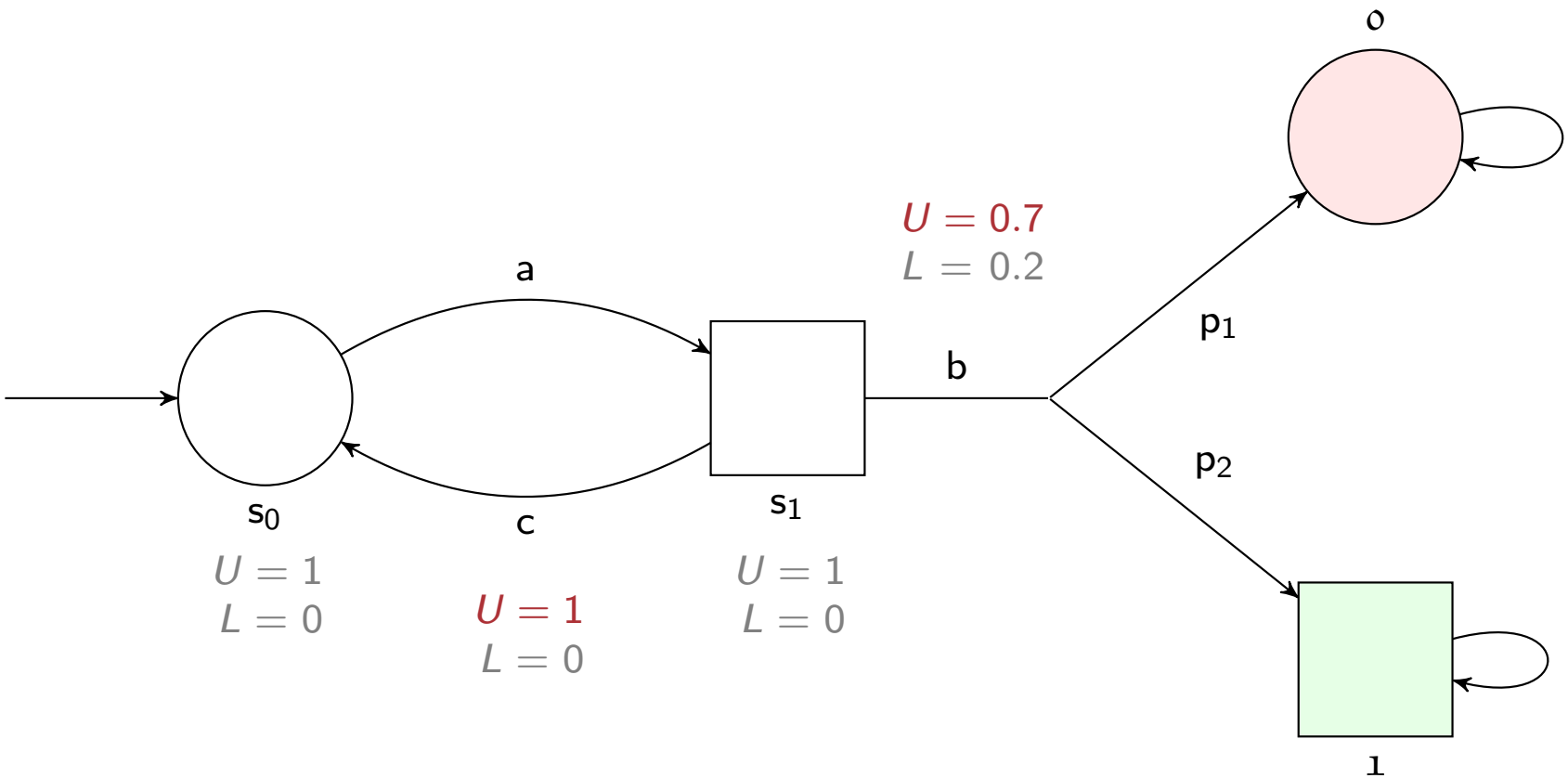
Action at s_1 =
$\arg\max_b U(s_1, b)$

Handling non-determinism

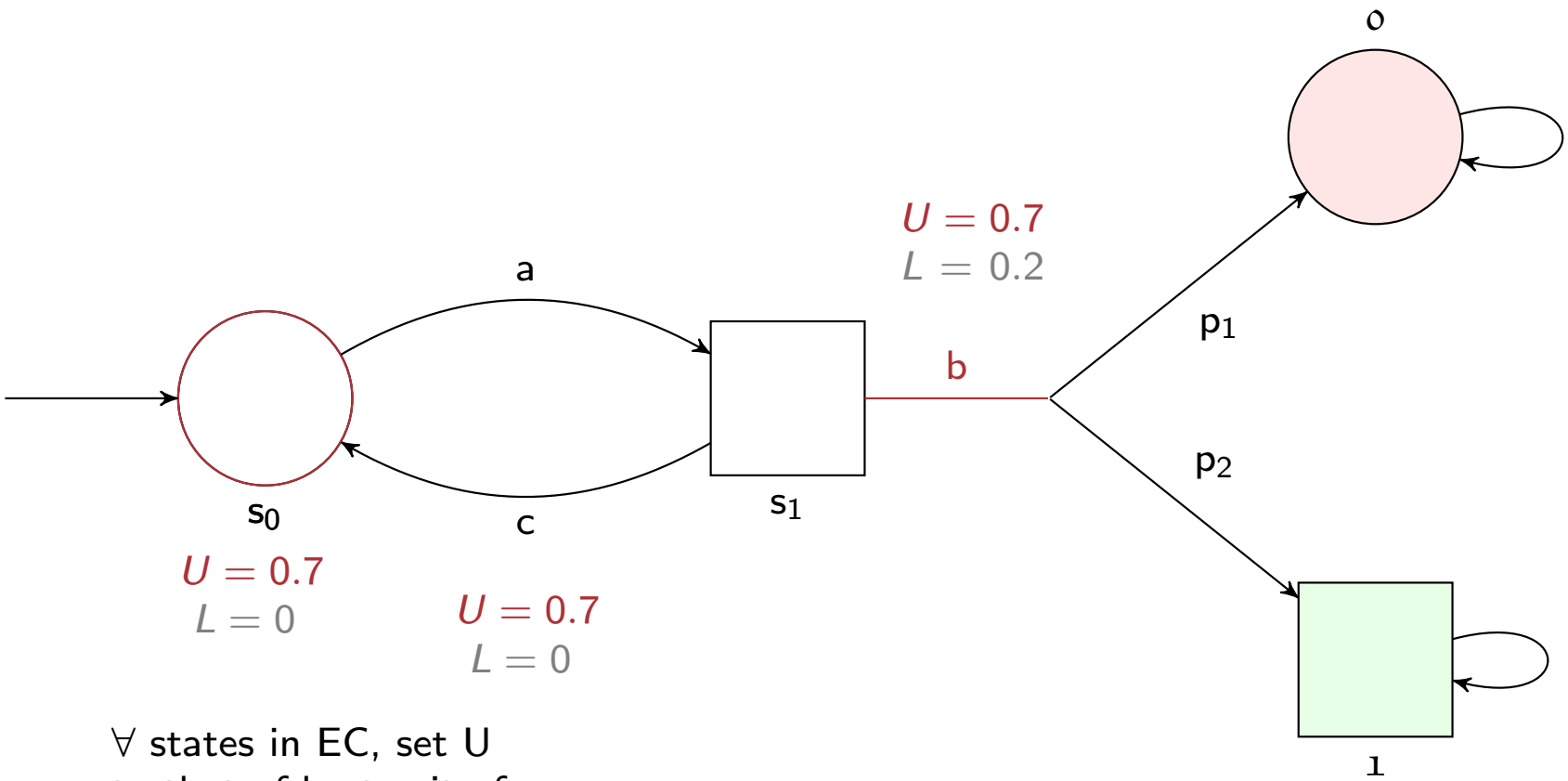


Action at s_1 =
$\arg\max_b U(s_1, b)$

Handling non-determinism

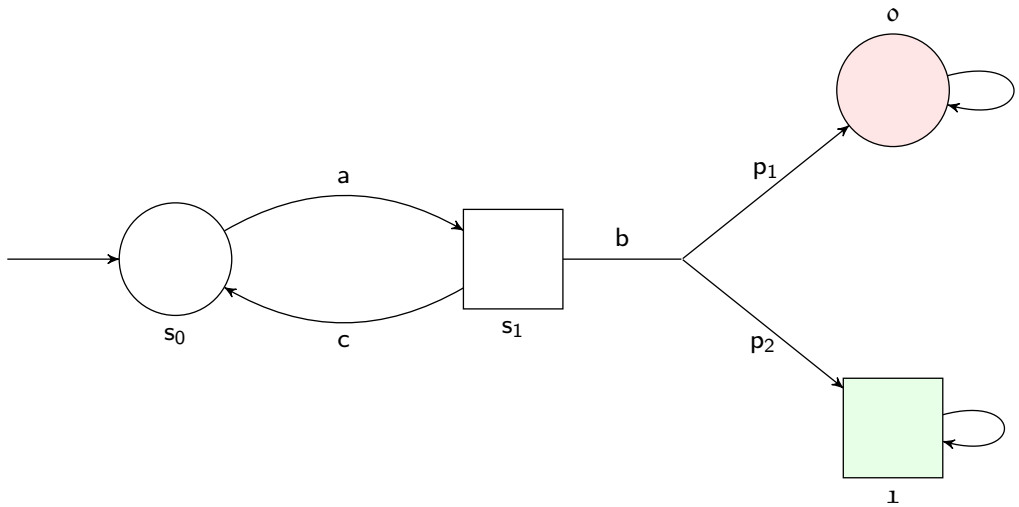


Handling non-determinism



$\forall$ states in EC, set U to that of best exit of maximizer

Handling infinite horizon



Identifying sink states

- 2016: Daca et. al. - fast SCC detection
- Extend to EC detection