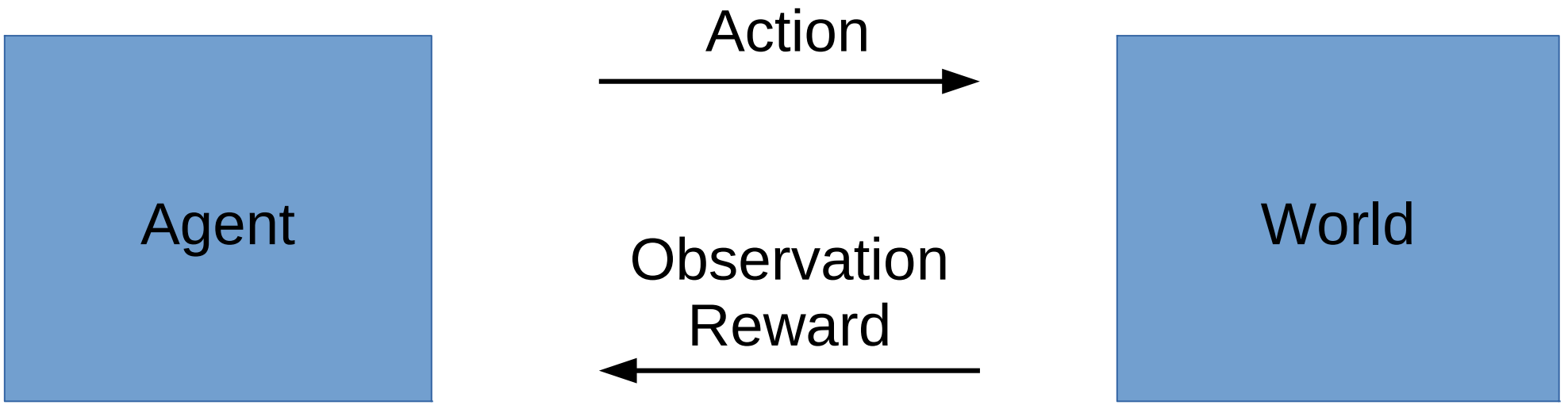
Markov decision process (MDP)

Robert Platt
Northeastern University

The RL Setting

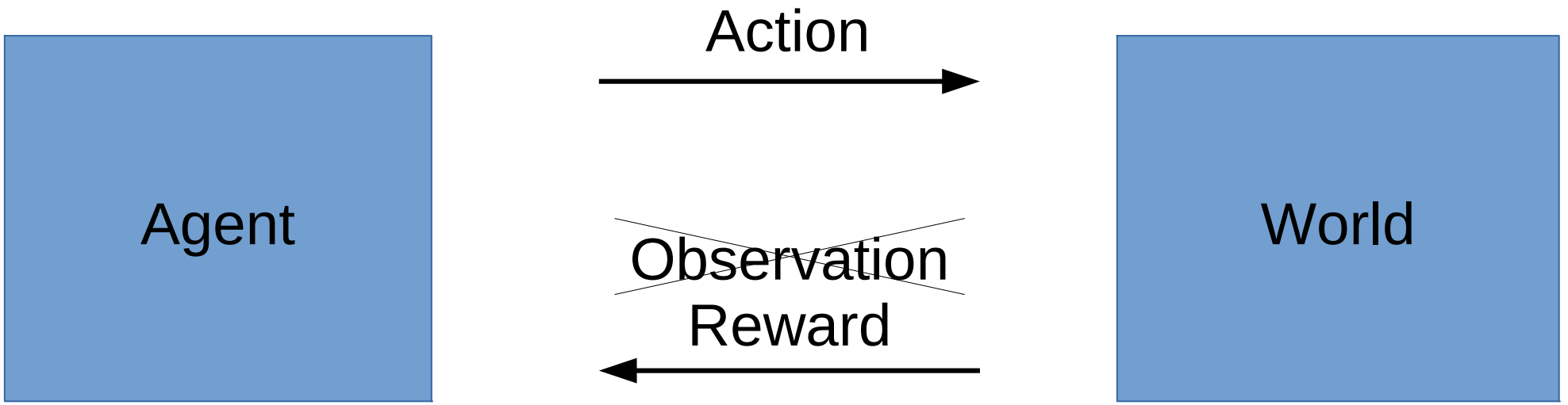


On a single time step, agent does the following:

1. observe some information
2. select an action to execute
3. take note of any reward

Goal of agent: select actions that maximize cumulative reward in the long run

Let’s turn this into an MDP

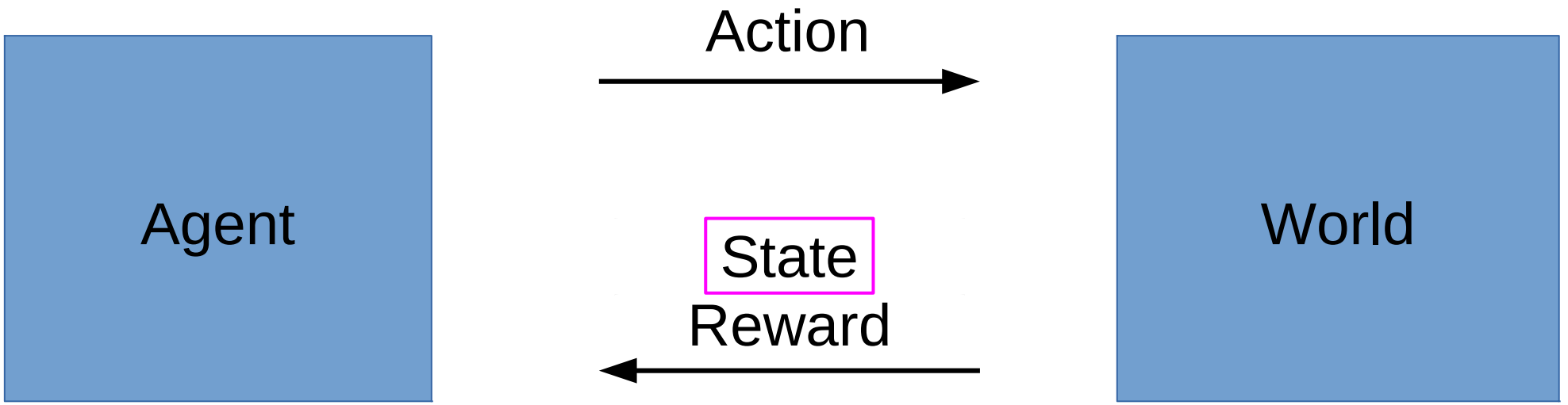


On a single time step, agent does the following:

1. observe some information
2. select an action to execute
3. take note of any reward

Goal of agent: select actions that maximize cumulative reward in the long run

Let's turn this into an MDP

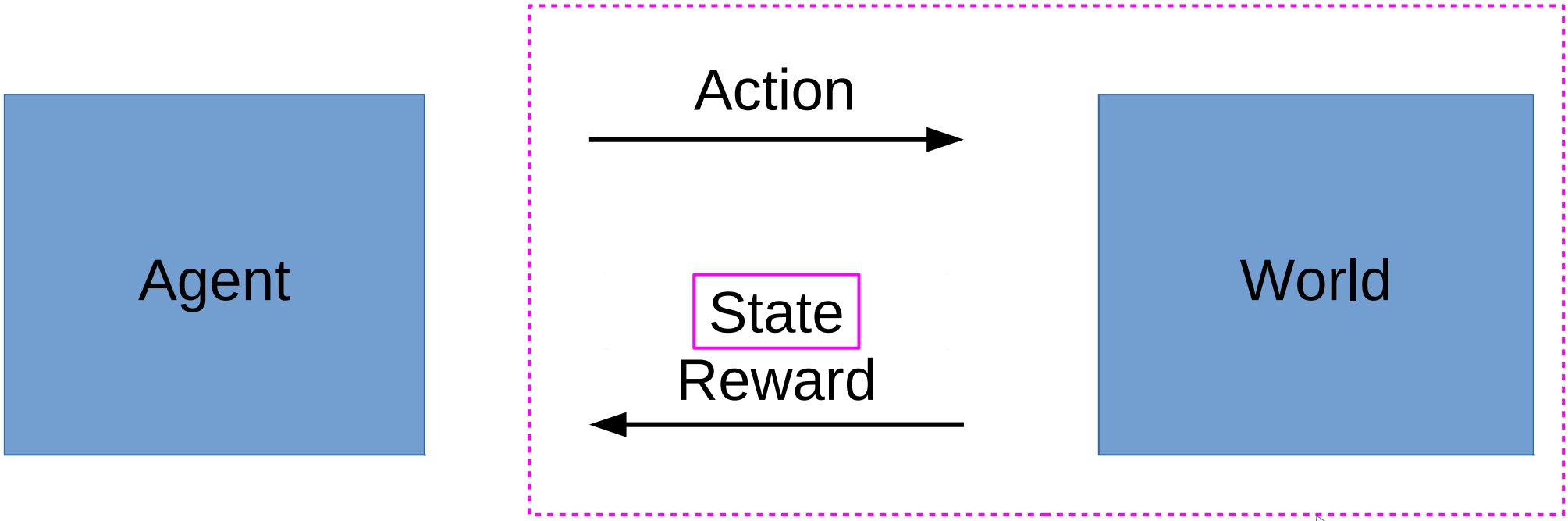


On a single time step, agent does the following:

1. observe state
2. select an action to execute
3. take note of any reward

Goal of agent: select actions that maximize cumulative reward in the long run

Let's turn this into an MDP



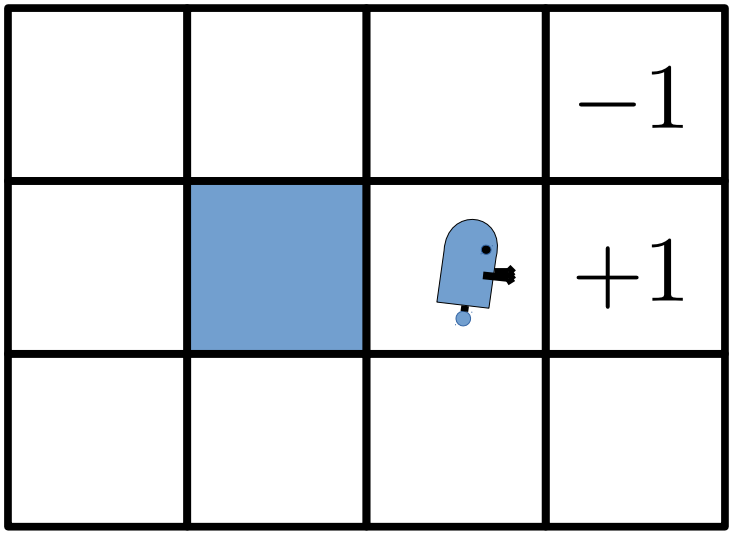
On a single time step, agent does the following:

1. observe state
2. select an action to execute
3. take note of any reward

This part is the MDP

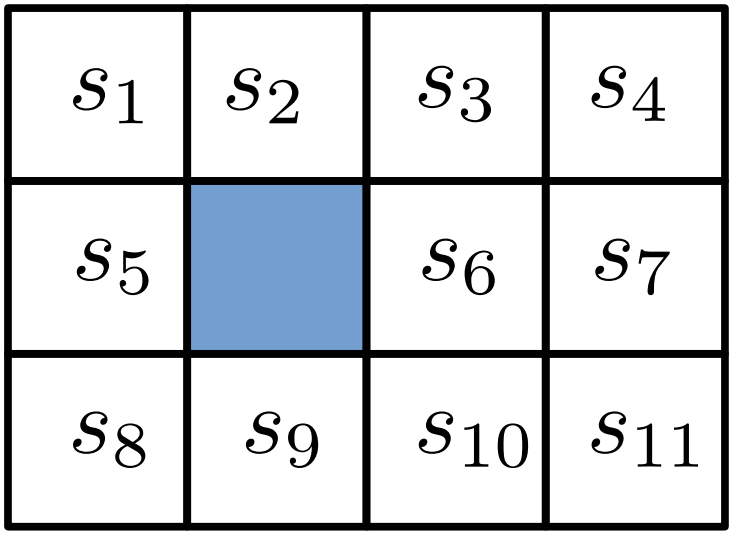
Goal of agent: select actions that maximize cumulative reward in the long run

Example: Grid world



Grid world:
– agent lives on grid
– always occupies a single cell
– can move left, right, up, down
– gets zero reward unless in “+1” or “-1” cells

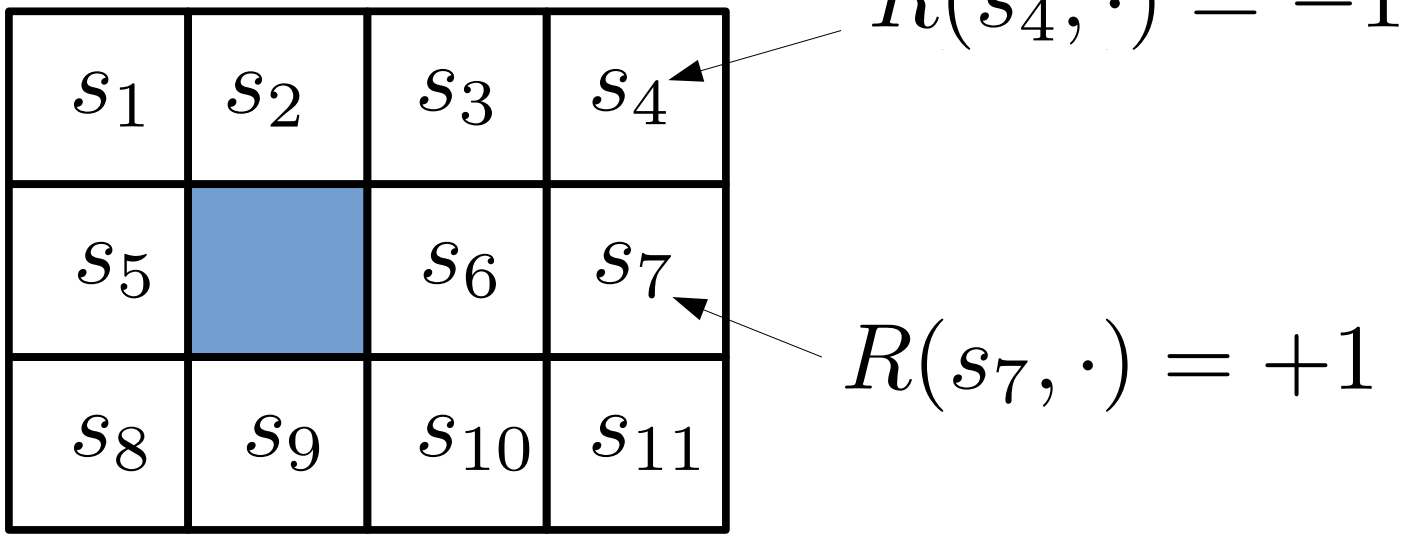
States and actions



State set: $S = \{s_1, \ldots, s_{11}\}$

Action set: $A = \{left, right, up, down\}$

Reward function

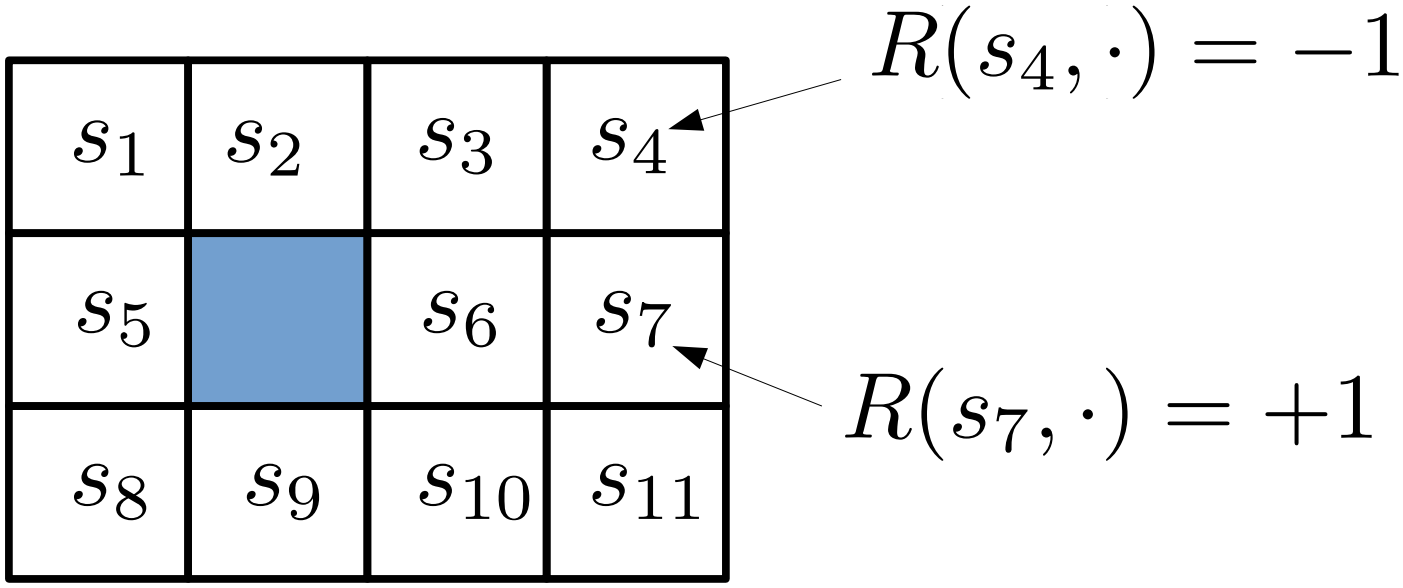


Reward function: $R(s_4, \cdot) = -1$

$R(s_7, \cdot) = +1$

Otherwise: $R(s, \cdot) = 0$

Reward function



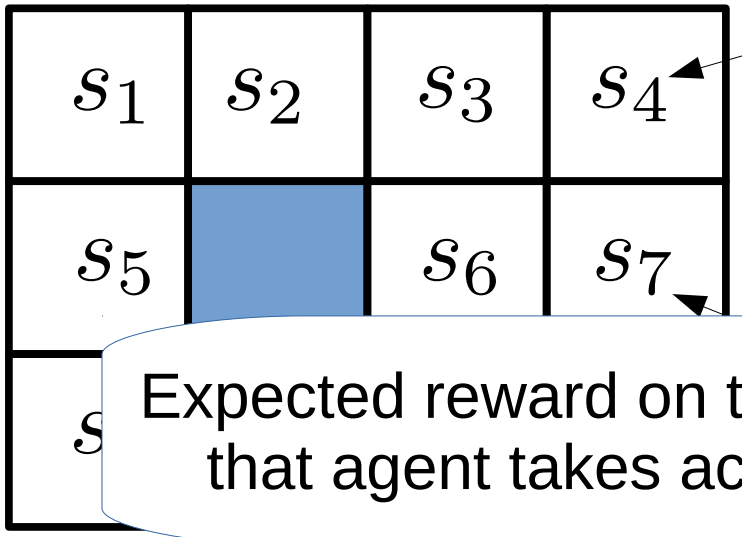
Reward function: $R(s_4, \cdot) = -1$

$R(s_7, \cdot) = +1$

Otherwise: $R(s, \cdot) = 0$

In general: $R(s, a) = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$

Reward function



$R(s_4, \cdot) = -1$

-1

Expected reward on this time step given that agent takes action a from state s

Reward function: $R(s_4$ -1

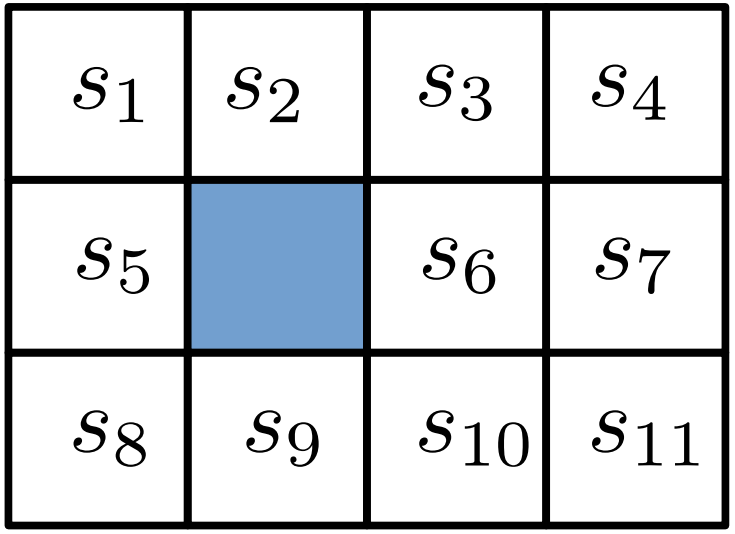
$R(s_7,$ $+1$

Otherwise: $R(s, \cdot) = 0$

In general: $R(s, a) = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$

Transition function

Transition model: $P(s_{t+1} = s' | s_t = s, a_t = a)$

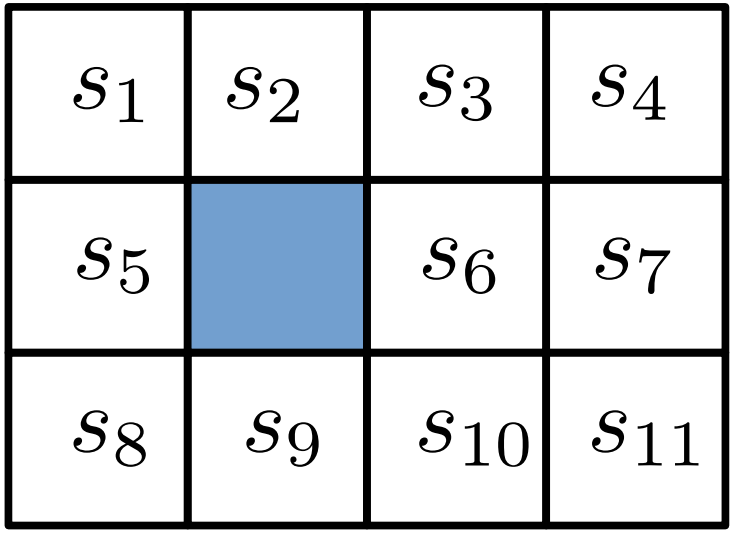


For example:

$$P(s_{t+1} = s_4 | s_t = s_3, a_t = left) = 0.1$$
$$P(s_{t+1} = s_2 | s_t = s_3, a_t = left) = 0.7$$
$$P(s_{t+1} = s_6 | s_t = s_3, a_t = left) = 0.1$$
$$P(s_{t+1} = s_3 | s_t = s_3, a_t = left) = 0.1$$

Transition function

Transition model: $P(s_{t+1} = s' | s_t = s, a_t = a)$



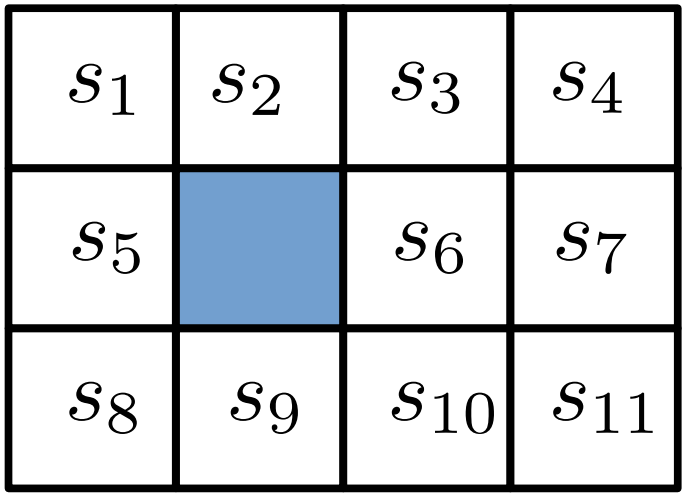
For example:

$$P(s_{t+1} = s_4 | s_t = s_3, a_t = left) = 0.1$$
$$P(s_{t+1} = s_2 | s_t = s_3, a_t = left) = 0.7$$
$$P(s_{t+1} = s_6 | s_t = s_3, a_t = left) = 0.1$$
$$P(s_{t+1} = s_3 | s_t = s_3, a_t = left) = 0.1$$

– This entire probability distribution can be written as a table over state, action, next state.

s_t	a_t	s_{t+1}	probability of this transition

Definition of an MDP



An MDP is a tuple: $\mathcal{M} = \langle S, A, R, P \rangle$

where

State set: $S = \{s_1, \ldots, s_{11}\}$

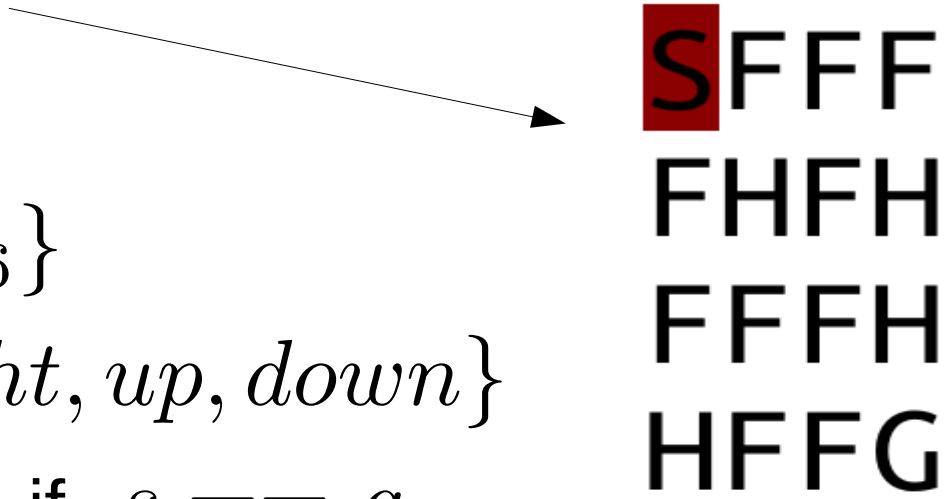
Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, a) = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$

Transition model: $P(s_{t+1} = s' | s_t = s, a_t = a)$

Example: Frozen Lake

Frozen Lake is this 4x4 grid



State set: $S = \{s_1, \ldots, s_{16}\}$

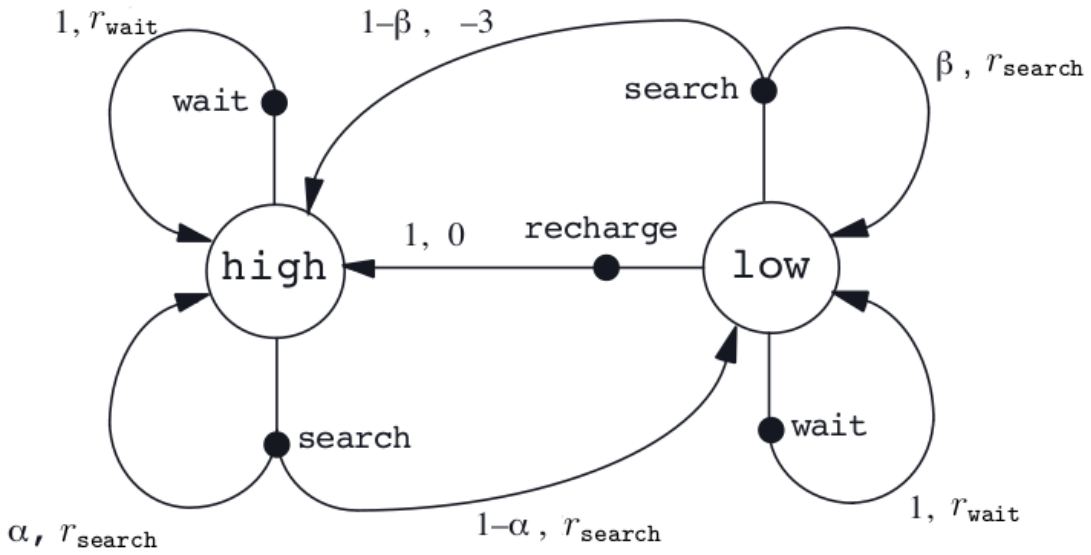
Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, \cdot) = 1$ if $s == g$

$R(s, \cdot) = 0$ otherwise

Transition model: only one third chance of going in specified direction
– one third chance of moving +90deg
– one third change of moving -90deg

Example: Recycling Robot



s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	r_{wait}
low	wait	high	0	r_{wait}
low	wait	low	1	r_{wait}
low	recharge	high	1	0
low	recharge	low	0	0.

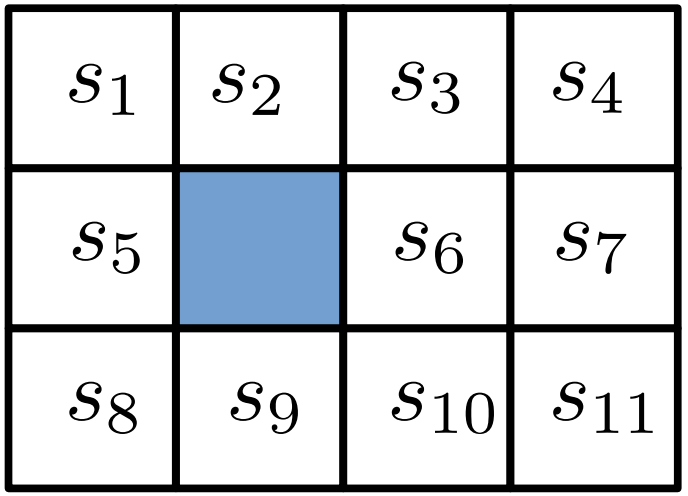
Example 3.4 in SB, 2nd Ed.

Think-pair-share

Mobile robot:
– the robot moves on a flat surface
– the robot can execute point turns either left or right. It can also go forward or back with fixed velocity
– it must reach a goal while avoiding obstacles

Express mobile robot control problem as an MDP

Definition of an MDP



An MDP is a tuple: $\mathcal{M} = \langle S, A, R, P \rangle$

where

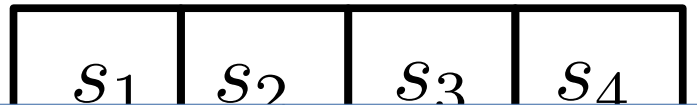
State set: $S = \{s_1, \ldots, s_{11}\}$

Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, a) = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$

Transition model: $P(s_{t+1} = s' | s_t = s, a_t = a)$

Definition of an MDP



Why is it called a *Markov* decision process?

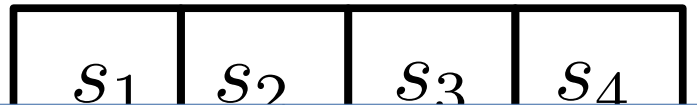
State set: $S = \{s_1, \ldots, s_{11}\}$

Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, a) = \mathbb{E}[r_{t+1} | s_t = s, a_t = a]$

Transition model: $P(s_{t+1} = s' | s_t = s, a_t = a)$

Definition of an MDP



Why is it called a *Markov* decision process?

Because we're making the following assumption:

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_t, a_t, \ldots, s_1, a_1)$$

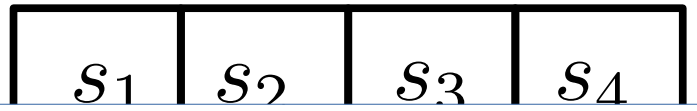
State set: $S = \{s_1, \ldots, s_{11}\}$

Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a]$

Transition model: $P(s_{t+1} = s'|s_t = s, a_t = a)$

Definition of an MDP



Why is it called a *Markov* decision process?

Because we're making the following assumption:

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_t, a_t, \ldots, s_1, a_1)$$

– this is called the "Markov" assumption

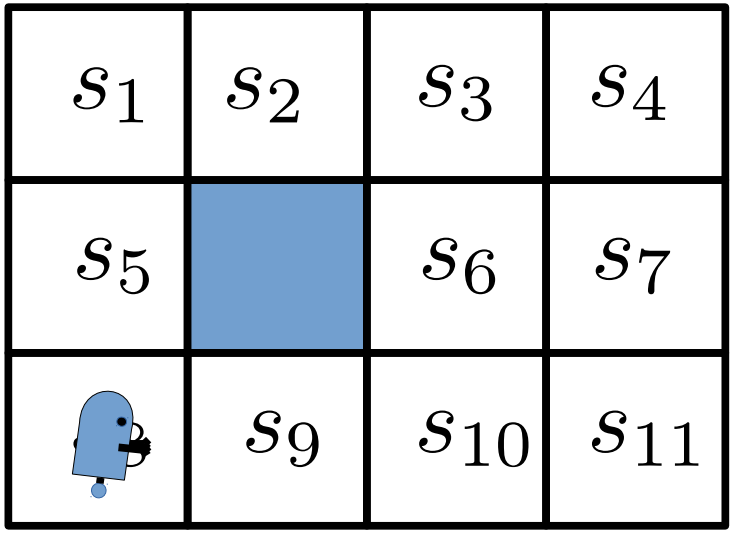
State set: $S = \{s_1, \ldots, s_{11}\}$

Action set: $A = \{left, right, up, down\}$

Reward function: $R(s, a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a]$

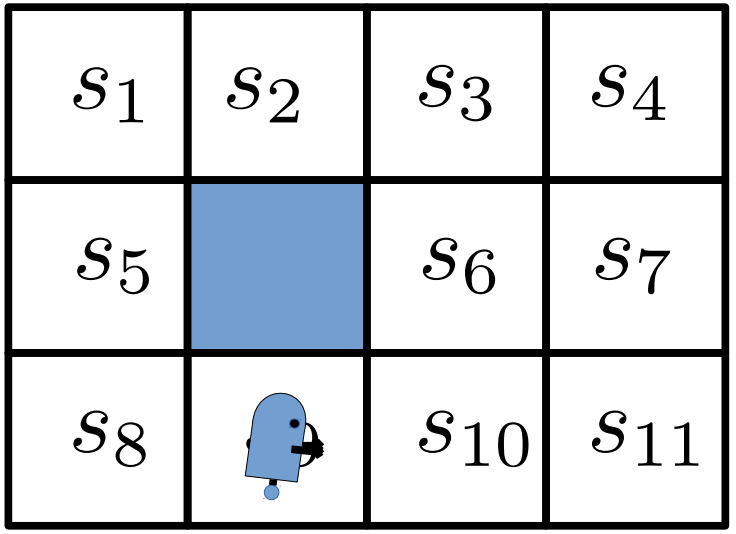
Transition model: $P(s_{t+1} = s'|s_t = s, a_t = a)$

The Markov Assumption



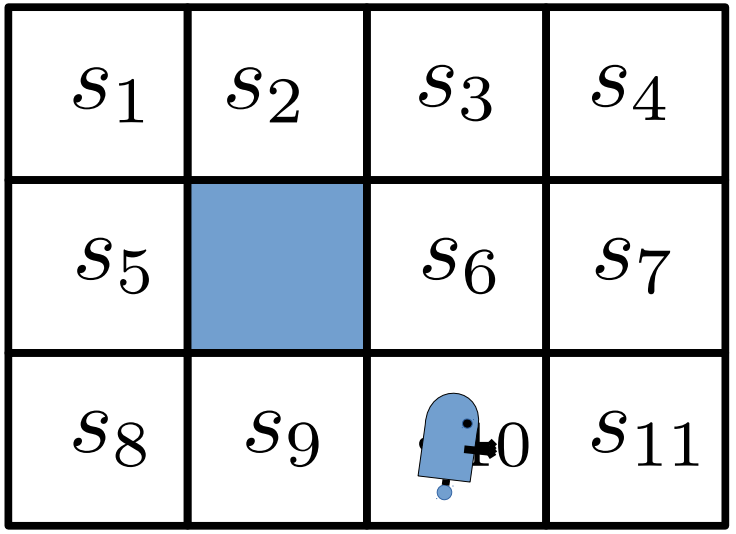
Suppose agent starts in s_8 and follows this path: s_8, s_9, s_{10}

The Markov Assumption



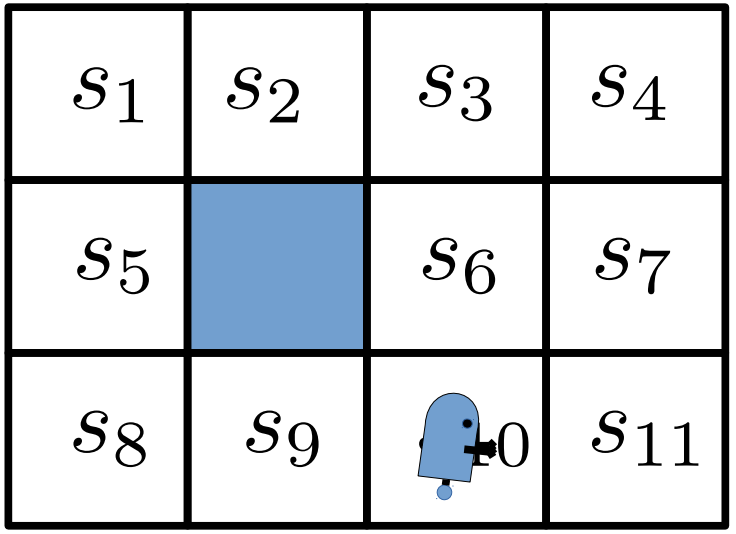
Suppose agent starts in s_8 and follows this path: s_8, s_9, s_{10}

The Markov Assumption



Suppose agent starts in s_8 and follows this path: s_8, s_9, s_{10}

The Markov Assumption

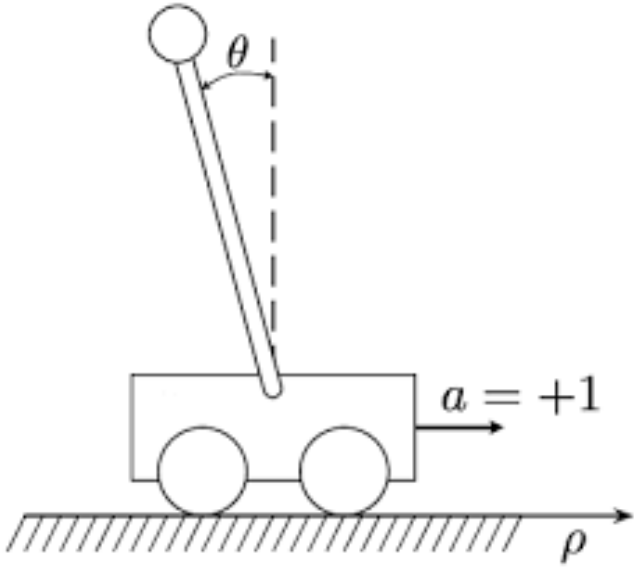


Suppose agent starts in s_8 and follows this path: s_8, s_9, s_{10}

Notice that probability of arriving in s_{11} if agent executes right action does not depend on path taken to get to s_{10} :

$$P(s_{11}|s_{10}, right) = P(s_{11}|s_{10}, right, s_9, right, s_8, right)$$

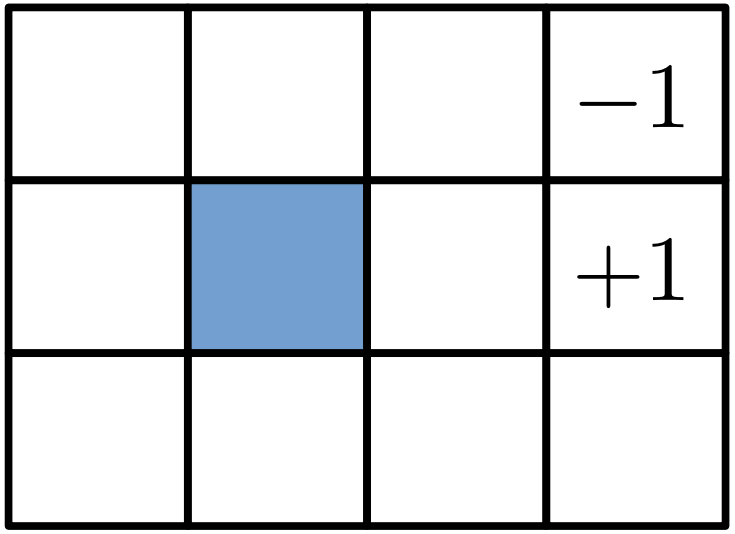
Think-pair-share



Cart-pole robot:
– state is the position of the cart and the orientation of the pole
– cart can execute a constant acceleration either left or right

1. Is this system Markov?
2. Why / Why not?
3. If not, how do you change it to make it Markov?

Policy

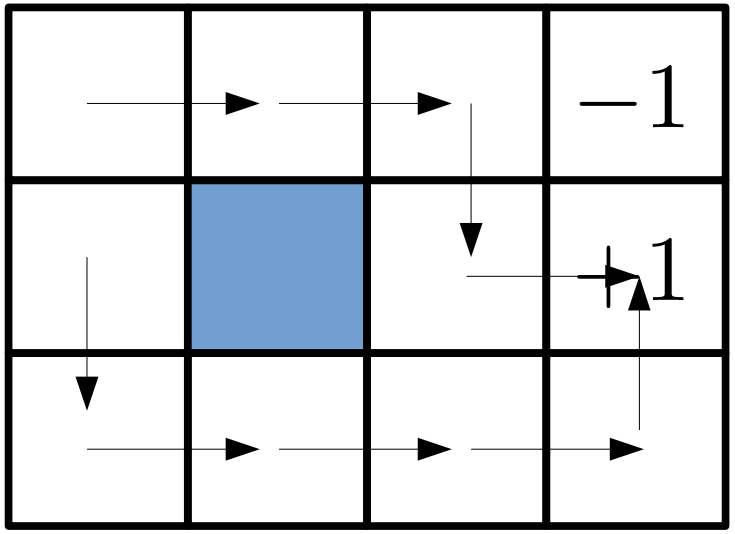


A *policy* is a rule for selecting actions: $\pi : S \rightarrow A$

$\pi(s) = a$

If agent is in *this* state, then take *this* action

Policy

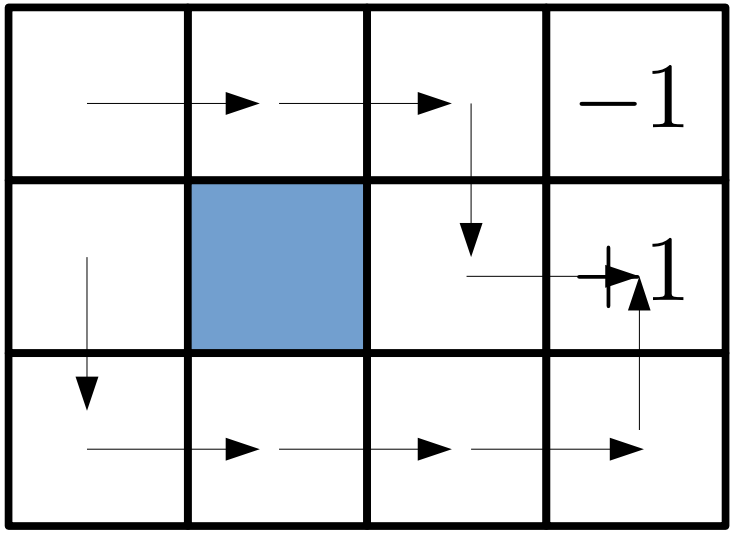


A *policy* is a rule for selecting actions: $\pi : S \rightarrow A$

$$\pi(s) = a$$

If agent is in *this* state, then take *this* action

Policy



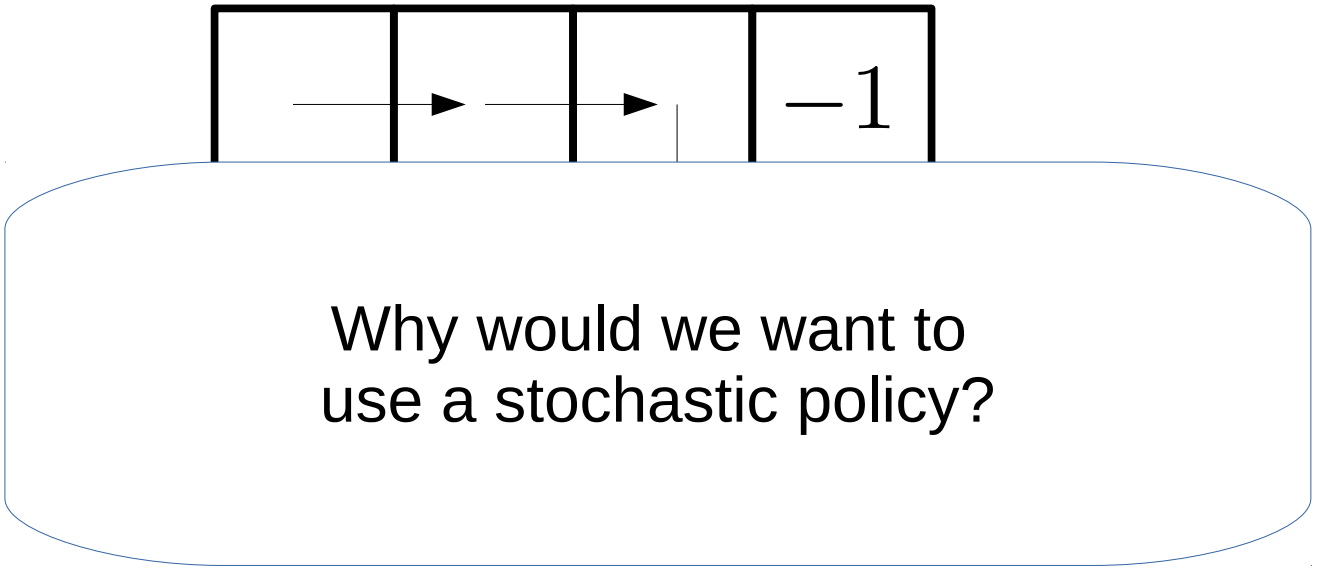
A *policy* is a rule for selecting actions: $\pi : S \rightarrow A$

$$\pi(s) = a$$

If agent is in *this* state, then take *this* action

A policy can be stochastic: $\pi(a|s) = P(a_t = a|s_t = s)$

Question



A *policy* is a rule for selecting actions: $\pi : S \rightarrow A$

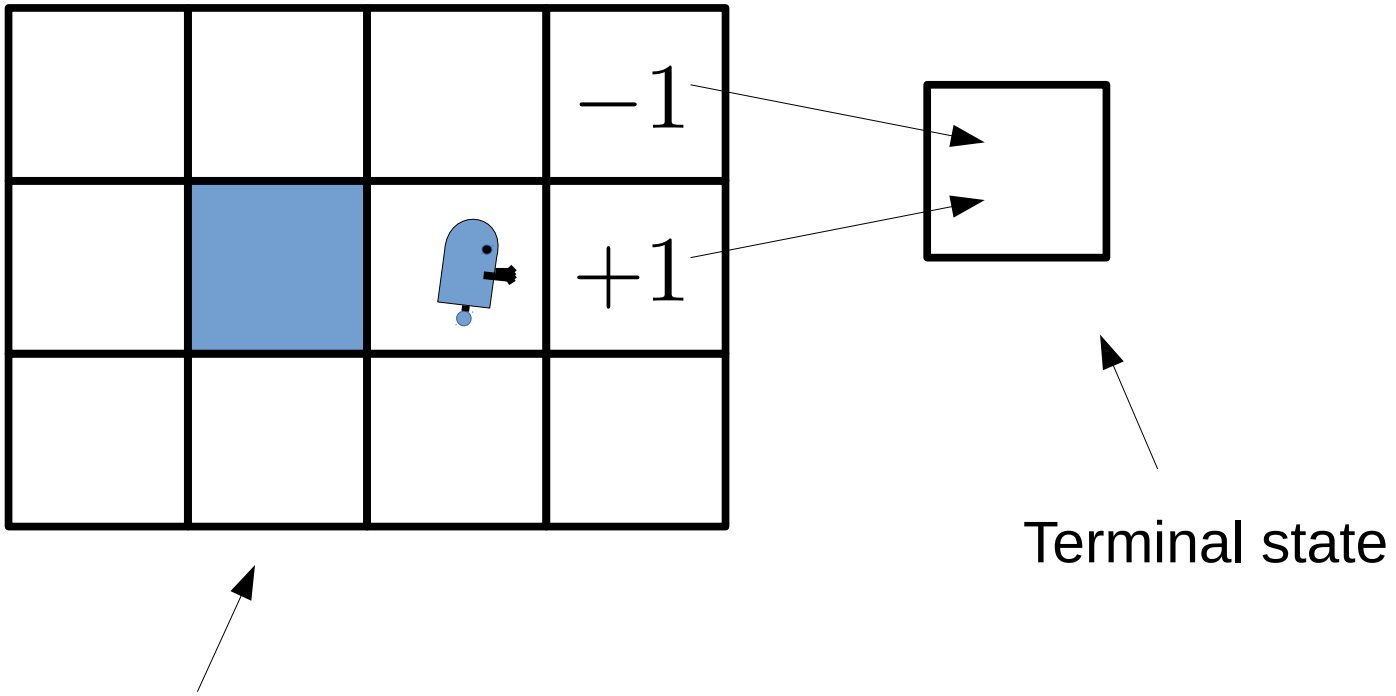
$$\pi(s) = a$$

If agent is in *this* state, then take *this* action

A policy can be stochastic: $\pi(a|s) = P(a_t = a|s_t = s)$

Episodic vs Continuing Process

Episodic process: execution ends at some point and starts over.
- after a fixed number of time steps
- upon reaching a terminal state

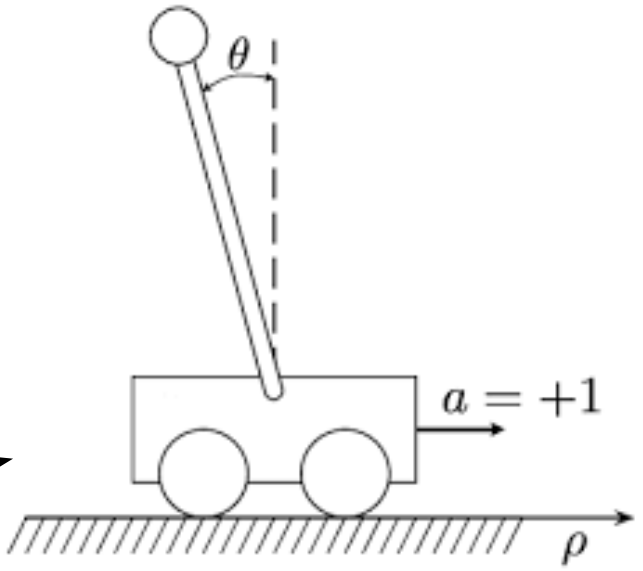


Example of an episodic task:
- execution ends upon reaching terminal state OR after 15 time steps

Episodic vs Continuing Process

Continuing process: execution goes on forever.

Example of a continuing task



Process doesn’t stop
– keep getting rewards

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

– could have positive reward at goal, zero reward elsewhere
– could have negative reward on every time step
– could have an arbitrary reward function

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

Rewards and Return

Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

But, it is often a *discounted* sum of rewards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$$

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

But, it is often a *discounted* sum of rewards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$$

What effect does gamma have?

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

But, it is often a *discounted* sum of rewards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$$

Reward received *k* time steps in the future is only worth γ^{k-1} of what it would have been worth immediately

Rewards and Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

But, it is often a *discounted* sum of rewards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$$

Rewards and Return

Return

On each time step, the agent gets a reward: $r_1, r_2, r_3, \ldots, r_T$

Return can be a simple sum of rewards:

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

But, it is often a *discounted* sum of rewards:

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_T$$

Return is often evaluated over an infinite horizon:

$$\begin{aligned} G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \\ &= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \end{aligned}$$

Think-pair-share

Exercise 3.8 Suppose $\gamma = 0.5$ and the following sequence of rewards is received $R_1 = -1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, and $R_5 = 2$, with $T = 5$. What are G_0, G_1, ..., G_5? Hint: Work backwards. □

Value Function

Value of state s when acting according to policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

Value Function

Value of state s when acting according to policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

Value of a state == expected return from that state
if agent follows policy π

Value Function

Value of state s when acting according to policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

Value of a state == expected return from that state
if agent follows policy π

Value of taking action a from state s when acting according to policy π :

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$$

Value Function

Value of state s when acting according to policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

Value of a state == expected return from that state if agent follows policy π

Value of taking action a from state s when acting according to policy π :

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$$

Value of a state/action pair == expected return when taking action *a* from state *s* and following π after that

Value Function

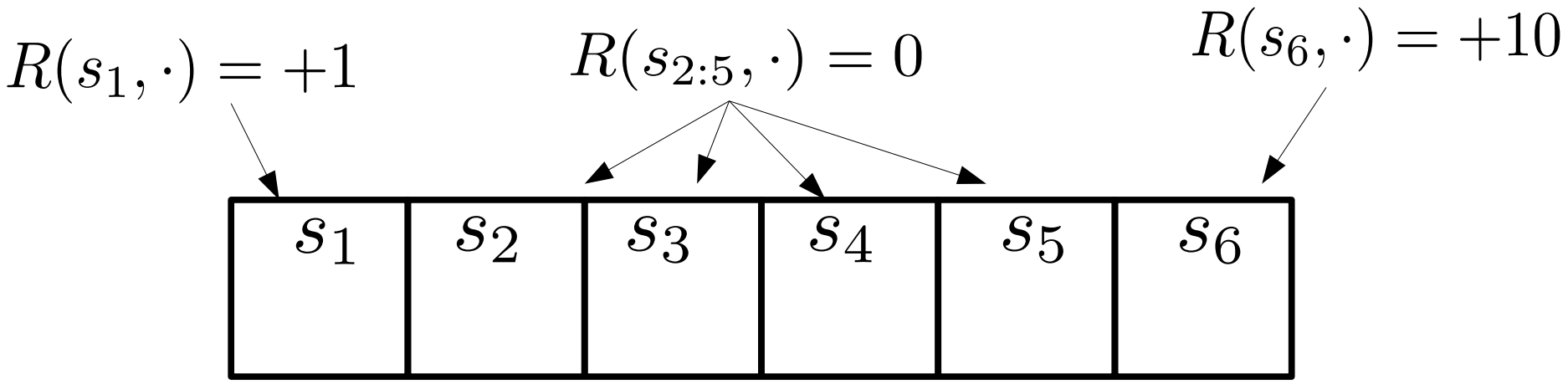
Value of state s when acting according to policy π :

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$
$$= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right]$$

Value of taking action a from state s when acting according to policy π :

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$$
$$= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right]$$

Value function example 1



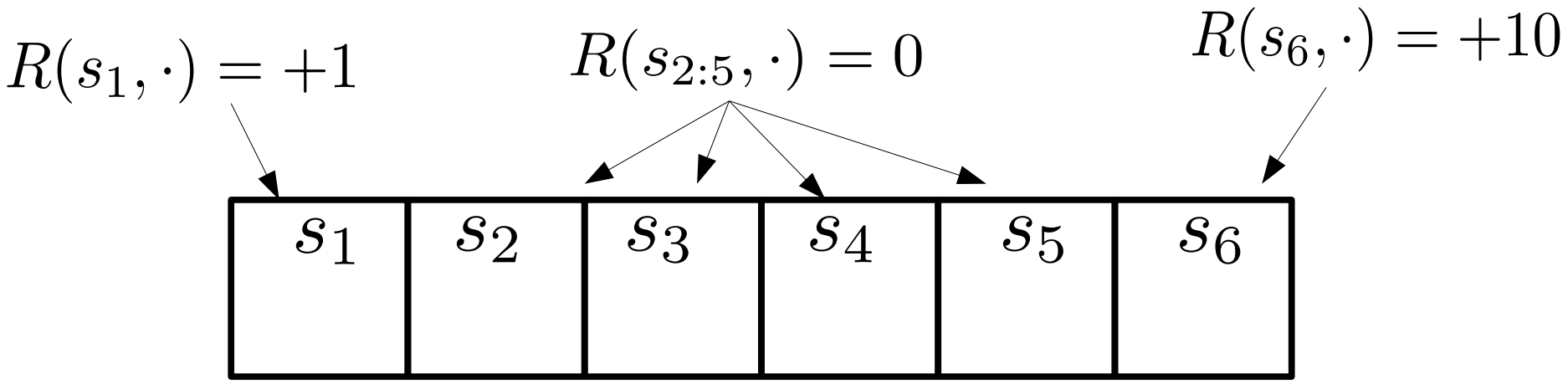
Policy: 

Discount factor: $\gamma = 0.9$

Value fn:

6.9	6.6	7.3	8.1	9	10

Value function example 2



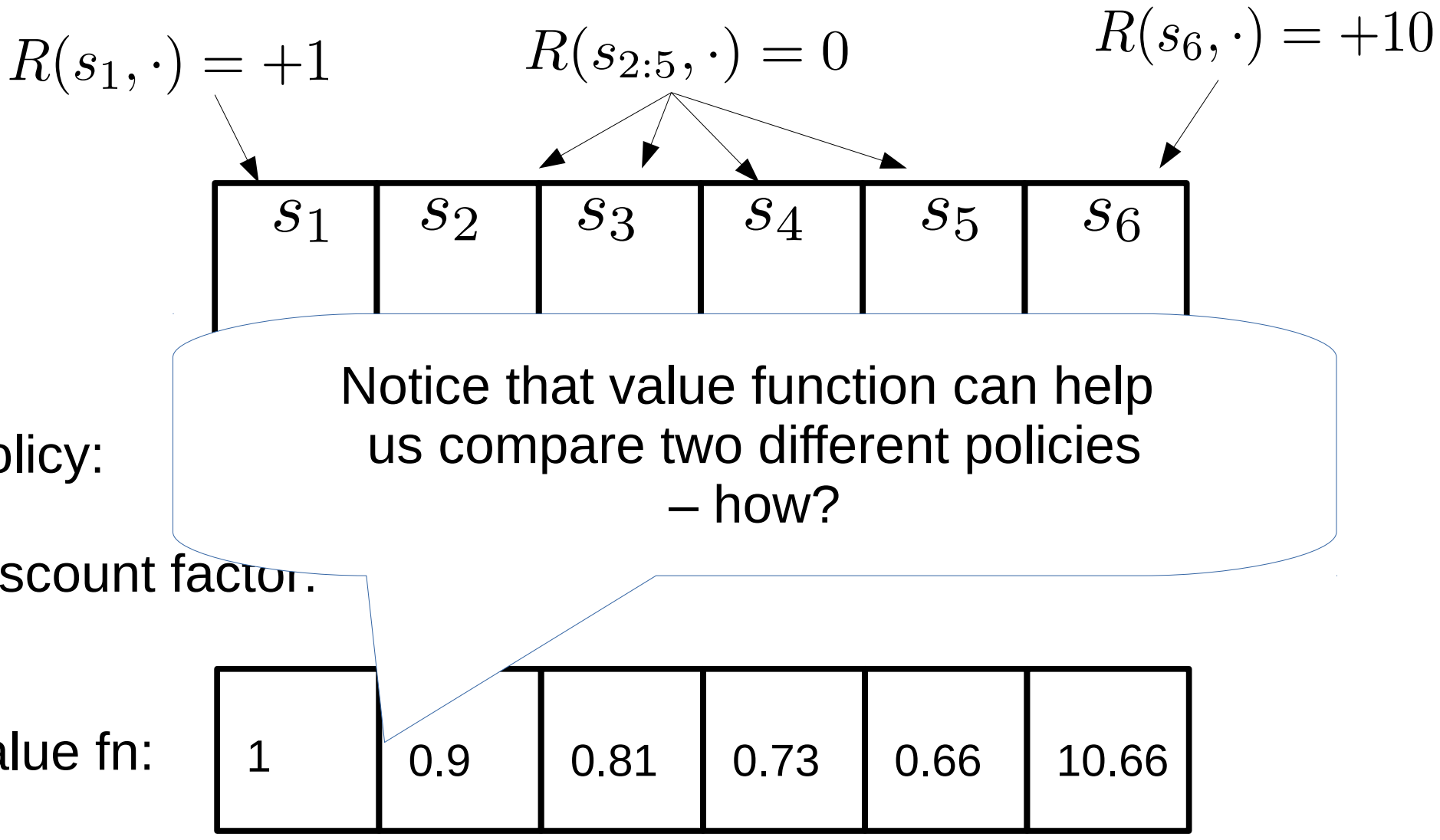
Policy:

Discount factor: $\gamma = 0.9$

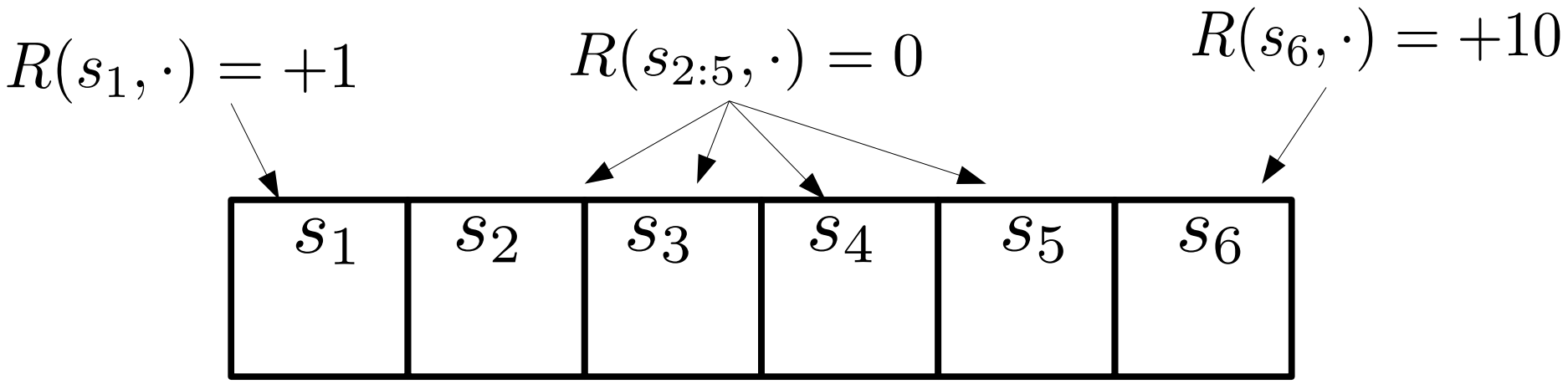
Value fn:

1	0.9	0.81	0.73	0.66	10.66

Value function example 2



Value function example 3



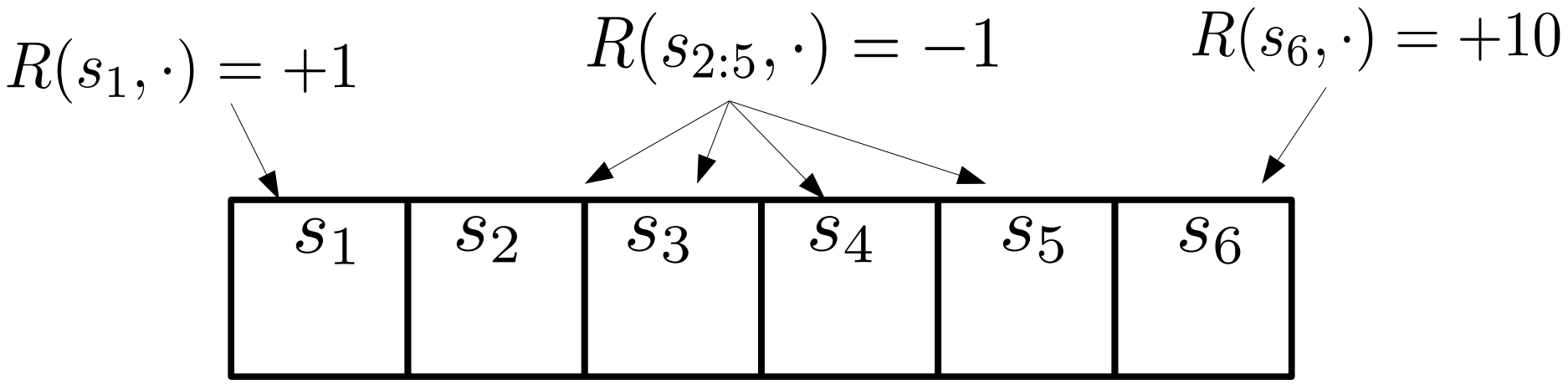
Policy: 

Discount factor: $\gamma = 1$

Value fn:

11	10	10	10	10	10

Think-pair-share



Policy: 

Discount factor: $\gamma = 1$

Value fn:

?	?	?	?	?	?

Value Function Revisited

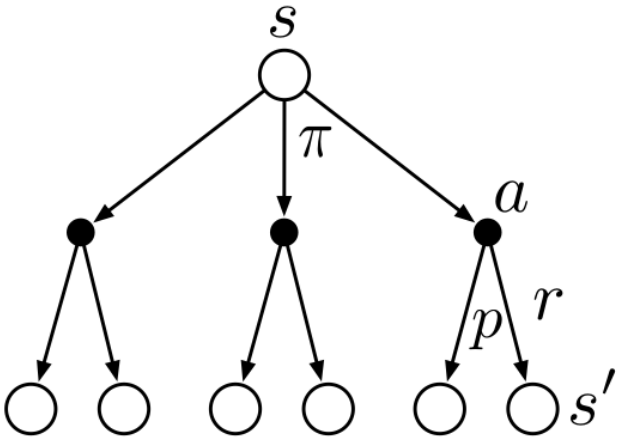
Value of state s when acting according to policy π :

$$V^{\pi}(s) \quad = \quad \mathbb{E}_{\pi}[G_t | s_t = s]$$

Value Function Revisited

Value of state s when acting according to policy π :

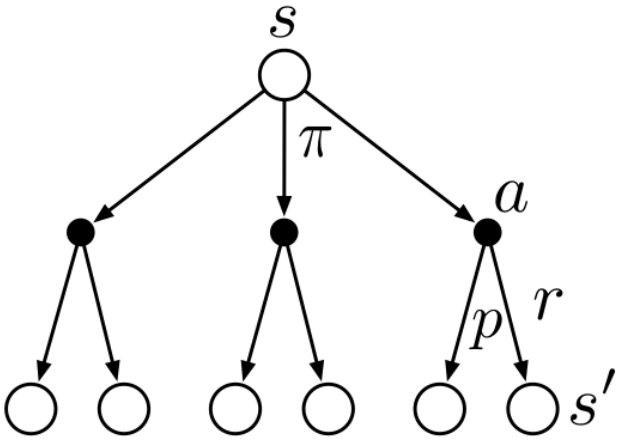
$$V^{\pi}(s) \quad = \quad \mathbb{E}_{\pi}[G_t | s_t = s]$$



Value Function Revisited

Value of state s when acting according to policy π :

$$V^{\pi}(s) \quad = \quad \mathbb{E}_{\pi}[G_t | s_t = s]$$

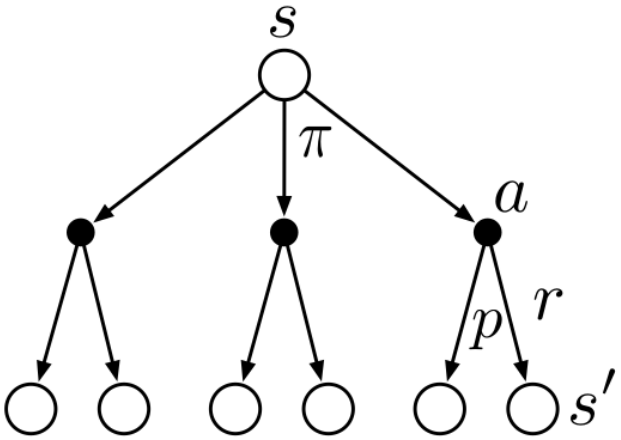


This is called a “backup diagram”

Value Function Revisited

Value of state s when acting according to policy π :

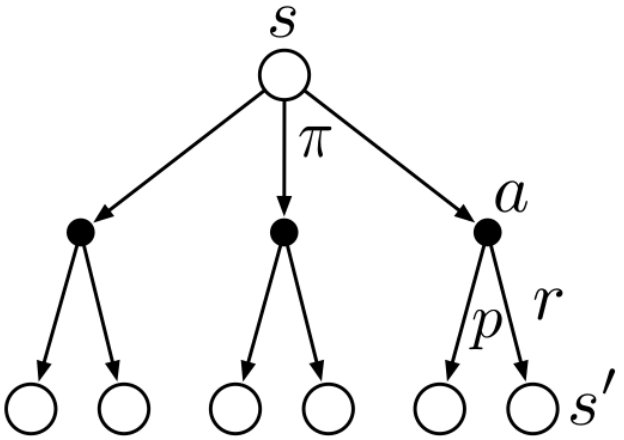
$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}[G_t | s_t = s] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right]
\end{aligned}
$$



Value Function Revisited

Value of state s when acting according to policy π :

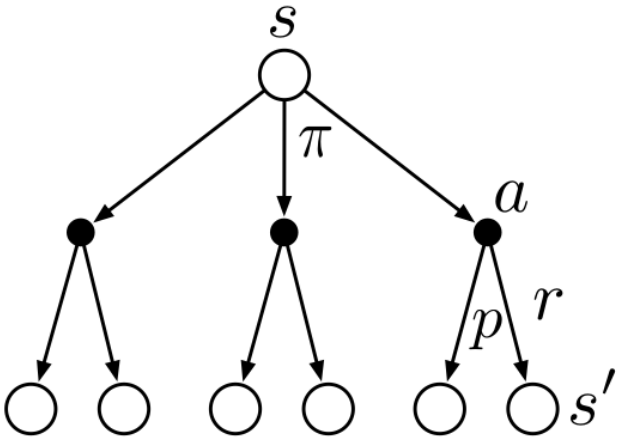
$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}[G_t | s_t = s] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right] \\
&= \mathbb{E}_{s',r}\left[r + \gamma V^{\pi}(s')\right]
\end{aligned}
$$



Think-pair-share 1

Value of state s when acting according to policy π :

$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}[G_t | s_t = s] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right] \\
&= \mathbb{E}_{s',r}\left[r + \gamma V^{\pi}(s')\right] \\
&= \qquad\qquad
\end{aligned}
$$

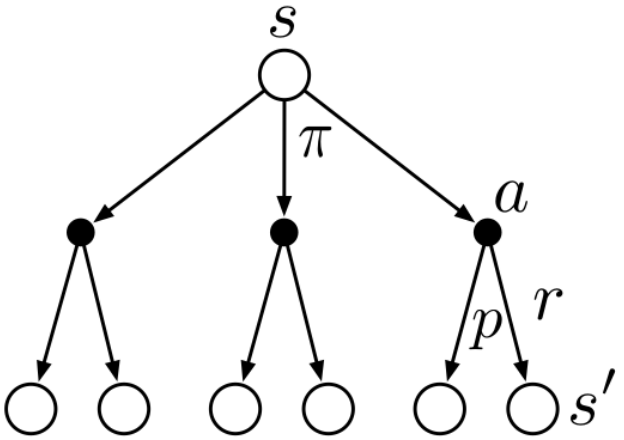


Write this expectation in terms of P(*s'*,*r*|*s*,*a*) for a deterministic policy, $\pi(s)$

Think-pair-share 2

Value of state s when acting according to policy π :

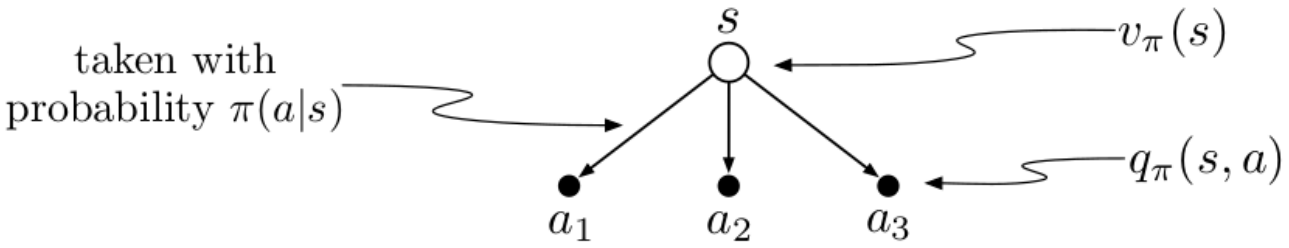
$$
\begin{aligned}
V^{\pi}(s) &= \mathbb{E}_{\pi}[G_t | s_t = s] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right] \\
&= \mathbb{E}_{s',r}\left[r + \gamma V^{\pi}(s')\right] \\
&= \qquad\qquad
\end{aligned}
$$



Write this expectation in terms of P(*s'*,*r*|*s*,*a*) for a stochastic policy, $\pi(a|s)$

Think-pair-share

Exercise 3.18 The value of a state depends on the values of the actions possible in that state and on how likely each action is to be taken under the current policy. We can think of this in terms of a small backup diagram rooted at the state and considering each possible action:



Give the equation corresponding to this intuition and diagram for the value at the root node, $v_\pi(s)$, in terms of the value at the expected leaf node, $q_\pi(s,a)$, given $S_t = s$. This equation should include an expectation conditioned on following the policy, π. Then give a second equation in which the expected value is written out explicitly in terms of $\pi(a|s)$ such that no expected value notation appears in the equation. □

Value Function Revisited

Can we calculate *Q* in terms of *V*?

$$Q^{\pi}(s,a) \quad = \quad \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$$

Value Function Revisited

Can we calculate *Q* in terms of *V*?

$$
\begin{aligned}
Q^{\pi}(s,a) &= \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right]
\end{aligned}
$$

Think-pair-share

Can we calculate *Q* in terms of *V*?

$$
\begin{aligned}
Q^{\pi}(s,a) &= \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a] \\
&= \mathbb{E}_{s',r}\left[r + \gamma \mathbb{E}_{\pi}[G_{t+1} | s_{t+1} = s']\right] \\
&= \qquad\qquad
\end{aligned}
$$

Write this expectation in terms of P(*s'*,*r*|*s*,*a*) and V^{π}

Optimal policies

Given a policy, π, we know how to compute the value function, V

But, how do we compute the optimal policy, π^*?

Optimal policies

Given a policy, π, we know how to compute the value function, V

But, how do we compute the optimal policy, π^*?

Definition: $\pi^*(s) = \arg\max_{\pi} V^{\pi}(s)$

Optimal policies

Given a pol ction, V

But, how d

Best out of all possible policies

Definition: $\pi^*(s) = \arg\max_{\pi} V^{\pi}(s)$

Optimal policies

Given a policy, π, we know how to compute the value function, V

But, how do we compute the optimal policy, π^*?

Definition: $\pi^*(s) = \arg\max_{\pi} V^{\pi}(s)$

Definition: $V^* = V^{\pi^*}, Q^* = Q^{\pi^*}$

Optimal policies

Given a policy, π, we know how to compute the value function, V

But, how do we compute the optimal policy, π^*?

Definition: $\pi^*(s) = \arg\max_\pi V^\pi(s)$

Definition: $V^* = V^{\pi^*}, Q^* = Q^{\pi^*}$

Bellman Equation: $\mathrm{Q}^\pi(s,a) = \sum_{s',r} P(s',r|s,a)\left[r + \gamma V^\pi(s')\right]$

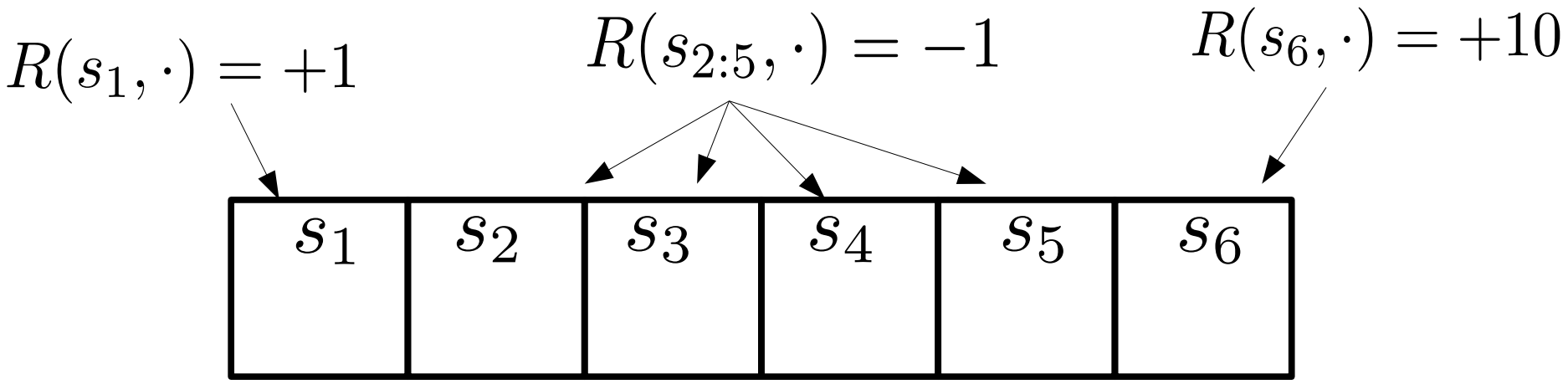
Bellman optimality condition:

$$\begin{aligned} Q^*(s,a) &= \sum_{s',r} P(s',r|s,a)\left[r + \gamma V^*(s')\right] \\ &= \sum_{s',r} P(s',r|s,a)\left[r + \gamma \max_{a'} Q^*(s',a')\right] \end{aligned}$$

Think-pair-share

Exercise 3.25 Give an equation for v_* in terms of q_*.

Value function example 3



Policy: 

Discount factor: $\gamma = 1$

Value fn:

7	6	7	8	9	10