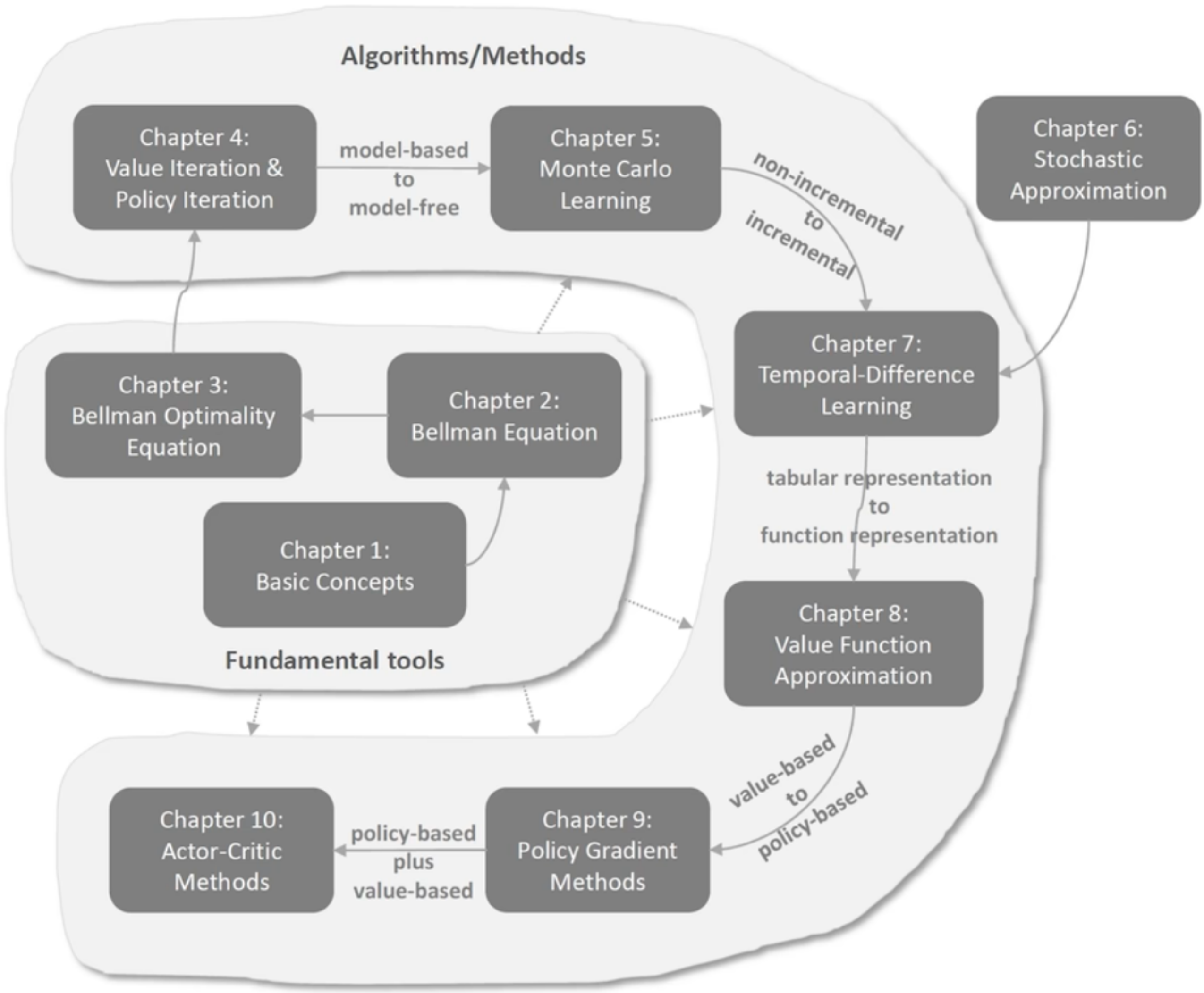
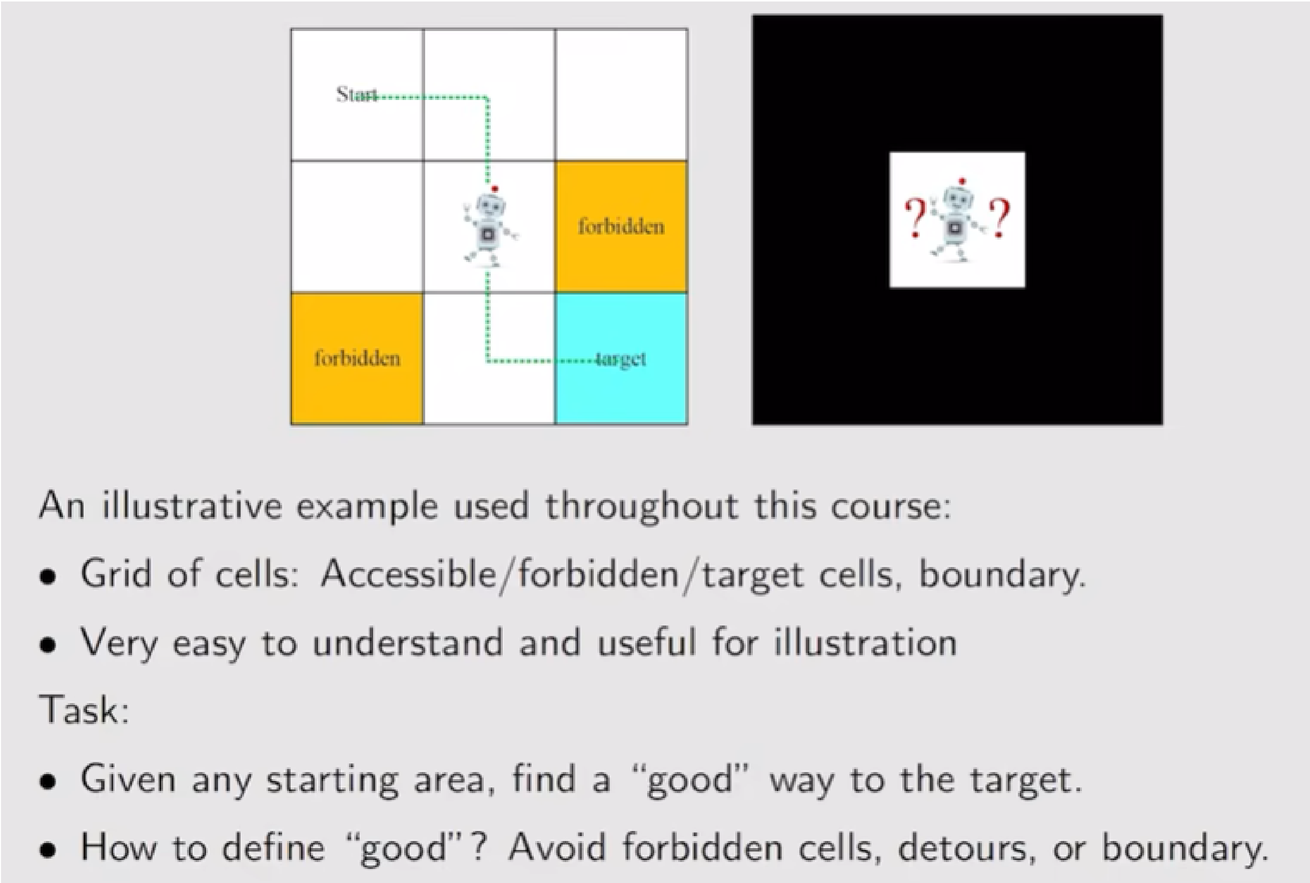
Lec1-Basic Concepts in RL



A grid-world example



Basic Concept——Part 1

State:The status of agent with respect to the environment.
(Agent相对于环境的变化)

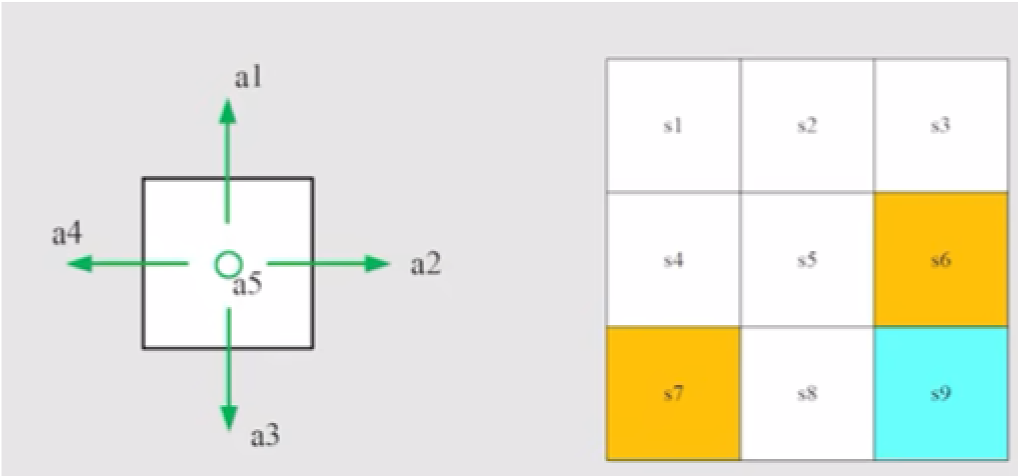
- For the gird-world example , the location of the agent is the state . There are nine possible locations and hence nine states. $s_1, s_2, \ldots, s_9$

State space:The set of all states $\mathrm{S} = \{s_i\}_{i=1}^{9}$

Action:For each state , there are five possible actions: $a_1, a_2, \ldots, a_5$

- a_1:move upwards
- a_2:move rightwards
- a_3:move downwards
- a_4:move leftwards

- a_5:stay unchanged



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

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

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

$$s_1 \xrightarrow{a_2} s_2$$

- 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 defines the interaction with the environment.

Forbidden Area: At state *s_5* , if we choose action *a_2* , then what is the next state ?

- Case1: the forbidden area is accessible but with penalty . Then,

$$s_5 \xrightarrow{a_2} s_6$$

- Case2: 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.

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

	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	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

（比如，在状态s_1采取行动a_1，有可能反弹到不同的位置，这样就无法用表格表示）

State transition probability: use probability to describe state transition

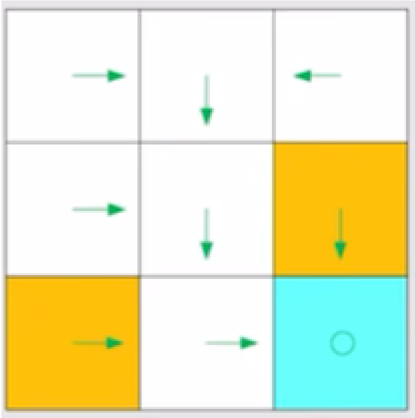
- Intuition : At state s_1 , if we choose action a_2 , the next state is s_2
- Math:

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

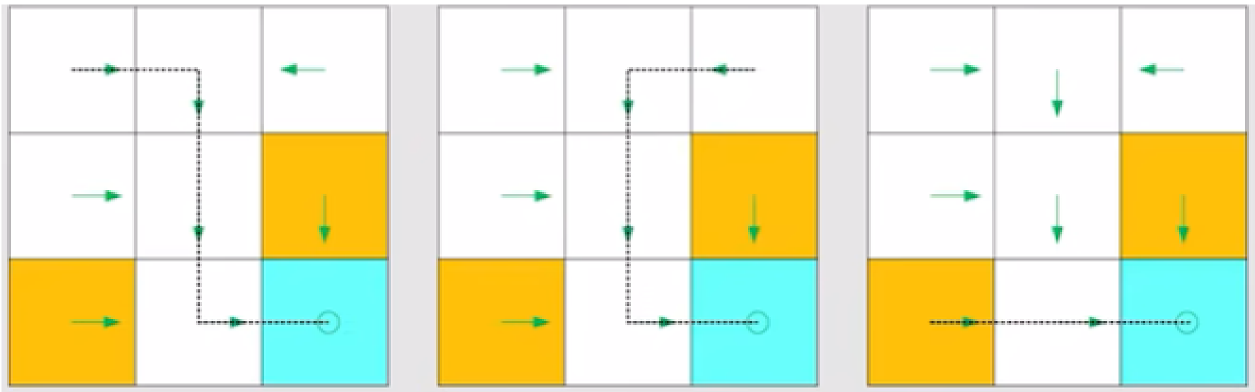
Here it is a deterministic case . The state transition could be stochastic (for example , wind gust).

Policy : tell the agent what actions to take at a state.

- Intuitive representation : The arrows demonstrate a policy.



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



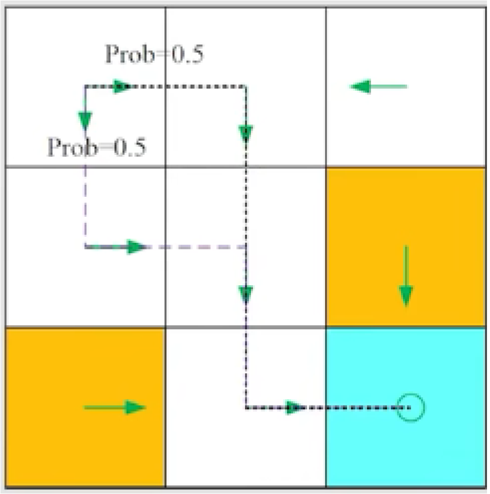
- Mathematical representation : using conditional probability
For example , for state s_1:

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

It is a deterministic policy.

（在强化学习中，π指的是策略，本质上是条件概率，指定了任何一个状态下任何一个action的概率）

There are stochastic policies.
For example:



In this policy , for s_1:

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

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

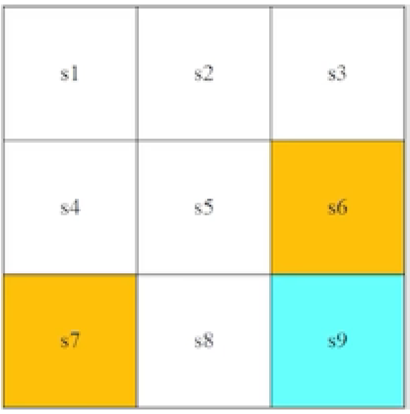
	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	a_5 (unchanged)
s_1	0	0.5	0.5	0	0
s_2	0	0	1	0	0
s_3	0	0	0	1	0
s_4	0	1	0	0	0
s_5	0	0	1	0	0
s_6	0	0	1	0	0
s_7	0	1	0	0	0
s_8	0	1	0	0	0
s_9	0	0	0	0	1

Can represent either deterministic or stochastic cases.

Basic Concept——Part 2

Reward: a real number we get after taking an action (It is one of the most unique concepts of RL)

- 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.



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

- If the agent attempts to get out of the boundary , let $r_{bound} = -1$
- If the agent attempts to enter a forbidden cell , let $r_{forbid} = -1$
- If the agent reaches the target cell , let $r_{target} = +1$
- 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 boundary or stepping into the forbidden cells.

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

	a_1 (upwards)	a_2 (rightwards)	a_3 (downwards)	a_4 (leftwards)	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

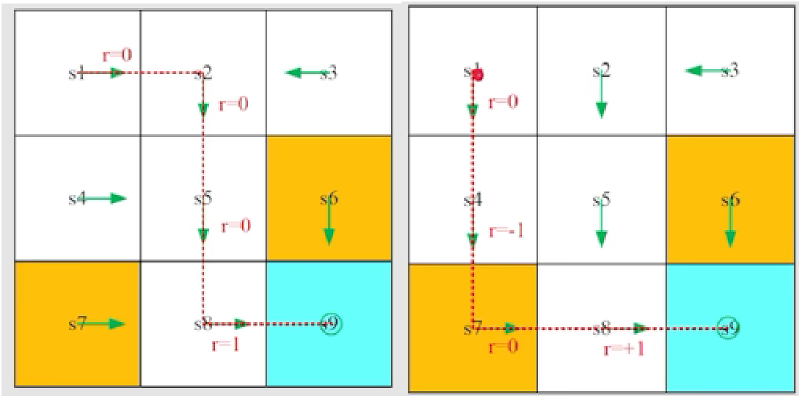
- Intuition : At state s_1 , if we choose action a_1 , the reward is -1

- 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 reward . But how much is uncertain.
- The reward depends on the state and action , but not the next state(for example , considers_1,a_1ands_1,a_5)

  (在s_1状态下采取两种行动，下一个状态还是s_1，但是两者的reward不同)



Trajectory: 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 he trajectory:

$$return = 0+0+0+1$$

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:

$$return = 0-1+0+1=0$$

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

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

$$return = 0 + 0 + 0 + 1 + 1 + 1 + \ldots = \infty$$

The definition is invalid since the return diverges(发散)!

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

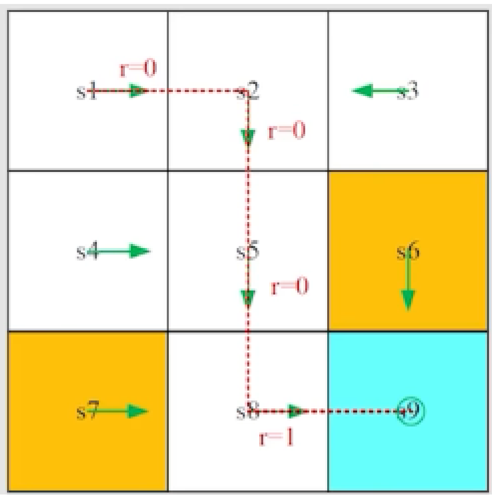
$$\text{Discounted\_return} = 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \ldots = \gamma^3 \left(1 + \gamma + \gamma^2 + \ldots\right) = \gamma^3 \frac{1}{1-\gamma}$$

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.
  (近视与远视)

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.

Markov decision process(MDP)

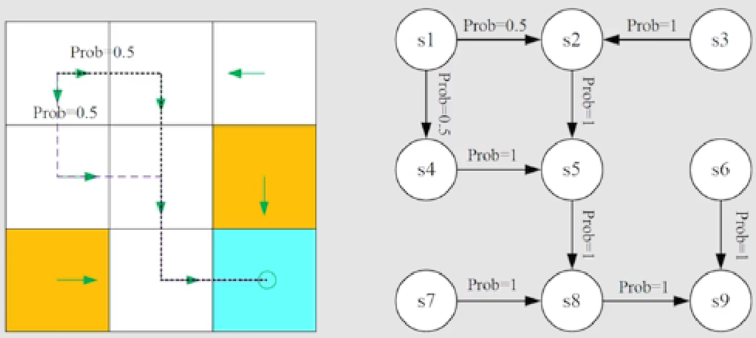
- Sets:
  - State : the set of states S
  - Action : the set of actions $A(s)$ is associated for state $s \in S$
  - Reward : the set of rewards $R(s, a)$
- Probability distribution:
  - State transition probability : at state s, taking action a ,the probability to transit to state s' is $p(s'|s, a)$
  - 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

$$
p(s_{t+1}|a_{t+1}, s_t, \ldots, a_1, s_0) = p(s_{t+1}|a_{t+1}, s_t)
$$
$$
p(r_{t+1}|a_{t+1}, s_t, \ldots, a_1, s_0) = p(r_{t+1}|a_{t+1}, s_t)
$$

All the concepts introduced in this lecture can be put in the framework in 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.
Markov decision process becomes Markov process once the policy is given!

By using grid-world examples , we demonstrate 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