Sistemi Intelligenti
Corso di Laurea in Informatica, A.A. 2017-2018
Università degli Studi di Milano

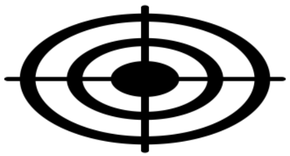
Discrete planning (an introduction)

Nicola Basilico
Dipartimento di Informatica
Via Comelico 39/41 - 20135 Milano (MI)
Ufficio S242
nicola.basilico@unimi.it
+39 02.503.16294

Sito per queste lezioni

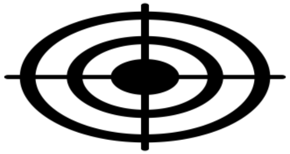
Planning under uncertainty

- Action selection is often affected by uncertainty
- Example:



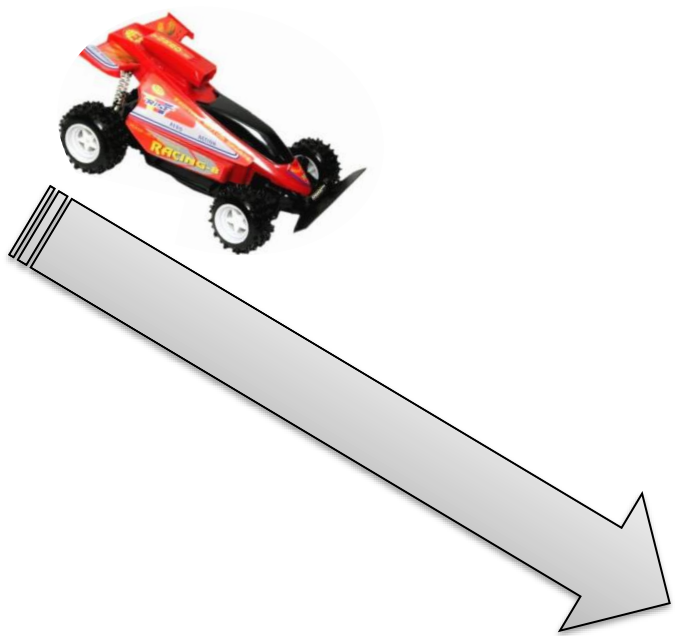
Planning under uncertainty

- Action selection is often affected by uncertainty
- Example:

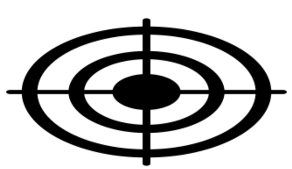


Planning under uncertainty

- Action selection is often affected by uncertainty
- Example:



Fast forward for 2 seconds



Planning under uncertainty

- Action selection is often affected by uncertainty
- Example:

Fast forward for 2 seconds

Planning under uncertainty

Are the effects of my actions perfectly predictable?

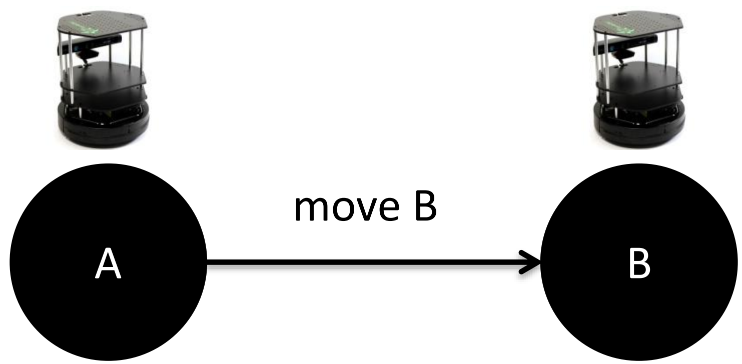
Deterministic
vs
Stochastic
transitions

Am I always sure about what's going on?

Fully observable
vs
Partially observable
states

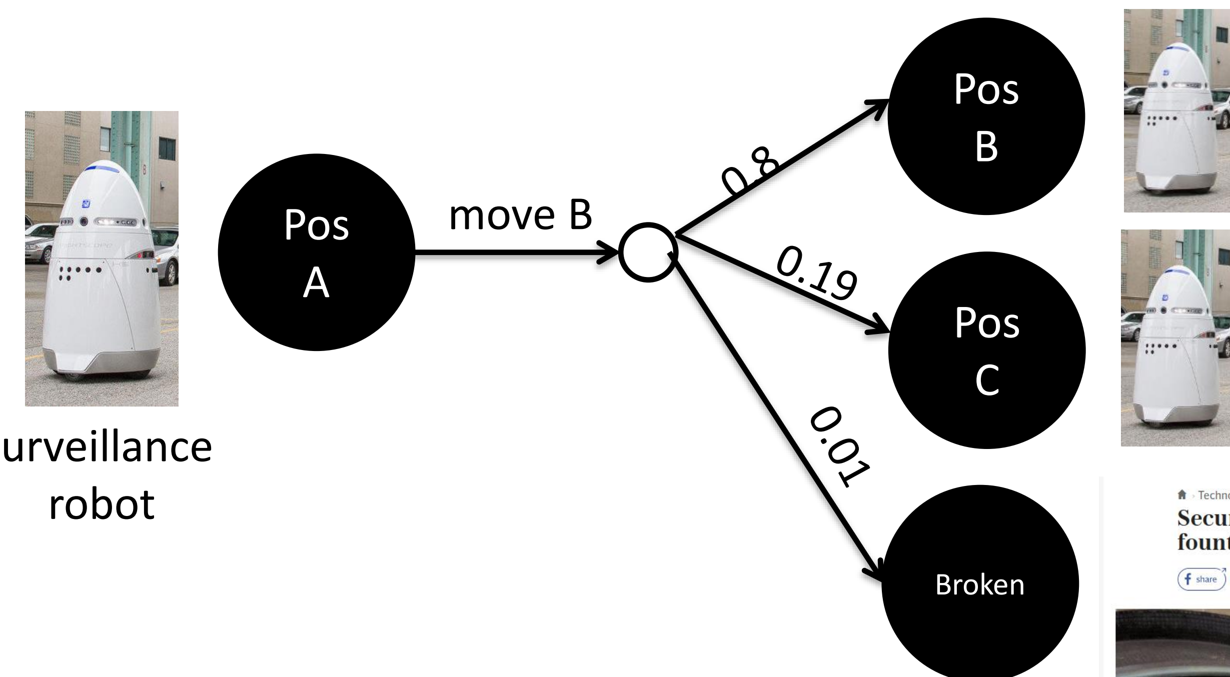
Examples

- Deterministic transitions, fully observable states
- Only actuation is needed, no sensing!

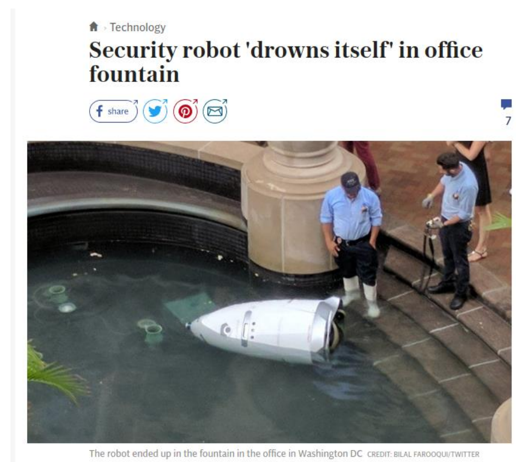


Examples

- Stochastic transitions, fully observable states



Surveillance
robot



Technology

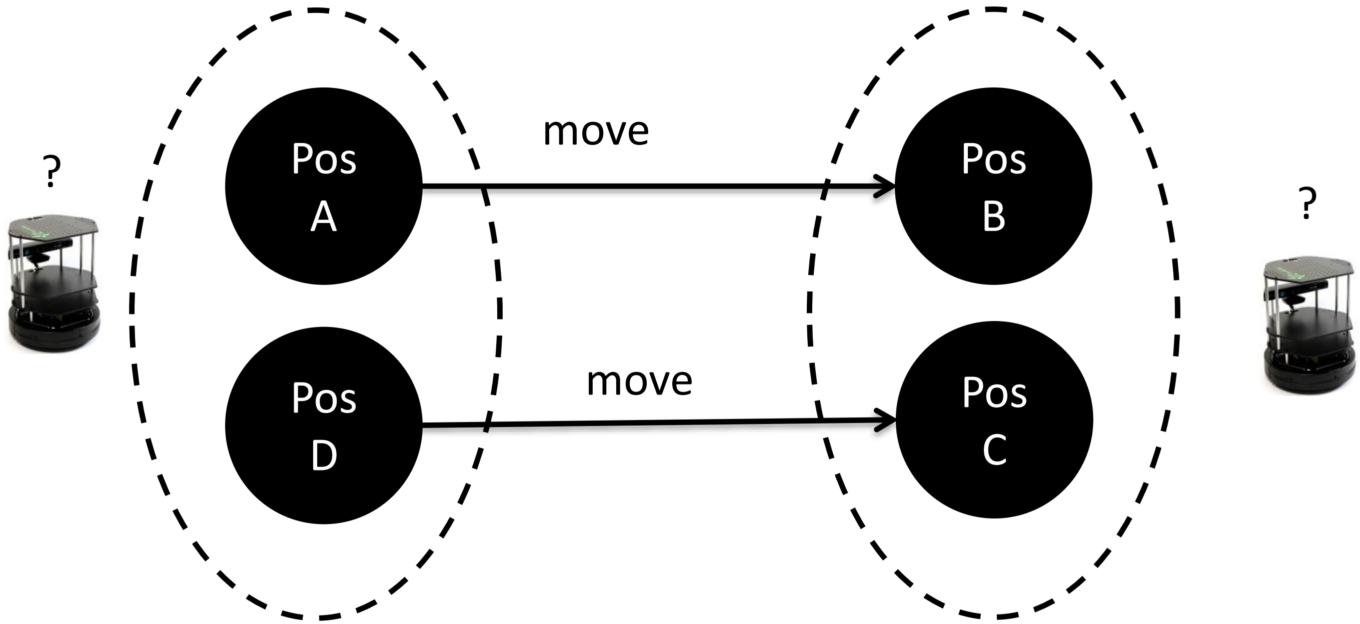
Security robot 'drowns itself' in office fountain

share

7

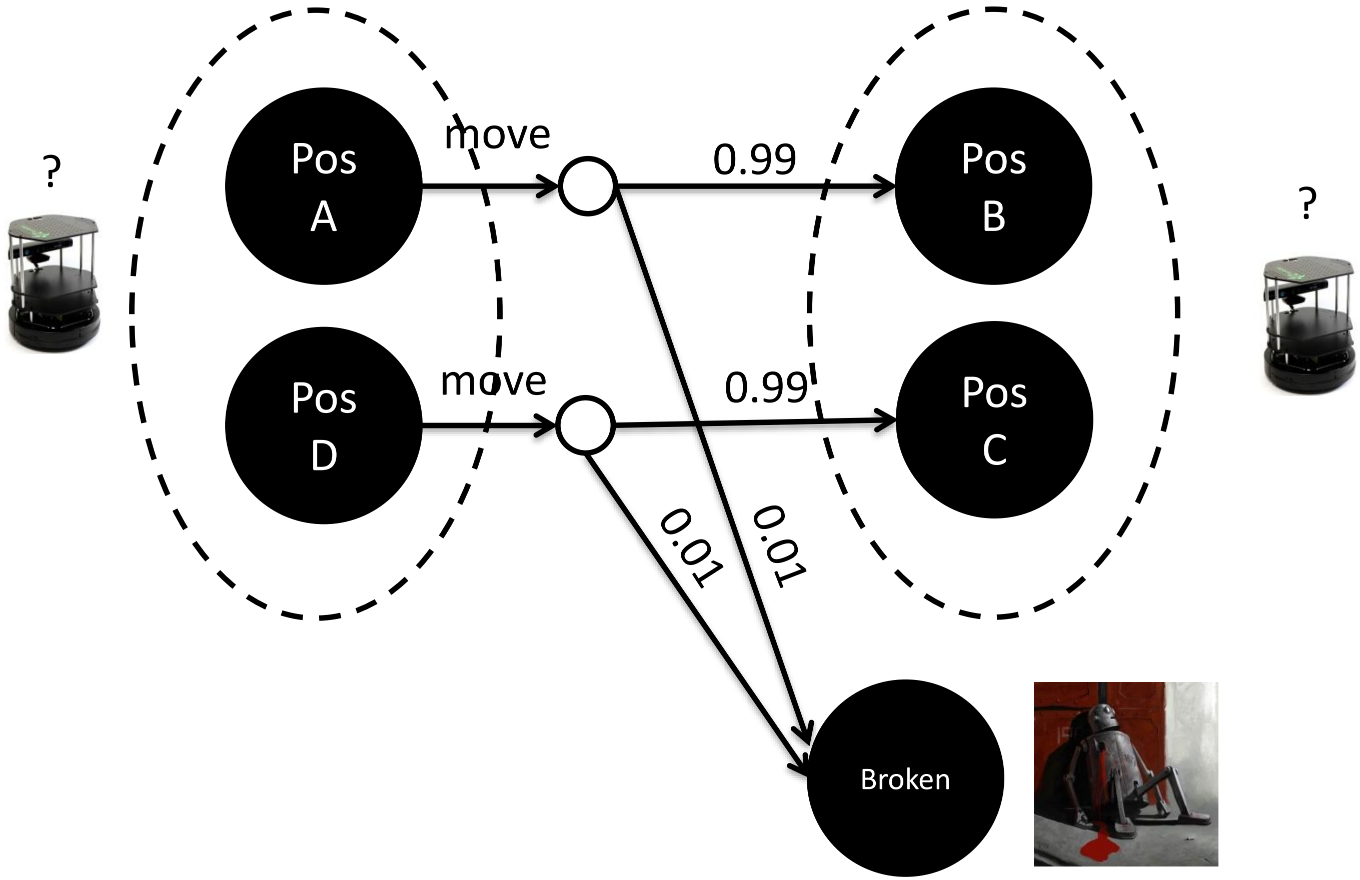
The robot ended up in the fountain in the office in Washington DC CREDIT: BILAL FAROOQUI/TWITTER

Examples



Deterministic transitions, partially observable states

Examples



Stochastic transitions, partially observable states

Markov Decision Processes (MDP)

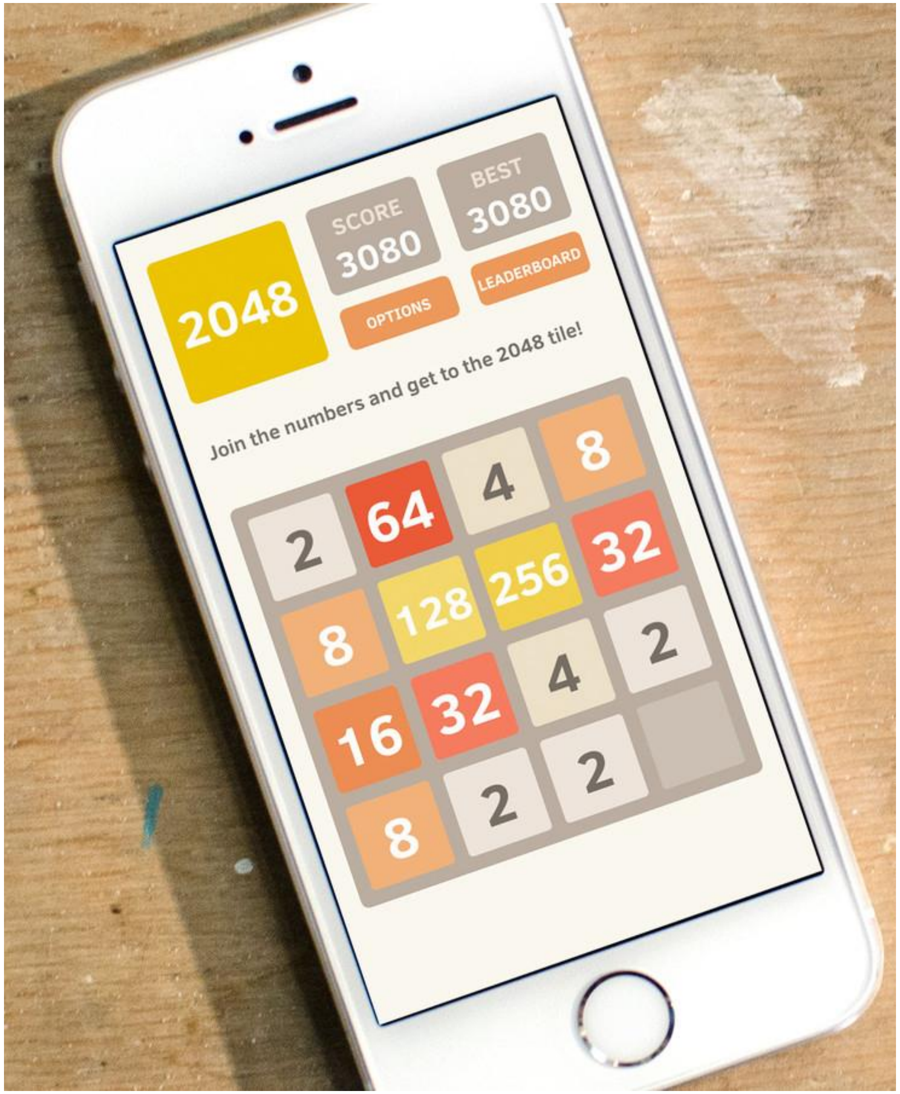
- We assume full observability of states, but non-deterministic actions
- We cannot specify a transition function like before, instead we give a set of transition probabilities

$$P(s'|s,a)$$

Probability of reaching state s', given that current state is s and action a is taken

- State transitions satisfy the **Markov property**: they depend only on the current state and not on states visited before

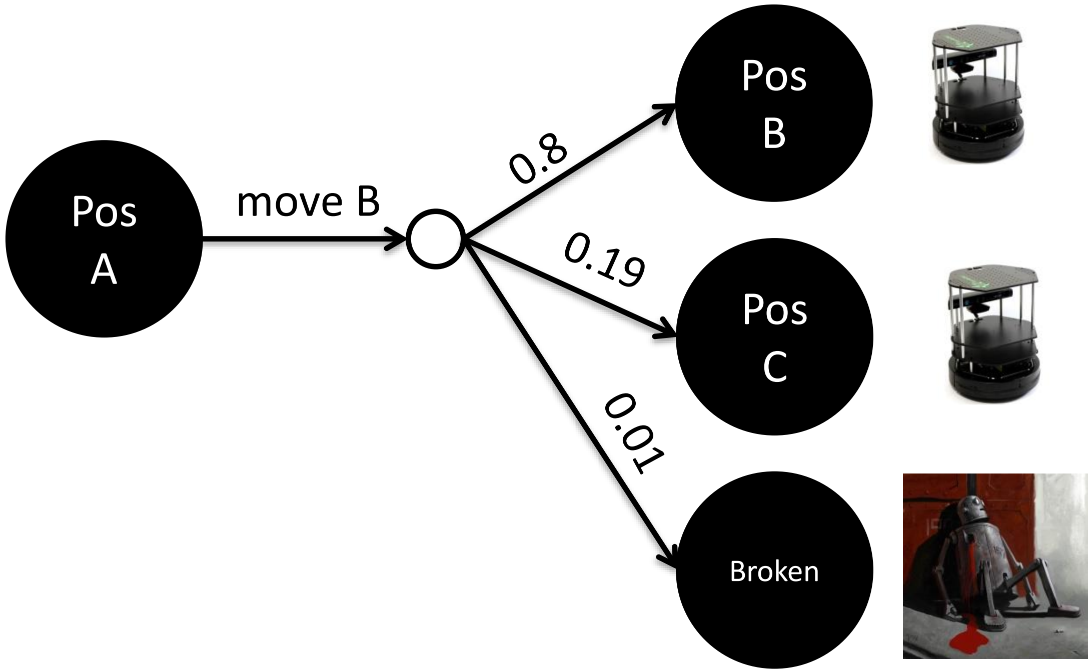
Example (Markovian, deterministic)



Example (Non-Markovian (?), deterministic)

If the snake world is finite, then …

Example (Markovian, stochastic)



Stochastic transitions, fully observable states

MDPs

- Can we formulate the problem asking for a plan?
- Plans are unfit for this situation: we cannot tell how to reach some goal by giving a mere sequence of actions
- We need a **policy**

$$\pi : X \rightarrow a$$ Given the current state, returns what action to play

- It's **deterministic**: given a state it does not randomize on which action to take
- It's **stationary**: it does not change over time

- These assumption are not restrictive in MDPs

MDPs

Policy execution:

1. Observe current state s
2. Execute $\pi(s)$
3. Repeat from 1

MDPs

- We previously spoken about action costs, in MDPs we speak about immediate rewards

$$R_a(s, s')$$ It’s a payoff the agent gets when she transitions from state s, to state s’ with an action a

- Rewards generalize in some sense the notion of goal states

- The objective is to find a policy that maximizes the expected reward over some **time horizon H**

MDP Value iteration

- Idea: let's introduce the concept of **value function**

- How does it work?

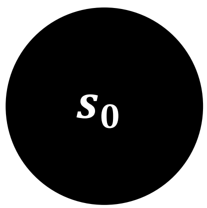
$$V_\pi : X \to \mathbb{R}$$

An agent executing policy π is in state s: how happy is she?

$$V_\pi(s)$$

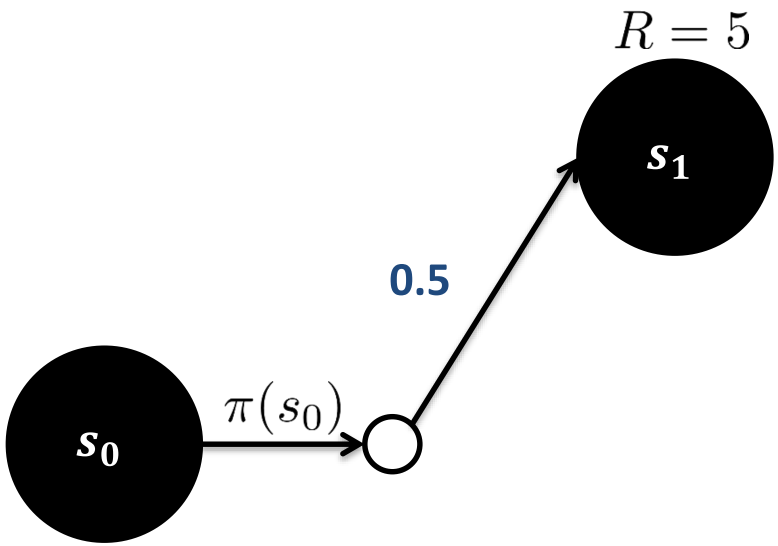
- This quantity is defined as **the expected cumulative reward that can be obtained by executing π from s**

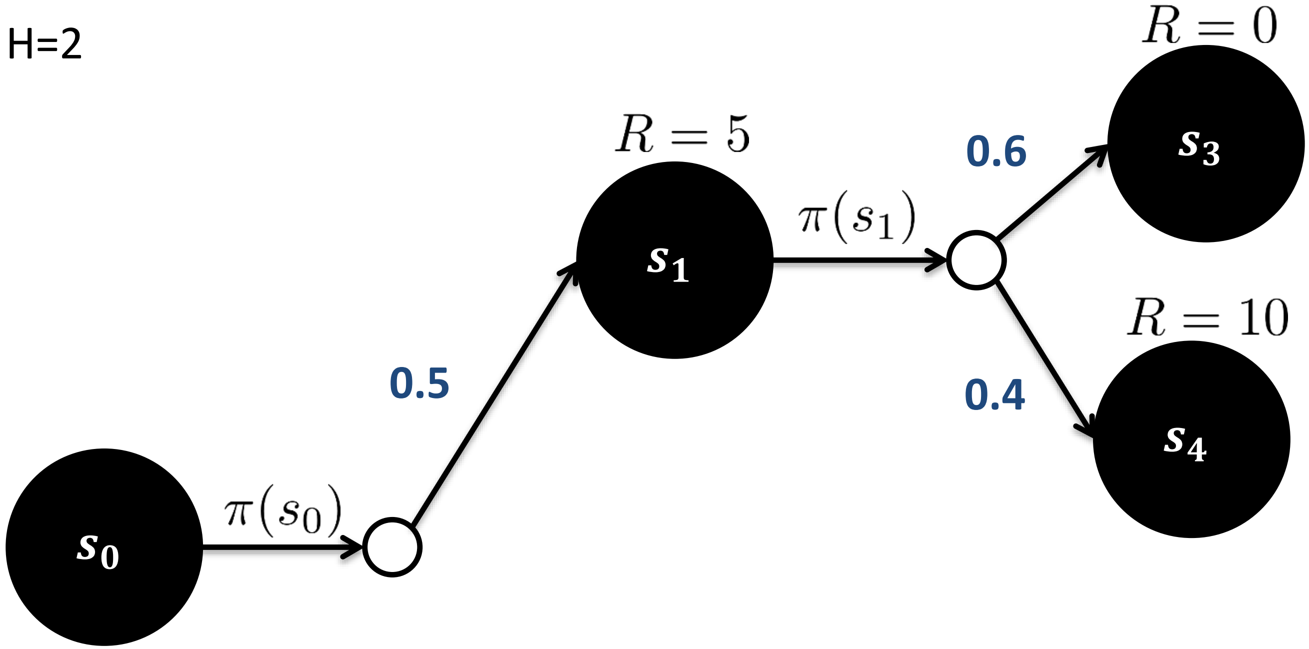
H=2



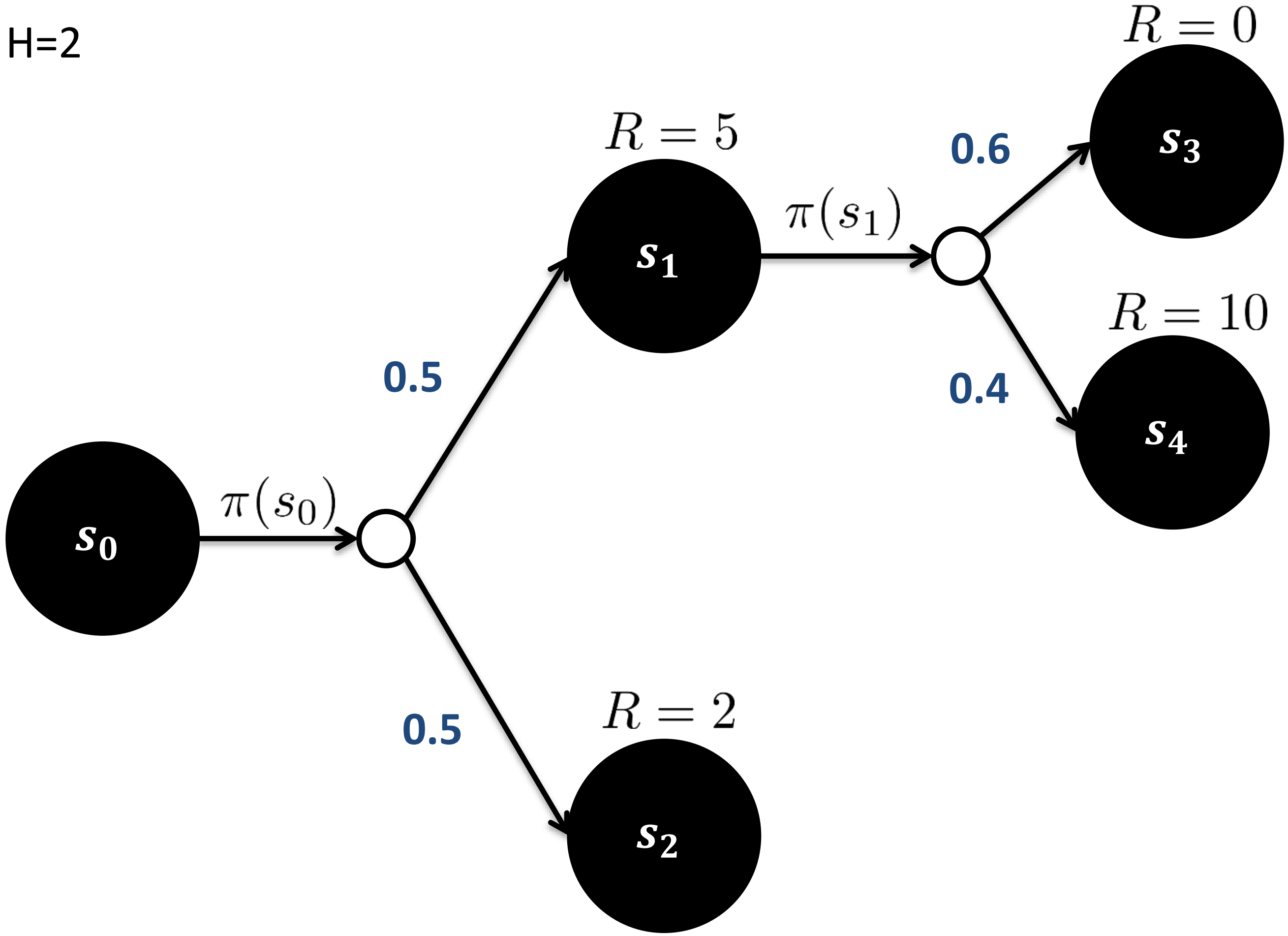
$$V_\pi(s_0) = ?$$

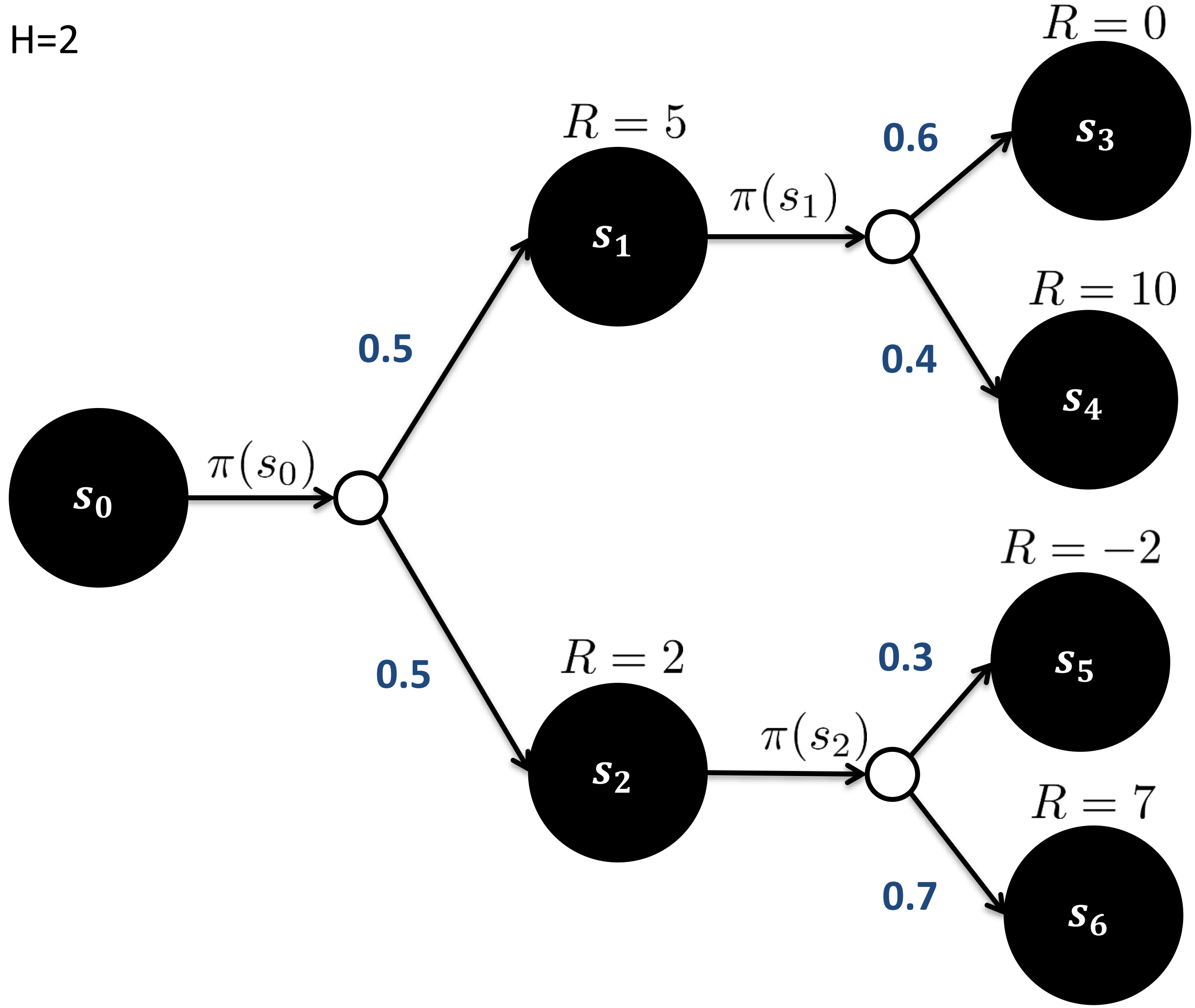
H=2



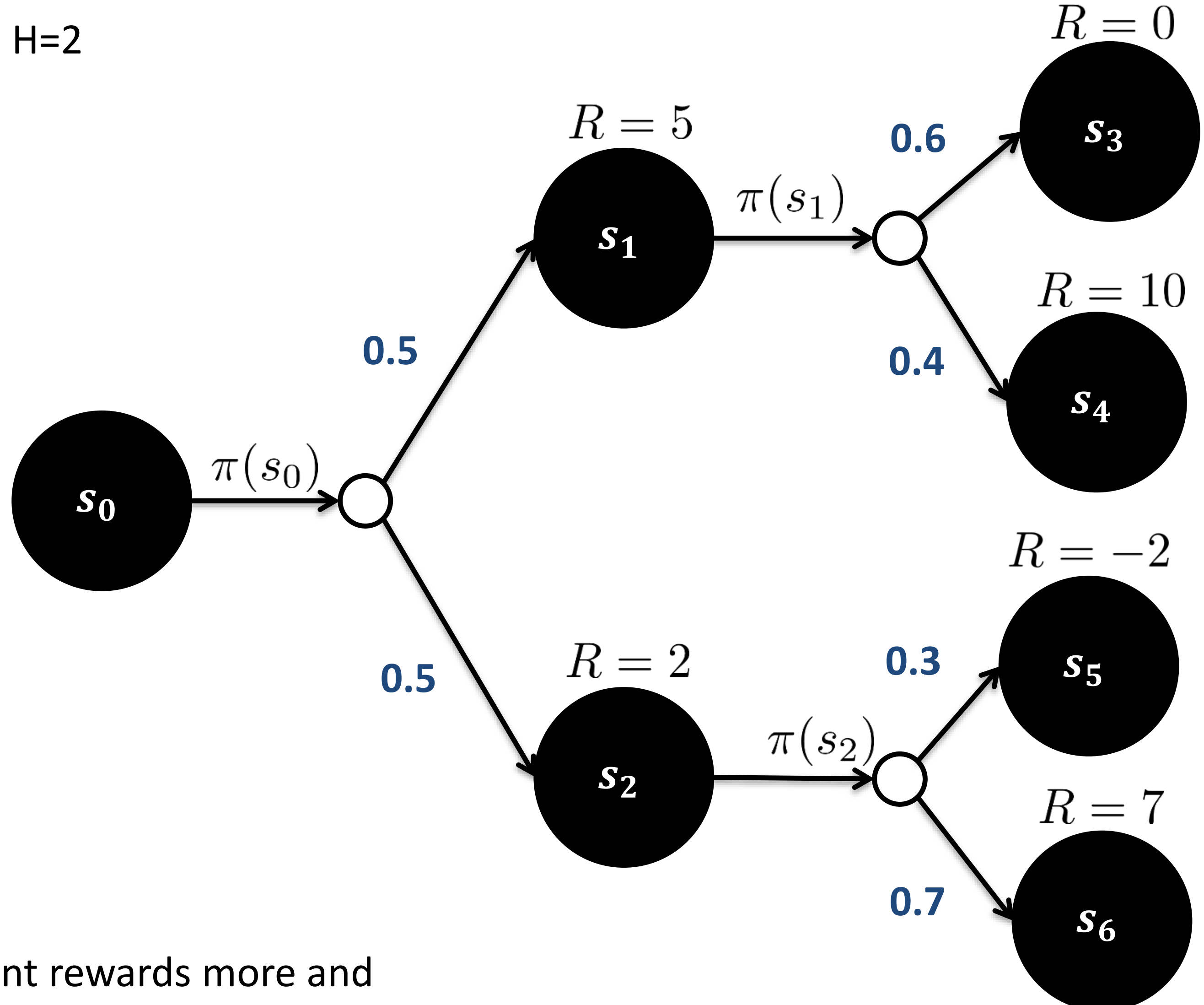

H=2
$R = 5$
$R = 0$
$R = 10$
s_0
s_1
s_3
s_4
$\pi(s_0)$
$\pi(s_1)$
0.5
0.6
0.4

H=2




H=2
$\pi(s_0)$
s_0
0.5
0.5
$R = 5$
s_1
$\pi(s_1)$
$R = 2$
s_2
$\pi(s_2)$
$R = 0$
0.6
s_3
$R = 10$
0.4
s_4
$R = -2$
0.3
s_5
$R = 7$
0.7
s_6

H=2



Discount rewards more and more as they happen in the long future $\gamma \in [0, 1]$

$$V_\pi(s_0) = 0.5(\gamma^0 5 + (0.4(\gamma^1 10))) + 0.5(\gamma^0 2 + (0.3\gamma^1(-2) + 0.7\gamma^1(7)))$$

MDP Value iteration

- Clearly we search for a policy π that yields the maximum value in every state $V^*_\pi(s)$

- We can exploit the Bellman principle of optimality to define a value iteration procedure

$V^*_{\pi,i}(s)$ It's the value of an optimal policy with horizon i

MDP Value iteration

We can write:

$$V^*_{\pi,0}(s) = 0 \quad \forall s \in X$$

The horizon is zero, no action no reward

MDP Value iteration

We can write:

$$V^*_{\pi,0}(s) = 0 \quad \forall s \in X$$

$$V^*_{\pi,1}(s) = \max_a \left\{ \sum_{s' \in X} P(s'|s,a) R_a(s,s') \right\}$$

The horizon is 1, there's room for just one action. The best thing to do is selecting the actions that maximize the **immediate expected reward**.

MDP Value iteration

We can write:

$$V^*_{\pi,0}(s) = 0 \quad \forall s \in X$$

$$V^*_{\pi,1}(s) = \max_a \left\{ \sum_{s' \in X} P(s'|s,a) R_a(s,s') \right\}$$

$$V^*_{\pi,2}(s) = \max_a \left\{ \sum_{s' \in X} P(s'|s,a) \big(R_a(s,s') + \gamma V^*_{\pi,1}(s') \big) \right\}$$

Now the horizon is 2. The optimal policy would select the action that maximizes the immediate expected reward plus the **expected discounted reward of acting optimally from the arrival state.**

MDP Value iteration

We can write:

$$V^*_{\pi,0}(s) = 0 \quad \forall s \in X$$

$$V^*_{\pi,1}(s) = \max_a \Big\{ \sum_{s' \in X} P(s'|s,a) R_a(s,s') \Big\}$$

$$V^*_{\pi,2}(s) = \max_a \Big\{ \sum_{s' \in X} P(s'|s,a) \big(R_a(s,s') + \gamma V^*_{\pi,1}(s') \big) \Big\}$$

$$V^*_{\pi,3}(s) = \max_a \Big\{ \sum_{s' \in X} P(s'|s,a) \big(R_a(s,s') + \gamma V^*_{\pi,2}(s') \big) \Big\}$$

$$\vdots$$

MDP Value iteration

- Bellman's equation

$$V^*_{\pi,H}(s) = \max_a \Big\{ \sum_{s' \in X} P(s'|s,a)\big(R_a(s,s') + \gamma V^*_{\pi,H-1}(s')\big) \Big\}$$

$$\pi^*(s) = \arg\max_a \Big\{ \sum_{s' \in X} P(s'|s,a)\big(R_a(s,s') + \gamma V^*_{\pi,H-1}(s')\big) \Big\}$$

Sistemi Intelligenti
Corso di Laurea in Informatica, A.A. 2017-2018
Università degli Studi di Milano

Nicola Basilico
Dipartimento di Informatica
Via Comelico 39/41 - 20135 Milano (MI)
Ufficio S242
nicola.basilico@unimi.it
+39 02.503.16294