Lecture 15 - March 2, 2023

- Temporal Difference Learning
  - TD(0)
  - SARSA
  - Q-Learning
  - On-Policy Vs. off-policy

HW3 → Due March 17

Project 2 → Due March 7

TA's office hour: Wendsdays, 2pm-3pm (in-person)
Fridays, 2pm-3pm (virtual)

| | | | |
|---|---|---|---|
| Known MDPs | Finite S&A | Dynamic Programing: Policy Iteration (PI) | |
| | | Dynamic Programing: Value Iteration (VI) | |
| | Large S&A | Approximate Dynamic Programing (ADP) | |
| Unknown MDPs | Finite S & A | Monte Carlo Methods (MC) | |
| | | Temporal Difference (TD) Learning: Q-Learning | |
| | | Temporal Difference (TD) Learning: SARSA | |
| | | Temporal Difference (TD) Learning: Double Q-Learning | |
| | | Temporal Difference (TD) Learning: SARSA ($\lambda$) | |
| | | Temporal Difference (TD) Learning: Actor Critic | |
| | Large S & Finite A | Batch Learning | Least Squares Policy Iteration (LSPI) |
| | | | Neural Fitted Q Iteration (NFQI) |
| | | Interactive Learning | Deep Reinforcement Learning (DRL): Deep Q Network (DQN) |
| | | | Deep Reinforcemat Learning (DRL): Double DQN |
| | | | Deep Reinforcement Learing (DRL): Dueling DQN |
| | | | Deep Reinforcement Learning (DRL): Prioritized DQN |
| | Larges Continous A | Policy Gradient (PG): REINFORCE | |
| | | Policy Gradient (PG): REINFORCE with Baseline | |
| | | Policy Gradient (PG): one-Step Actor Critic | |
| | | Policy Gradient (PG): Deep Deterministic Policy Gradient (DDPG) | |

# Temporal Difference Learning (TD Learning)

MDP: $(S, A, \cancel{T(s'|s,a)}, \cancel{R(s,a,s')})$

Known Transition $\Rightarrow$ DP

Unknown Trans $\Rightarrow$ MC

$$V_\pi(s) = E[G_t \mid s_t = s, \pi] \approx \frac{1}{N}\sum_{i=1}^{N} G_t^i \quad \text{MC}$$

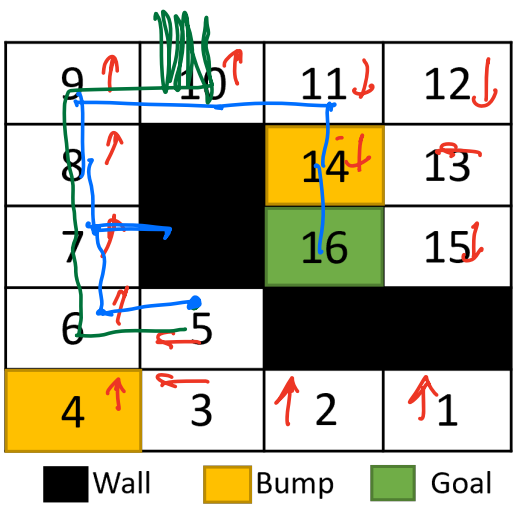
$$= E[R_{t+1} + \gamma V_\pi(s_{t+1}) \mid s_t = s, \pi]$$

$$= \sum_{s'} P(s'|s, \pi(s)) [R_{t+1} + \gamma V_\pi(s')] \quad \text{DP}$$

$$V_\pi(s) = \frac{1}{N}\sum_{i=1}^{N} G_t^i$$

$$V_\pi(s) = 0.9[-1 + \gamma V_\pi(s)] + \frac{0.1}{3}[-1 + \gamma V_\pi(5)] + \frac{0.1}{3}[-1 + \gamma V_\pi(3)]$$

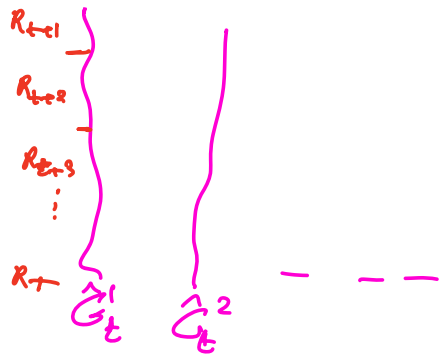
$$\rightarrow \frac{0.1}{3}[-1 + \gamma V_\pi(s)]$$



Monte Carlo

$$V(s) = V(s) + \alpha \left[ G_t^n - V(s) \right]$$

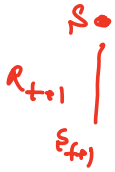
$$V(s) = \frac{G_t^1 + G_t^2 - - - + G_t^N}{N}$$




$$G_t^1 = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

Temporal Difference Learning

$$V(s) = V(s) + \alpha \left[ R_{t+1} + \gamma V(s_{t+1}) - V(s) \right]$$




$$V^{\pi}(s) = E\left[ \underbrace{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots}_{G_t} \mid s_t = s, \pi \right]$$

$$V(S_t) = V(S_t) + \alpha \left[ R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

$R_{t+1} + \gamma V(S_{t+1})$ : TD Target

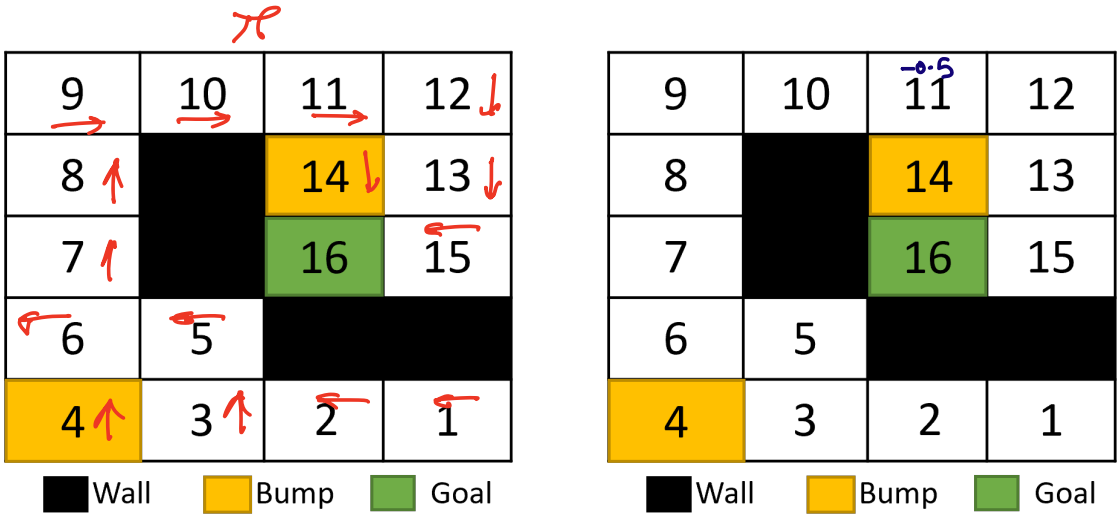
$R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ : TD Error

$$V_\pi(s) = \sum_{s'} P(s' | s, \pi(s)) \left[ R + \gamma V_\pi(s') \right] \rightarrow DP$$

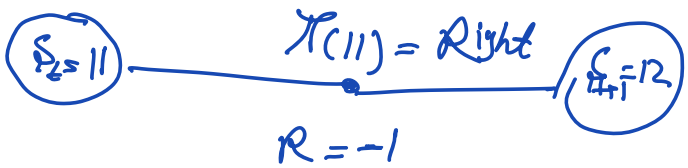
$$V(s) = V(s) + \alpha \left[ G_t - V(s) \right] \rightarrow MC$$

$$V(S_t) = V(S_t) + \alpha \left[ R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

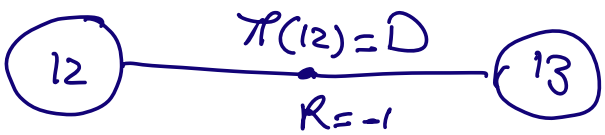
↳ TD



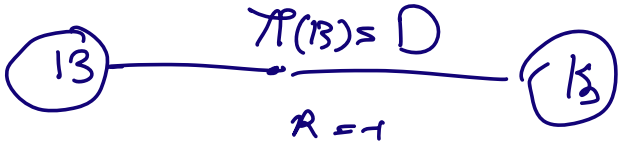

Random state



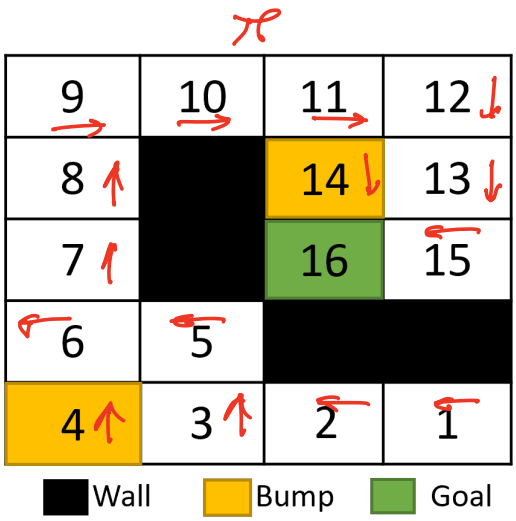

$$V(11) = \underbrace{V(11)}_{0} + \underbrace{\alpha}_{0.5}\left[\underbrace{R}_{-1} + \underbrace{\gamma}_{0.9}\underbrace{V(12)}_{0} - \underbrace{V(11)}_{0}\right] = -0.5$$

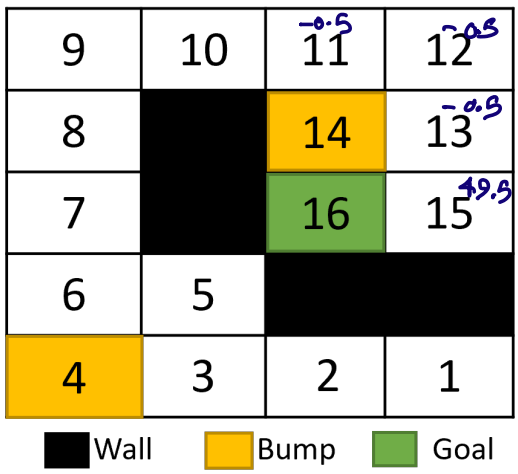


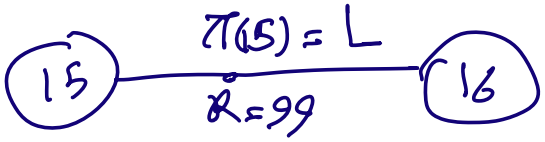

$$V(12) = \underbrace{V(12)}_{0} + \underbrace{\alpha}_{0.5}\left[\underbrace{R}_{-1} + \gamma\underbrace{V(13)}_{0} - \underbrace{V(12)}_{0}\right] = -0.5$$



$V(13) = V(13) + \alpha [R + \gamma V(15) - V(13)] = -0.5$



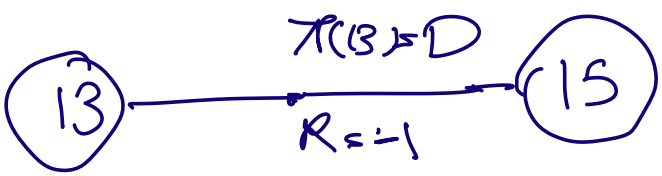




[end of episode]

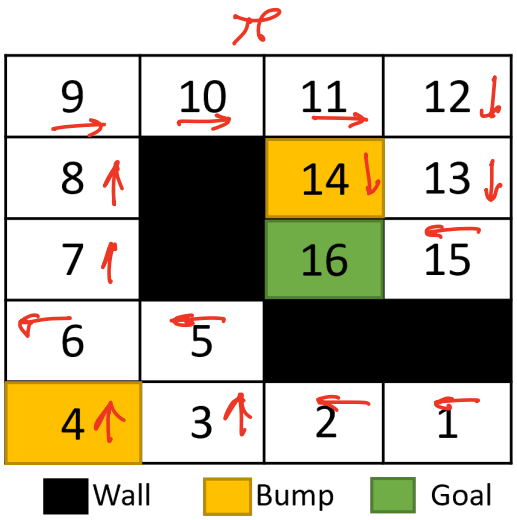
$V(15) = \underset{0}{V(15)} + \underset{0.5}{\alpha} [\underset{99}{R} + \gamma \underset{0}{V(Goal)} - \underset{0}{V(15)}] = 49.5$

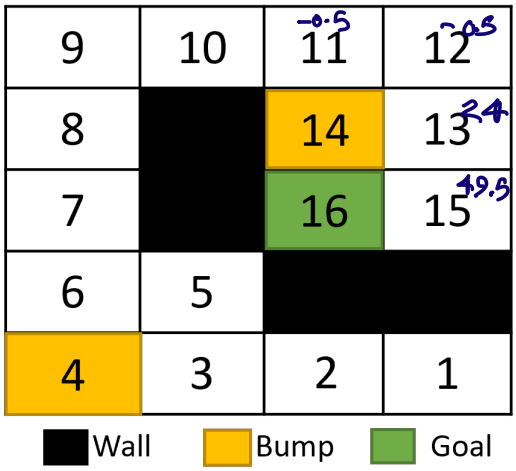
## Episode 2

Random state




$V(13) = \underset{-0.5}{V(13)} + \underset{0.5}{\alpha} [\underset{-1}{R} + \gamma \underset{49.5}{V(15)} - \underset{-0.5}{V(13)}] = 24$




(15) —— $\pi(15)=L$ ——> (Goal)

$R=99$

[Terminal End of episode]

$$V(15) = \underbrace{V(15)}_{49.5} + \underbrace{\alpha}_{0.5}\left[\underbrace{R}_{99} + \gamma \underbrace{V(\text{Goal})}_{0} - \underbrace{V(15)}_{49.5}\right]$$

$$= 74.25$$

---

episode 3

(3) —— $\pi(3)=0$ ——> (5)

$R=-1$

$V(3) = -0.5$

**Tabular TD(0) for estimating $v_\pi$**

Input: the policy $\pi$ to be evaluated
Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0$, for all $s \in \mathcal{S}^+$)
Repeat (for each episode):
  Initialize $S$
  Repeat (for each step of episode):
    $A \leftarrow$ action given by $\pi$ for $S$
    Take action $A$, observe $R$, $S'$
    $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    $S \leftarrow S'$
  until $S$ is terminal

$\pi \longrightarrow V^\pi \Longrightarrow \pi' > \pi$

$\pi'(s) = \underset{a \in A}{\operatorname{argmax}} \sum_{s'} p(s'|s,a)\,[R + \gamma V_\pi(s')]$

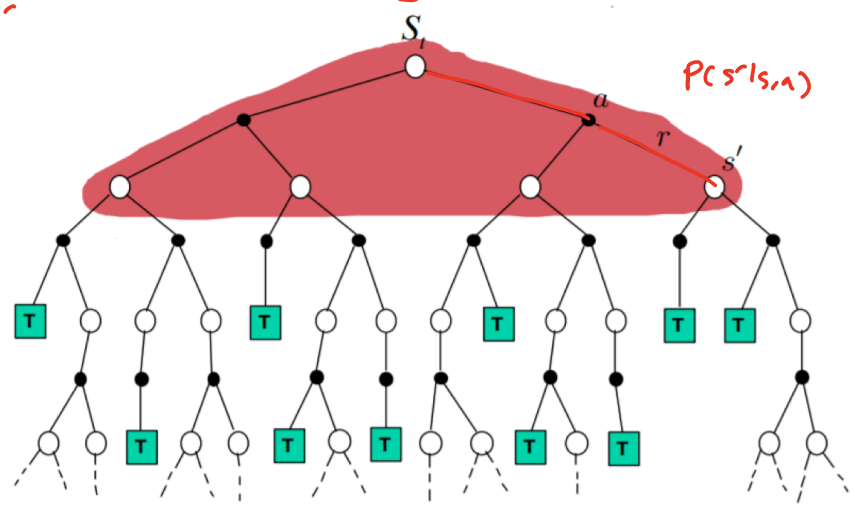
Unknown

$Q_\pi(s,a)$

$\pi \longrightarrow Q^\pi(s,a) \longrightarrow \pi'$

DP: $V_{k+1}(s) = \max_{a \in A} \sum_{s'} P(s'|s,a)[R + \gamma V_k(s')]$

Model of system

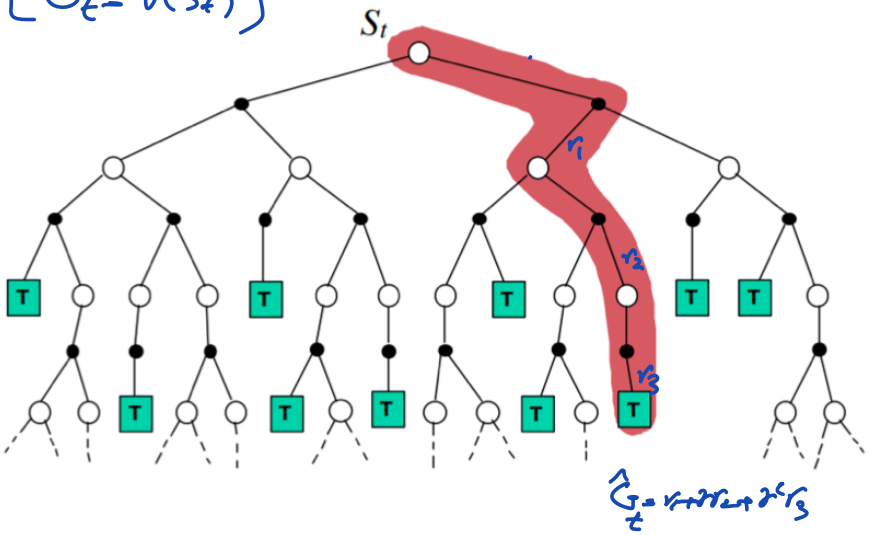



MC: $V(S_t) = V(S_t) + \alpha[\hat{G}_t - V(S_t)]$

Model X

Delay ✓

Interative X

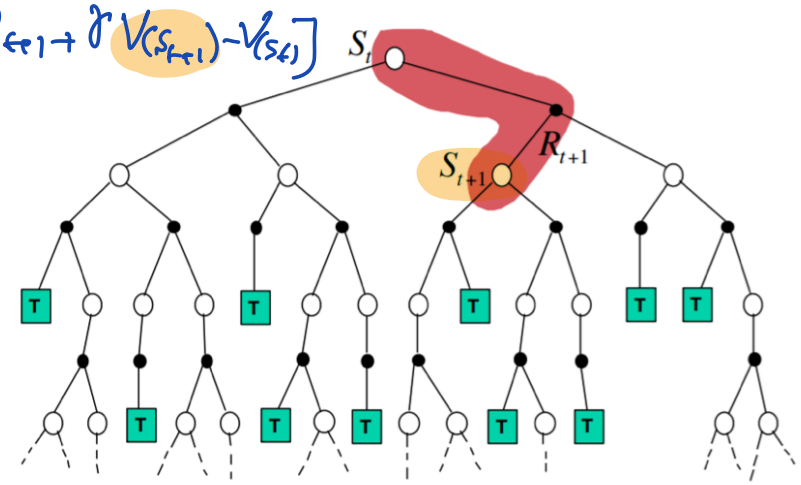


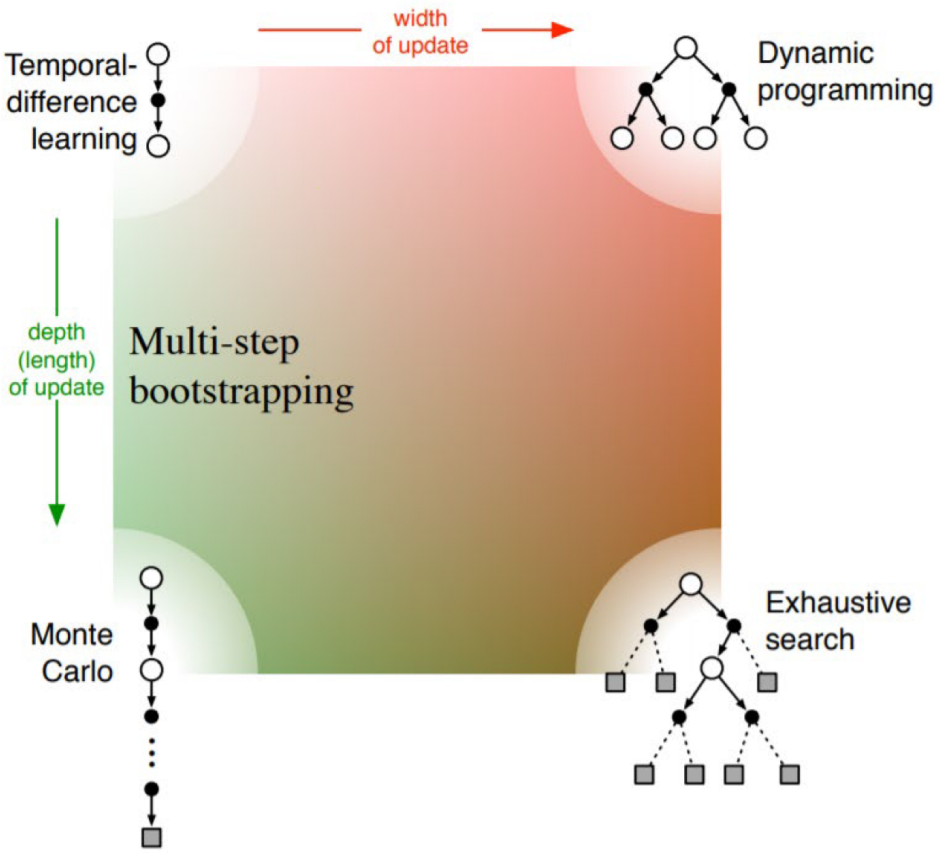

TD: $V(S_t) = V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$

Model X

Delay one step

Interative ✓



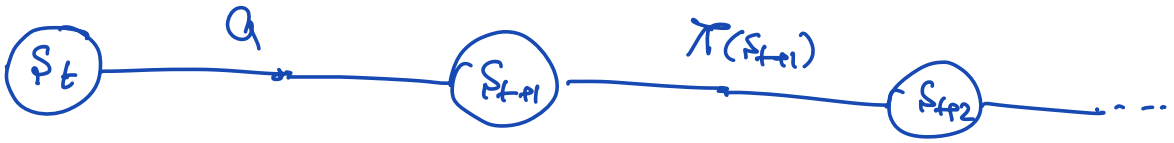



| | | | |
|---|---|---|---|
| 9 | 10 | 11 | 12 |
| 8 | | 14 | 13 |
| 7 | | 16 | 15 |
| 6 | 5 | | |
| 4 | 3 | 2 | 1 |

Wall Bump Goal

$V_\pi(s) \longrightarrow \pi'(s) = \sum P(s'|s)$ ~~$\cdot$~~

$Q_\pi(s,a) \longrightarrow \pi'(s) = \underset{a \in A}{\text{argmax}}\ Q_\pi(s,a)$




TD: $V(S_t) = V(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right]$

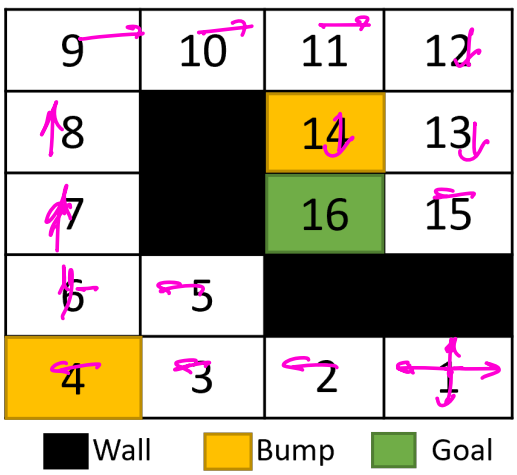
$Q(S_t, \pi(S_t)) = Q(S_t, \pi(S_t)) + \alpha\left[R_{t+1} + \gamma Q(S_{t+1}, \pi(S_{t+1})) - Q(S_t, \pi(S_t))\right]$

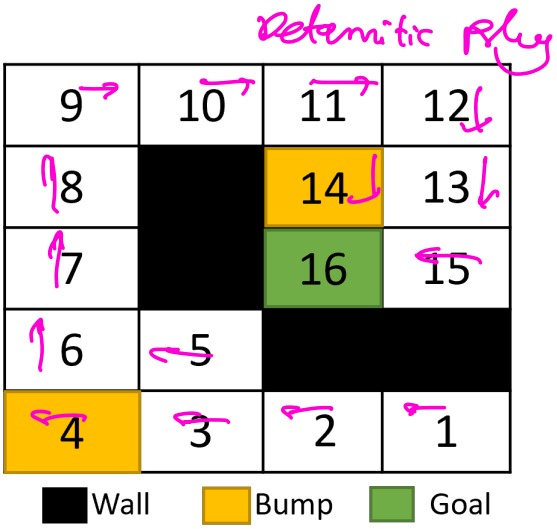
1- How to improve policy

2- How to update $Q$ in interative way

$\varepsilon$-greedy

$$\pi(S_t) = \begin{cases} \underset{a \in A}{\text{argmax}}\ Q(S_t, a) & \text{w.p. } 1-\varepsilon \\ \text{Random} & \text{w.p. } \varepsilon \end{cases}$$

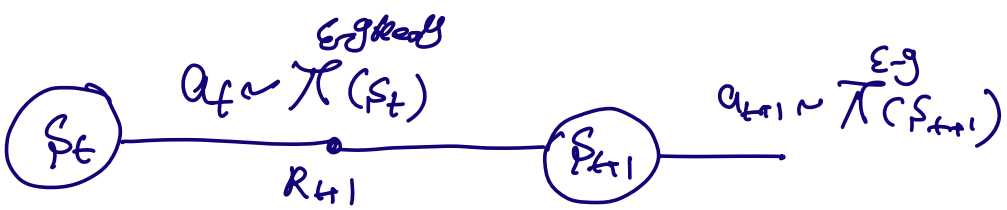



$$V^{\varepsilon\text{-greedy}}(s) = E[R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s, \pi^{\varepsilon\text{-greedy}}]$$

$$Q^{\varepsilon\text{-greedy}}(s,a) = E[R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s, a_t = a, \pi^{\varepsilon\text{-greedy}}]$$

---




$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left[R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]$$

State – Action – Reward – State – Action

# SARSA algorithm

**SARSA**

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize $S$
  Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
  Repeat (for each step of episode):
    Take action $A$, observe $R$, $S'$
    Choose $A'$ from $S'$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma Q(S', A') - Q(S, A)]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until $S$ is terminal