Lecture 1: Introduction and Basic Concepts

Dr. Wen Fuxi

Teaching Team

Dr. Wen Fuxi

Ph.D. - School of Electrical and Electronic Engineering | 2013
Nanyang Technological University, Singapore

- Email: wenfuxi@tsinghua.edu.cn
- WeChat: 18618188257

Work Experience

- 2021 - Now | Associate Professor (research-track)
  School of Vehicle and Mobility, Tsinghua University, China
- 2017 - 2020 | Marie Skłodowska-Curie Individual Fellow
  Chalmers University of Technology, Sweden
- 2015 - 2017 | Postdoctoral
  Singapore University of Technology and Design, Singapore
- 2015 - 2015 | Research Associate
  University of Sheffield, UK
- 2011 - 2014 | Research Associate, Research Fellow
  Nanyang Technological University, Singapore

Why you should consider taking this course?

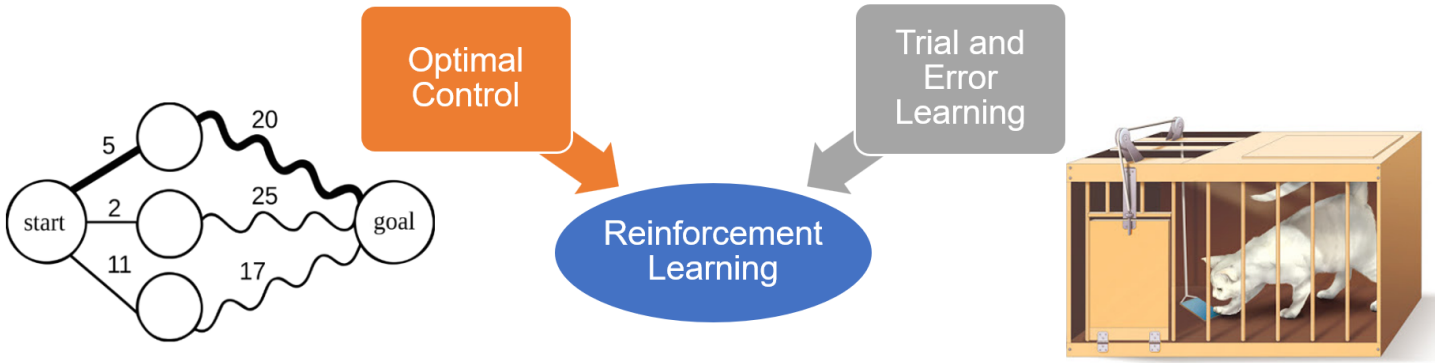
- There will be quite a few THEOREMS and PROOFS ...
  - Promote a deeper understanding of scientific/engineering results.
- Nonrigorous/heuristic from time to time
  - "Nonrigorous" but grounded in rigorous theory
  - Help develop intuition

Supervised learning

Given training data, make prediction on unseen data:

Primarily deal with **pattern recognition**

Reinforcement Learning



1. Optimal Control

   Find a control law for a given dynamic system that minimizes a certain criterion.

2. Trial and Error Learning

   Repeat behaviors that result in rewards and avoid behaviors that result in punishment.

Reinforcement Learning

In RL, an agent learns by interacting with an environment.

1. - No training data
2. - Maximize total rewards
3. - Trial-and-error
4. - Sequential and online

Deal with decision making, sometimes with constraints

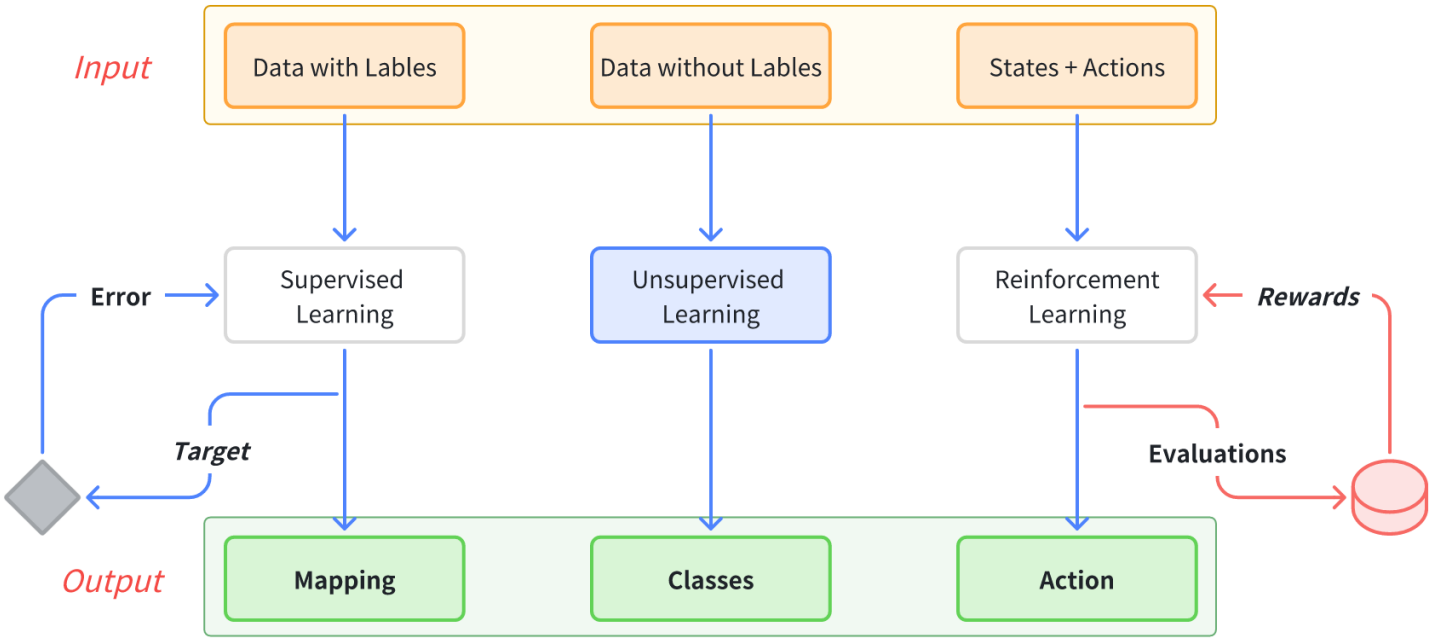
Learn from the past to predict and optimize future performance

Supervised, Unsupervised, Reinforcement learning

- Supervised: Labelled data with an external teacher
- Unsupervised: Find hidden structure in unlabeled data

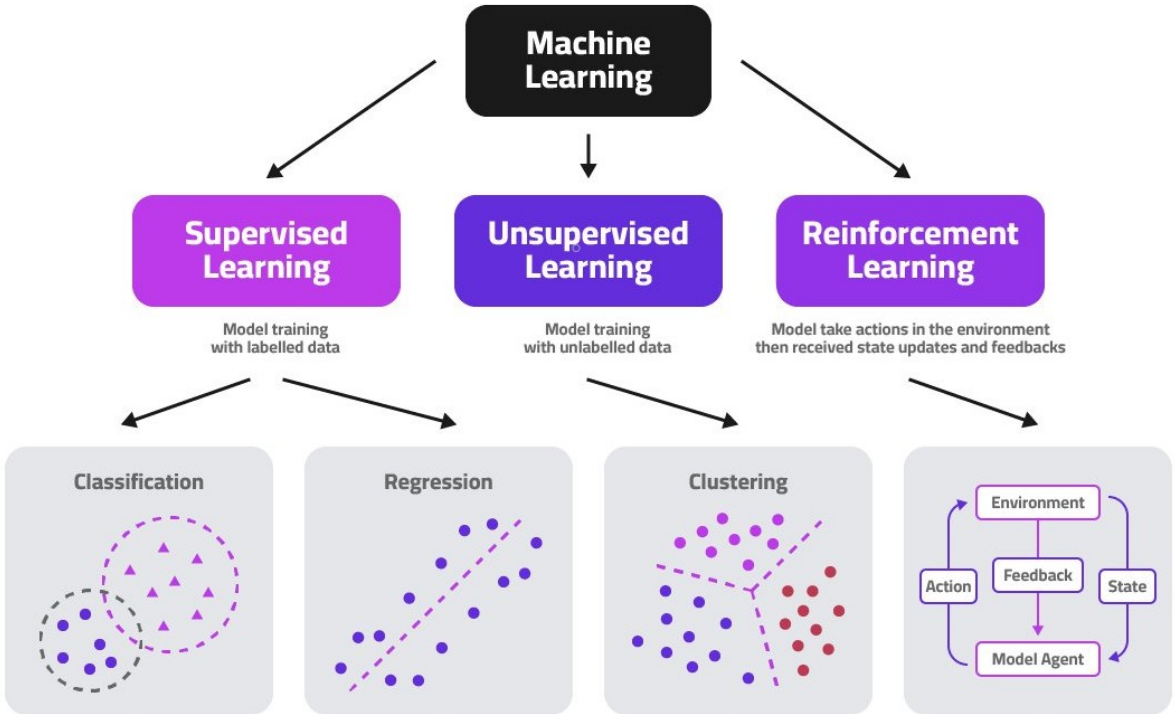
  Reinforcement learning:

  - Not enough labelled data, not even any data
  - Feedback by reward or punishment signals



Comparison between supervised, unsupervised, and reinforcement learning

	Supervised	Unsupervised	Reinforcement learning
Definition	Makes predictions from data	Segments and groups data	Reward-punishment system and interactive environment
Types of data	Labeled data	Unlabeled data	Acts according to a policy with a final goal to reach (No or predefined data)
Problems	Regression and classification	Association and Clustering	Exploitation or Exploration
Aim	Calculate outcomes	Discover underlying patterns	Learn a series of action



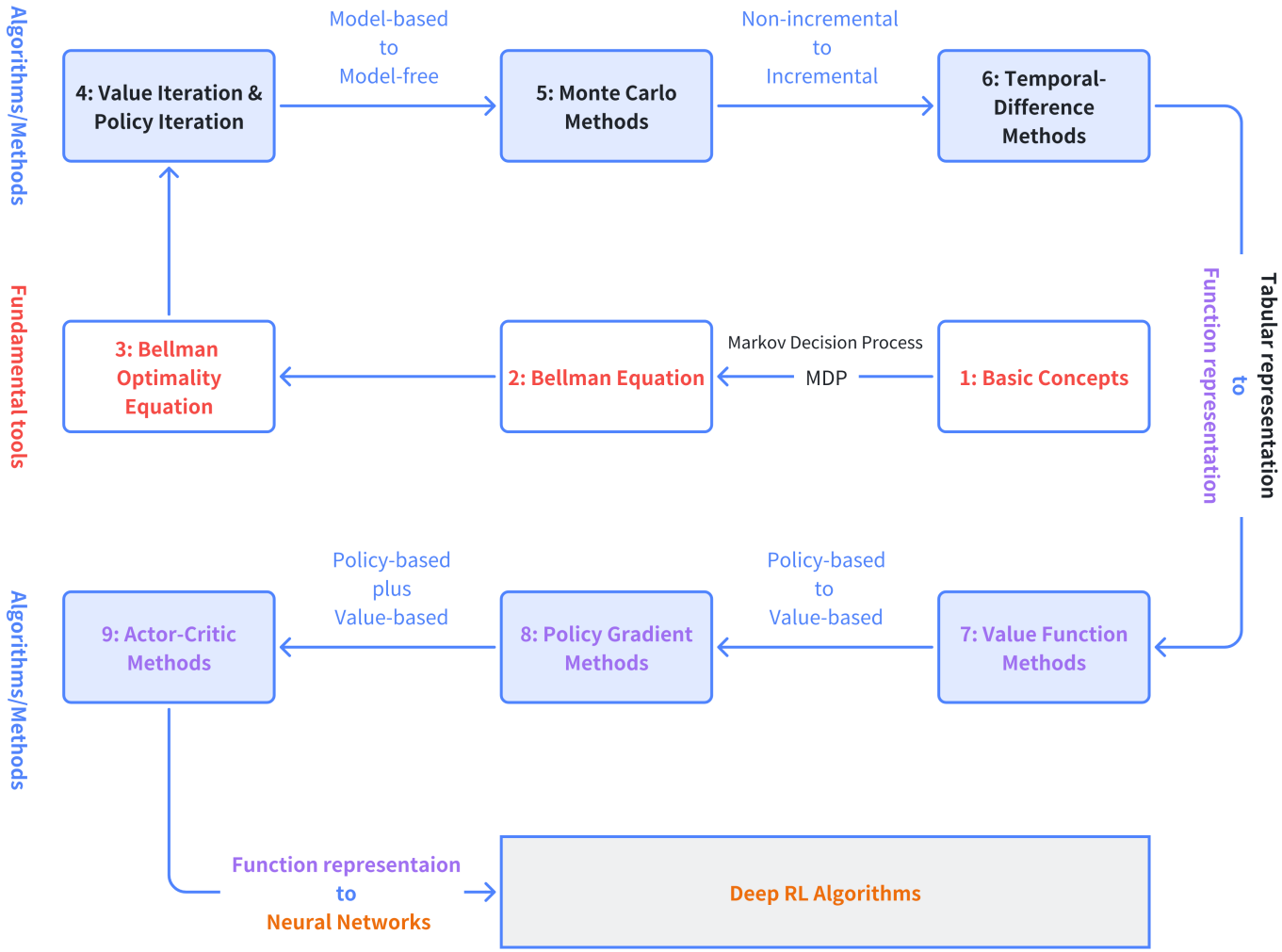
Key Differences Between RL and RLHF (from Human Feedback)

1. Source of Feedback:
   - RL: The feedback comes from the environment, typically through numerical rewards or penalties based on the agent's actions.
   - RLHF: Humans provide feedback, offering more subjective and nuanced guidance.
2. Learning Focus:
   - RL: The agent focuses solely on maximizing rewards within the environment.
   - RLHF: The agent learns to align its actions with human values and preferences, ensuring more complex tasks are performed appropriately.
3. Use Cases:
   - RL: Ideal for tasks with clear rewards and penalties, such as games or robotics with straightforward objectives.
   - RLHF: Best suited for tasks requiring ethical decision-making or where human values are critical, such as autonomous vehicles or natural language generation.

Tentative topics

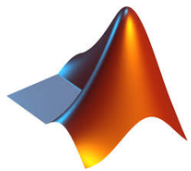
1. Basic Concepts
2. State Values and Bellman Equation
3. Optimal State Values and Bellman Optimality Equation
4. Value Iteration and Policy Iteration
5. Monte Carlo Methods
6. Stochastic Approximation
7. Temporal-Difference Methods
8. Value Function Methods
9. Policy Gradient Methods
10. Actor-Critic Methods
11. Deep RL
12. Typical Deep RL Algorithms
13. Paper sharing seminar

Course Structure

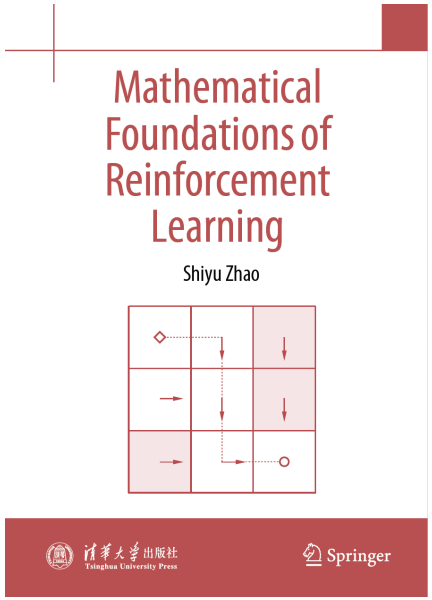


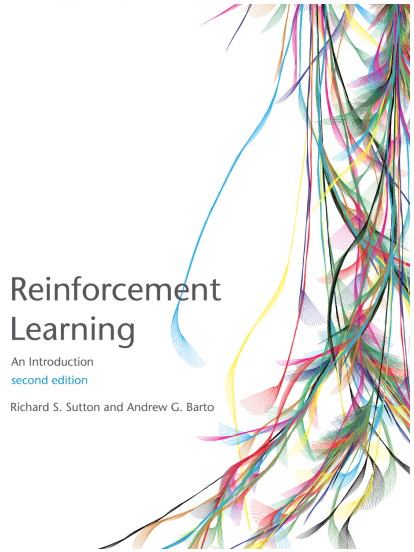
Prerequisites

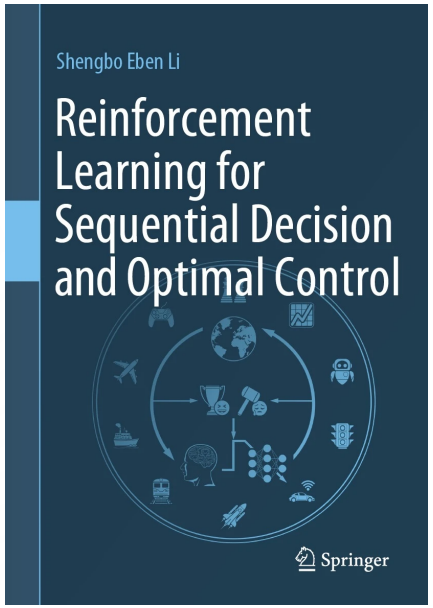
- A programming language, Python / MATLAB / C++
- Probability
- Linear algebra
- Basic optimization



References







1. **Mathematical Foundations of Reinforcement Learning**, by Zhao Shiyu, Tsinghua University 2024
2. **Reinforcement Learning: An Introduction**, by Sutton and Barto, MIT Press, 2018
3. **Reinforcement Learning for Sequential Decision and Optimal Control**, by Shengbo Li, Springer, 2023

Online Course Materials

- Shengbo Li, Tsinghua University
- Yuejie Chi, Carnegie Mellon University
- Shiyu Zhao, Westlake University

Acknowledgement

Parts of the lecture materials are adapted from online course materials with permission for teaching purpose, thanks Prof. Li, Prof. Chi and Prof. Zhao.

Recent successes in RL

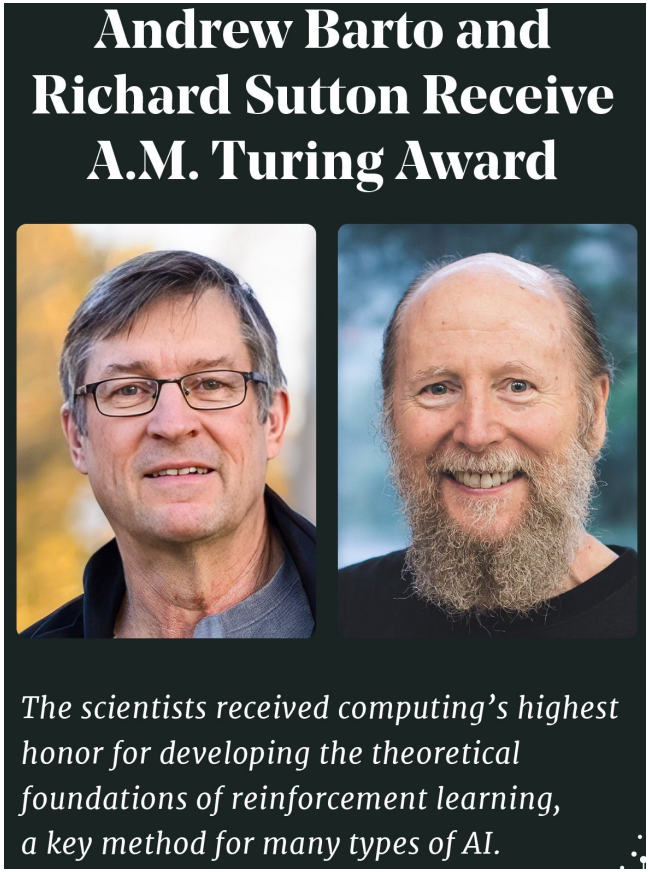
ChatGPT

deepseek

Chat with DeepSeek AI

RL holds great promise in the next era of artificial intelligence.

Recent successes in RL



ACM A.M. Turing Award Honors Two Researchers Who Led the Development of Cornerstone AI Technology

Andrew Barto and Richard Sutton Recognized as Pioneers of Reinforcement Learning

Challenges in Reinforcement Learning

Sample Inefficiency

Foremost among these is the issue of sample inefficiency. Many RL algorithms rely on extensive trial-and-error interactions with the environment to learn effective policies. This requirement imposes significant costs, particularly in physical systems such as robotics, where repeated failures—e.g., mechanical falls—are not only resource-intensive but may also lead to hardware degradation.

How to design a reward function?

Another critical challenge lies in the design of the reward function. Constructing a reward signal that accurately encodes task objectives is both scientifically non-trivial and problem-specific. Improper reward design can result in unintended agent behaviors, often referred to as "reward hacking," where the agent maximizes reward through strategies that violate the spirit of the intended task.

Challenges in Reinforcement Learning

Curse of dimensionality: Enormous state and action space

The curse of dimensionality further complicates the learning process. Many RL environments feature extremely large or continuous state and action spaces, making exhaustive exploration computationally infeasible. Effective learning thus requires mechanisms for directed exploration and continual policy refinement in high-dimensional spaces.

Constrained infeasibility & safety

- Ensuring safe and reliable decision-making is paramount in safety-critical domains, such as autonomous driving and robotic-assisted surgery.
- RL algorithms must be designed to avoid unsafe actions during both training and deployment, necessitating advances in safe exploration and robust policy learning.

Challenges in Reinforcement Learning

1. Unknown or changing environments
2. Partial observability
3. Lack of generalizability
4. Exploration-exploitation dilemma
5. Temporally delayed rewards or feedback
6. Nonconvex optimization
7. ...

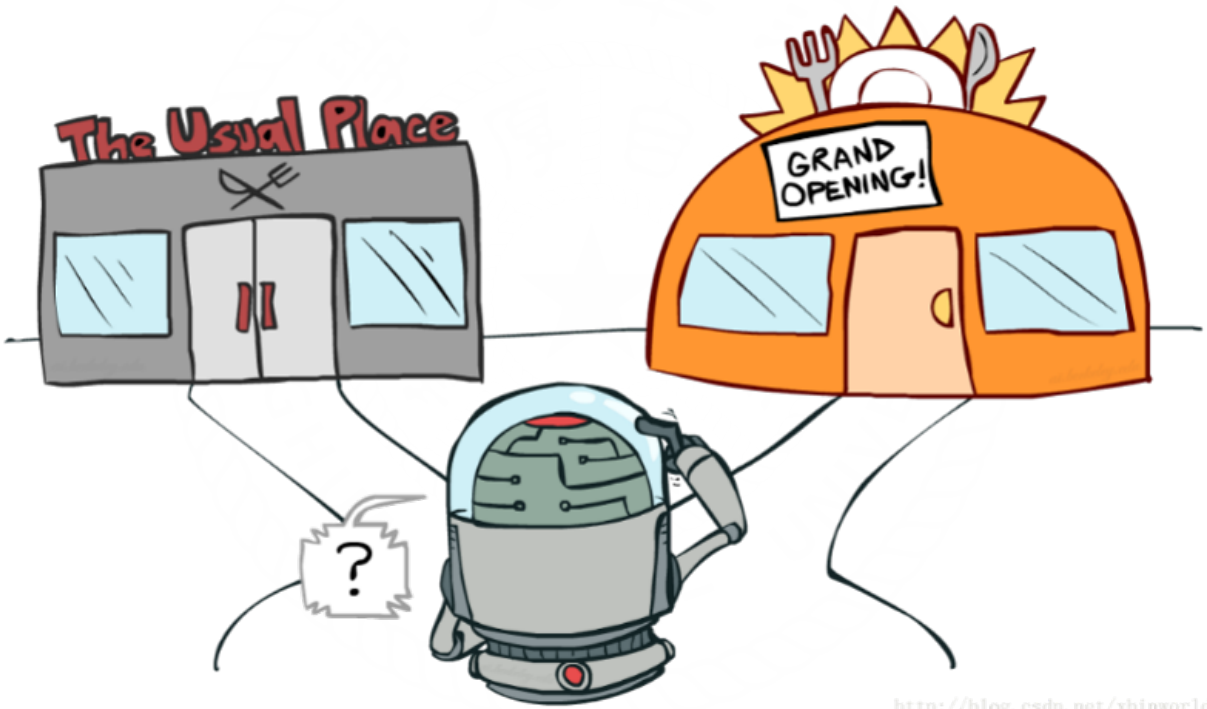
Exploitation and Exploration

Exploitation: Utilize known information to maximize reward

Go to your favorite restaurant?

Exploration: Dig out more information about the environment

Try a new restaurant?



We need both to maximize the total reward.

Partial observability in RL

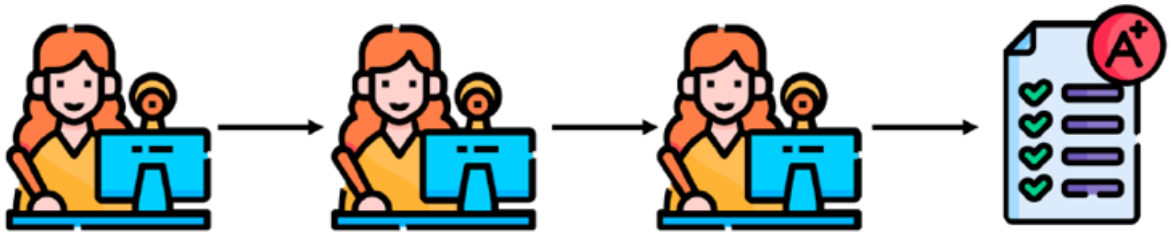
Full observability is beneficial with a few advantages, like theoretical completeness and easy implementation.

The entire state of the environment is not fully visible to an external observer.

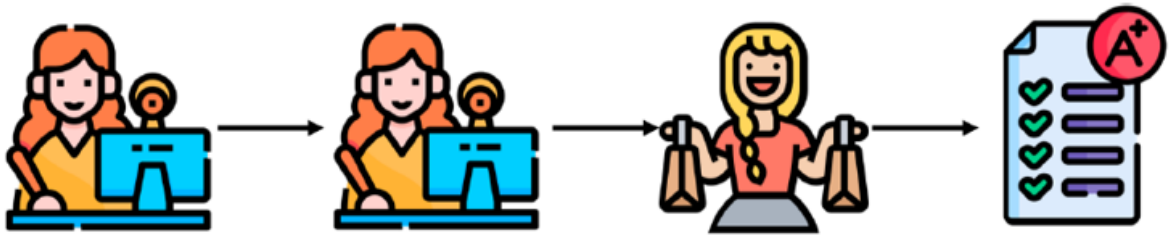
Delayed Rewards or Feedback

Credit assignment problem: results (win or lose) are released at the end of the assignment, and the accumulated reward signal

What is the action that leads to the desired outcome?

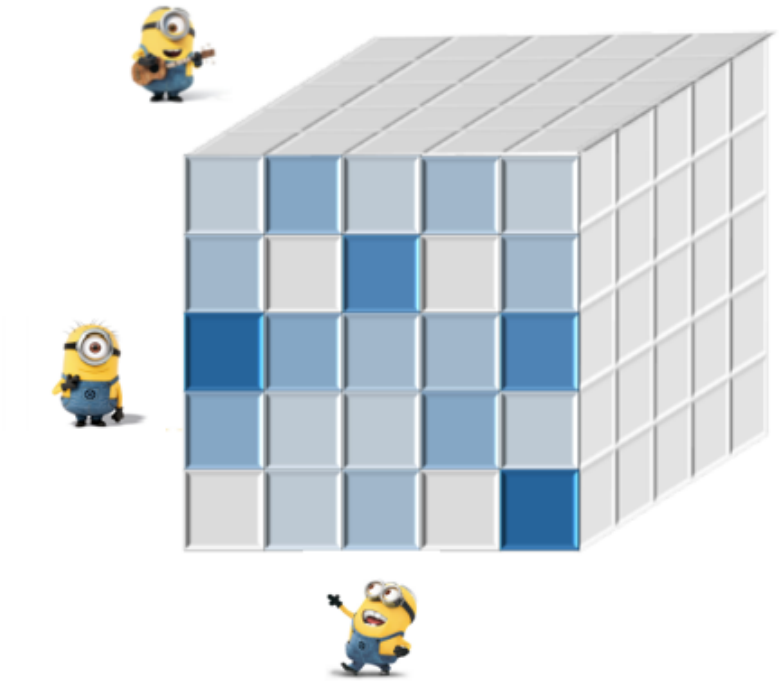


What if....



Curse of Dimensionality

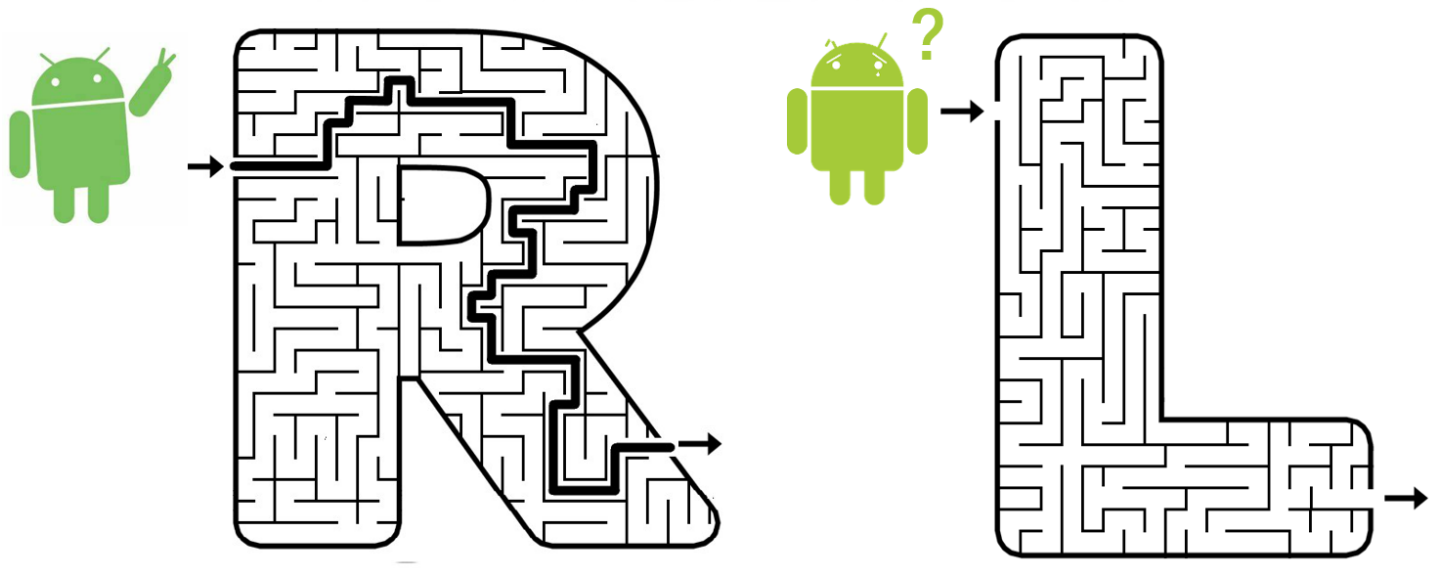
Computational requirements grow exponentially with the number of state variables.



The explosion of choices:
The joint action space grows **exponentially** with the agents!

Lack of Generalizability

- Existing RL cannot adapt to unseen scenarios.
- Policies tend to specialize in their training domain.
- There may be serious sim-to-real generalization issues.



Training scenario

Implementation

Goal of this course

- Not a **deep** RL course
- Aim to build the "**foundations**".
- Research-oriented
- Models, algorithms, and their analyses

You are strongly encouraged to combine it with your research.

Sample Efficiency

Collecting data samples might be expensive or time-consuming

clinical trials

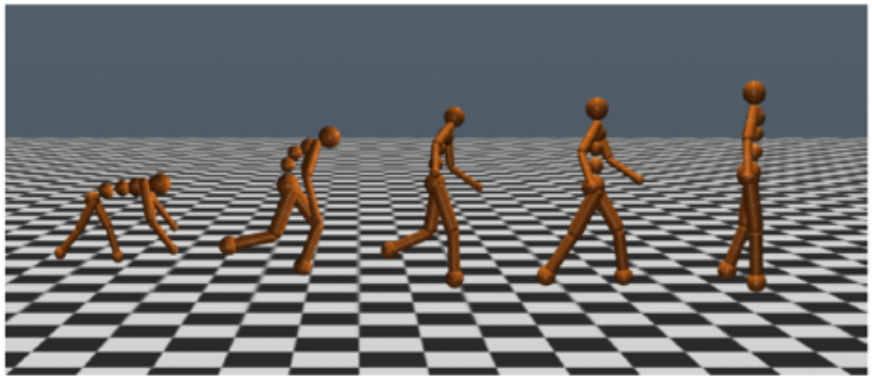
autonomous driving

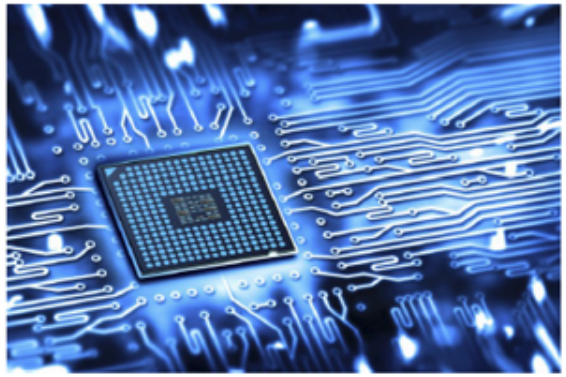
online ads

Calls for design of sample-efficient RL algorithms!

Computational Efficiency

Running RL algorithms might take a long time and space





many CPUs / GPUs / TPUs + computing hours

Calls for computationally efficient RL algorithms!

From Asymptotic to Non-asymptotic Analyses

Non-asymptotic analyses are key to understand sample and computational efficiency in modern RL.

Constrained Infeasibility & Safety

Feasibility and safety are strongly coupled:

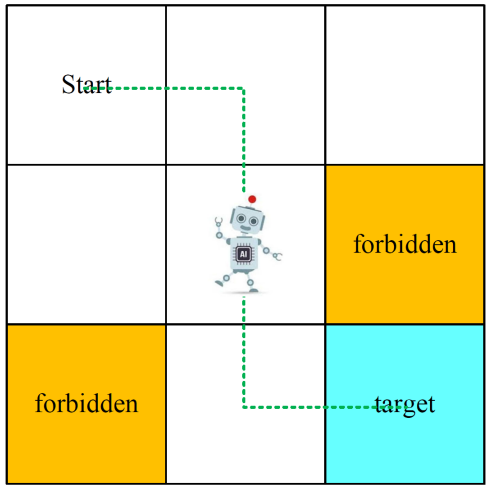
Both are caused by hard state constraints.

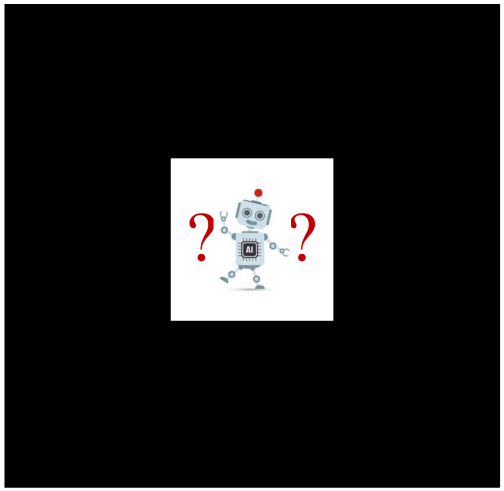
Safety must be guaranteed in feasible working regions.

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

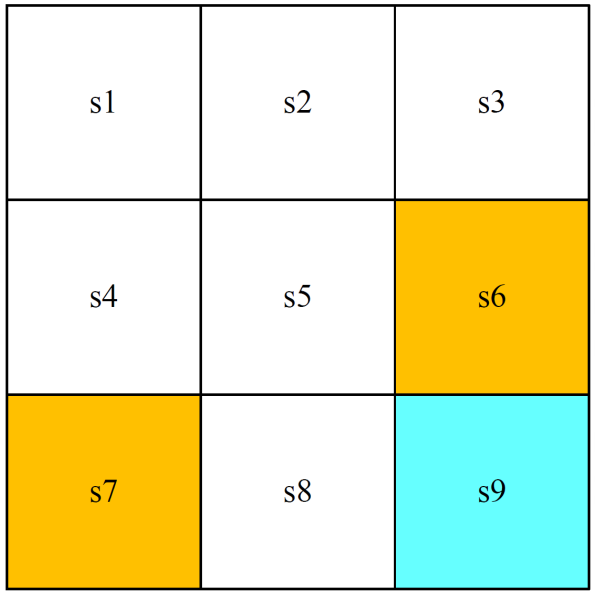
Task:

1. From any starting cell, find a "good" way to the target.
2. How to define "good"? Avoid forbidden cells or boundaries.

State

The status of the agent with respect to the environment.

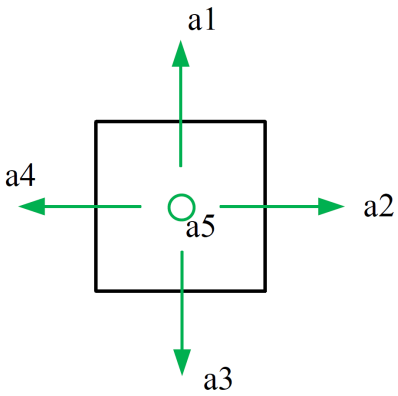
- For the grid-world example, the location of the agent is the state. There are nine possible locations and hence nine states: $s_1, s_2, \ldots, s_9$.

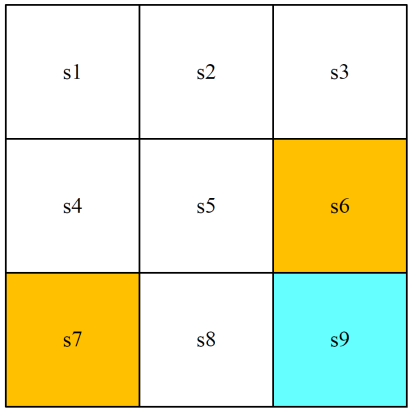


State space: the set of all states $\mathcal{S} = \{s_i\}_{i=1}^{9}$.

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 still;

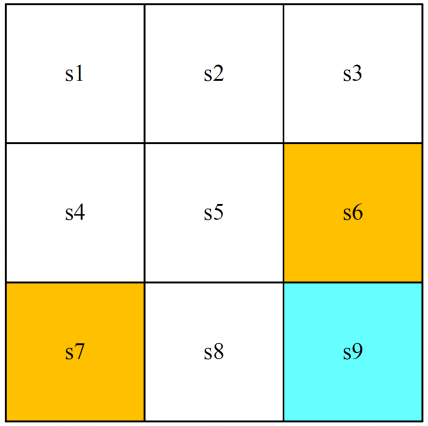
Action space of a state:

The set of all possible actions of a state. $\mathcal{A}(s_i) = \{a_k\}_{k=1}^{5}$.

Question:

Can different states have different sets of actions?

State transition



When taking an action, the agent may move from one state to another.

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

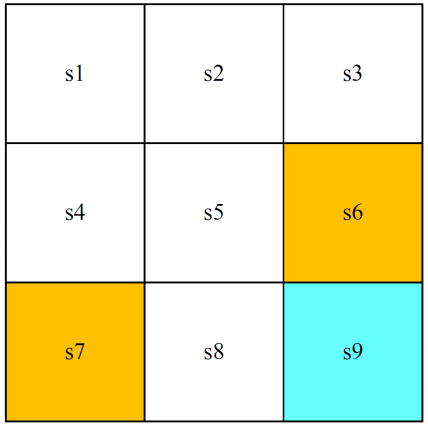
$$s_1 \xrightarrow{a_2} s_2$$

- 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.

State transition

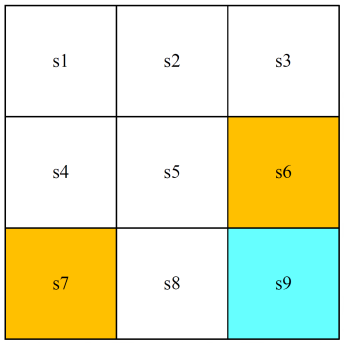


Pay attention to forbidden areas: Example: at state s_5, if we choose action a_2, then what is the next state?

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

$$s_5 \xrightarrow{a_2} s_6$$

State transition

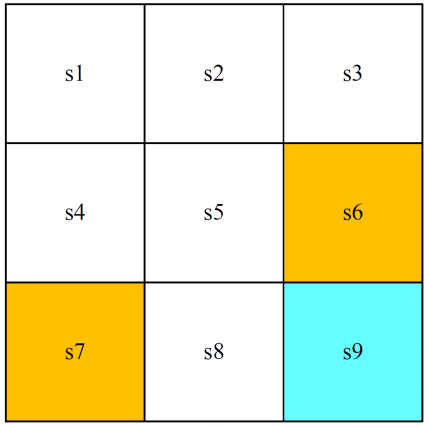


Tabular representation: We can use a table to describe the state transition:

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
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.

$$\begin{cases} p(s_2 \mid s_1, a_2) = 1 \\ p(s_i \mid s_1, a_2) = 0 \quad \forall i \neq 2 \end{cases}$$

Here it is a deterministic case. The state transition could be stochastic.

Two ways to describe the stochastic environment:

Probabilistic model

$$\mathcal{P}_{ss'}^{a} = p\left(s' \mid s, a\right) \overset{\text{def}}{=} \Pr\left\{s_{t+1} = s' \mid s_t = s, a_t = a\right\}, s \in \mathcal{S}, a \in \mathcal{A}$$

- s: state, $\mathcal{S}$: a finite set of states
- a: action, $\mathcal{A}$: a finite set of actions

State-action samples

- One sample: $< s, a, s' >$
- One trajectory:

$$\mathcal{D} = \{\underbrace{s_0, a_0, s_1}_{\text{Sample}}, a_1, s_2, a_2, \cdots\cdots, \underbrace{s_t, a_t, s_{t+1}}_{\text{Sample}}, a_{t+1}, \cdots\}$$

Policy

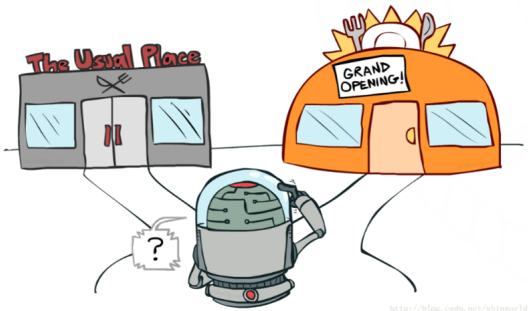
A mapping from the state space to the action space that the agent has to respond to

- Stochastic policy: probability of selecting action a at state s

$$\sum_{a \in \mathcal{A}} \pi(a \mid s) = 1$$

- Deterministic policy: mapping from $s \in \mathcal{S}$ to $a \in \mathcal{A}$

$$a = \pi(s)$$



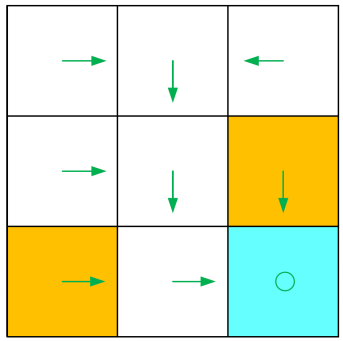
Try a new restaurant with a probability of $p = 0.2$.
Go to your favorite restaurant? $p = 0.8$

Deterministic policy in Robot:
The current state determines the action.

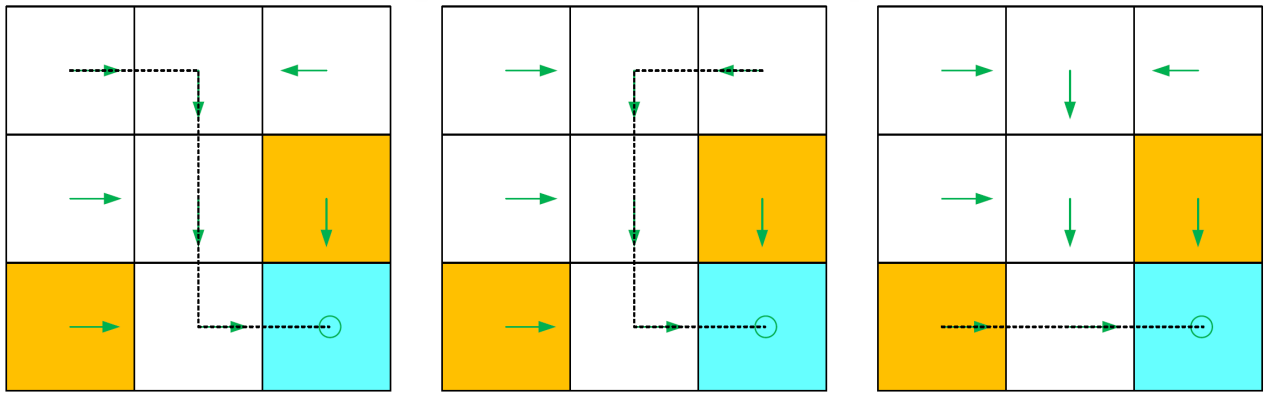
Policy

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

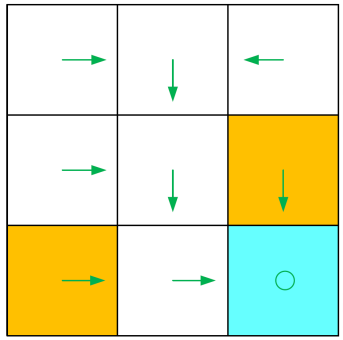
Intuitive representation: We use arrows to describe a policy.



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 :

$$\pi(a_1 \mid s_1) = 0$$

$$\pi(a_2 \mid s_1) = 1$$

$$\pi(a_3 \mid s_1) = 0$$

$$\pi(a_4 \mid s_1) = 0$$

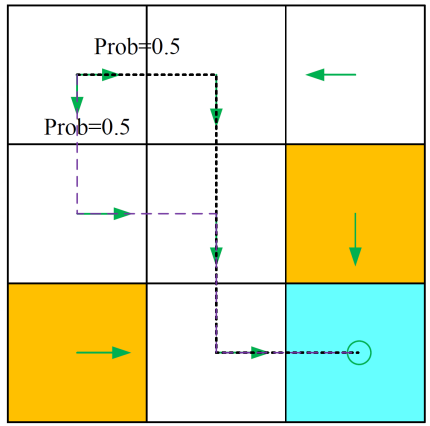
$$\pi(a_5 \mid s_1) = 0$$

It is a deterministic policy.

Policy

There are stochastic policies.

For example:



In this policy, for s_1 :

$$\pi(a_1 \mid s_1) = 0$$

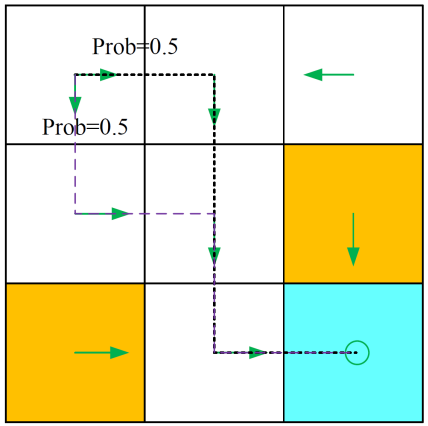
$$\pi(a_2 \mid s_1) = 0.5$$

$$\pi(a_3 \mid s_1) = 0.5$$

$$\pi(a_4 \mid s_1) = 0$$

$$\pi(a_5 \mid s_1) = 0$$

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 (still)
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.

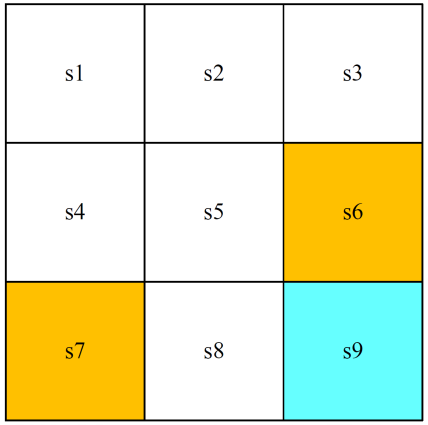
A positive reward represents encouragement to take such actions.

A negative reward represents punishment for taking such actions.

Questions:

- Can positive indicate punishment and negative indicate encouragement?
  - Yes.
  - In this case, reward may be called cost.
- What about a zero reward?
  - Relative values matter, not absolute values.
  - $r = \{+1, -1\}$ becomes $r = \{+2, 0\}$ will not change the optimal policy.

Reward



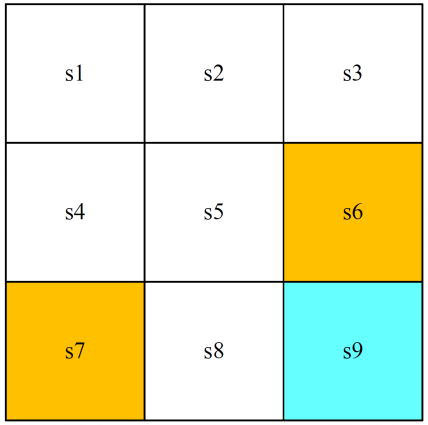
In the grid-world example, the rewards are designed as follows:

- 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

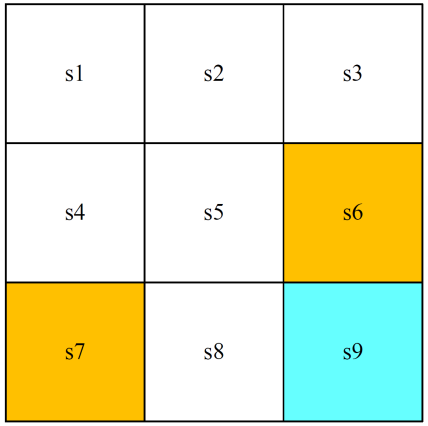


Tabular representation of reward transition: how to use the table?

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
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



Mathematical description: conditional probability

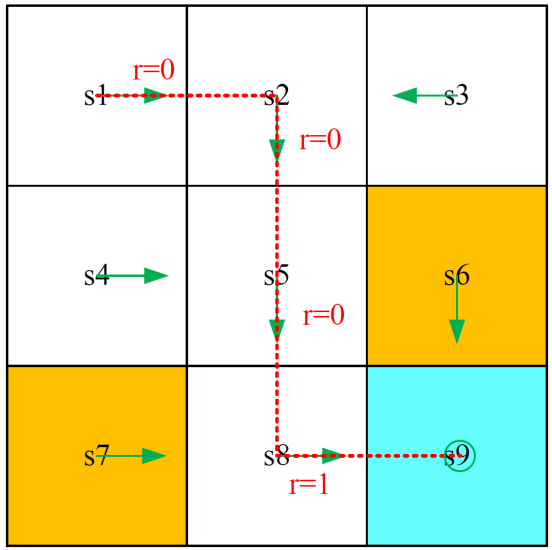
Intuition: At state s_1, if we choose action a_1, the reward is -1 .

Math: $p(r = -1 \mid s_1, a_1) = 1$ and $p(r \neq -1 \mid s_1, a_1) = 0$

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



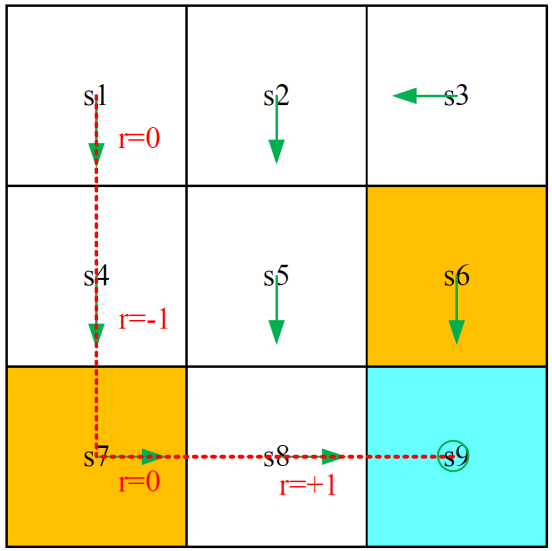
A trajectory is a state-action-reward chain:

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

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

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

Trajectory and return



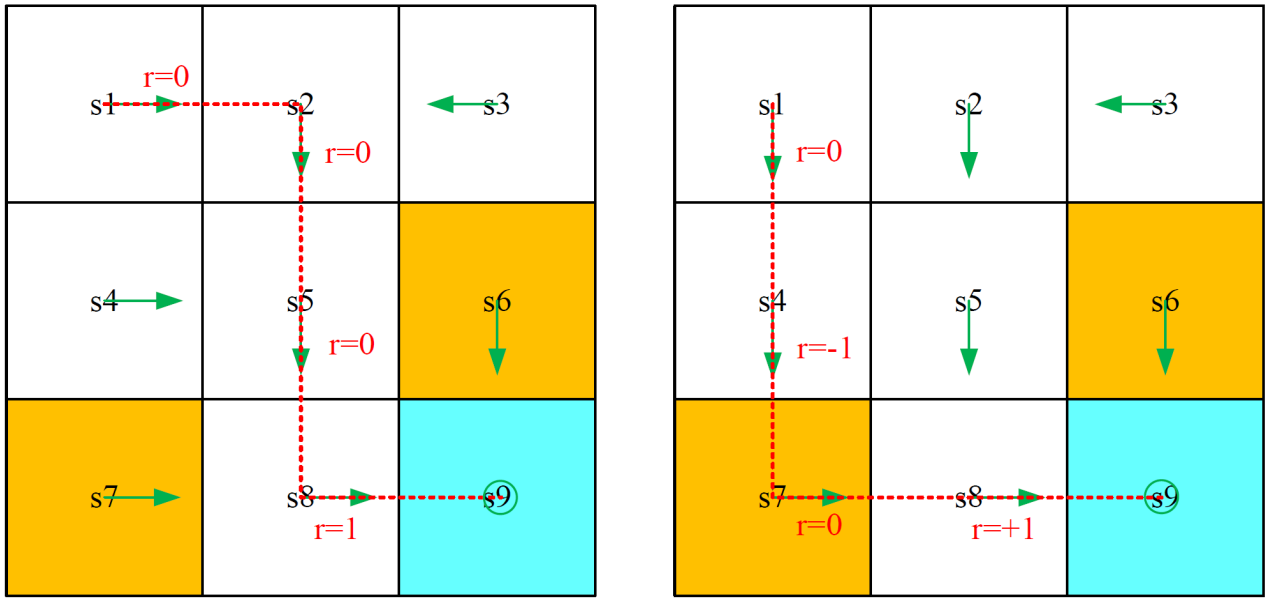
A different policy gives a different trajectory:

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

The return of this path is:

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

Trajectory and return

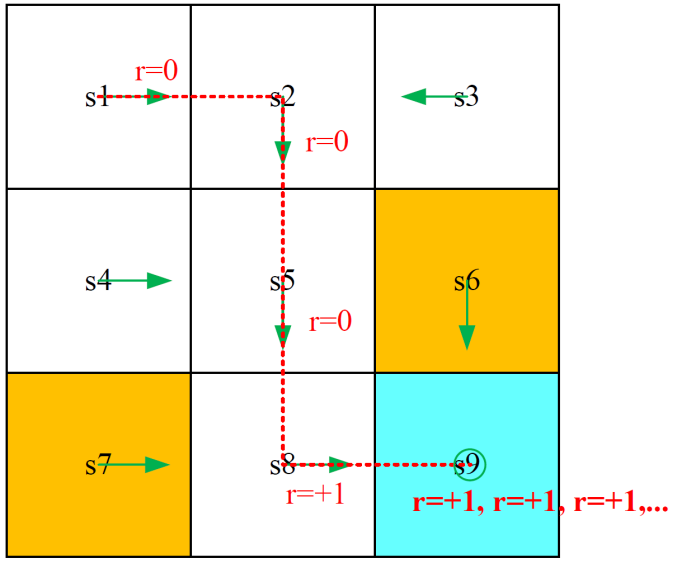


Which policy is better?

- 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)!

Discounted return



A trajectory may be infinite:

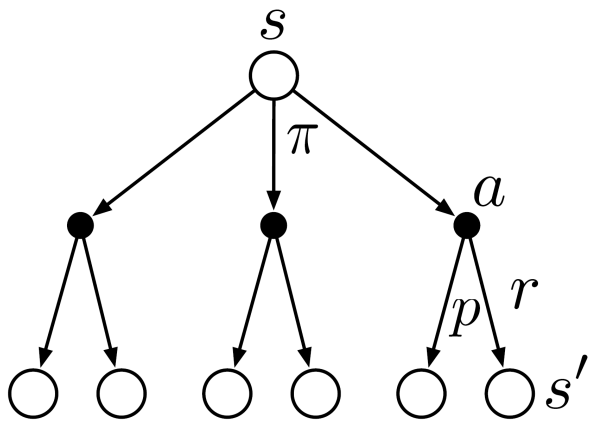
$$s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_5 \xrightarrow{a_3} s_8 \xrightarrow{a_2} s_9 \xrightarrow{a_5} s_9 \xrightarrow{a_5} s_9 \ldots$$

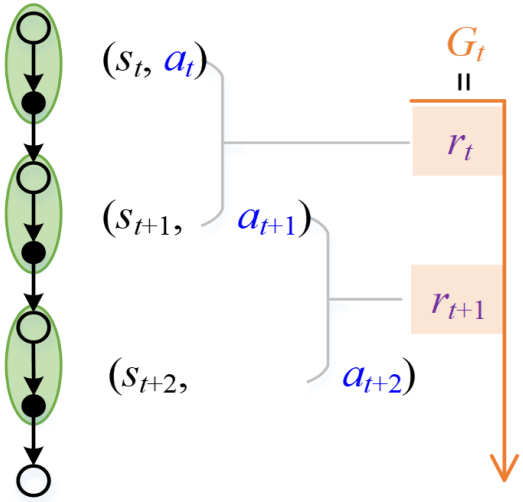
The return is

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

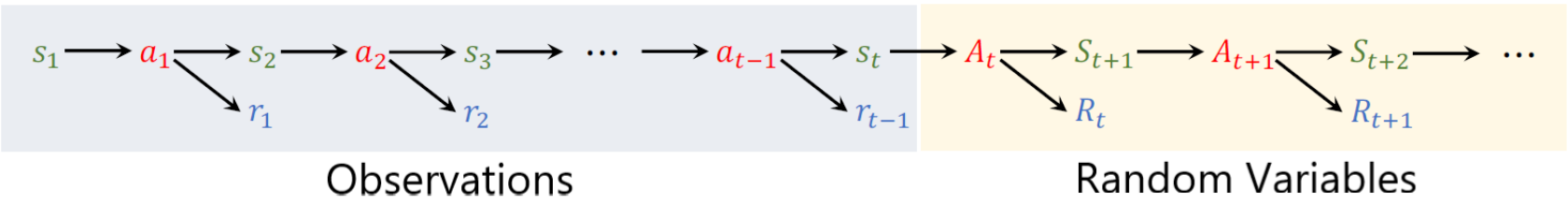
The definition is invalid since the return diverges! *Discounted return?*

Relationship of state, action and reward in MDPs





$$s_t \xrightarrow{a_t, r_t} s_{t+1} \xrightarrow{a_{t+1}, r_{t+1}} s_{t+2} \xrightarrow{a_{t+2}, r_{t+2}} s_{t+3} \to \dots$$

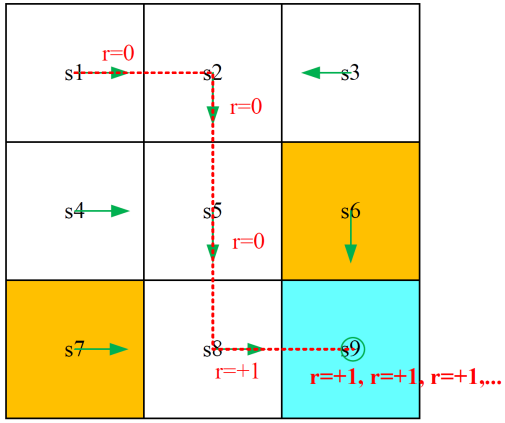


Observations Random Variables

Why discounted return?

1. Mathematically convenient: the limit always exists.
2. Immediate rewards earn more interest than future rewards.
3. Account for variability and uncertainty in the future, which may not be fully captured.

Discounted return G_t



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

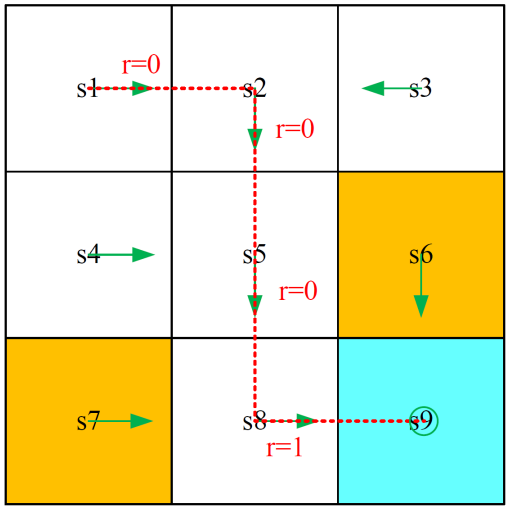
$$G_t = 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \dots$$
$$= \gamma^3 \left(1 + \gamma + \gamma^2 + \dots\right) = \gamma^3 \frac{1}{1-\gamma}$$

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

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

Episode

When interacting with the environment according to a policy, the agent may reach some terminal states. The resulting trajectory is referred to as an episode (or 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 referred to as episodic tasks.

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 once we arrive at the target?

- 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 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 $p\left(s' \mid s, a\right)$

Reward probability: at state s, taking action a, the probability to get reward r is $p(r \mid s, a)$

Policy: at state s, the probability to choose action a is $\pi(a \mid s)$

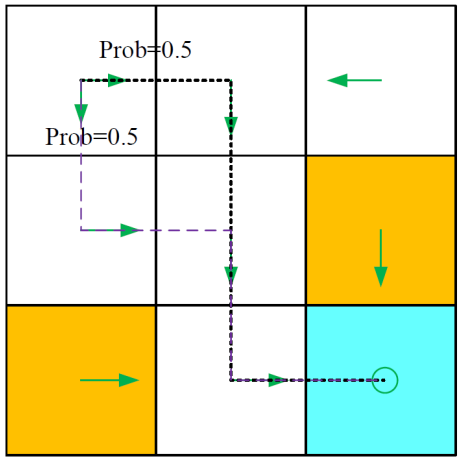
Markov decision process (MDP)

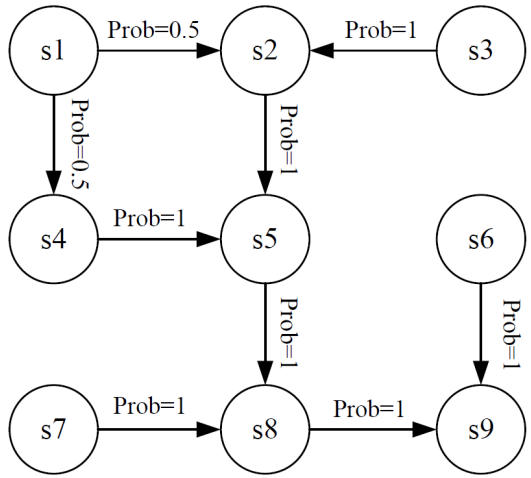
Markov property: memoryless property

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

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, a Markov process.





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

Summary

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

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