Markov Decision Process

Announcements

TA office hours are posted

HW0 is due Wednesday

Programming assignment 1 will be out on Wednesday

Reading Materials:
Reinforcement Learning: Theory & Algorithms

https://rltheorybook.github.io/

This is an extremely advanced RL book, so we will pick **specific subsections** for you to read

Please let us know if you find any typos or mistakes in the book

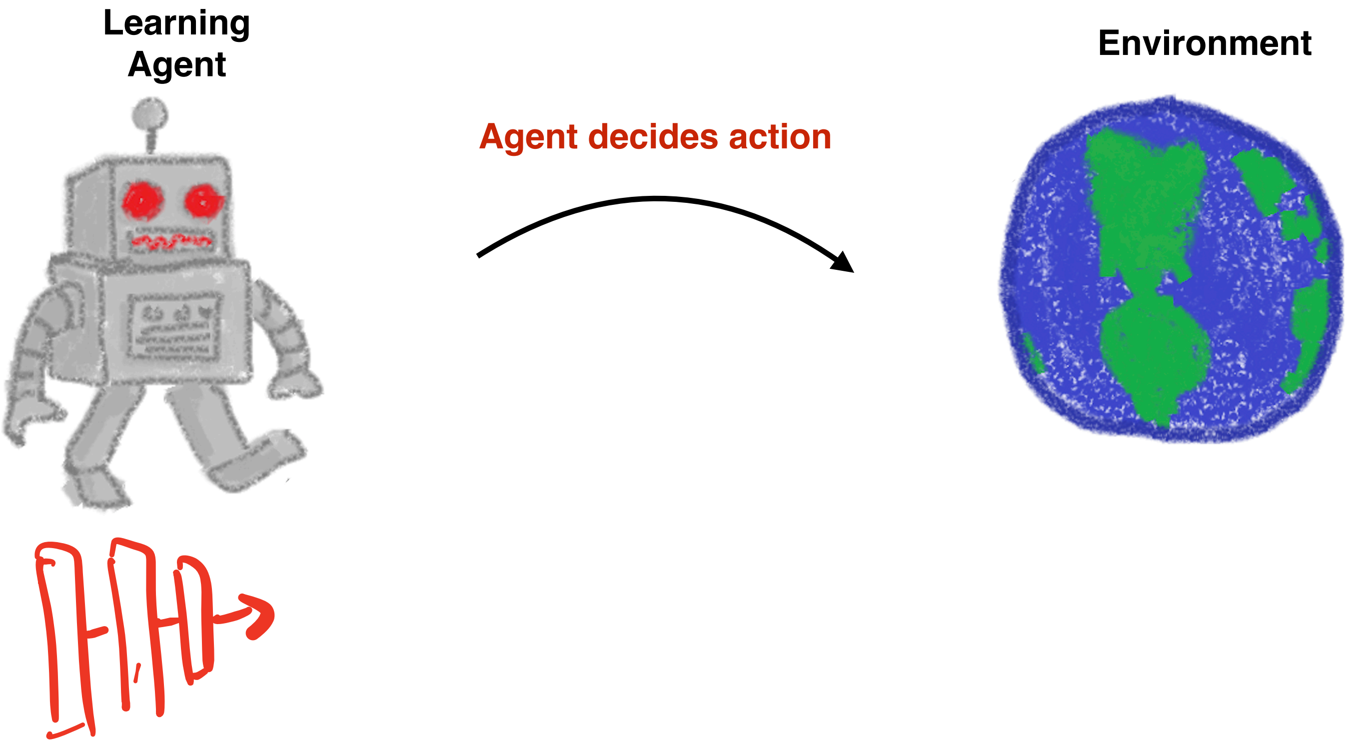
Outlines:

1. Definitions of Markov Decision Process

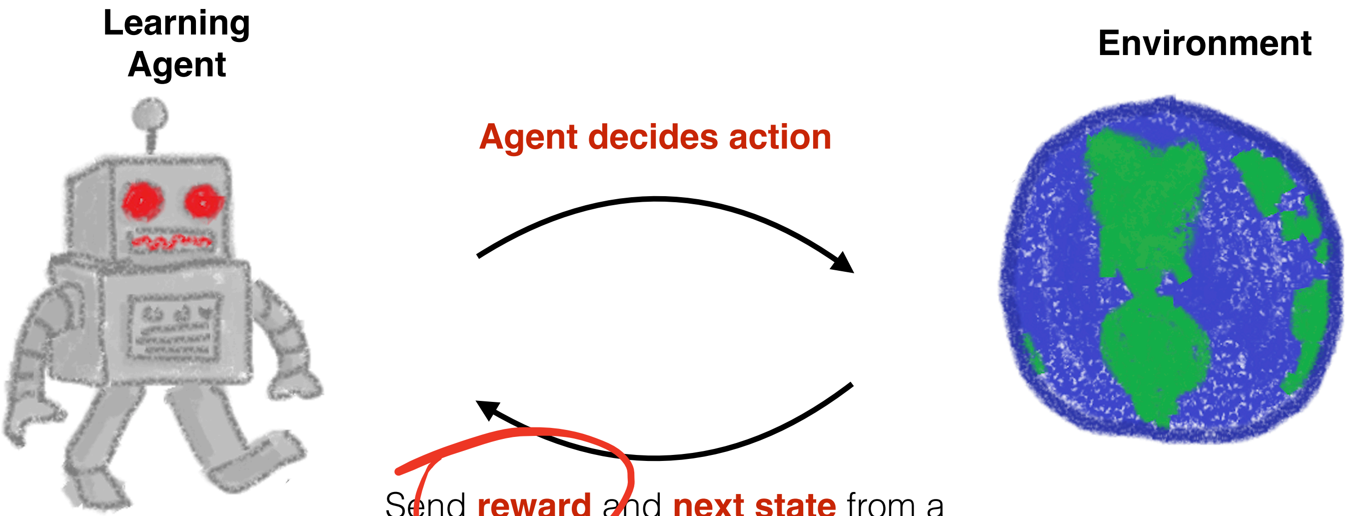
2. Value functions (V and Q functions)

3. Bellman equations

The Mathematical framework:
Infinite horizon **Markov Decision Process**



The Mathematical framework:
Infinite horizon **Markov Decision Process**



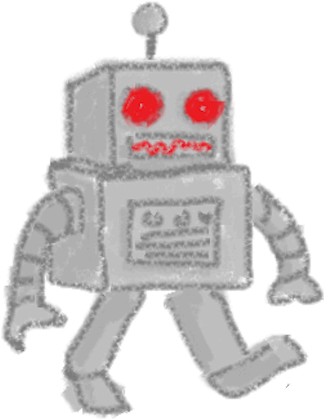
Send **reward** and **next state** from a Markovian transition dynamics

$$r(s,a), s' \sim P(\cdot \mid s,a)$$

$P(. \mid s,a)$: distribution over the next state

The Mathematical framework: Infinite horizon **Markov Decision Process**

Learning Agent

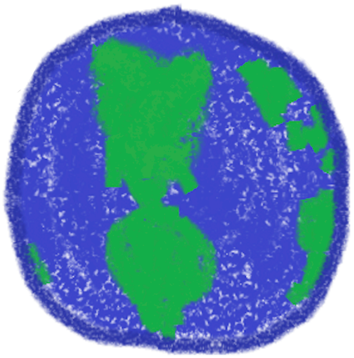


Agent decides action

Infinity many Steps

Send **reward** and **next state** from a Markovian transition dynamics

Environment

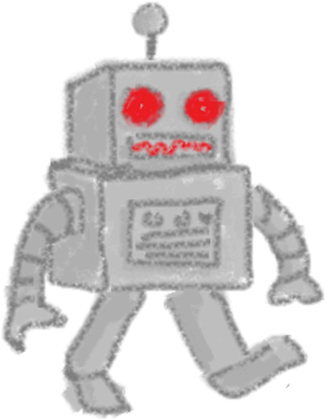


$$r(s,a), s' \sim P(\cdot \,|\, s,a)$$

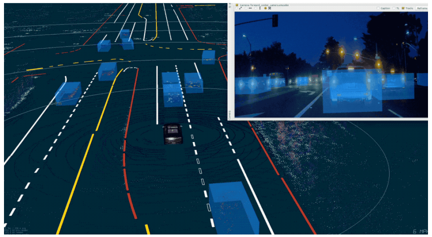
$P(\,.\,|\,s,a)$: distribution over the next state

The Mathematical framework:
Infinite horizon **Markov Decision Process**

Learning Agent



Environment



Agent decides action

Infinity many Steps

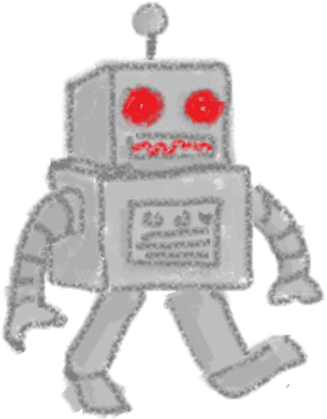
Send **reward** and **next state** from a Markovian transition dynamics

$$r(s,a), s' \sim P(\cdot \mid s,a)$$

$P(\,.\mid s,a)$: distribution over the next state

The Mathematical framework:
Infinite horizon **Markov Decision Process**

Learning Agent



Environment

Agent decides action

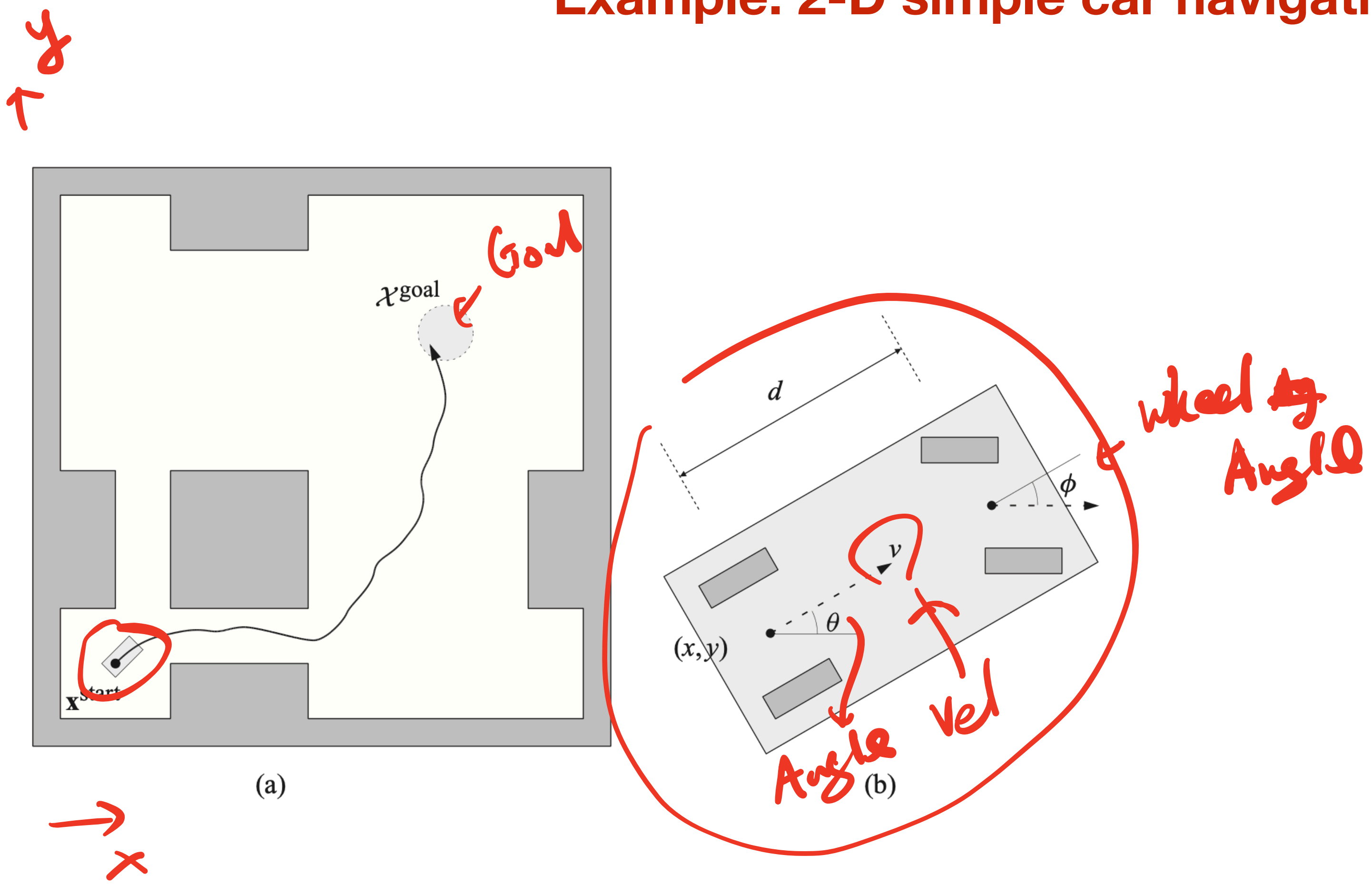
Infinity many Steps

Send **reward** and **next state** from a Markovian transition dynamics

$$r(s, a), s' \sim P(\cdot \mid s, a)$$

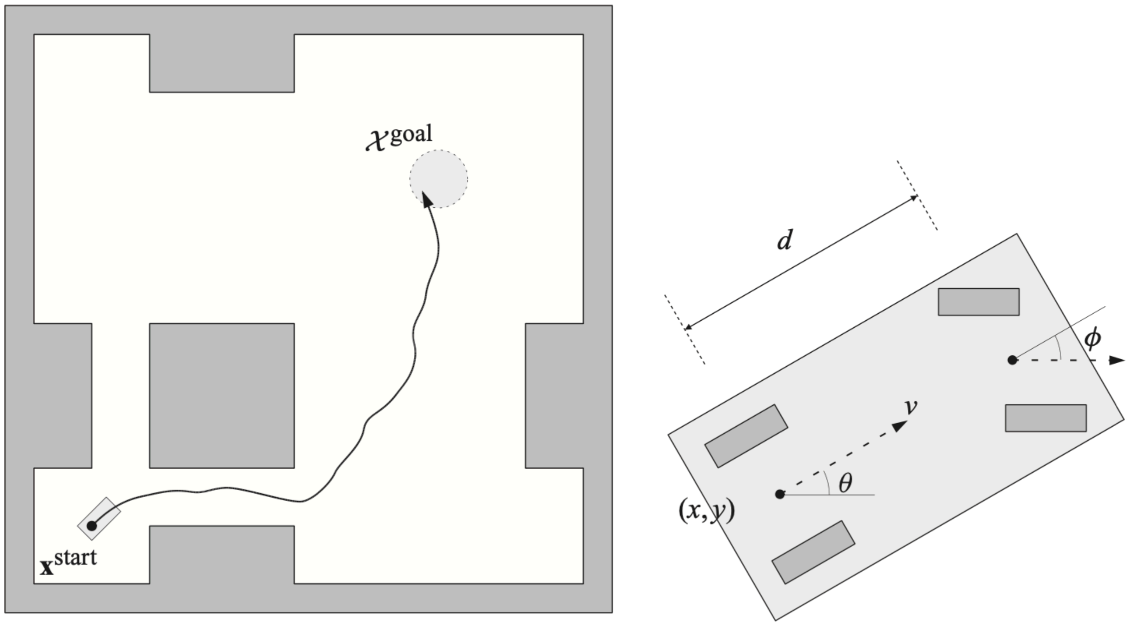
$P(\,.\mid s, a)$: distribution over the next state

Example: 2-D simple car navigation



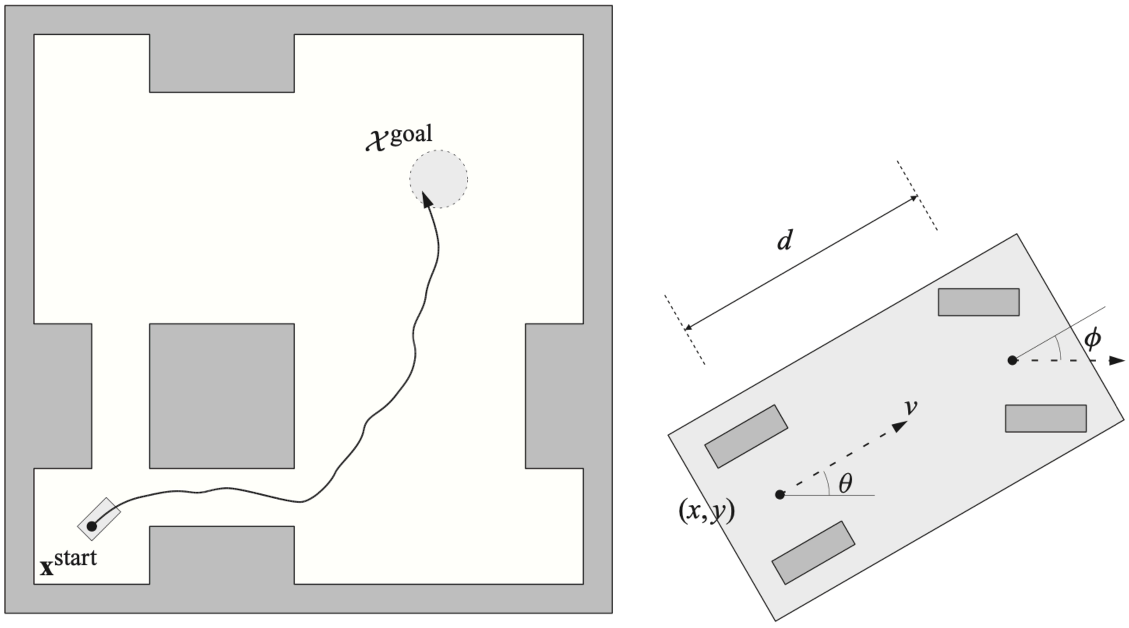
Example: 2-D simple car navigation

$$s = [x, y, \theta, v]^\top \in \mathbb{R}^4$$



(a) (b)

Example: 2-D simple car navigation



(a) (b)

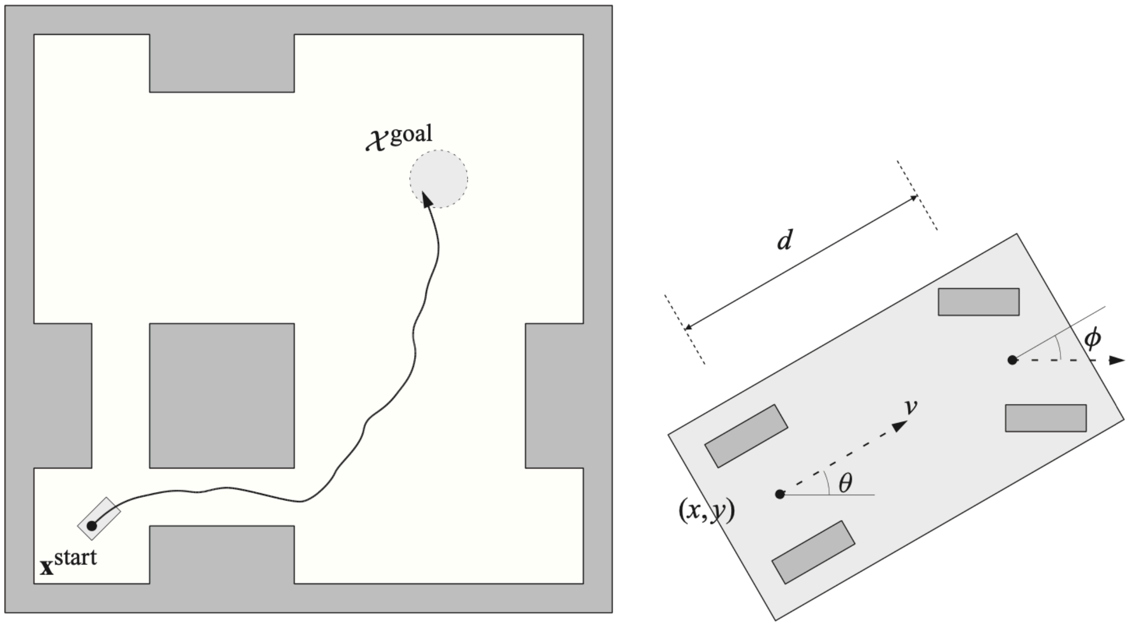
$$s = [x, y, \theta, v]^\top \in \mathbb{R}^4$$

$$a = [\alpha, \phi]^\top \in \mathbb{R}^2$$

Acc

Wheel Angle

Example: 2-D simple car navigation



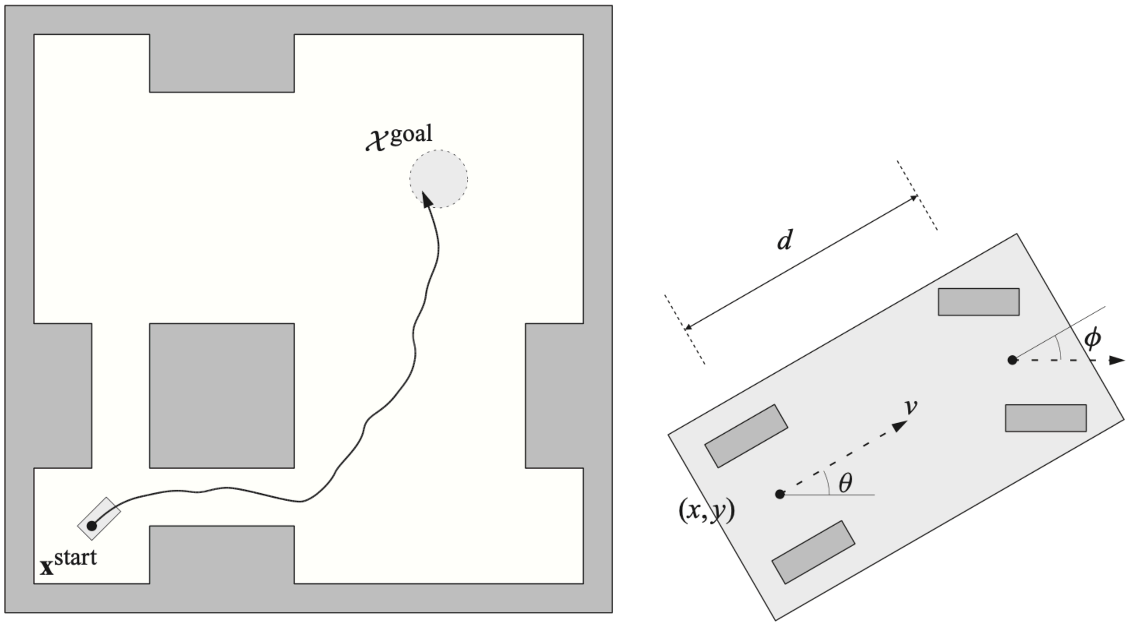
(a) (b)

$$s = [x, y, \theta, v]^\top \in \mathbb{R}^4$$

$$a = [\alpha, \phi]^\top \in \mathbb{R}^2$$

$$r(s, a) = \begin{cases} 100 & (x, y) \in \mathcal{X}_{goal} \\ -1 & \text{hit obstacles} \\ 0 & \text{else} \end{cases}$$

Example: 2-D simple car navigation



(a) (b)

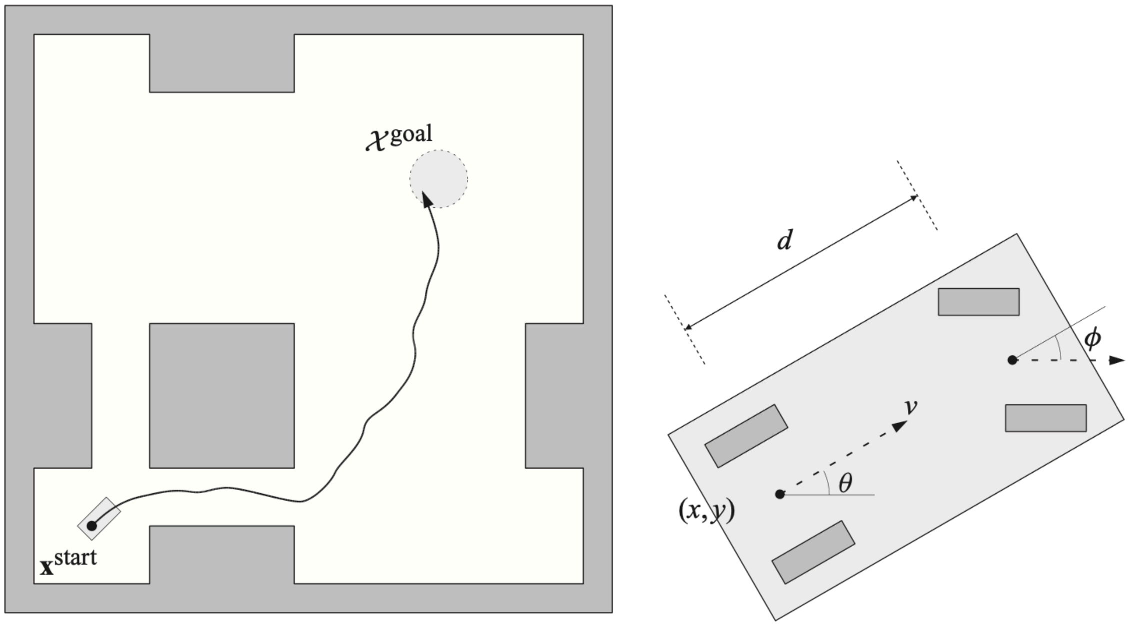
$$s = [x, y, \theta, v]^\top \in \mathbb{R}^4$$

$$a = [\alpha, \phi]^\top \in \mathbb{R}^2$$

$$r(s, a) = \begin{cases} 100 & (x, y) \in \mathcal{X}_{goal} \\ -1 & \text{hit obstacles} \\ 0 & \text{else} \end{cases}$$

$$s' = f(s, a) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

Example: 2-D simple car navigation



(a) (b)

$$s = [x, y, \theta, v]^\top \in \mathbb{R}^4$$

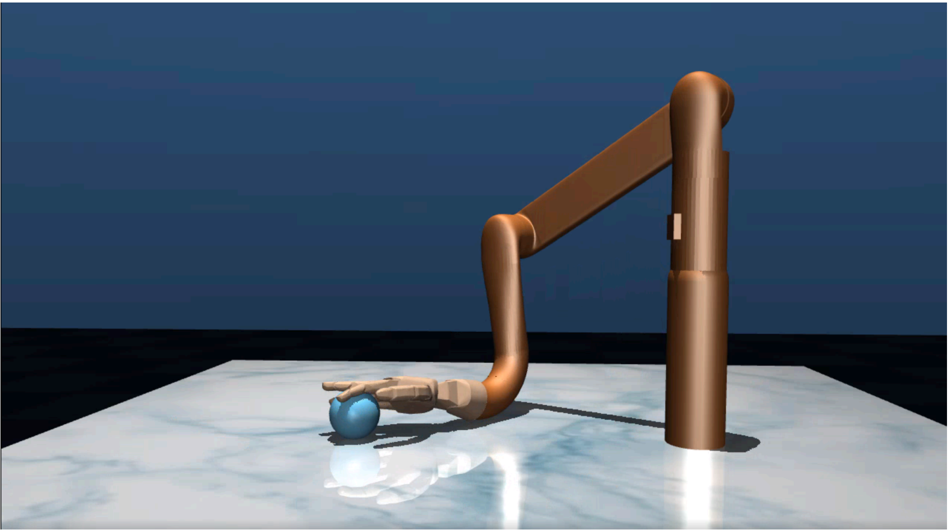
$$a = [\alpha, \phi]^\top \in \mathbb{R}^2$$

$$r(s,a) = \begin{cases} 100 & (x,y) \in \mathcal{X}_{goal} \\ -1 & \text{hit obstacles} \\ 0 & \text{else} \end{cases}$$

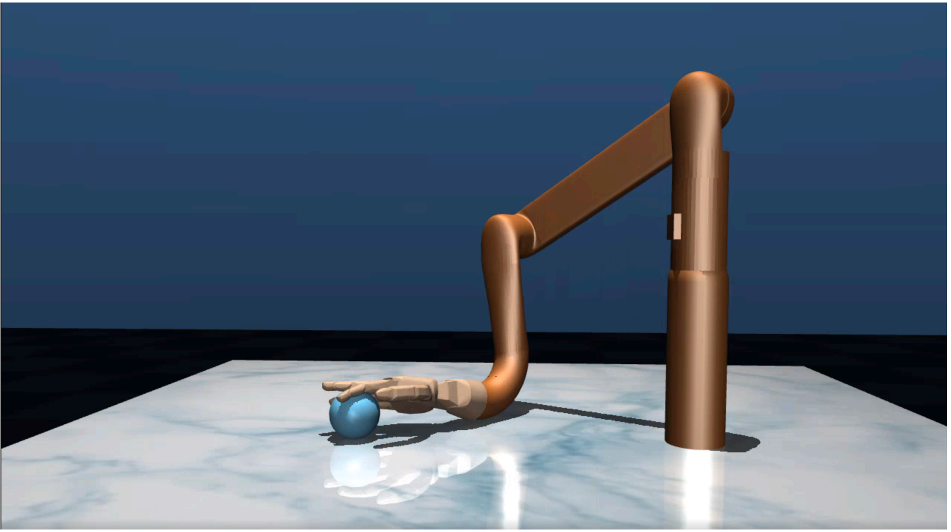
$$s' = f(s,a) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I)$$

$$f(s,a) = \begin{bmatrix} x + \tau v \cos\theta \\ y + \tau v \sin\theta \\ \theta + \tau v \tan(\phi)/d \\ v + \tau\alpha \end{bmatrix}$$

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position

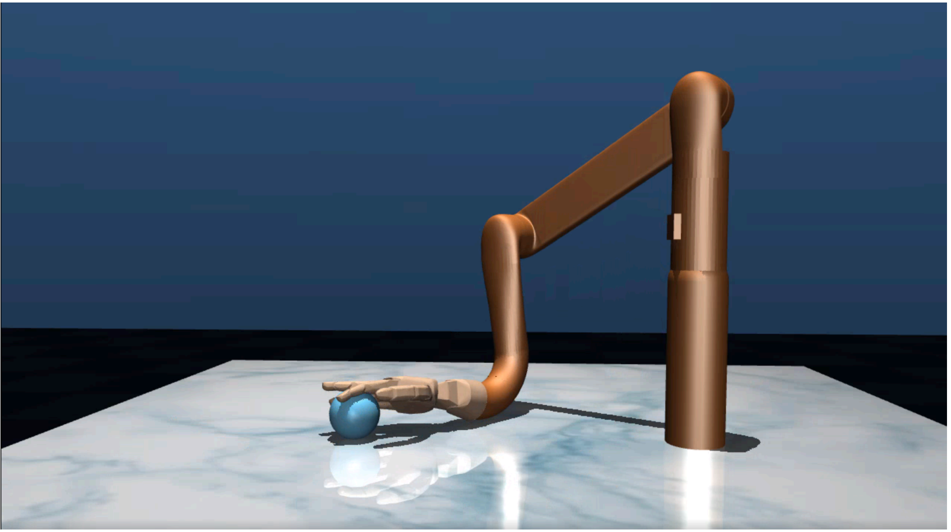


Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

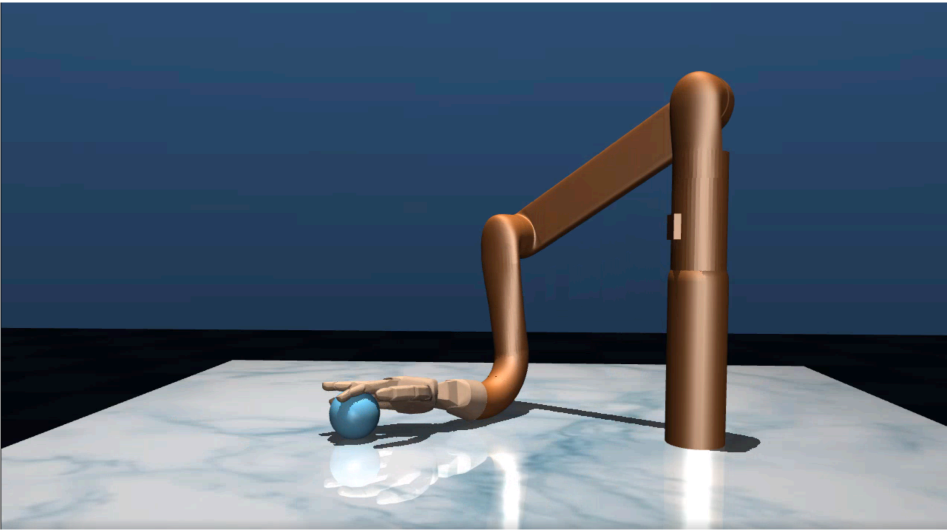
Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position

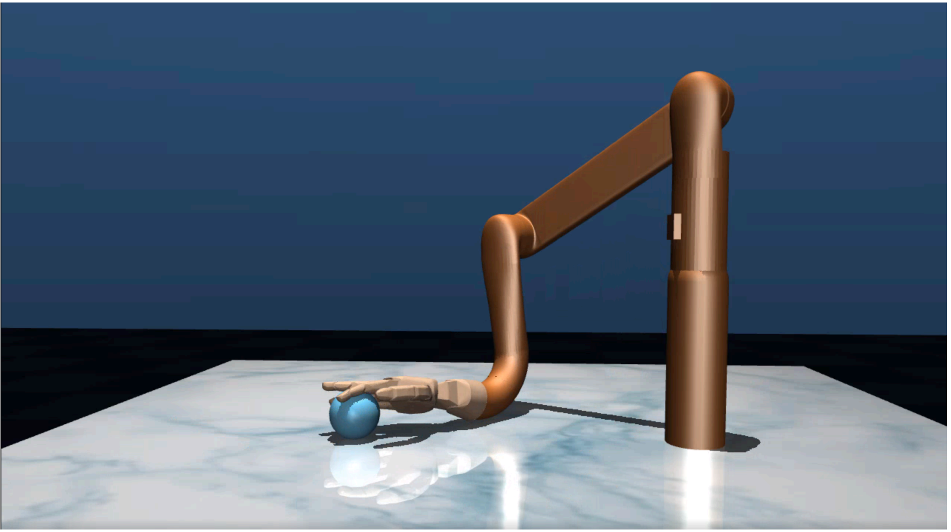


State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

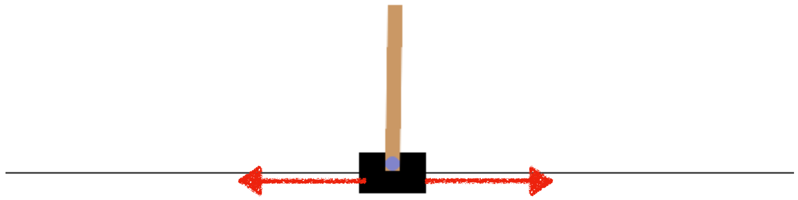
Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

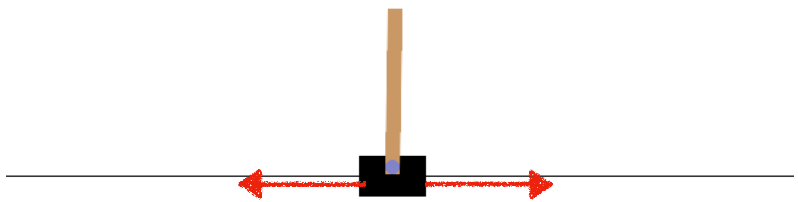
Cost $c(s, a)$: torque magnitude + dist to goal

Example: OpenAI Gym demonstrations

cart pole

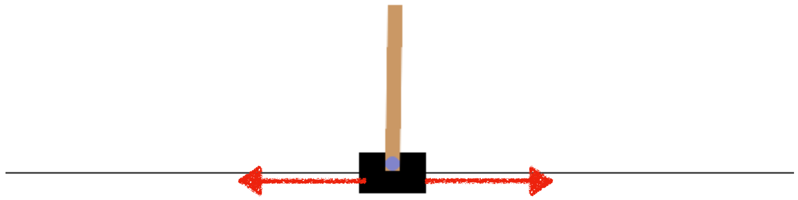


Example: OpenAI Gym demonstrations



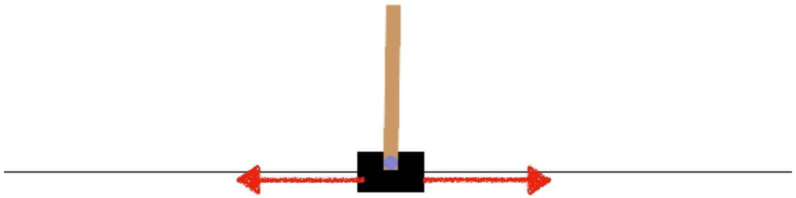
Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity, pole angle, pole angular velocity] $\in \mathbb{R}^4$



Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity, pole angle, pole angular velocity]



Push left

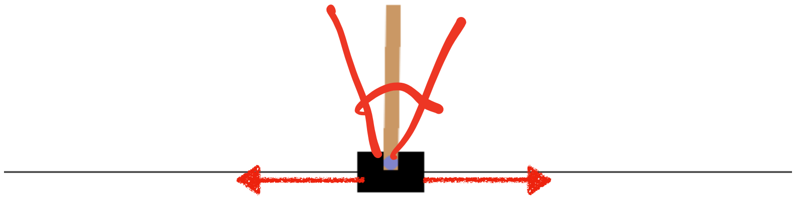
$a = 0$

Push right

$a = 1$

Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity,
pole angle, pole angular velocity]



Push left $a = 0$

Push right $a = 1$

$$r(s, a) = \begin{cases} 1 & \text{pole angle} \in [-12^o, 12^o], \\ 0 & \text{else} \end{cases}$$

Policy

$$\mathscr{S} \rightarrow \mathscr{A}$$

A mapping from state to action (what action should I take if I'm in this state…)

$$a \sim \pi(\cdot \mid s)$$

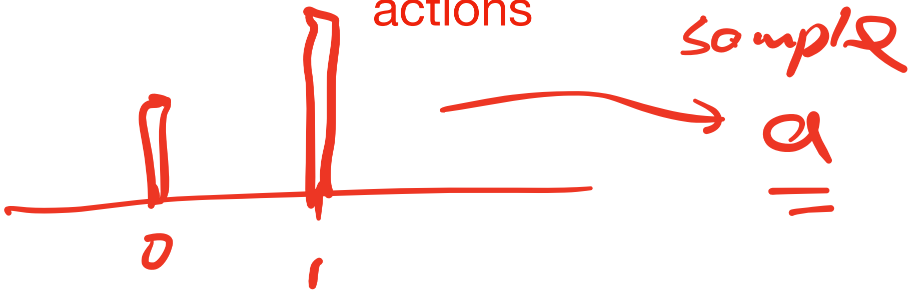
Policy

$$\mathcal{S} \rightarrow \mathcal{A}$$

A mapping from state to action (what action should I take if I'm in this state…)

$$a \sim \pi(\cdot \mid s)$$

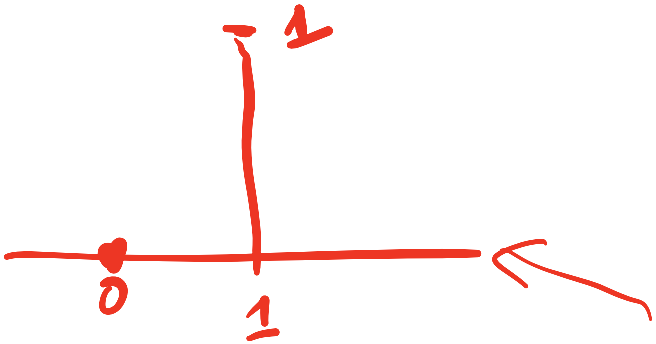
conditional distribution over all actions



Policy

$$\mathcal{S} \rightarrow \mathcal{A}$$

A mapping from state to action (what action should I take if I'm in this state…)



$$a \sim \pi(\cdot \mid s)$$

conditional distribution over all actions

Deterministic vs stochastic?

Policy

$$\mathcal{S} \to \mathcal{A}$$

A mapping from state to action (what action should I take if I'm in this state…)

$$a \sim \pi(\cdot \mid s)$$

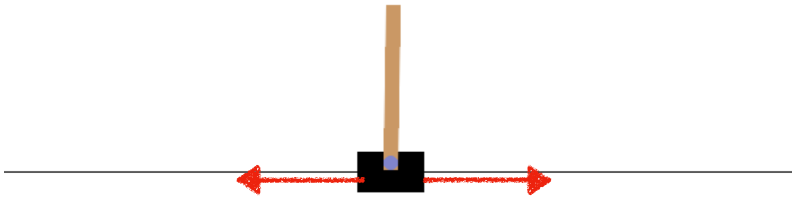
conditional distribution over all actions

Deterministic vs stochastic?

Q: Assume S state and A actions, how many different deterministic policies we can have?

Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity, pole angle, pole angular velocity]



Push left $a = 0$

Push right $a = 1$

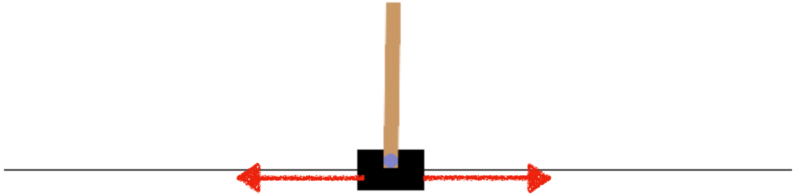
$$r(s,a) = \begin{cases} 1 & \text{pole angle} \in [-12^o, 12^o], \\ 0 & \text{else} \end{cases}$$

Policy 1: uniform random

$$\pi(0 \mid s) = \pi(1 \mid s) = 0.5, \forall s$$

Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity, pole angle, pole angular velocity]



Push left

$a = 0$

Push right

$a = 1$

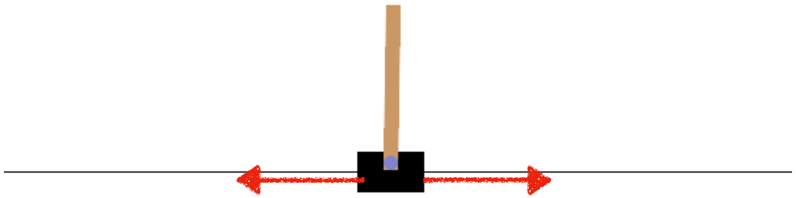
$$r(s, a) = \begin{cases} 1 & \text{pole angle} \in [-12^o, 12^o], \\ 0 & \text{else} \end{cases}$$

Policy 1: uniform random

$$\pi(0 \mid s) = \pi(1 \mid s) = 0.5, \forall s$$

Example: OpenAI Gym demonstrations

State = [cart pos, cart velocity,
pole angle, pole angular velocity]



$$r(s, a) = \begin{cases} 1 & \text{pole angle} \in [-12^o, 12^o], \\ 0 & \text{else} \end{cases}$$

Policy 1: uniform random

$$\pi(0 \mid s) = \pi(1 \mid s) = 0.5, \forall s$$

Policy 2: adaptive

$$\pi(s) = \begin{cases} 0(\text{left}) & \text{if pole angle} < 0 \\ 1(\text{right}) & \text{else} \end{cases}$$

Outlines:

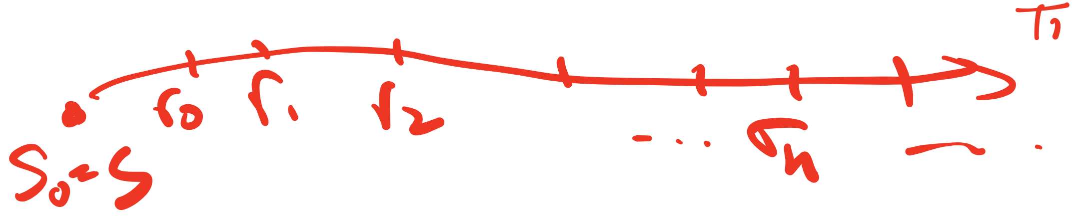
1. Definitions of Markov Decision Process
2. Value functions (V and Q functions)
3. Bellman equations

Performance of a policy π

Expected total reward of a policy π:

$$V^\pi(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \middle| s_0 = s, \pi \right]$$

$\gamma \in [0,1)$: discount factor (value future reward less and less)



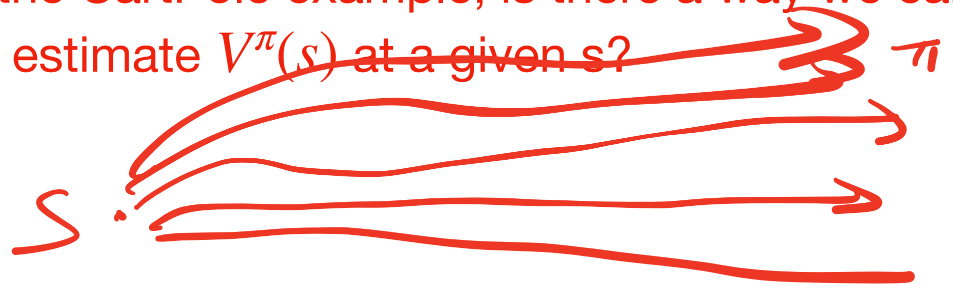
Performance of a policy π

Expected total reward of a policy π:

$$V^{\pi}(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \middle| s_0 = s, \pi \right]$$

$\gamma \in [0,1)$: discount factor (value future reward less and less)

Q: think about the CartPole example, is there a way we can estimate $V^{\pi}(s)$ at a given s?



Optimal policy

$\pi^\star$: the policy that maximizes expected future reward at all states

$$V^\star(s) \geq V^\pi(s), \forall s, \forall \pi$$

Optimal policy

$\pi^\star$: the policy that maximizes expected future reward at all states

$$V^\star(s) \geq V^\pi(s), \forall s, \forall \pi$$

Fact: $\pi^\star$ is deterministic, only depends on state (i.e., Markovian);
$\pi^\star$always exists for any infinite horizon discounted MDP

Optimal policy

$\pi^\star$: the policy that maximizes expected future reward at all states

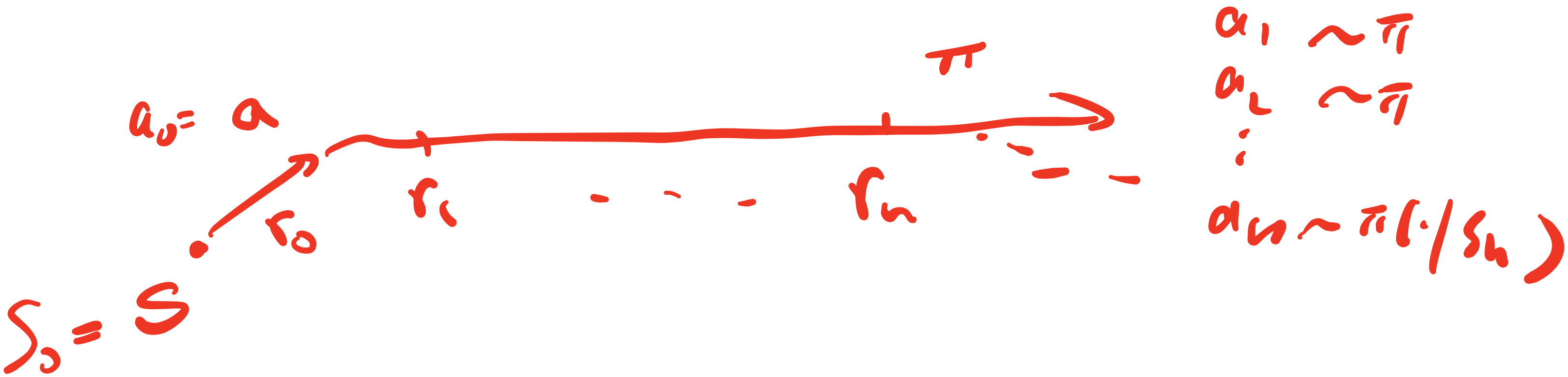
$$V^\star(s) \geq V^\pi(s), \forall s, \forall \pi$$

Fact: $\pi^\star$ is deterministic, only depends on state (i.e., Markovian);
$\pi^\star$always exists for any infinite horizon discounted MDP

Q: what is the optimal policy when $\gamma = 0$?

State-action Q function

$$Q^{\pi}(s, a) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \mid s_0 = s, a_0 = a, \pi\right]$$



Outlines:

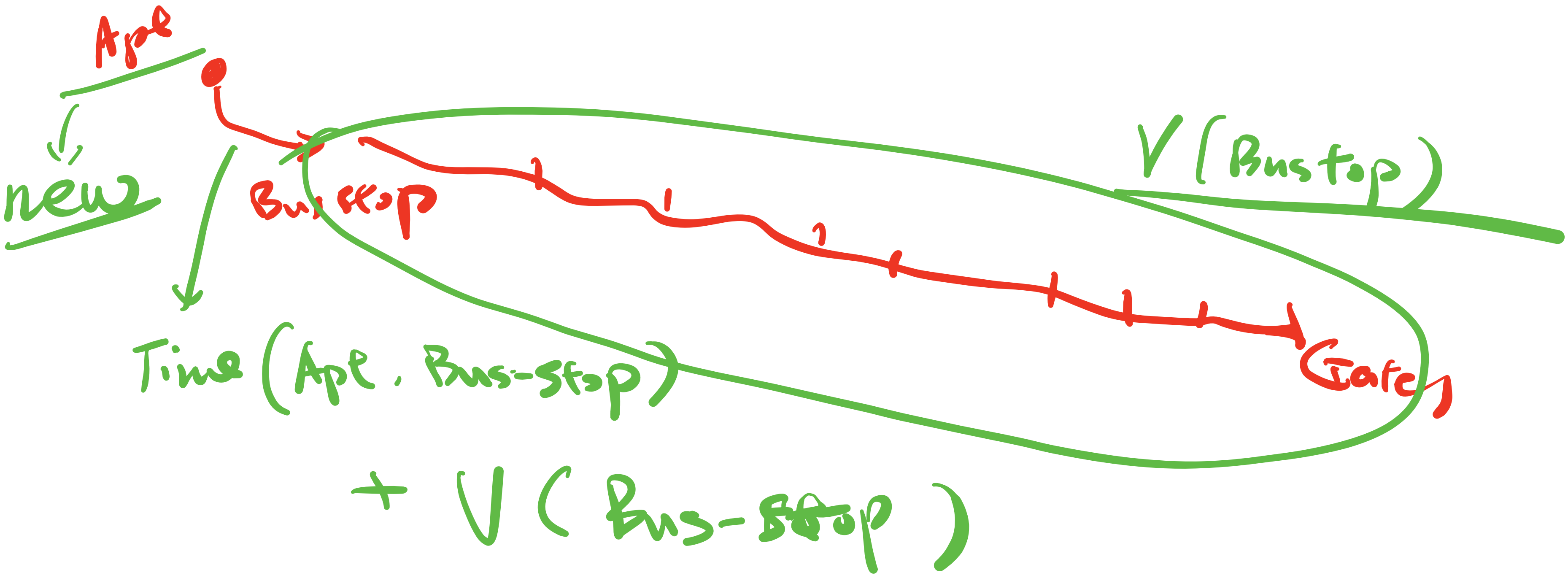
✓ 1. Definitions of Markov Decision Process

✓ 2. Value functions (V and Q functions)

3. Bellman equations

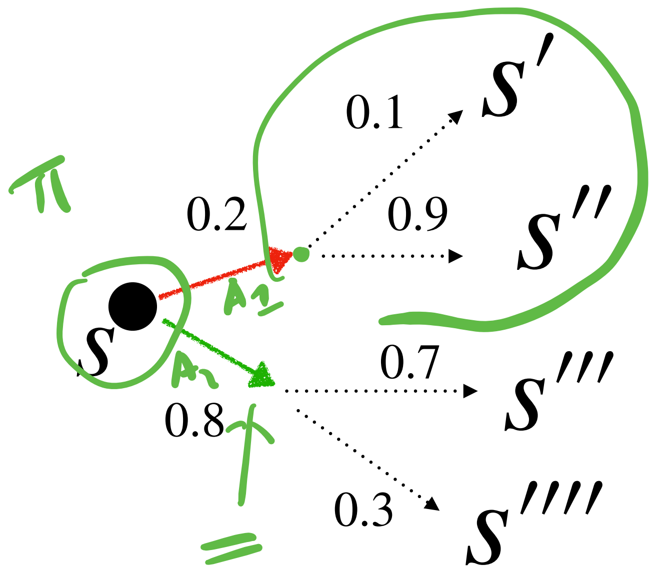
Can we quantify V / Q using one-step transition?

$$V^{\pi}(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \mid s_0 = s, a \sim \pi\right]$$



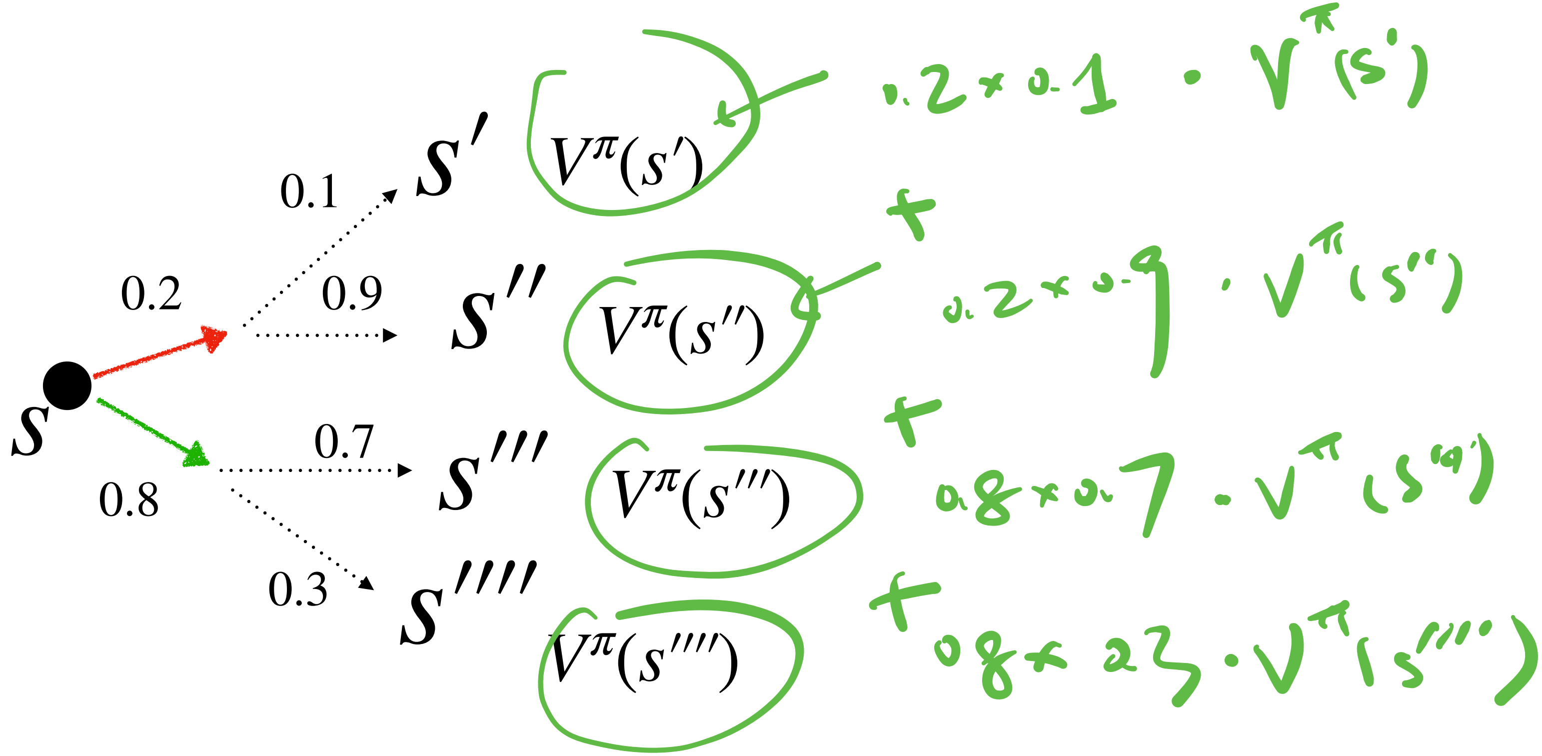
Can we quantify V / Q using one-step transition?

$$V^{\pi}(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \mid s_0 = s, a \sim \pi\right]$$



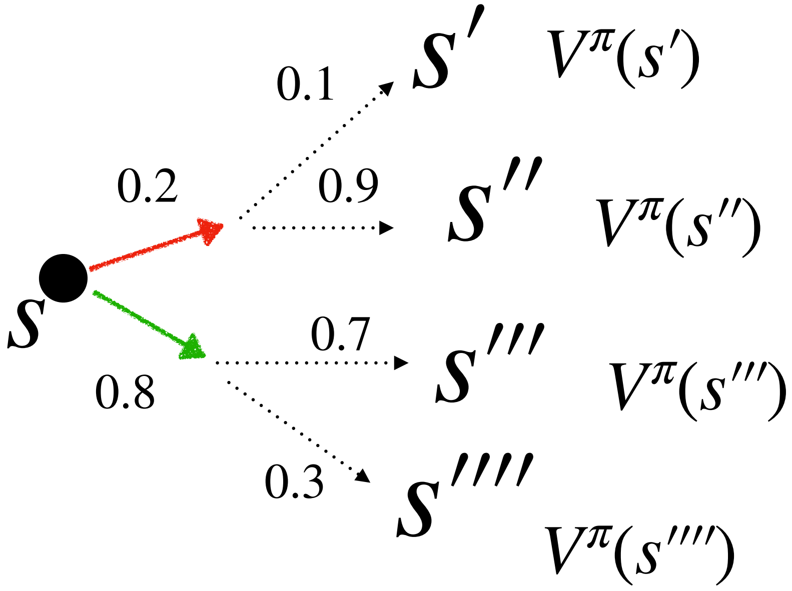
Can we quantify V / Q using one-step transition?

$$V^\pi(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \mid s_0 = s, a \sim \pi\right]$$



Can we quantify V / Q using one-step transition?

$$V^\pi(s) = \mathbb{E}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^h r_h + \ldots \mid s_0 = s, a \sim \pi\right]$$



$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \mathbb{E}_{s'\sim P(.|s,a)}\left[V^\pi(s')\right]\right]$$

Bellman equation for value function

Can we quantify V / Q using one-step transition?

Your homework: understand the one-step relationship between V and Q

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(.|s,a)} \left[V^\pi(s') \right]$$

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} Q^\pi(s,a)$$

Summary:

- **Discounted infinite horizon MDP:**
  - State, action, policy, transition, reward (or cost), discount factor
  - **V function and Q function**
  - Key concept: **Bellman equation**