*This lecture will be recorded!!!*

Welcome to

# *DS595 Reinforcement Learning*

Prof. Yanhua Li

Time: 6:00pm –8:50pm W
Zoom Lecture
Spring 2022

# Quiz 1 today Week 4 (2/9 W)

- Model-based Control
  - Policy Evaluation, Policy Iteration, Value Iteration
  - 30 min at the beginning
  - For the second question Q#2, the initial policy is still a random policy, and the policy evaluation only needs one step to get $V_1$ as what you did in Q#1.
  - For those students who are observing the class, sorry, this time, we don't have time to let you do the Quiz 1. Next time, we will send you guys pdf to do the quizzes.

# So far

- Review: Model based control
  - Policy Iteration, and Value iteration

---

- Model-Free Policy Evaluation
  - Monte Carlo policy evaluation
  - Temporal-difference (TD) policy evaluation

# This Lecture

- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description.
- Value Function Approximation

# This Lecture

- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description.
- Value Function Approximation

# Quiz 2 next Thursday Week #6 (2/23 W)

- Model-free Control
  - Model-free policy evaluation
    - Monte Carlo policy evaluation
    - TD policy evaluation
  - Model-free control
    - SARSA
    - Q-Learning
    - Double-Q-Learning

# Model-based vs -free RL algorithms

## Tabular Representation Value Function

- Model-based control
  - Policy evaluation (DP)
  - Policy iteration
  - Value iteration

- Model-Free Control
  - Policy evaluation
    - Monte Carlo (MC)
      - First visit
      - Every visit
    - Temporal Difference (TD)
  - Value/Policy Iteration
    - MC Policy Iteration
    - TD Policy Iteration
      - SARSA
      - Q-Learning
      - Double Q-Learning

# Model-Free Control Problems

- Examples:


Autonomous vehicles (AVs)

Robocomp

Patient treatment

- Model-Free Control
  - For many of these and other problems
    - Either MDP model is unknown but can be sampled (e.g., AVs)
    - Or MDP model is known but it is computationally infeasible to use directly, except through sampling (e.g., Go Game).

# Recall Policy Iteration

Loop

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $V^{\pi}$
  - Policy improvement: update $\pi$

Policy evaluation

Policy improvement

$$\pi'(s) = \arg\max_a \left[ R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi}(s') \right] = \arg\max_a Q^{\pi}(s,a)$$

- Now want to do the above two steps without access to the true dynamics and reward models
- Last lecture introduced methods for model-free policy evaluation

# Model Free Policy Iteration

?

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^\pi$ With model-free policy evaluation
  - Policy improvement: update $\pi$

# Model Free Policy Iteration

?

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^\pi$ With model-free policy evaluation
  - Policy improvement: update $\pi$ $\pi'(s) = \arg\max_a Q^\pi(s, a)$

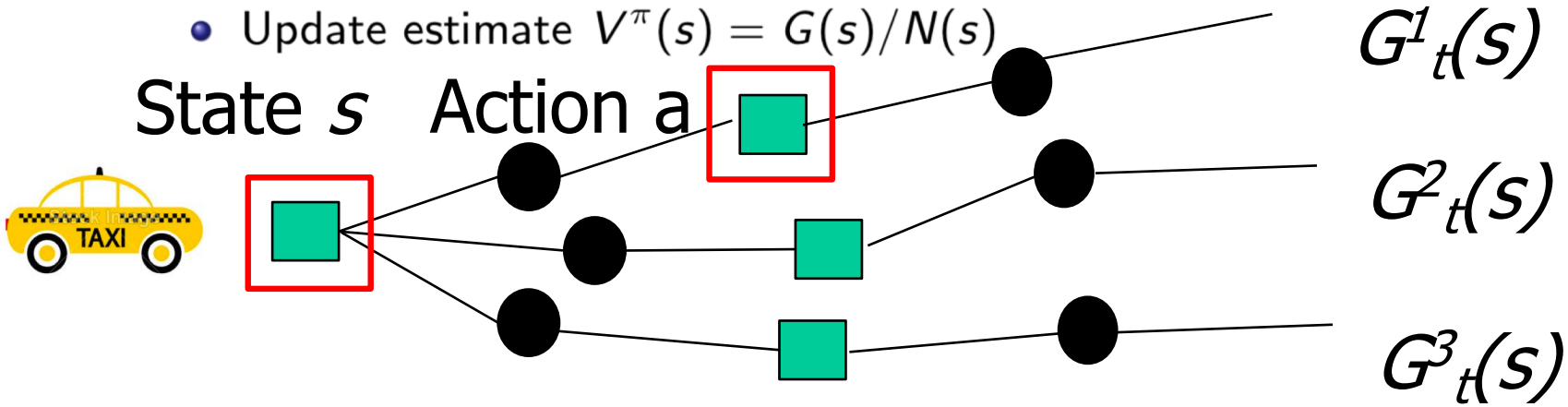
# First-Visit Monte Carlo (MC) On Policy Evaluation

## Evaluation for $V^{\pi}(s)$

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step $t$ onwards in $i$th episode
- For each state $s$ visited in episode $i$
  - For **first** time $t$ that state $s$ is visited in episode $i$
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

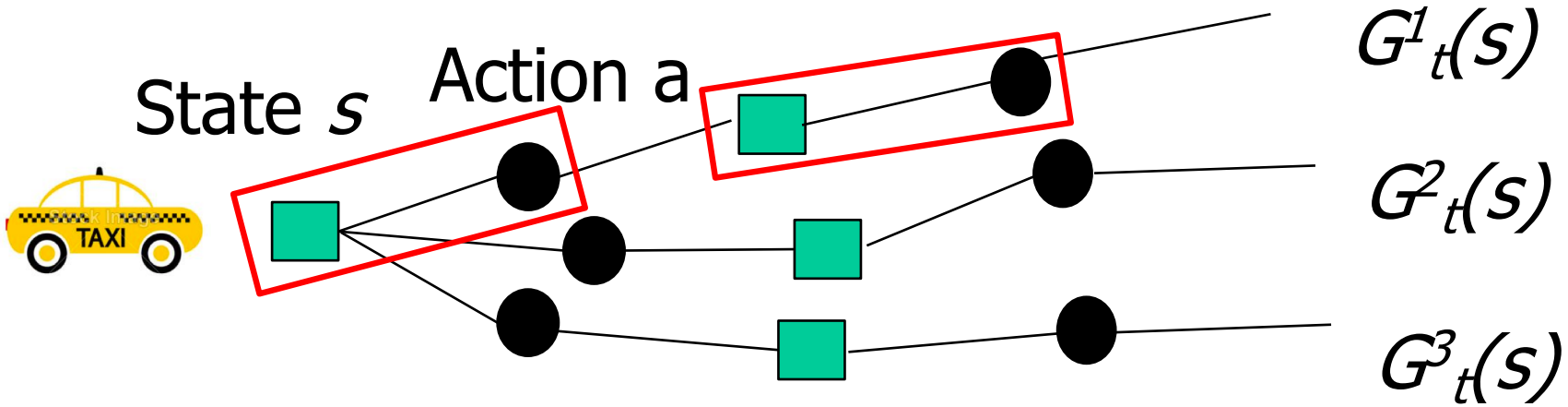


# MC for On Policy Q Evaluation

## Evaluation for $Q^{\pi}(s,a)$

Initialize $N(s,a) = 0,\ G(s,a) = 0,\ Q^{\pi}(s,a) = 0,\ \forall s \in S,\ \forall a \in A$
Loop

- Using policy $\pi$ sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$
- For each **state,action** $(s,a)$ visited in episode $i$
  - For **first or every** time $t$ that $(s,a)$ is visited in episode $i$
    - $N(s,a) = N(s,a) + 1,\ G(s,a) = G(s,a) + G_{i,t}$
    - Update estimate $Q^{\pi}(s,a) = G(s,a)/N(s,a)$



# Model-free Policy Iteration

?

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^\pi$ (MC On Policy Q-evaluation)
  - Policy improvement: update $\pi$ given $Q^\pi$

$$\pi'(s) = \arg\max_a Q^\pi(s, a)$$

Problem with this algorithm?

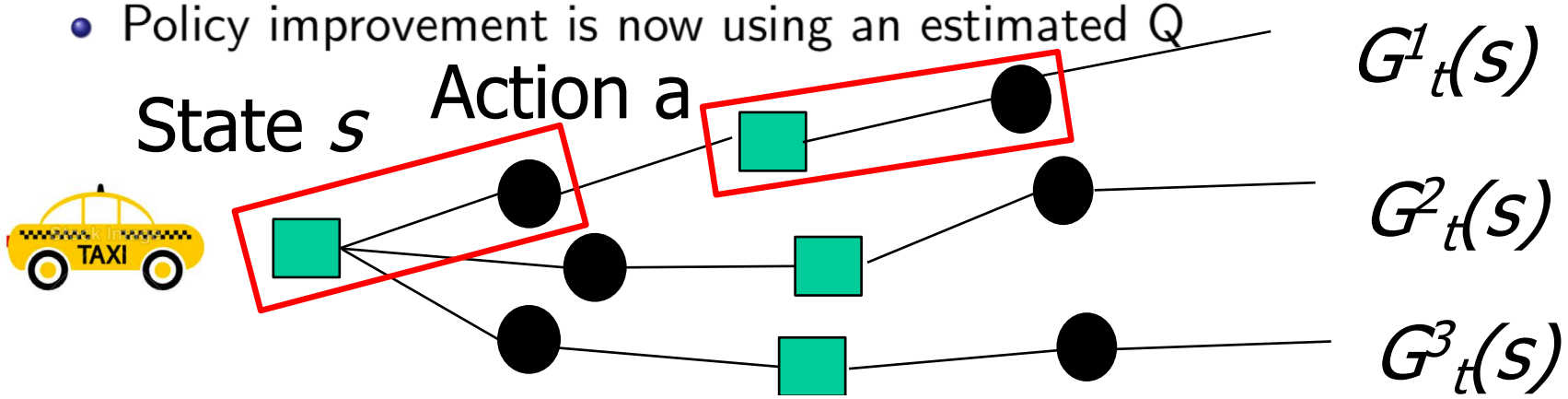
# Model-free Policy Iteration

?

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^\pi$ (MC On Policy Q-evaluation)
  - Policy improvement: update $\pi$ given $Q^\pi$

$$\pi'(s) = \arg\max_a Q^\pi(s, a)$$

- May need to modify policy evaluation:
  - If $\pi$ is deterministic, can't compute $Q(s, a)$ for any $a \neq \pi(s)$
- How to interleave policy evaluation and improvement?
  - Policy improvement is now using an estimated Q



# Road map

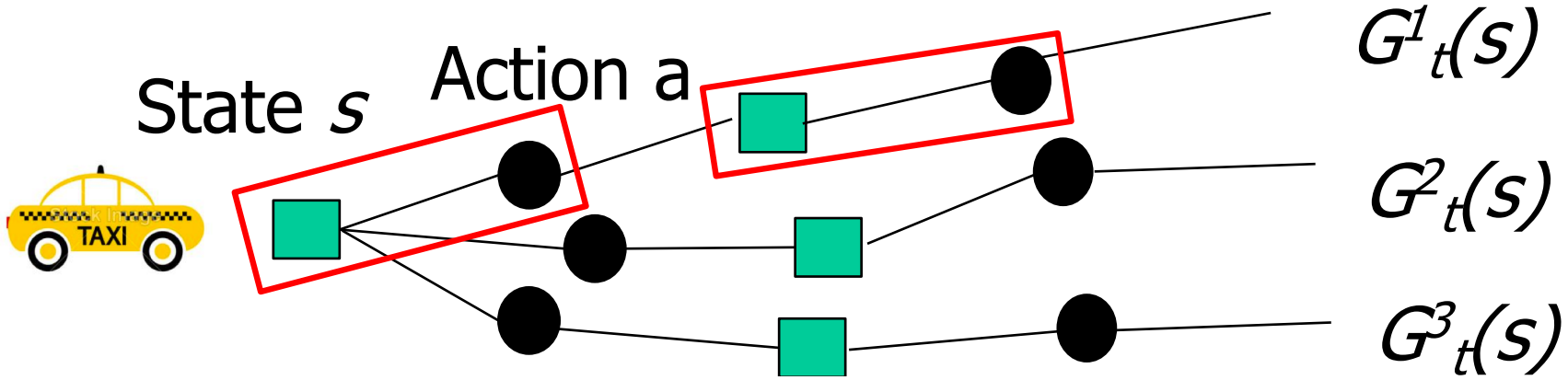
- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description.
- Value Function Approximation

# $\epsilon$-greedy Policies

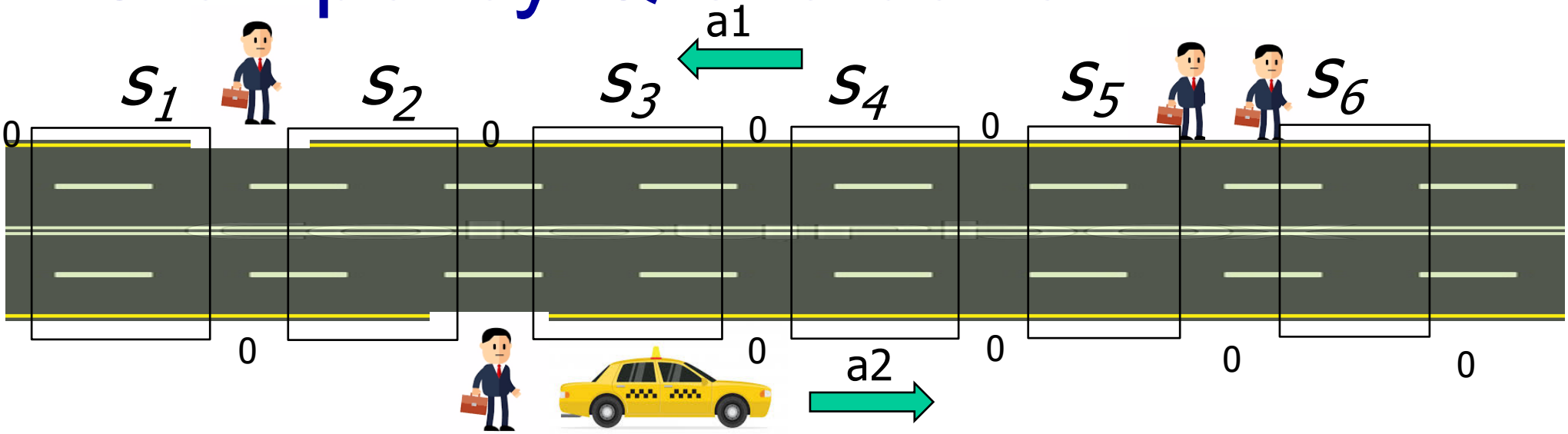
- Simple idea to balance exploration and exploitation
- Let $|A|$ be the number of actions
- Then an $\epsilon$-greedy policy w.r.t. a state-action value $Q(s, a)$ is $\pi(a|s) = [\arg\max_a Q(s, a)$, w. prob $1 - \epsilon$; a w. prob $\frac{\epsilon}{|A|}]$

Greedy Policy vs Behavior policy

- Model-Free Control with exploration
- Exploration ratio ε in (0,1)



# MC on policy Q evaluation




Taxi passenger-seeking process: $R(-,a_1)=[0,1,0,0,0,2]$, $R(-,a_2)=[0,1,0,0,0,0]$, $\gamma = 1$,

Assume the current greedy policy: $\pi(s) = a_1, \forall s.\ \varepsilon = 1.$

Sample trajectories from ε-greedy policy.

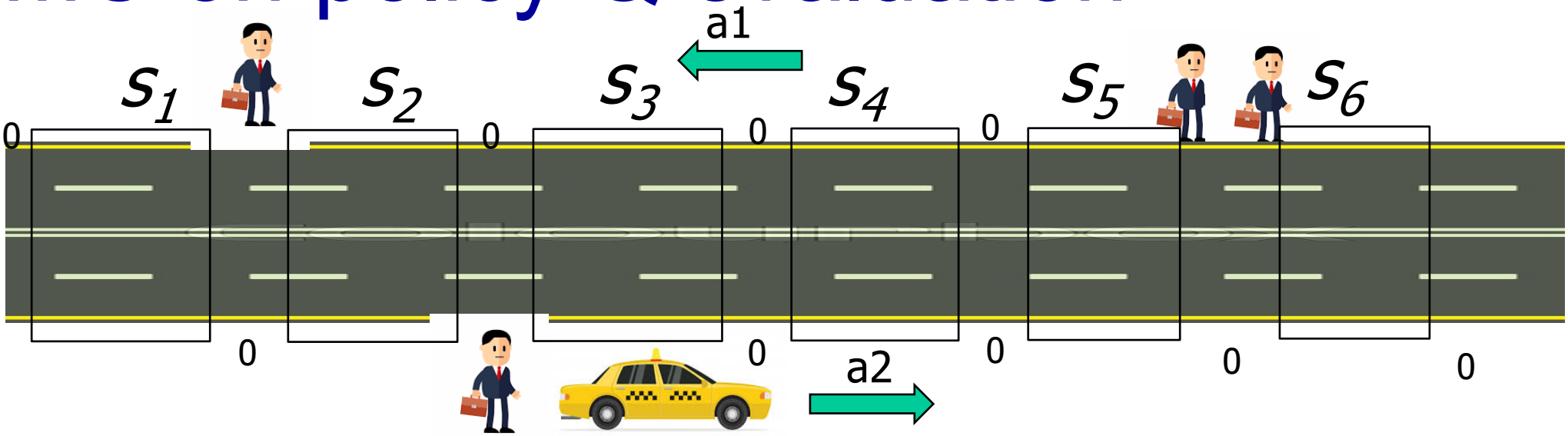
any action from $s_1$ or $s_6$ terminates an episode

Given ($s_3, a_1, 0, s_2, a_2, 1, s_3, a_1, 0, s_2, a_1, 1, s_1, a_1, 0, T$);

*Q:* First visit MC estimate Q of each state-action? Q init 0

*A:*

# MC on policy Q evaluation




Taxi passenger-seeking process: $R(-,a_1)=[0,1,0,0,0,2]$, $R(-,a_2)=[0,1,0,0,0,0]$, $\gamma = 1$,

Assume the current greedy policy: $\pi(s) = a_1, \forall s$, $\varepsilon = 1$.

Sample trajectories from ε-greedy policy.

any action from $s_1$ or $s_6$ terminates an episode

Given ($s_3, a_1, 0, s_2, a_2, 1, s_3, a_1, 0, s_2, a_1, 1, s_1, a_1, 0, T$);

*Q:* First visit MC estimate Q of each state-action? Q init 0

*A: $Q(-,a_1)=[0,1,2,0,0,0]$; $Q(-,a_2)=[0,2,0,0,0,0]$.*

# Greedy in the Limit of Infinite Exploration (GLIE)

- Choice of the exploration ratio ε for convergence.

## Definition of GLIE

- All state-action pairs are visited an infinite number of times

$$\lim_{i \to \infty} N_i(s, a) \to \infty$$

- Behavior policy converges to greedy policy
  $\lim_{i \to \infty} \pi(a|s) \to \arg\max_a Q(s, a)$ with probability 1

- A simple GLIE strategy is $\epsilon$-greedy where $\epsilon$ is reduced to 0 with the following rate: $\epsilon_i = 1/i$

# Road map

- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description.
- Value Function Approximation

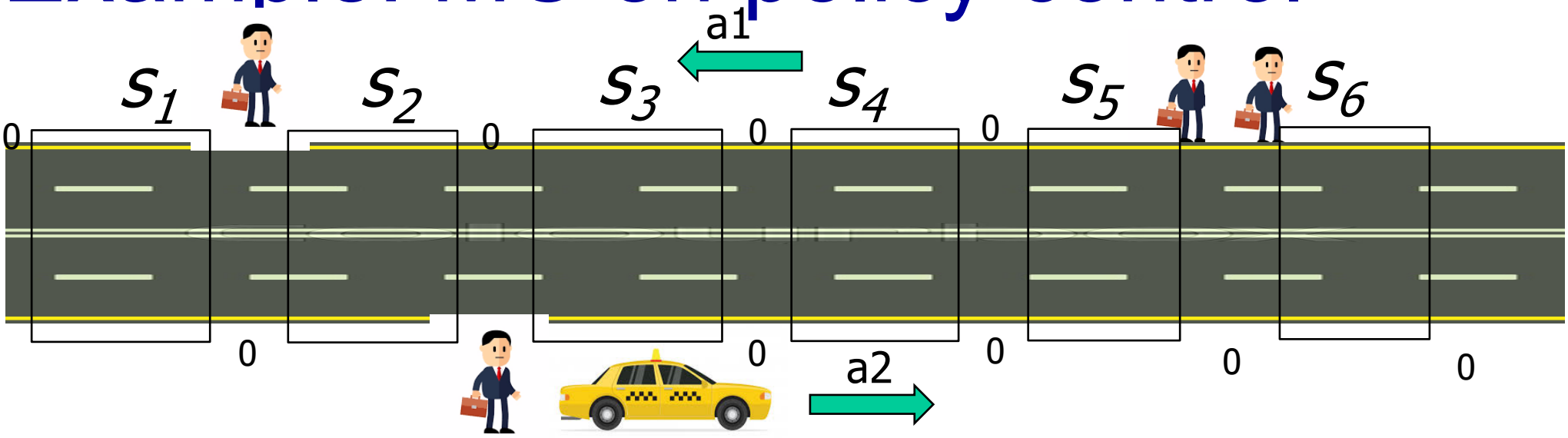
# Monte Carlo Online Control / On Policy Improvement

1: Initialize $Q(s,a) = 0, N(s,a) = 0\ \forall (s,a)$, Set $\epsilon = 1$, $k = 1$
2: $\pi_k = \epsilon$-greedy$(Q)$ // Create initial $\epsilon$-greedy policy
3: **loop**
4: Sample $k$-th episode $(s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, \ldots, s_{k,T})$ given $\pi_k$
4: $G_{k,t} = r_{k,t} + \gamma r_{k,t+1} + \gamma^2 r_{k,t+2} + \cdots \gamma^{T_i - 1} r_{k,T_i}$
5: **for** $t = 1, \ldots, T$ **do**
6: **if** First visit to $(s,a)$ in episode $k$ **then**
7: $N(s,a) = N(s,a) + 1$
8: $Q(s_t, a_t) = Q(s_t, a_t) + \frac{1}{N(s,a)}(G_{k,t} - Q(s_t, a_t))$
9: **end if**
10: **end for**
11: $k = k + 1$, $\epsilon = 1/k$
12: $\pi_k = \epsilon$-greedy$(Q)$ // Policy improvement
13: **end loop**

?

GLIE Monte-Carlo control converges to the optimal state-action value function $Q(s, a) \rightarrow Q^*(s, a)$. If you choose $\varepsilon = 1/k$, GLIE is guaranteed.

# Example: MC on policy control

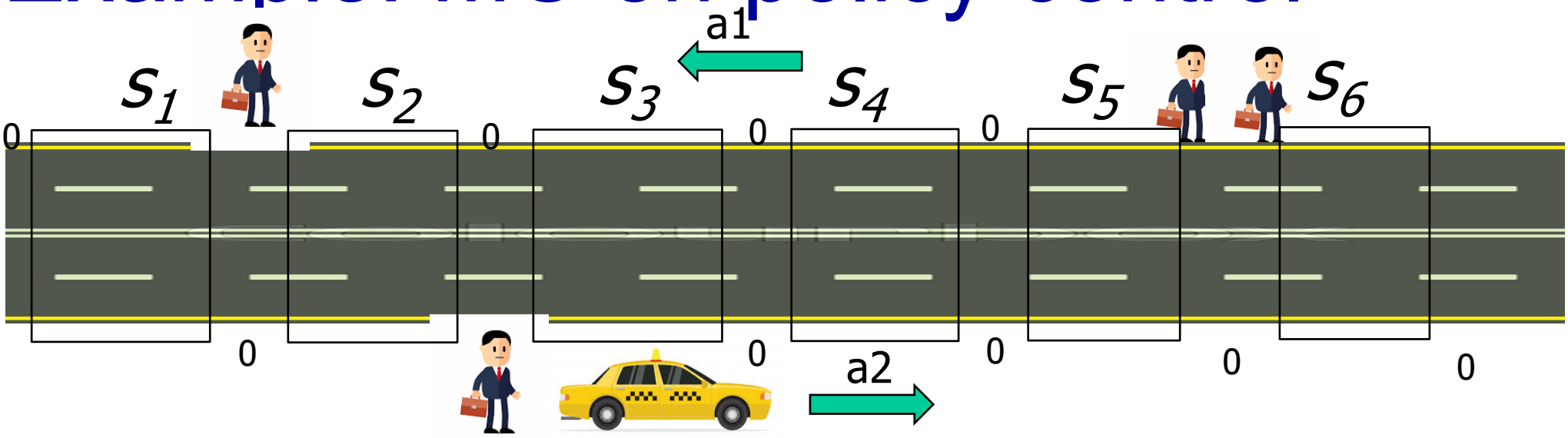



First visit MC estimate of Q of each state-action as $Q(-,a_1)=[0,1,1,0,0,0]$; $Q(-,a_2)=[0,1,0,0,0,0]$.

*Q1: What is greedy policy π(s)?*

*Q2: What is ε-greedy policy, given k=4, ε=1/k?*

# Example: MC on policy control




First visit MC estimate of Q of each state-action as $Q(-,a_1)=[0,1,1,0,0,0]$; $Q(-,a_2)=[0,1,0,0,0,0]$.

*Q1: What is greedy policy π(s)?*

*A1: π(s) = [tie, tie, a1, tie, tie, tie]*

*Q2: What is ε-greedy policy, given k=4, ε=1/k?*

*With probability ¾ following π(s), and ¼ random.*

*A2: $\pi(a_1|s)$ = [tie, tie, 7/8, tie, tie, tie]*

*$\pi(a_2|s)$ = [tie, tie, 1/8, tie, tie, tie]*

# Road map

- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description.
- Value Function Approximation

# Model-free Policy Iteration

- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^\pi$
  - Policy improvement: update $\pi$ given $Q^\pi$

- What about TD methods?

# MC + DP = TD

- Dynamic Programming (DP) policy evaluation

$$V_k^\pi(s) = \sum_{a \in A} \sum_{s' \in S} \pi(a|s) P(s'|s,a)(r + \gamma V_{k-1}^\pi(s'))$$

$$V_k^\pi(s) = \mathbb{E}_\pi[r + \gamma V_{k-1}^\pi(s'))]$$

- Monte Carlo (MC) policy evaluation
  - $V^\pi(s) = V^\pi(s) + \alpha(G_{i,t} - V^\pi(s))$
- Temporal Difference (TD)

$$V^\pi(s) = V^\pi(s) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s))$$

# Model-free Policy Iteration with TD Methods

- Use temporal difference methods for policy evaluation step
- Initialize policy $\pi$
- Repeat:
  - Policy evaluation: compute $Q^{\pi}$ using temporal difference updating with $\epsilon$-greedy policy
  - Policy improvement: Same as Monte carlo policy improvement, set $\pi$ to $\epsilon$-greedy $(Q^{\pi})$

# General Form of SARSA Algorithm

1: Set initial $\epsilon$-greedy policy $\pi$ randomly, $t = 0$, initial state $s_t = s_0$
2: Take $a_t \sim \pi(s_t)$ // Sample action from policy
3: Observe $(r_t, s_{t+1})$
4: **loop**
5: Take action $a_{t+1} \sim \pi(s_{t+1})$
6: Observe $(r_{t+1}, s_{t+2})$
7: Update Q given $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$:
8: Perform policy improvement:
9: $t = t + 1$
10: **end loop**

# General Form of SARSA Algorithm

1: Set initial $\epsilon$-greedy policy $\pi$, $t = 0$, initial state $s_t = s_0$
2: Take $a_t \sim \pi(s_t)$ // Sample action from policy
3: Observe $(r_t, s_{t+1})$
4: **loop**
5: Take action $a_{t+1} \sim \pi(s_{t+1})$
6: Observe $(r_{t+1}, s_{t+2})$
7: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
8: $\pi(s_t) = \arg\max_a Q(s_t, a)$ w.prob $1 - \epsilon$, else random
9: $t = t + 1$
10: **end loop**

What are the benefits to improving the policy after each step?

Convergence?

# General Form of SARSA Algorithm

1: Set initial $\epsilon$-greedy policy $\pi$, $t = 0$, initial state $s_t = s_0$
2: Take $a_t \sim \pi(s_t)$ // Sample action from policy
3: Observe $(r_t, s_{t+1})$
4: **loop**
5: Take action $a_{t+1} \sim \pi(s_{t+1})$
6: Observe $(r_{t+1}, s_{t+2})$
7: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
8: $\pi(s_t) = \arg\max_a Q(s_t, a)$ w.prob $1 - \epsilon$, else random
9: $t = t + 1$
10: **end loop**

Convergence? *Yes, if ε and α follow 1/t to decay.*
*ε= 1/t leads to* GLIE, and *α=1/t leads to a* Robbins-Munro sequence.
*This is a sufficient condition.*

# TD-based model-free control
# Q-learning

- Recall SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha((r_t + \gamma Q(s_{t+1}, a_{t+1})) - Q(s_t, a_t))$$

- Q-learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha((r_t + \gamma \max_{a'} Q(s_{t+1}, a')) - Q(s_t, a_t))$$

# Q-Learning with $\epsilon$-greedy Exploration

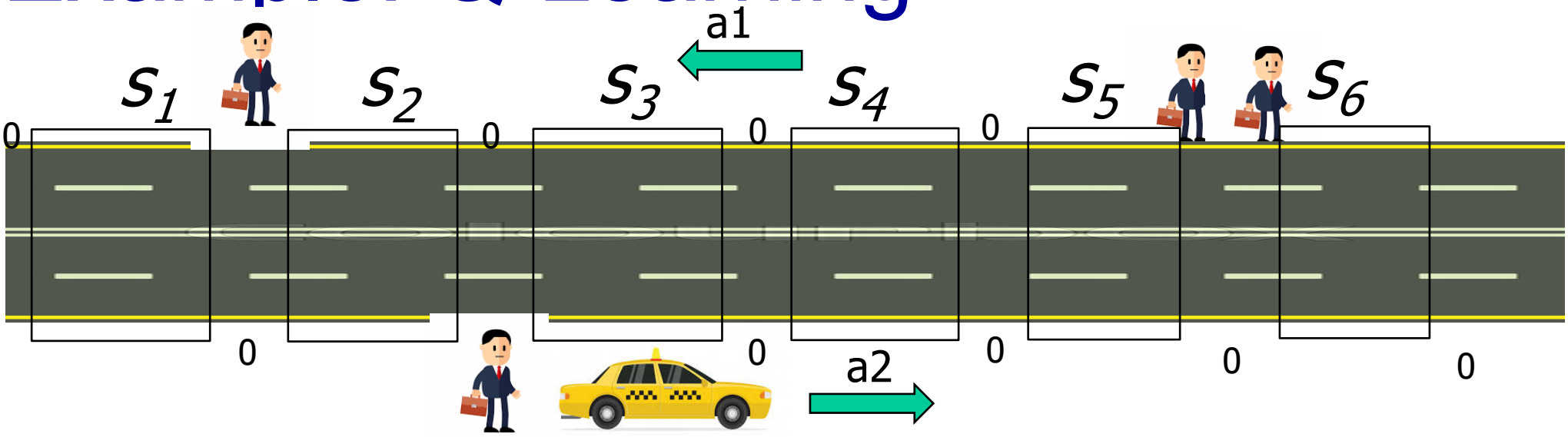
1: Initialize $Q(s, a), \forall s \in S, a \in A$ $t = 0$, initial state $s_t = s_0$
2: Set $\pi_b$ to be $\epsilon$-greedy w.r.t. $Q$
3: **loop**
4: Take $a_t \sim \pi_b(s_t)$ // Sample action from policy
5: Observe $(r_t, s_{t+1})$
6: Update Q given $(s_t, a_t, r_t, s_{t+1})$:
7: Perform policy improvement: set $\pi_b$ to be $\epsilon$-greedy w.r.t. $Q$
8: $t = t + 1$
9: **end loop**

# Q-Learning with $\epsilon$-greedy Exploration

1: Initialize $Q(s, a), \forall s \in S, a \in A$ $t = 0$, initial state $s_t = s_0$
2: Set $\pi_b$ to be $\epsilon$-greedy w.r.t. $Q$
3: **loop**
4: Take $a_t \sim \pi_b(s_t)$ // Sample action from policy
5: Observe $(r_t, s_{t+1})$
6: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \quad \max_a Q(s_{t_1}, a) - Q(s_t, a_t))$
7: $\pi(s_t) = \arg\max_a Q(s_t, a)$ w.prob $1 - \epsilon$, else random
8: $t = t + 1$
9: **end loop**

Does how $Q$ is initialized matter?
Asymptotically no, under mild condiditions, but at the beginning, yes

# Example: Q-Learning




π is random with probability ε=1/k, else $\pi_k$, α=0.5, γ=1,

Q-learning: Init $Q(-,a_1) = Q(-,a_2) = [0,0,0,0,0,0]$;

*1st step (k=1):* First tuple: $(s_3, a_1, 0, s_2)$

$Q(s_3,a_1) \leftarrow Q(s_3,a_1) + \alpha(r + \gamma \max_a Q(s_2,a) - Q(s_3,a_1))$

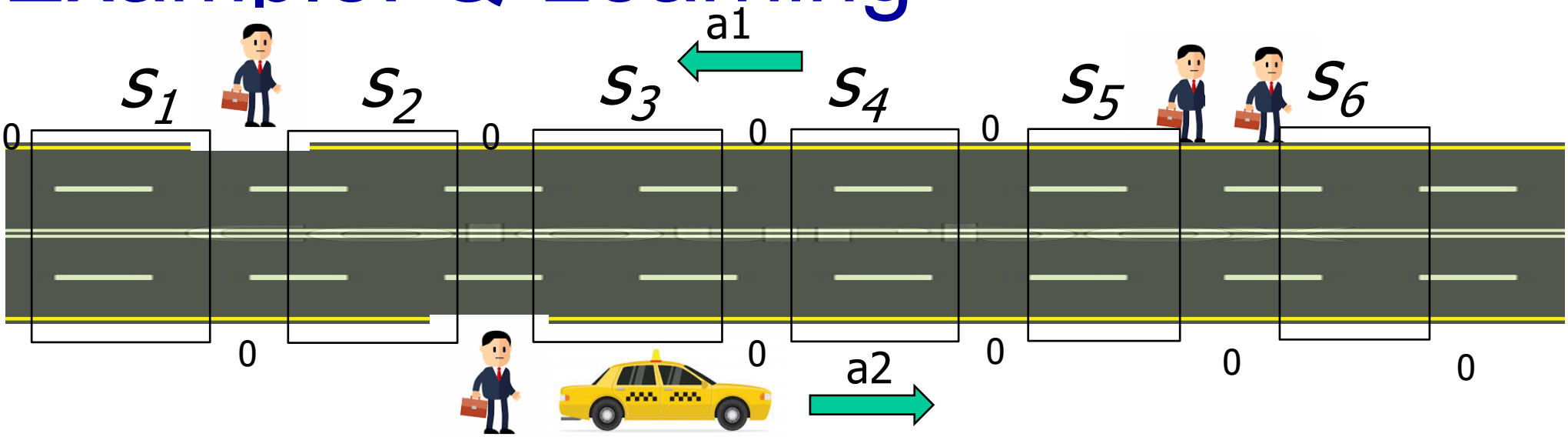
Update $Q(s_3,a_1)$ to ?.

*2nd step (k=2):* Second tuple: $(s_2, a_1, 1, s_1)$

$Q(s_2,a_1) \leftarrow Q(s_2,a_1) + \alpha(r + \gamma \max_a Q(s_1,a) - Q(s_2,a_1))$

Update $Q(s_2,a_1)$ to?.

# Example: Q-Learning




π is random with probability ε=1/k, else $\pi_k$, α=0.5, γ=1,

Q-learning: Init $Q(-,a_1) = Q(-,a_2) = [0,0,0,0,0,0]$;

*1st step (k=1):* First tuple: $(s_3, a_1, 0, s_2)$

$Q(s_3,a_1) \leftarrow Q(s_3,a_1) + \alpha(r + \gamma \max_a Q(s_2,a) - Q(s_3,a_1)) = 0$

Update $Q(s_3,a_1)$ to 0.

*2nd step (k=2):* Second tuple: $(s_2, a_1, 1, s_1)$

$Q(s_2,a_1) \leftarrow Q(s_2,a_1) + \alpha(1 + \gamma \max_a Q(s_1,a) - Q(s_2,a_1)) = 1/2$

Update $Q(s_2,a_1)$ to ½.

# Road map

- Model-Free Control
  - Generalized policy iteration
  - Control with Exploration
  - Monte Carlo (MC) Policy Iteration
  - Temporal-Difference (TD) Policy Iteration
    - SARSA
    - Q-Learning
  - Project 2 description
- Value Function Approximation

# Project 2 starts next Wed
# Due 3/2 Week #7 (W) mid-night

- https://github.com/yingxue-zhang/DS595-RL-Projects/tree/master/Project2

- Beyond the requirement from Project 2, you are encouraged to try different methods, such as SARSA, Q-Learning, for both scenarios.

Questions?