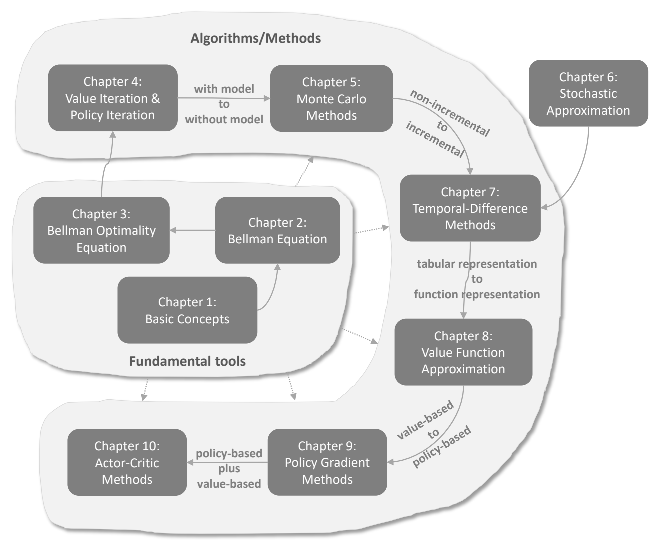
Lecture 1: Basic Concepts

Shiyu Zhao

School of Engineering, Westlake University

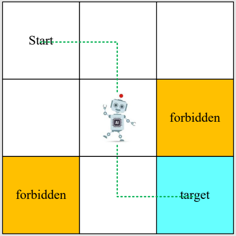
Outline

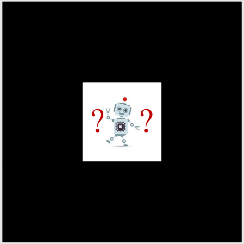

Algorithms/Methods
Chapter 4:
Value Iteration &
Policy Iteration
with model
to
without model
Chapter 5:
Monte Carlo
Methods
non-incremental
to
incremental
Chapter 6:
Stochastic
Approximation
Chapter 3:
Bellman Optimality
Equation
Chapter 2:
Bellman Equation
Chapter 1:
Basic Concepts
Fundamental tools
Chapter 7:
Temporal-Difference
Methods
tabular representation
to
function representation
Chapter 8:
Value Function
Approximation
value-based
to
policy-based
Chapter 10:
Actor-Critic
Methods
policy-based
plus
value-based
Chapter 9:
Policy Gradient
Methods

Contents

- First, introduce fundamental concepts in reinforcement learning (RL) by examples.
- Second, formalize the concepts in the context of Markov decision processes.

A grid-world example





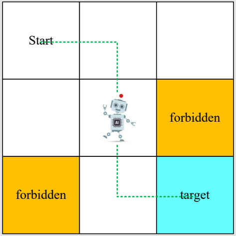
An illustrative example used throughout this course:

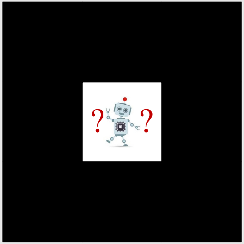
- Grid of cells: Accessible/forbidden/target cells, boundary.
- Very easy to understand and useful for illustration

Task:

- Given any starting area, find a "good" way to the target.
- How to define "good"? Avoid forbidden cells, detours, or boundary.

A grid-world example





An illustrative example used throughout this course:

- Grid of cells: Accessible/forbidden/target cells, boundary.
- Very easy to understand and useful for illustration

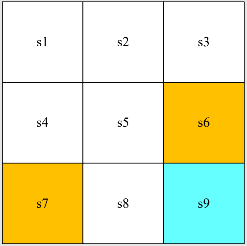
Task:

- Given any starting area, find a "good" way to the target.
- How to define "good"? Avoid forbidden cells, detours, or boundary.

State

State: The status of the agent with respect to the environment.

- For the grid-world example, the location of the agent is the state. There are nine possible locations and hence nine states: $s_1, s_2, \ldots, s_9$.



State space: the set of all states $\mathcal{S} = \{s_i\}_{i=1}^{9}$.

State: The status of the agent with respect to the environment.

- For the grid-world example, the location of the agent is the state. There are nine possible locations and hence nine states: $s_1, s_2, \ldots, s_9$.

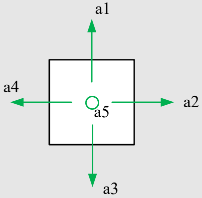
s1	s2	s3
s4	s5	s6
s7	s8	s9

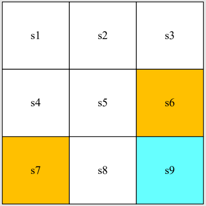
State space: the set of all states $\mathcal{S} = \{s_i\}_{i=1}^{9}$.

Action

Action: For each state, there are five possible actions: $a_1, \ldots, a_5$

- a_1: move upward;
- a_2: move rightward;
- a_3: move downward;
- a_4: move leftward;
- a_5: stay unchanged;



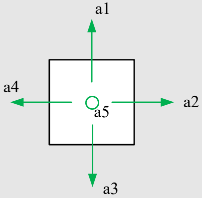


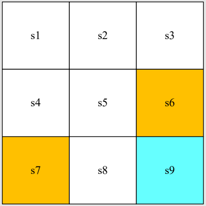
Action space of a state: the set of all possible actions of a state.
$\mathcal{A}(s_i) = \{a_i\}_{i=1}^{5}$.
Question: can different states have different sets of actions?

Action

Action: For each state, there are five possible actions: $a_1, \ldots, a_5$

- a_1: move upward;
- a_2: move rightward;
- a_3: move downward;
- a_4: move leftward;
- a_5: stay unchanged;





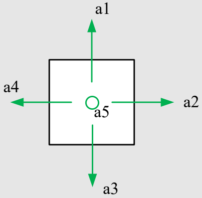
Action space of a state: the set of all possible actions of a state.
$\mathcal{A}(s_i) = \{a_i\}_{i=1}^5$.

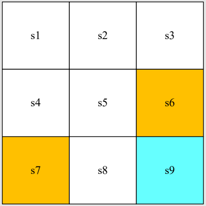
Question: can different states have different sets of actions?

Action

Action: For each state, there are five possible actions: $a_1, \ldots, a_5$

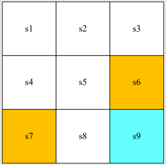
- a_1: move upward;
- a_2: move rightward;
- a_3: move downward;
- a_4: move leftward;
- a_5: stay unchanged;





Action space of a state: the set of all possible actions of a state.
$\mathcal{A}(s_i) = \{a_i\}_{i=1}^5$.
Question: can different states have different sets of actions?

State transition



When taking an action, the agent may move from one state to another. Such a process is called *state transition*.

- Example: At state s_1, if we choose action a_2, then what is the next state?

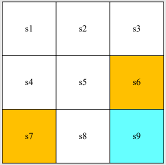
$$s_1 \xrightarrow{a_2} s_2$$

- Example: At state s_1, if we choose action a_1, then what is the next state?

$$s_1 \xrightarrow{a_1} s_1$$

- State transition describes the interaction with the environment.
- Question: Can we define the state transition in other ways? Simulation vs physics

State transition



When taking an action, the agent may move from one state to another. Such a process is called *state transition*.

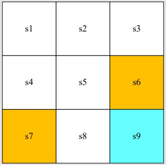
- Example: At state s_1, if we choose action a_2, then what is the next state?

$$s_1 \xrightarrow{a_2} s_2$$

- Example: At state s_1, if we choose action a_1, then what is the next state?

$$s_1 \xrightarrow{a_1} s_1$$

- State transition describes the interaction with the environment.
- Question: Can we define the state transition in other ways? Simulation vs physics



When taking an action, the agent may move from one state to another. Such a process is called *state transition*.

- Example: At state s_1, if we choose action a_2, then what is the next state?

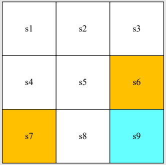
$$s_1 \xrightarrow{a_2} s_2$$

- Example: At state s_1, if we choose action a_1, then what is the next state?

$$s_1 \xrightarrow{a_1} s_1$$

- State transition describes the interaction with the environment.
- Question: Can we define the state transition in other ways? Simulation vs physics

State transition



When taking an action, the agent may move from one state to another. Such a process is called *state transition*.

- Example: At state s_1, if we choose action a_2, then what is the next state?

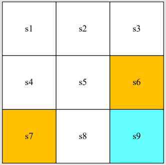
$$s_1 \xrightarrow{a_2} s_2$$

- Example: At state s_1, if we choose action a_1, then what is the next state?

$$s_1 \xrightarrow{a_1} s_1$$

- State transition describes the interaction with the environment.
- Question: Can we define the state transition in other ways? Simulation vs physics

State transition



When taking an action, the agent may move from one state to another. Such a process is called *state transition*.

- Example: At state s_1, if we choose action a_2, then what is the next state?

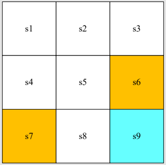
$$s_1 \xrightarrow{a_2} s_2$$

- Example: At state s_1, if we choose action a_1, then what is the next state?

$$s_1 \xrightarrow{a_1} s_1$$

- State transition describes the interaction with the environment.
- Question: Can we define the state transition in other ways? Simulation vs physics

State transition



Pay attention to *forbidden areas*: Example: at state s_5, if we choose action a_2, then what is the next state?

- Case 1: the forbidden area is accessible but with penalty. Then,

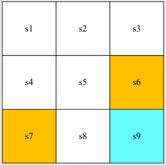
$$s_5 \xrightarrow{a_2} s_6$$

- Case 2: the forbidden area is inaccessible (e.g., surrounded by a wall)

$$s_5 \xrightarrow{a_2} s_5$$

We consider the first case, which is more general and challenging.

State transition



Pay attention to *forbidden areas*: Example: at state s_5, if we choose action a_2, then what is the next state?

- Case 1: the forbidden area is accessible but with penalty. Then,

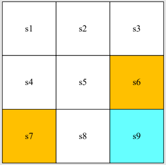
$$s_5 \xrightarrow{a_2} s_6$$

- Case 2: the forbidden area is inaccessible (e.g., surrounded by a wall)

$$s_5 \xrightarrow{a_2} s_5$$

We consider the first case, which is more general and challenging.

State transition



Pay attention to *forbidden areas*: Example: at state s_5, if we choose action a_2, then what is the next state?

- Case 1: the forbidden area is accessible but with penalty. Then,

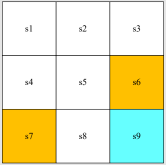
$$s_5 \xrightarrow{a_2} s_6$$

- Case 2: the forbidden area is inaccessible (e.g., surrounded by a wall)

$$s_5 \xrightarrow{a_2} s_5$$

We consider the first case, which is more general and challenging.

State transition



Pay attention to *forbidden areas*: Example: at state s_5, if we choose action a_2, then what is the next state?

- Case 1: the forbidden area is accessible but with penalty. Then,

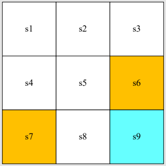
$$s_5 \xrightarrow{a_2} s_6$$

- Case 2: the forbidden area is inaccessible (e.g., surrounded by a wall)

$$s_5 \xrightarrow{a_2} s_5$$

We consider the first case, which is more general and challenging.

State transition

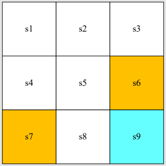


Tabular representation: We can use a table to describe the state transition:

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

Can only represent *deterministic* cases.

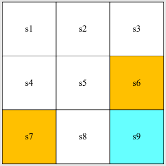
State transition



Tabular representation: We can use a table to describe the state transition:

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

Can only represent *deterministic* cases.

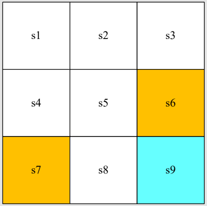


Tabular representation: We can use a table to describe the state transition:

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

Can only represent *deterministic* cases.

State transition



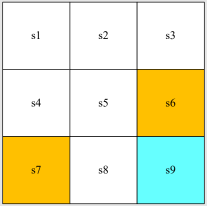
State transition probability: use probability to describe state transition!

- Intuition: At state s_1, if we choose action a_2, the next state is s_2.
- Math:

$$p(s_2|s_1, a_2) = 1$$
$$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$$

Here it is a deterministic case. The state transition could be stochastic (for example, wind gust).

State transition



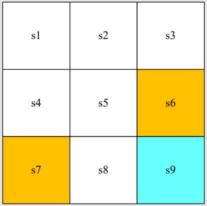
State transition probability: use probability to describe state transition!

- Intuition: At state s_1, if we choose action a_2, the next state is s_2.
- Math:

$$p(s_2|s_1, a_2) = 1$$
$$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$$

Here it is a deterministic case. The state transition could be stochastic (for example, wind gust).

State transition



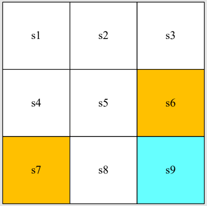
State transition probability: use probability to describe state transition!

- Intuition: At state s_1, if we choose action a_2, the next state is s_2.
- Math:

$$p(s_2|s_1, a_2) = 1$$
$$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$$

Here it is a deterministic case. The state transition could be stochastic (for example, wind gust).

State transition



State transition probability: use probability to describe state transition!

- Intuition: At state s_1, if we choose action a_2, the next state is s_2.
- Math:

$$p(s_2|s_1, a_2) = 1$$
$$p(s_i|s_1, a_2) = 0 \quad \forall i \neq 2$$

Here it is a deterministic case. The state transition could be stochastic (for example, wind gust).

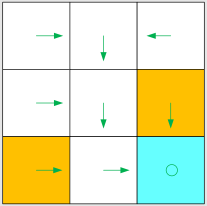
Policy

Policy tells the agent what actions to take at a state.

Policy

Policy tells the agent what actions to take at a state.

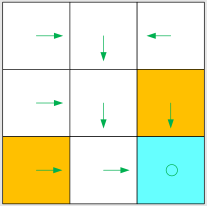
Intuitive representation: We use *arrows* to describe a policy.



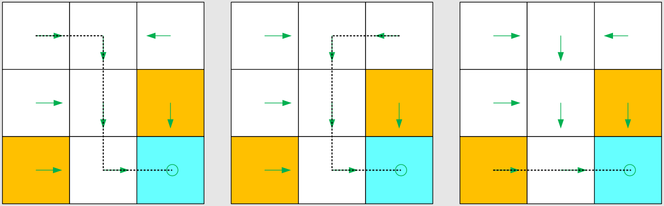
Policy

Policy tells the agent what actions to take at a state.

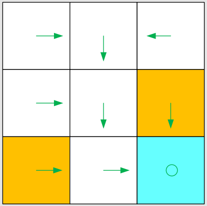
Intuitive representation: We use *arrows* to describe a policy.



Based on this policy, we get the following trajectories with different starting points.



Policy



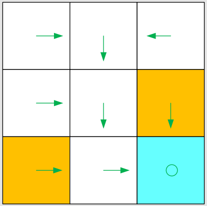
Mathematical representation: using conditional probability

For example, for state s_1:

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 1$$
$$\pi(a_3|s_1) = 0$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$

It is a deterministic policy.

Policy



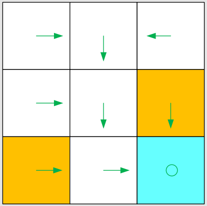
Mathematical representation: using conditional probability

For example, for state s_1:

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 1$$
$$\pi(a_3|s_1) = 0$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$

It is a deterministic policy.

Policy



Mathematical representation: using conditional probability

For example, for state s_1:

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 1$$
$$\pi(a_3|s_1) = 0$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$

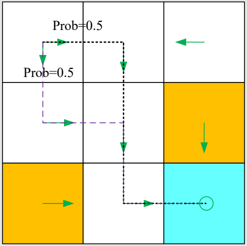
It is a deterministic policy.

Policy

There are stochastic policies.

There are stochastic policies.

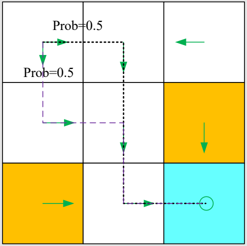
For example:



Policy

There are stochastic policies.

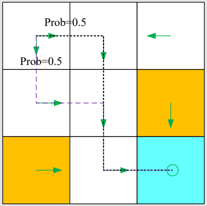
For example:



In this policy, for s_1:

$$\pi(a_1|s_1) = 0$$
$$\pi(a_2|s_1) = 0.5$$
$$\pi(a_3|s_1) = 0.5$$
$$\pi(a_4|s_1) = 0$$
$$\pi(a_5|s_1) = 0$$

Policy



Tabular representation of a policy: how to use this table.

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	0	0.5	0.5	0	0
s_2	0	0	1	0	0
s_3	0	0	0	1	0
s_4	0	1	0	0	0
s_5	0	0	1	0	0
s_6	0	0	1	0	0
s_7	0	1	0	0	0
s_8	0	1	0	0	0
s_9	0	0	0	0	1

Can represent either *deterministic* or *stochastic* cases.

Reward

Reward is one of the most unique concepts of RL.

Reward: a real number we get after taking an action.

- A positive reward represents encouragement to take such actions.
- A negative reward represents punishment to take such actions.

Questions:

- What about a zero reward? No punishment.
- Can positive mean punishment? Yes.

Reward

Reward is one of the most unique concepts of RL.

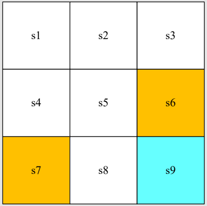
Reward: a real number we get after taking an action.

- A positive reward represents encouragement to take such actions.
- A negative reward represents punishment to take such actions.

Questions:

- What about a zero reward? No punishment.
- Can positive mean punishment? Yes.

Reward



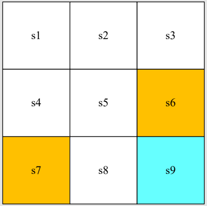
In the grid-world example, the rewards are designed as follows:

- If the agent attempts to get out of the boundary, let $r_{\text{bound}} = -1$
- If the agent attempts to enter a forbidden cell, let $r_{\text{forbid}} = -1$
- If the agent reaches the target cell, let $r_{\text{target}} = +1$
- Otherwise, the agent gets a reward of $r = 0$.

Reward can be interpreted as a **human-machine interface**, with which we can guide the agent to behave as what we expect.

For example, with the above designed rewards, the agent will try to avoid getting out of the boundary or stepping into the forbidden cells.

Reward

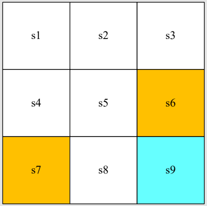


In the grid-world example, the rewards are designed as follows:

- If the agent attempts to get out of the boundary, let $r_{\text{bound}} = -1$
- If the agent attempts to enter a forbidden cell, let $r_{\text{forbid}} = -1$
- If the agent reaches the target cell, let $r_{\text{target}} = +1$
- Otherwise, the agent gets a reward of $r = 0$.

Reward can be interpreted as a **human-machine interface**, with which we can guide the agent to behave as what we expect.
For example, with the above designed rewards, the agent will try to avoid getting out of the boundary or stepping into the forbidden cells.

Reward

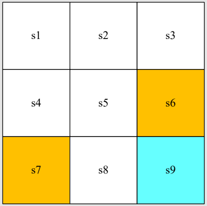


Tabular representation of *reward transition*: how to use the table?

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	r_{bound}	0	0	r_{bound}	0
s_2	r_{bound}	0	0	0	0
s_3	r_{bound}	r_{bound}	r_{forbid}	0	0
s_4	0	0	r_{forbid}	r_{bound}	0
s_5	0	r_{forbid}	0	0	0
s_6	0	r_{bound}	r_{target}	0	r_{forbid}
s_7	0	0	r_{bound}	r_{bound}	r_{forbid}
s_8	0	r_{target}	r_{bound}	r_{forbid}	0
s_9	r_{forbid}	r_{bound}	r_{bound}	0	r_{target}

Can only represent *deterministic* cases.

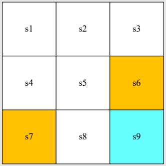
Reward



Tabular representation of *reward transition*: how to use the table?

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (unchanged)
s_1	r_{bound}	0	0	r_{bound}	0
s_2	r_{bound}	0	0	0	0
s_3	r_{bound}	r_{bound}	r_{forbid}	0	0
s_4	0	0	r_{forbid}	r_{bound}	0
s_5	0	r_{forbid}	0	0	0
s_6	0	r_{bound}	r_{target}	0	r_{forbid}
s_7	0	0	r_{bound}	r_{bound}	r_{forbid}
s_8	0	r_{target}	r_{bound}	r_{forbid}	0
s_9	r_{forbid}	r_{bound}	r_{bound}	0	r_{target}

Can only represent *deterministic* cases.



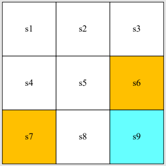
Mathematical description: conditional probability

- Intuition: At state s_1, if we choose action a_1, the reward is -1.
- Math: $p(r=-1|s_1,a_1)=1$ and $p(r\neq -1|s_1,a_1)=0$

Remarks:

- Here it is a deterministic case. The reward transition could be stochastic. For example, if you study hard, you will get rewards. But how much is uncertain.

Reward



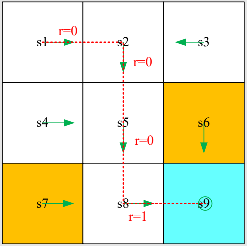
Mathematical description: conditional probability

- Intuition: At state s_1, if we choose action a_1, the reward is -1.
- Math: $p(r = -1|s_1, a_1) = 1$ and $p(r \neq -1|s_1, a_1) = 0$

Remarks:

- Here it is a deterministic case. The reward transition could be stochastic. For example, if you study hard, you will get rewards. But how much is uncertain.

Trajectory and return



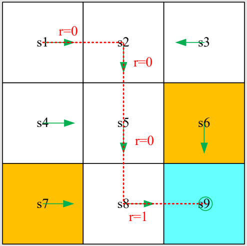
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

Trajectory and return



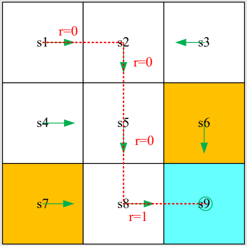
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

Trajectory and return



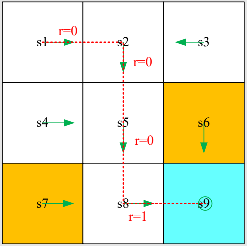
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

Trajectory and return



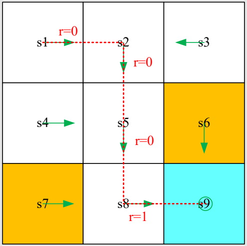
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

Trajectory and return



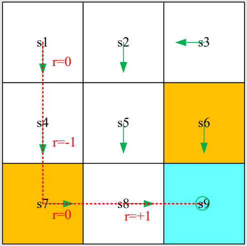
A *trajectory* is a state-action-reward chain:

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

The *return* of this trajectory is the sum of all the rewards collected along the trajectory:

$$\text{return} = 0 + 0 + 0 + 1 = 1$$

Trajectory and return

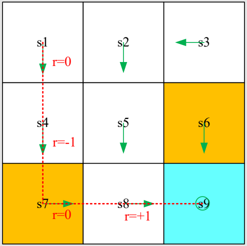


A different policy gives a different trajectory:

$$s_1 \xrightarrow[r=0]{a_3} s_4 \xrightarrow[r=-1]{a_3} s_7 \xrightarrow[r=0]{a_2} s_8 \xrightarrow[r=+1]{a_2} s_9$$

The return of this path is:

$$\text{return} = 0 - 1 + 0 + 1 = 0$$



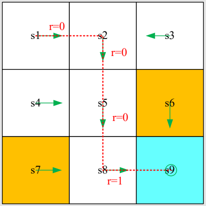
A different policy gives a different trajectory:

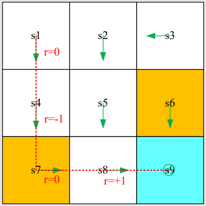
$$s_1 \xrightarrow[r=0]{a_3} s_4 \xrightarrow[r=-1]{a_3} s_7 \xrightarrow[r=0]{a_2} s_8 \xrightarrow[r=+1]{a_2} s_9$$

The return of this path is:

$$\text{return} = 0 - 1 + 0 + 1 = 0$$

Trajectory and return

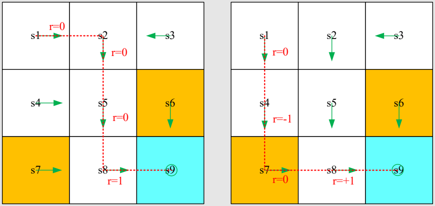




Which policy is better?

- **Intuition**: the first is better, because it avoids the forbidden areas.
- **Mathematics**: the first one is better, since it has a greater return!
- Return could be used to evaluate whether a policy is good or not (see details in the next lecture)!

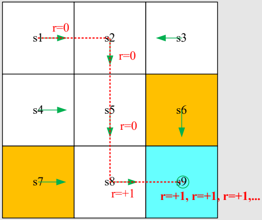
Trajectory and return



Which policy is better?

- **Intuition**: the first is better, because it avoids the forbidden areas.
- **Mathematics**: the first one is better, since it has a greater return!
- Return could be used to evaluate whether a policy is good or not (see details in the next lecture)!

Discounted return



A trajectory may be infinite:

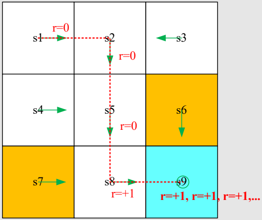
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges!

Discounted return



A trajectory may be infinite:

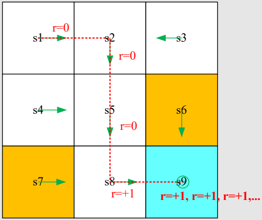
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges!

Discounted return



A trajectory may be infinite:

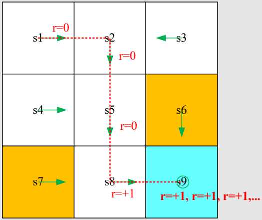
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges!

Discounted return



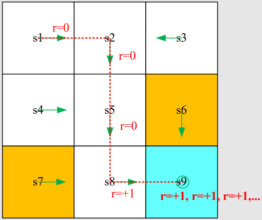
A trajectory may be infinite:

$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges!



A trajectory may be infinite:

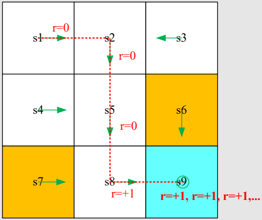
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges!

Discounted return



A trajectory may be infinite:

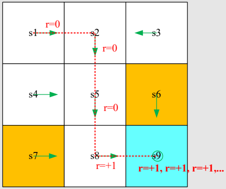
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \dots$$

The return is

$$\text{return} = 0 + 0 + 0 + 1 + 1 + 1 + \dots = \infty$$

The definition is invalid since the return diverges!

Discounted return



Need to introduce a *discount rate* $\gamma \in [0, 1)$:

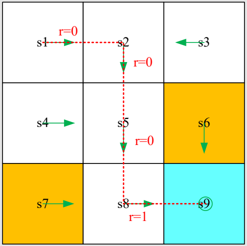
$$\begin{aligned}\text{discounted return} &= 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \ldots \\ &= \gamma^3 (1 + \gamma + \gamma^2 + \ldots) = \gamma^3 \frac{1}{1-\gamma}.\end{aligned}$$

Roles: 1) the sum becomes finite; 2) balance the far and near future rewards:

- If γ is close to 0, the value of the discounted return is dominated by the rewards obtained in the near future.
- If γ is close to 1, the value of the discounted return is dominated by the rewards obtained in the far future.

Episode

When interacting with the environment following a policy, the agent may stop at some *terminal states*. The resulting trajectory is called an *episode* (or a trial).



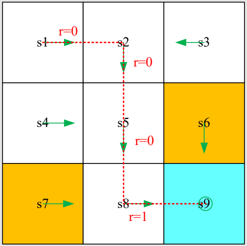
Example: episode

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

An episode is usually assumed to be a finite trajectory. Tasks with episodes are called *episodic tasks*.

Episode

When interacting with the environment following a policy, the agent may stop at some *terminal states*. The resulting trajectory is called an *episode* (or a trial).



Example: episode

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

An episode is usually assumed to be a finite trajectory. Tasks with episodes are called *episodic tasks*.

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Episode

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Episode

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Episode

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Episode

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Episode

Some tasks may have no terminal states, meaning the interaction with the environment will never end. Such tasks are called *continuing tasks*.

In the grid-world example, should we stop after arriving the target?

In fact, we can treat episodic and continuing tasks in a unified mathematical way by converting episodic tasks to continuing tasks.

- Option 1: Treat the target state as a special absorbing state. Once the agent reaches an absorbing state, it will never leave. The consequent rewards $r = 0$.
- Option 2: Treat the target state as a normal state with a policy. The agent can still leave the target state and gain $r = +1$ when entering the target state.

We consider option 2 in this course so that we don't need to distinguish the target state from the others and can treat it as a normal state.

Markov decision process (MDP)

Key elements of MDP:

- Sets:
  - State: the set of states $\mathcal{S}$
  - Action: the set of actions $\mathcal{A}(s)$ is associated for state $s \in \mathcal{S}$.
  - Reward: the set of rewards $\mathcal{R}(s, a)$.
- Probability distribution (or called system model):
  - State transition probability: at state s, taking action a, the probability to transit to state s' is $p(s'|s, a)$
  - Reward probability: at state s, taking action a, the probability to get reward r is $p(r|s, a)$
- Policy: at state s, the probability to choose action a is $\pi(a|s)$
- *Markov property*: memoryless property

$$p(s_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(s_{t+1}|a_t, s_t),$$
$$p(r_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(r_{t+1}|a_t, s_t).$$

All the concepts introduced in this lecture can be put in the framework in MDP.

Markov decision process (MDP)

Key elements of MDP:

- Sets:
  - State: the set of states $\mathcal{S}$
  - Action: the set of actions $\mathcal{A}(s)$ is associated for state $s \in \mathcal{S}$.
  - Reward: the set of rewards $\mathcal{R}(s, a)$.
- Probability distribution (or called system model):
  - State transition probability: at state s, taking action a, the probability to transit to state s' is $p(s'|s, a)$
  - Reward probability: at state s, taking action a, the probability to get reward r is $p(r|s, a)$
- Policy: at state s, the probability to choose action a is $\pi(a|s)$
- *Markov property*: memoryless property

$$
\begin{aligned}
p(s_{t+1}|a_t, s_t, \ldots, a_0, s_0) &= p(s_{t+1}|a_t, s_t), \\
p(r_{t+1}|a_t, s_t, \ldots, a_0, s_0) &= p(r_{t+1}|a_t, s_t).
\end{aligned}
$$

All the concepts introduced in this lecture can be put in the framework in MDP.

Key elements of MDP:

- Sets:
  - State: the set of states $\mathcal{S}$
  - Action: the set of actions $\mathcal{A}(s)$ is associated for state $s \in \mathcal{S}$.
  - Reward: the set of rewards $\mathcal{R}(s, a)$.
- Probability distribution (or called system model):
  - State transition probability: at state s, taking action a, the probability to transit to state s' is $p(s'|s, a)$
  - Reward probability: at state s, taking action a, the probability to get reward r is $p(r|s, a)$
- Policy: at state s, the probability to choose action a is $\pi(a|s)$
- *Markov property*: memoryless property

$$p(s_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(s_{t+1}|a_t, s_t),$$
$$p(r_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(r_{t+1}|a_t, s_t).$$

All the concepts introduced in this lecture can be put in the framework in MDP.

Markov decision process (MDP)

Key elements of MDP:

- Sets:
  - State: the set of states $\mathcal{S}$
  - Action: the set of actions $\mathcal{A}(s)$ is associated for state $s \in \mathcal{S}$.
  - Reward: the set of rewards $\mathcal{R}(s, a)$.
- Probability distribution (or called system model):
  - State transition probability: at state s, taking action a, the probability to transit to state s' is $p(s'|s, a)$
  - Reward probability: at state s, taking action a, the probability to get reward r is $p(r|s, a)$
- Policy: at state s, the probability to choose action a is $\pi(a|s)$
- *Markov property*: memoryless property

$$p(s_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(s_{t+1}|a_t, s_t),$$
$$p(r_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(r_{t+1}|a_t, s_t).$$

All the concepts introduced in this lecture can be put in the framework in MDP.

Markov decision process (MDP)

Key elements of MDP:

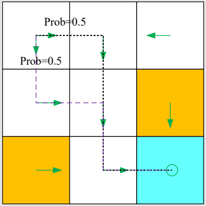
- Sets:
  - State: the set of states $\mathcal{S}$
  - Action: the set of actions $\mathcal{A}(s)$ is associated for state $s \in \mathcal{S}$.
  - Reward: the set of rewards $\mathcal{R}(s, a)$.
- Probability distribution (or called system model):
  - State transition probability: at state s, taking action a, the probability to transit to state s' is $p(s'|s, a)$
  - Reward probability: at state s, taking action a, the probability to get reward r is $p(r|s, a)$
- Policy: at state s, the probability to choose action a is $\pi(a|s)$
- *Markov property*: memoryless property

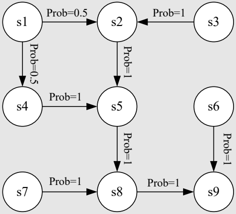
$$p(s_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(s_{t+1}|a_t, s_t),$$
$$p(r_{t+1}|a_t, s_t, \ldots, a_0, s_0) = p(r_{t+1}|a_t, s_t).$$

All the concepts introduced in this lecture can be put in the framework in MDP.

Markov decision process (MDP)

The grid world could be abstracted as a more general model, *Markov process*.





The circles represent states and the links with arrows represent the state transition.

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s, a)$
- Reward, reward probability $p(r|s, a)$
- Trajectory, episode, return, discounted return
- Markov decision process

Summary

By using grid-world examples, we demonstrated the following key concepts:

- State
- Action
- State transition, state transition probability $p(s'|s,a)$
- Reward, reward probability $p(r|s,a)$
- Trajectory, episode, return, discounted return
- Markov decision process