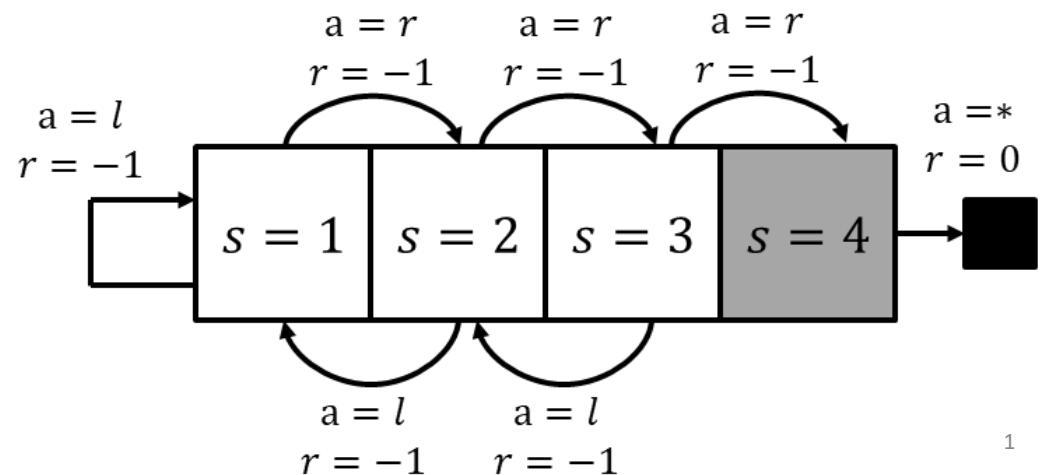
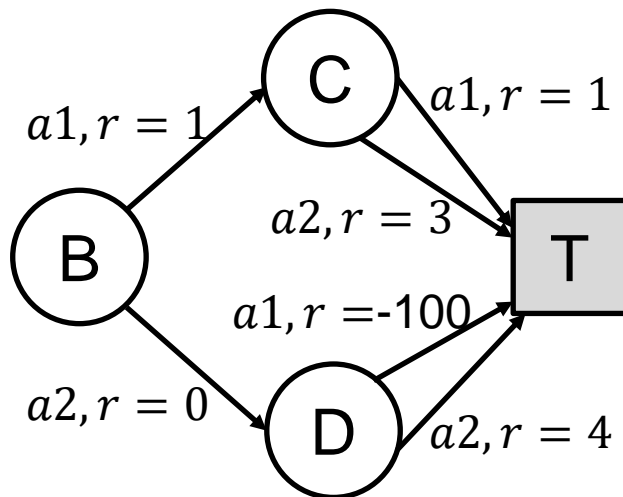


# L7.2.X Worked Examples

Zonghua Gu 2021



# Recall: Simplified Bellman Equations for Deterministic Env

- Bellman Equations:
  - $v_{\pi}(s) = \sum_a \pi(a|s) q_{\pi}(s, a)$
  - $q_{\pi}(s, a) = \sum_{r, s'} p(r, s'|s, a) [r + \gamma v_{\pi}(s')]$
  - $v_*(s) = \max_a q_*(s, a)$
  - $q_*(s, a) = \sum_{r, s'} p(r, s'|s, a) [r + \gamma v_*(s')]$
- For Deterministic Env: there is only one possible  $(r, s')$  for a given  $(s, a)$  (we use  $R_s^a$  to emphasize that reward  $r$  is specific to this  $(s, a)$ ):
  - $q_{\pi}(s, a) = R_s^a + \gamma v_{\pi}(s')$
  - $q_*(s, a) = R_s^a + \gamma v_*(s')$

# Recall: MC, TD, Sarsa, Q Learning

- MC (every-visit):
  - $V(S_t) \leftarrow V(S_t) + \alpha(G(S_t) - V(S_t))$ 
    - $G(S_t)$  can also be written as  $G_t$
- TD:
  - $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$
- Sarsa:
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$
- QL:
  - $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t))$

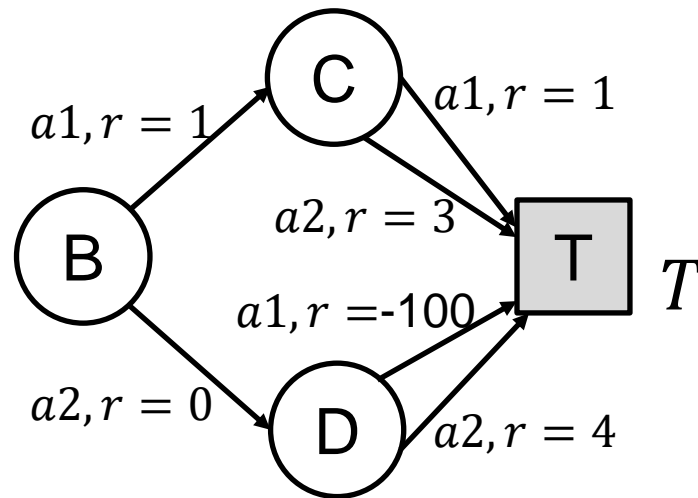
# MC, TD, Sarsa, QL w. $\alpha = 1$

- With learning rate  $\alpha = 1$ , each  $V(S_t)$  or  $Q(S_t, A_t)$  is completely overwritten in each update
  - The extreme case of “more recent visits are given more weight”
- update equations simplify to:
  - MC (every-visit):  $V(S_t) \leftarrow V(S_t) + \alpha(G(S_t) - V(S_t)) = G(S_t)$
  - TD:  $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)) = R_{t+1} + \gamma V(S_{t+1})$
  - Sarsa:  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)) = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$
  - QL:  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)) = R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')$

# Two-Branch Example

# Two-Branch Example

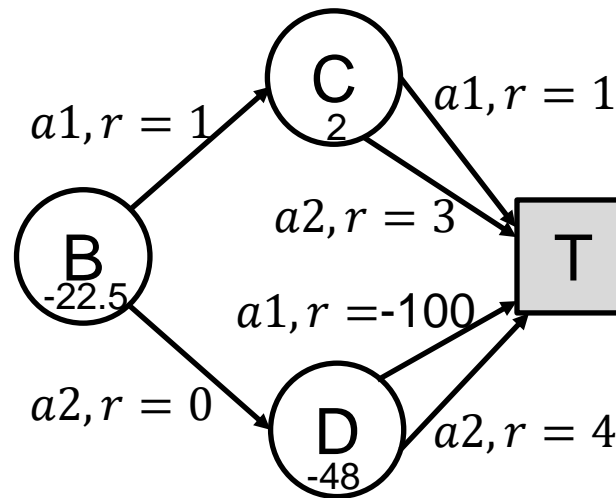
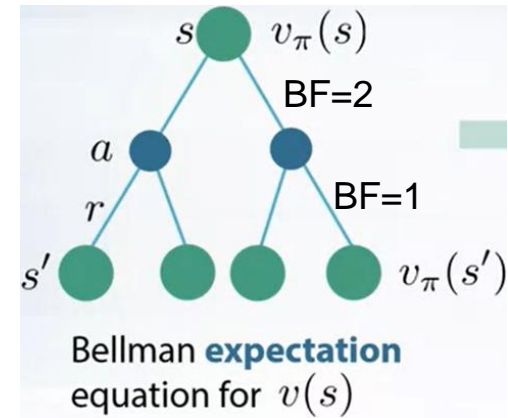
- An episodic MDP w. deterministic env, 3 states  $\{B, C, D\}$  and 2 actions  $\{1, 2\}$  at each state. Discount factor  $\gamma = 1$ , learning rate  $\alpha = 1$ . The initial state of each episode is  $B$ .



# Policy Iteration

# 1.1 Policy Evaluation of Random Policy

- Bellman Exp Equation:  $v_{\pi}(s) = \sum_a \pi(a|s) q_{\pi}(s, a); q_{\pi}(s, a) = R_s^a + \gamma v_{\pi}(s')$
- $v_{\pi}(B) = .5[q_{\pi}(B, a1) + q_{\pi}(B, a2)] = .5[1 + v_{\pi}(C) + v_{\pi}(D)]$ 
  - $q_{\pi}(B, a1) = 1 + v_{\pi}(C), q_{\pi}(B, a2) = 0 + v_{\pi}(D)$
- $v_{\pi}(C) = .5[q_{\pi}(C, a1) + q_{\pi}(C, a2)] = 2$ 
  - $q_{\pi}(C, a1) = 1, q_{\pi}(C, a2) = 3$
- $v_{\pi}(D) = .5[q_{\pi}(D, a1) + q_{\pi}(D, a2)] = -48$ 
  - $q_{\pi}(D, a1) = -100, q_{\pi}(D, a2) = 4$
- Solution:  $v_{\pi}(B) = -22.5, v_{\pi}(C) = 2, v_{\pi}(D) = -48$  (using in-place updates)

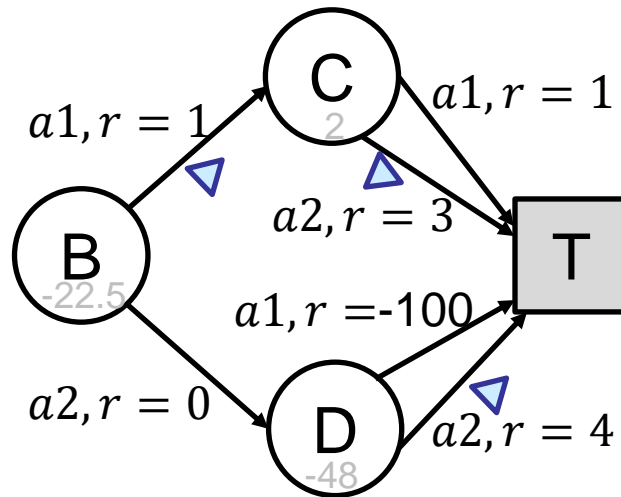


	$V_{\pi}(B)$	$V_{\pi}(C)$	$V_{\pi}(D)$
Iter1	-22.5	3	-48
Iter2	4	3	4
Iter3	4	3	4



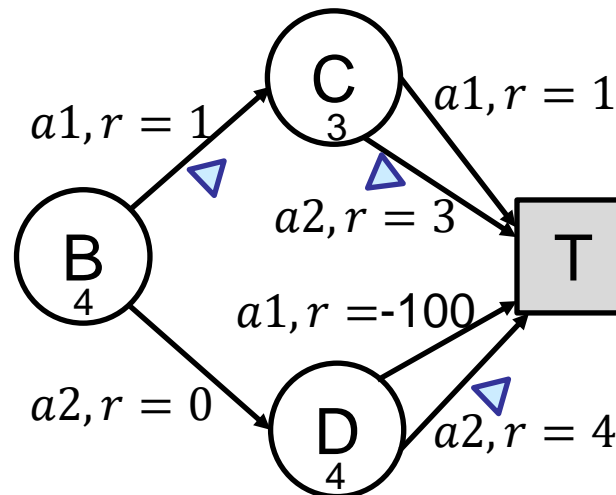
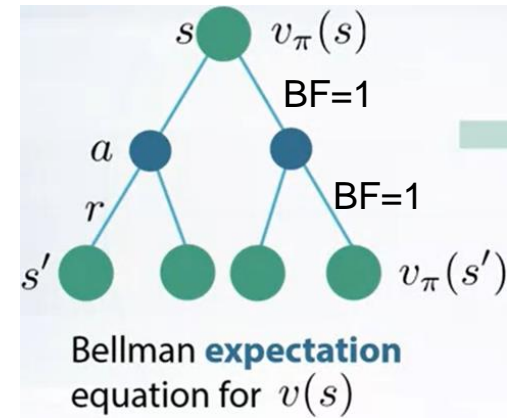
# 1.2 Policy Improvement

- Plug in values from PE to get new policy
- $\pi'(B) = \operatorname{argmax}_a (q_\pi(B, a1), q_\pi(B, a2)) = a1$ 
  - $q_\pi(B, a1) = 1 + v_\pi(C) = 3, q_\pi(B, a2) = 0 + v_\pi(D) = -22.5$
- $\pi'(C) = \operatorname{argmax}_a (q_\pi(C, a1), q_\pi(C, a2)) = a2$ 
  - $q_\pi(C, a1) = 1, q_\pi(C, a2) = 3$
- $\pi'(D) = \operatorname{argmax}_a (q_\pi(D, a1), q_\pi(D, a2)) = a2$ 
  - $q_\pi(D, a1) = -100, q_\pi(D, a2) = 4$



# 2.1 Policy Evaluation of Det Policy

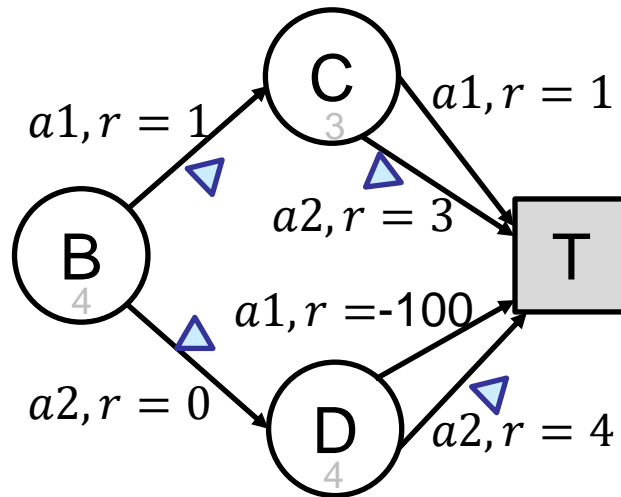
- Bellman Exp Equation:  $v_{\pi}(s) = \sum_a \pi(a|s)q_{\pi}(s, a); q_{\pi}(s, a) = R_s^a + \gamma v_{\pi}(s')$
- $v_{\pi}(B) = q_{\pi}(B, a1) = 1 + v_{\pi}(C)$ 
  - $q_{\pi}(B, a1) = 1 + v_{\pi}(C)$
- $v_{\pi}(C) = q_{\pi}(C, a2) = 3$ 
  - $q_{\pi}(C, a2) = 3$
- $v_{\pi}(D) = q_{\pi}(D, a2) = 4$ 
  - $q_{\pi}(D, a2) = 4$
- Solution:  $v_{\pi}(B) = 4, v_{\pi}(C) = 3, v_{\pi}(D) = 4$  (using in-place updates)



	$V_{\pi}(B)$	$V_{\pi}(C)$	$V_{\pi}(D)$
Iter1	-22.5	3	-48
Iter2	4	3	4
Iter3	4	3	4

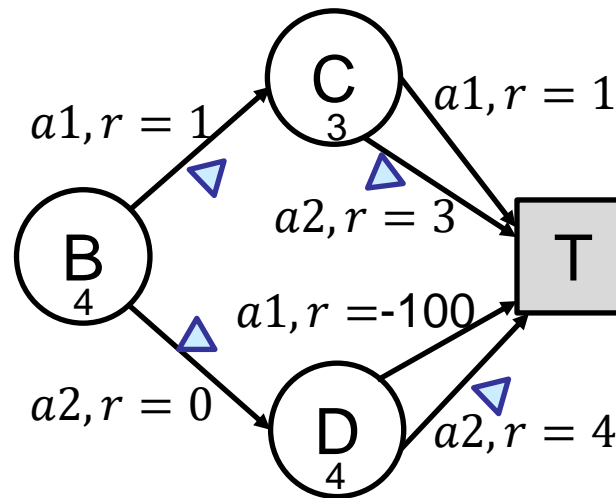
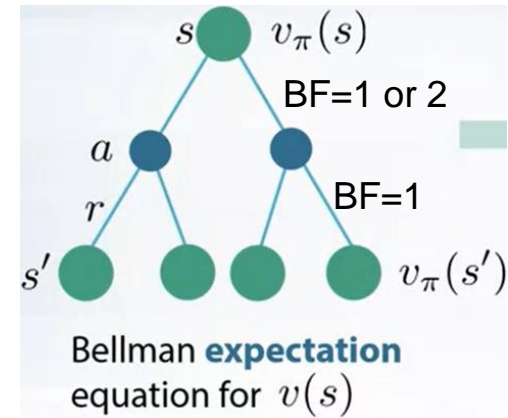
# 2.2 Policy Improvement

- Plug in values from PE to get new policy
- $\pi'(B) = \operatorname{argmax}_a (q_\pi(B, a1), q_\pi(B, a2)) = a1 \text{ or } a2$ 
  - $q_\pi(B, a1) = 1 + v_\pi(C) = 4, q_\pi(B, a2) = 0 + v_\pi(D) = 4$
- $\pi'(C) = \operatorname{argmax}_a (q_\pi(C, a1), q_\pi(C, a2)) = a2$ 
  - $q_\pi(C, a1) = 1, q_\pi(C, a2) = 3$
- $\pi'(D) = \operatorname{argmax}_a (q_\pi(D, a1), q_\pi(D, a2)) = a2$ 
  - $q_\pi(D, a1) = -100, q_\pi(D, a2) = 4$



# 3.1 Policy Evaluation

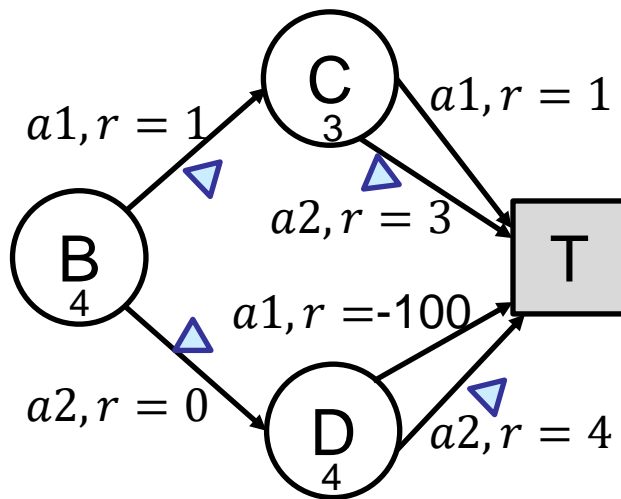
- Bellman Exp Equation:  $v_{\pi}(s) = \sum_a \pi(a|s) q_{\pi}(s, a)$ ;  $q_{\pi}(s, a) = R_s^a + \gamma v_{\pi}(s')$
- $v_{\pi}(B) = .5[q_{\pi}(B, a1) + q_{\pi}(B, a2)] = .5[1 + v_{\pi}(C) + v_{\pi}(D)]$ 
  - $q_{\pi}(B, a1) = 1 + v_{\pi}(C), q_{\pi}(B, a2) = 0 + v_{\pi}(D)$
- $v_{\pi}(C) = q_{\pi}(C, a2) = 3$ 
  - $q_{\pi}(C, a2) = 3$
- $v_{\pi}(D) = q_{\pi}(D, a2) = 4$ 
  - $q_{\pi}(D, a2) = 4$
- Solution:  $v_{\pi}(B) = 4, v_{\pi}(C) = 3, v_{\pi}(D) = 4$  (using in-place updates)



	$V_{\pi}(B)$	$V_{\pi}(C)$	$V_{\pi}(D)$
Iter1	-22.5	3	-48
Iter2	4	3	4
Iter3	4	3	4

## 3.2 Policy Improvement

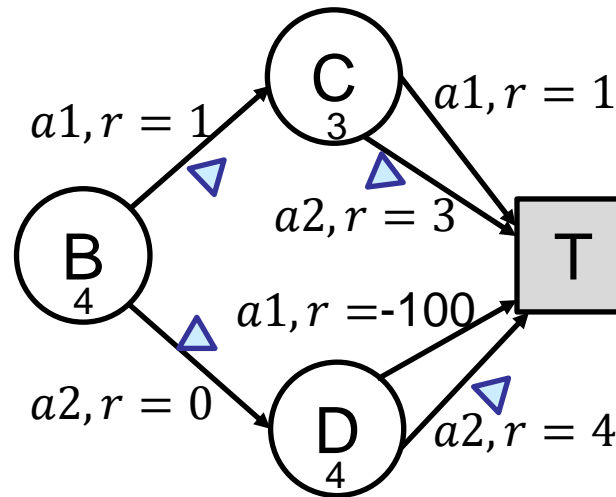
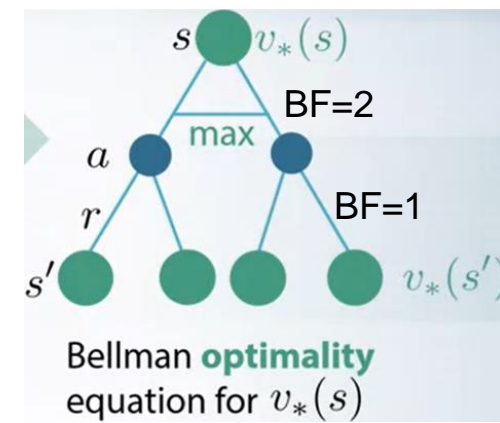
- Plug in values from PE to get new policy
- $\pi'(B) = \operatorname{argmax}_a (q_\pi(B, a1), q_\pi(B, a2)) = a1 \text{ or } a2$ 
  - $q_\pi(B, a1) = 1 + v_\pi(C) = 4, q_\pi(B, a2) = 0 + v_\pi(D) = 4$
- $\pi'(C) = \operatorname{argmax}_a (q_\pi(C, a1), q_\pi(C, a2)) = a2$ 
  - $q_\pi(C, a1) = 1, q_\pi(C, a2) = 3$
- $\pi'(D) = \operatorname{argmax}_a (q_\pi(D, a1), q_\pi(D, a2)) = a2$ 
  - $q_\pi(D, a1) = -100, q_\pi(D, a2) = 4$
- Policy has converged



# Value Iteration

# Value Iteration

- Bellman Opt Equation:  $v_*(s) = \max_a q_*(s, a); q_*(s, a) = R_s^a + \gamma v_*(s')$
- $v_*(B) = \max_a [q_*(B, a1), q_*(B, a2)] = \max[1 + v_*(C), v_*(D)]$ 
  - $q_*(B, a1) = 1 + v_*(C), q_*(B, a2) = 0 + v_*(D)$
- $v_*(C) = \max_a [q_*(C, a1), q_*(C, a2)] = q_*(C, a2) = 3$ 
  - $q_*(C, a1) = 1, q_*(C, a2) = 3$
- $v_*(D) = \max_a [q_*(D, a1), q_*(D, a2)] = 4$ 
  - $q_*(D, a1) = -100, q_*(D, a2) = 4$
- We use Value Iteration to solve it. Table shows the iteration process until convergence (not using in-place updates for clarity). Solution:  $v_*(B) = 4, v_*(C) = 3, v_*(D) = 4$
- Optimal policy:  $\pi_*(B) = \operatorname{argmax}_a q_*(B, a) = a1 \text{ or } a2; \pi_*(C) = \operatorname{argmax}_a q_*(C, a) = a2; \pi_*(D) = \operatorname{argmax}_a q_*(D, a) = a2$



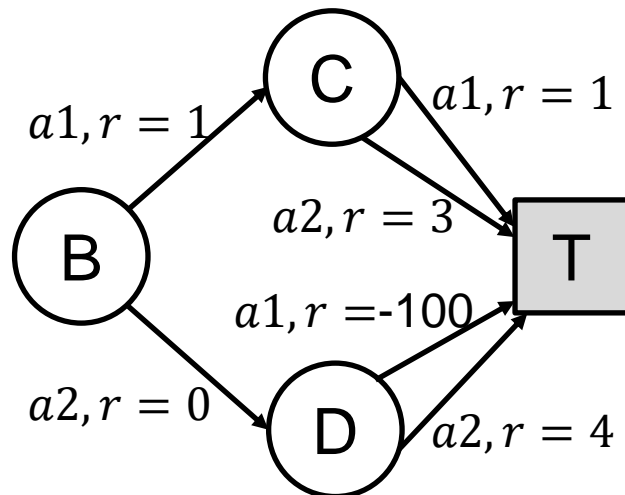
	$V_*(B)$	$V_*(C)$	$V_*(D)$
Init	0	0	0
Iter1	0	3	4
Iter2	4	3	4
Iter3	4	3	4

MC



MC, Episodes  $3 \times (B, a2, 0, D, a1, -100, T)$

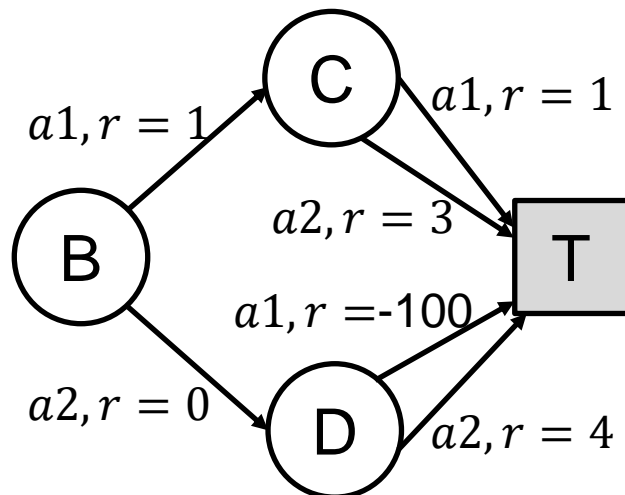
- MC update equation:  $V(S_t) \leftarrow G_t$
- EP1:
- $G(D) = -100 + V(T) = -100, G(B) = 0 + G(D) = -100$
- $V(B) = G(B) = -100, V(D) = G(D) = -100$
- EP2: same as EP1
- EP3: same as EP1



	$V(B)$	$V(D)$
Init	0	0
EP1	-100	-100
EP2	-100	-100
EP3	-100	-100

MC, Episodes  $3 \times (B, a2, 0, D, a2, 4, T)$

- MC update equation:  $V(S_t) \leftarrow G_t$
- EP1:
- $G(D) = 4 + V(T) = 4, G(B) = 0 + G(D) = 4,$
- $V(B) = G(B) = 4, V(D) = G(D) = 4$
- EP2: same as EP1
- EP3: same as EP1

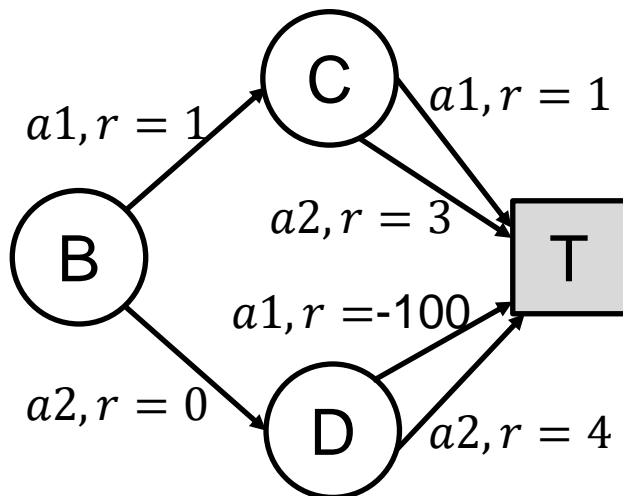


	$V(B)$	$V(D)$
Init	0	0
EP1	4	4
EP2	4	4
EP3	4	4

TD

# TD, Episodes $3 \times (B, a2, 0, D, a1, -100, T)$

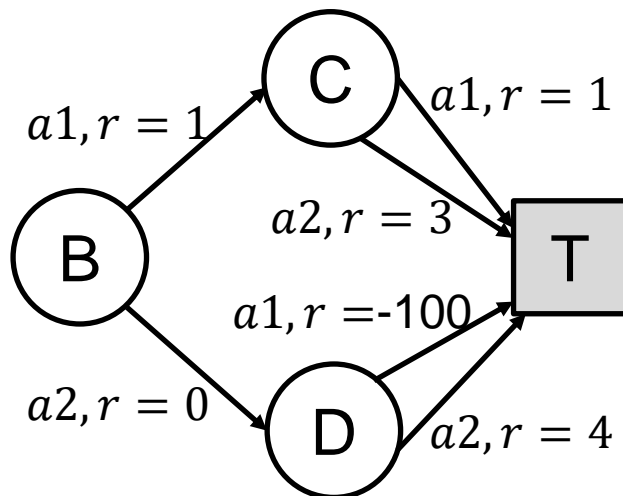
- TD update equation:  $V(S_t) \leftarrow R_{t+1} + \gamma V(S_{t+1})$
- EP1:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 + 0 = 0$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = -100 - 0 = -100$
- EP2:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 - 100 = -100$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = -100 - 0 = -100$
- EP3:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 - 100 = -100$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = -100 - 0 = -100$



	$V(B)$	$V(D)$
Init	0	0
EP1	0	-100
EP2	-100	-100
EP3	-100	-100

# TD, Episodes $3 \times (B, a2, 0, D, a2, 4, T)$

- TD update equation:  $V(S_t) \leftarrow R_{t+1} + \gamma V(S_{t+1})$
- EP1:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 + 0 = 0$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = 4 + 0 = 4$
- EP2:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 + 4 = 4$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = 4 + 0 = 4$
- EP3:
  - $V(B) \leftarrow R_{t+1} + \gamma V(D) = 0 + 4 = 4$
  - $V(D) \leftarrow R_{t+1} + \gamma V(T) = 4 + 0 = 4$

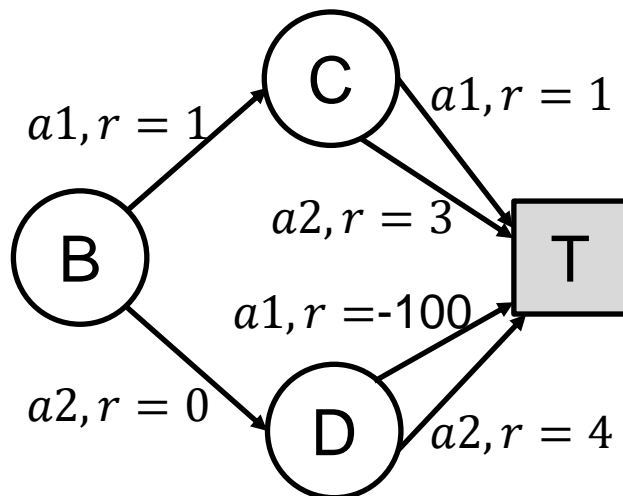


	$V(B)$	$V(D)$
Init	0	0
EP1	0	4
EP2	4	4
EP3	4	4

# Sarsa

# Sarsa, Episodes $3 \times (B, a2, 0, D, a1, -100, T)$

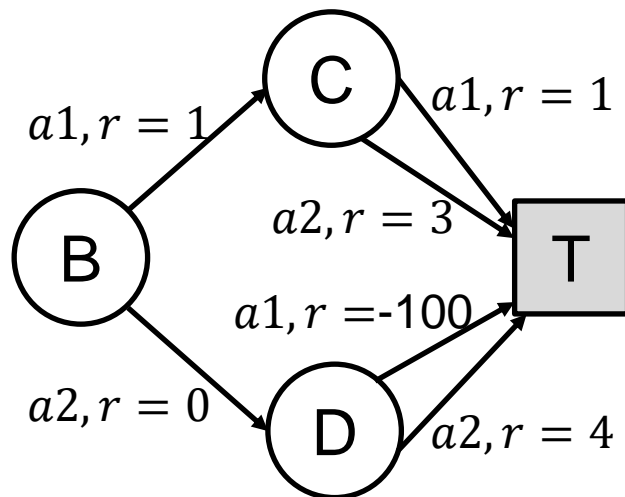
- Sarsa update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$
- EP1:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a1) = 0 - 0 = 0$
- $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- EP2:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a1) = 0 - 100 = -100$
- $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- EP3:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a1) = 0 - 100 = -100$
- $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$



	$Q(B, a1)$	$Q(B, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0
EP1	0	0	-100	0
EP2	0	-100	-100	0
EP3	0	-100	-100	0

# Sarsa, Episodes $3 \times (B, a2, 0, D, a2, 4, T)$

- Sarsa update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$
- EP1:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a2) = 0 - 0 = 0$
  - $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- EP2:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a2) = 0 + 4 = 4$
  - $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- EP3:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a2) = 0 + 4 = 4$
  - $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$

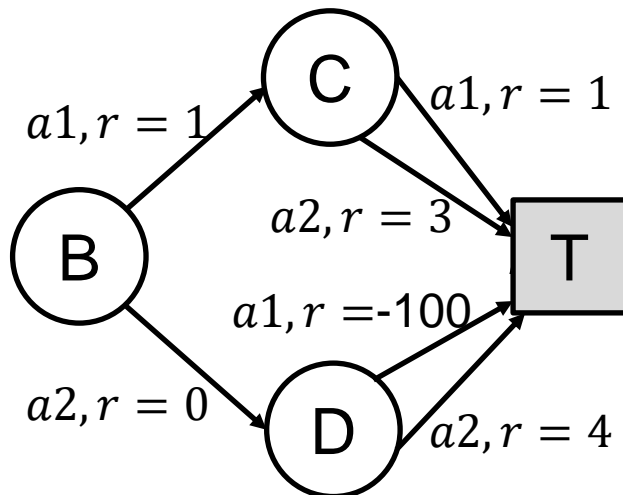


	$Q(B, a1)$	$Q(B, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0
EP1	0	0	0	4
EP2	0	4	0	4
EP3	0	4	0	4



# QL, Episodes $3 \times (B, a2, 0, D, a1, -100, T)$

- QL update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')$
- EP1:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(0, 0) = 0$
  - $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- EP2:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(-100, 0) = 0$
  - $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- EP3:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(-100, 0) = 0$
  - $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$

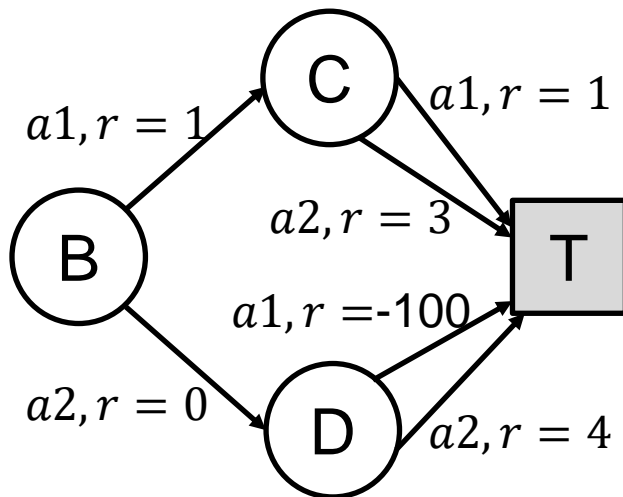


	$Q(B, a1)$	$Q(B, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0
EP1	0	0	-100	0
EP2	0	0	-100	0
EP3	0	0	-100	0

# Q Learning

# QL, Episodes $3 \times (B, 2, 0, D, 2, 4, T)$

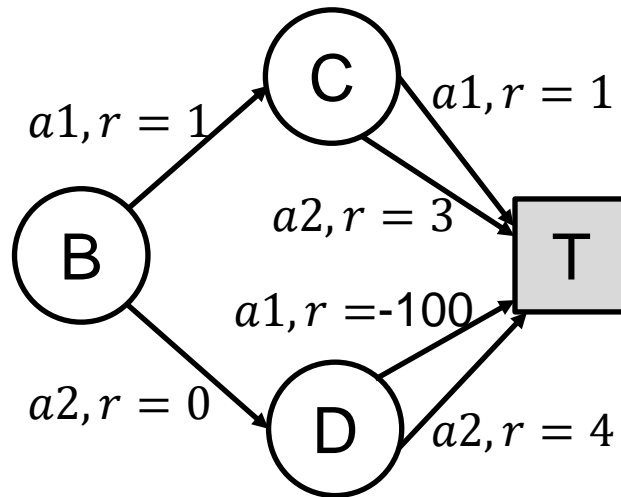
- QL update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')$
- EP1:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(0, 0) = 0$
- $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- EP2:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + 4 = 4$
- $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- EP3:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + 4 = 4$
- $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$



	$Q(B, a1)$	$Q(B, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0
EP1	0	0	0	4
EP2	0	4	0	4
EP3	0	4	0	4

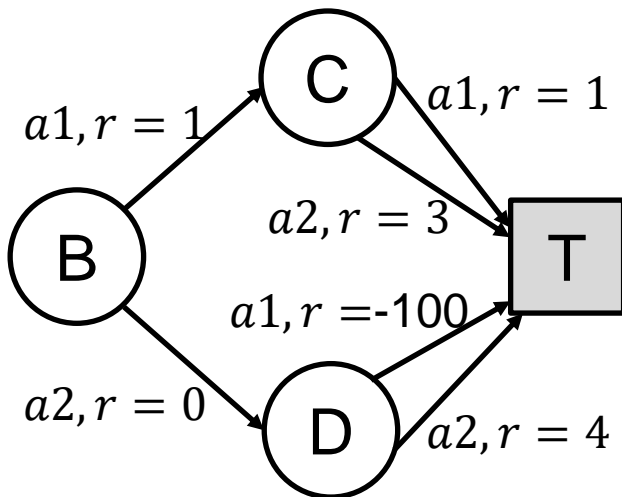
# Comparisons

- MC and TD:
  - Transition  $(D, a1, -100, T)$  drives  $V(D) \rightarrow -100$ ;  $V(D)$  drives  $V(B) \rightarrow -100$ .
  - Transition  $(D, a2, 4, T)$  drives  $V(D) \rightarrow 4$ ;  $V(D)$  drives  $V(B) \rightarrow 4$ .
  - Final values of  $V(B), V(D)$  depend on relative execution frequencies of the 2 transitions (e.g.,  $\epsilon$ -greedy).
- Sarsa:
  - Transition  $(D, a1, -100, T)$  drives  $Q(D, a1) \rightarrow -100$ ;  $Q(D, a1)$  drives  $Q(B, a2) \rightarrow -100$ .
  - Transition  $(D, a2, 4, T)$  drives  $Q(D, a2) \rightarrow 4$ ;  $Q(D, a2)$  drives  $Q(B, a2) \rightarrow 4$ .
  - Final value of  $Q(B, 2)$  depends on relative execution frequencies of the 2 transitions (e.g.,  $\epsilon$ -greedy).
- QL:
  - Transition  $(D, a1, -100, T)$  drives  $Q(D, a1) \rightarrow -100$ ;  $Q(D, a1)$  does not affect  $Q(B, a2)$  since
  - $\max_a Q(D, a) = \max(Q(D, a1), Q(D, a2)) = 0$ . (assuming  $Q(D, a2)$  is initialized to 0 and it never updated)
  - Transition  $(D, a2, 4, T)$  drives  $Q(D, a2) \rightarrow 4$ , which in turn drives  $Q(B, a2) \rightarrow 4$ .
- We perform policy evaluation for a given set of episodes, not control. If we consider control, e.g., Sarsa or QL uses  $\epsilon$ -greedy policy with small  $\epsilon$ , then the agent will likely avoid action  $a1$  in state  $D$  after taking it for the 1<sup>st</sup> time.



# Sarsa w. $\epsilon$ -greedy

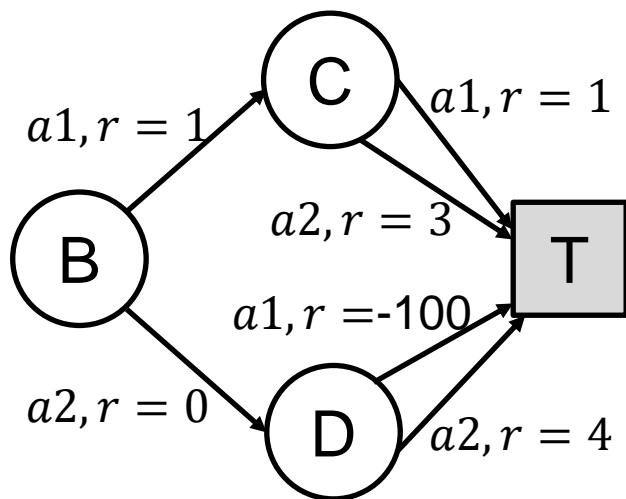
- Suppose EP1 is  $(B, a2, 0, D, a1, -100, T)$ . After EP1:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a1) = 0 - 0 = 0$ ,  $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- Suppose EP2 starts with  $(B, a2, 0, D)$ , then in state D, the agent is likely to select action  $\operatorname{argmax}_a \{Q(D, a1) = -100, Q(D, a2) = 0\} = a2$  based on  $\epsilon$ -greedy, so the episode is  $(B, a2, 0, D, a2, 4, T)$ . After EP2:
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a1) = 0 - 100 = -100$ ,  $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- In EP3, in initial state B, the agent is likely to select action  $\operatorname{argmax}_a \{Q(B, a1) = 0, Q(B, a2) = -100\} = a1$ . Suppose the episode is  $(B, a1, 1, C, a1, 1, T)$
- $Q(B, a1) \leftarrow R_{t+1} + \gamma Q(C, a1) = 1 + 0 = 1$ ,  $Q(C, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = 1 + 0 = 1$
- In EP4, in initial state B, the agent is likely to select action  $\operatorname{argmax}_a \{Q(B, a1) = 1, Q(B, a2) = -100\} = a1$ . in state C, the agent is likely to select action  $\operatorname{argmax}_a \{Q(C, a1) = 1, Q(C, a2) = 0\} = a1$ . Suppose the episode is again  $(B, a1, 1, C, a1, 1, T)$
- $Q(B, a1) \leftarrow R_{t+1} + \gamma Q(C, a1) = 1 + 1 = 2$ ,  $Q(C, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = 1 + 0 = 1$
- if the agent always follows the greedy policy, it will always follow the trajectory  $(B, a1, 1, C, a1, 1, T)$  and never learn anything new, e.g., it will never experience the trajectories  $(B, a1, 1, C, a2, 3, T)$ ,  $(B, a2, 0, D, a2, 4, T)$ . It got scared when  $Q(B, a2)$  was updated to  $-100$  after EP2 and never wanted to take action a2 in state B, but if it were more adventurous and tried it, it will likely experience EP5  $(B, a2, 0, D, a2, 4, T)$ :
- $Q(B, a2) \leftarrow R_{t+1} + \gamma Q(D, a2) = 0 + 4 = 4$ ,  $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- Now you can see the importance of exploration by selecting the non-greedy action occasionally.



	$Q(B, a1)$	$Q(B, a2)$	$Q(C, a1)$	$Q(C, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0	0	0
EP1	0	0	0	0	-100	0
EP2	0	-100	0	0	-100	4
EP3	1	-100	1	0	-100	4
EP4	2	-100	1	0	-100	4
EP5	2	4	1	0	-100	4

# QL w. $\epsilon$ -greedy

- Suppose EP1 is  $(B, a2, 0, D, a1, -100, T)$ . After EP1:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(0, 0) = 0$ ,  $Q(D, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = -100 + 0 = -100$
- Suppose EP2 starts with  $(B, a2, 0, D)$ , then in state D, the agent is likely to select action  $\text{argmax}_a \{Q(D, a1) = -100, Q(D, a2) = 0\} = a2$  based on  $\epsilon$ -greedy, so the episode is  $(B, a2, 0, D, a2, 4, T)$ . After EP2:
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(-100, 0) = 0$ ,  $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$
- In EP3, in initial state B, the agent is equally likely to select action  $a1$  and  $a2$  since  $Q(B, a1) = Q(B, a2) = 0$ . Suppose the episode is  $(B, a1, 1, C, a1, 1, T)$ 
  - $Q(B, a1) \leftarrow R_{t+1} + \gamma \max_a Q(C, a) = 1 + \max(0, 0) = 1$ ,  $Q(C, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = 1 + 0 = 1$
- In EP4, in initial state B, the agent is likely to select action  $\text{argmax}_a \{Q(B, a1) = 1, Q(B, a2) = 0\} = a1$ . in state D, the agent is likely to select action  $\text{argmax}_a \{Q(D, a1) = 1, Q(D, a2) = 0\} = a1$ . Suppose the episode is  $(B, a1, 1, C, a1, 1, T)$ 
  - $Q(B, a1) \leftarrow R_{t+1} + \gamma \max_a Q(C, a) = 1 + \max(1, 0) = 2$ ,  $Q(C, a1) \leftarrow R_{t+1} + \gamma Q(T, -) = 1 + 0 = 1$
- The difference from Sarsa lies in  $Q(B, a2)$ , which stays at 0 until the agent experienced EP5. So it got less scared than Sarsa (where  $Q(B, a2)$  was updated to  $-100$  after EP2), so QL agent is more likely to explore unseen states. Optimistic initialization of Q values encourages exploration, but may cause slow convergence.
- Suppose EP5 is  $(B, a2, 0, D, a2, 4, T)$ :
  - $Q(B, a2) \leftarrow R_{t+1} + \gamma \max_a Q(D, a) = 0 + \max(-100, 4) = 4$ ,  $Q(D, a2) \leftarrow R_{t+1} + \gamma Q(T, -) = 4 + 0 = 4$

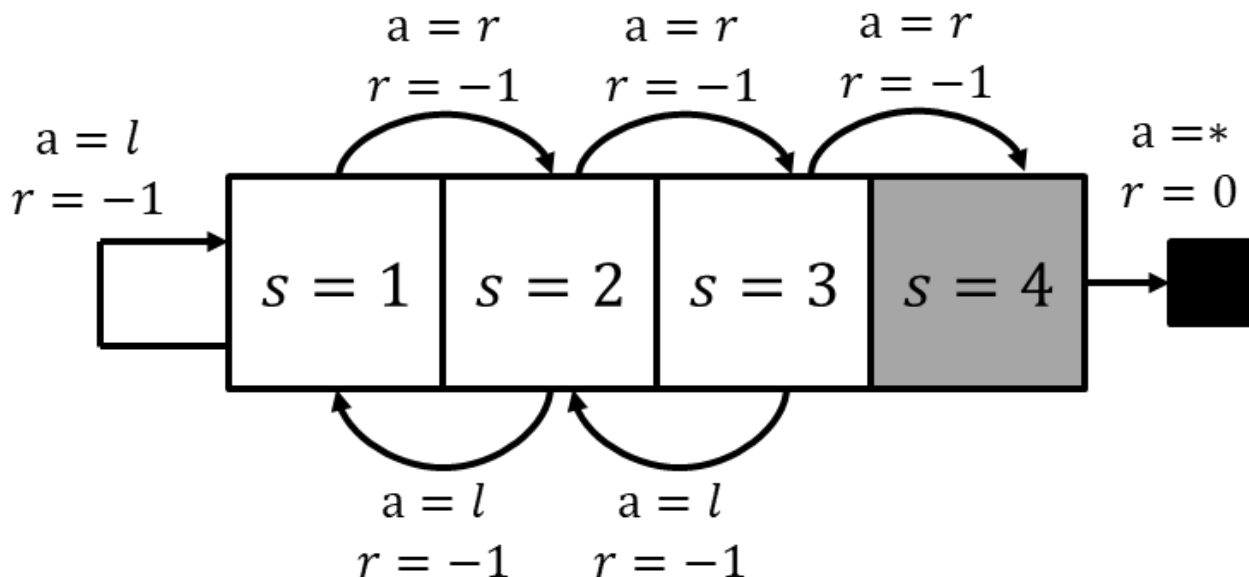


	$Q(B, a1)$	$Q(B, a2)$	$Q(C, a1)$	$Q(C, a2)$	$Q(D, a1)$	$Q(D, a2)$
Init	0	0	0	0	0	0
EP1	0	0	0	0	-100	0
EP2	0	0	0	0	-100	4
EP3	1	0	1	0	-100	4
EP4	2	0	1	0	-100	4
EP5	2	4	1	0	-100	4

# Linear Chain Example

# Linear Chain Example

- Consider the following MDP. Environment is deterministic. In each state, there are two possible actions  $a \in \{l, r\}$ , where  $l$  corresponds to moving left, and  $r$  corresponds to moving right. Each movement incurs a reward of  $r = -1$ . State  $s=4$  is the goal state: taking any action from  $s=4$  results in reward of  $r=0$  and ends the episode by going into the terminal state, hence  $V(4) \equiv 0, Q(4, a) \equiv 0$  for any action  $a$ . (Alternatively, we can view state 4 as the terminal state itself.) Assume discount factor  $\gamma = 1$ . All value functions are initialized to 0.
- A. Use Policy Iteration, Value Iteration to derive the optimal policy.

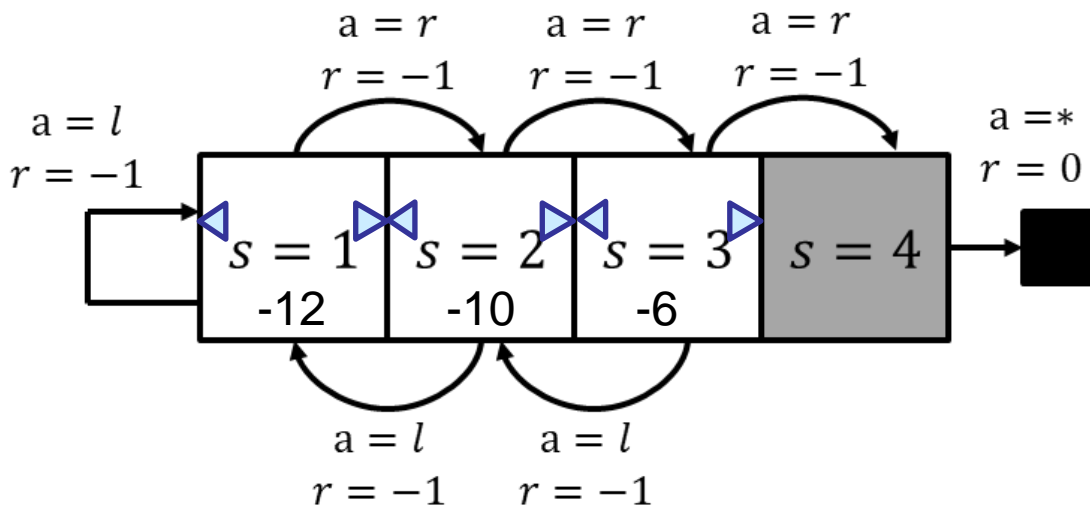
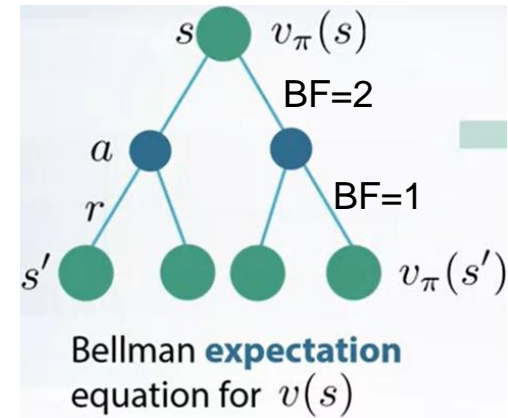




# Policy Iteration

# 1.1 Policy Evaluation of Random Policy

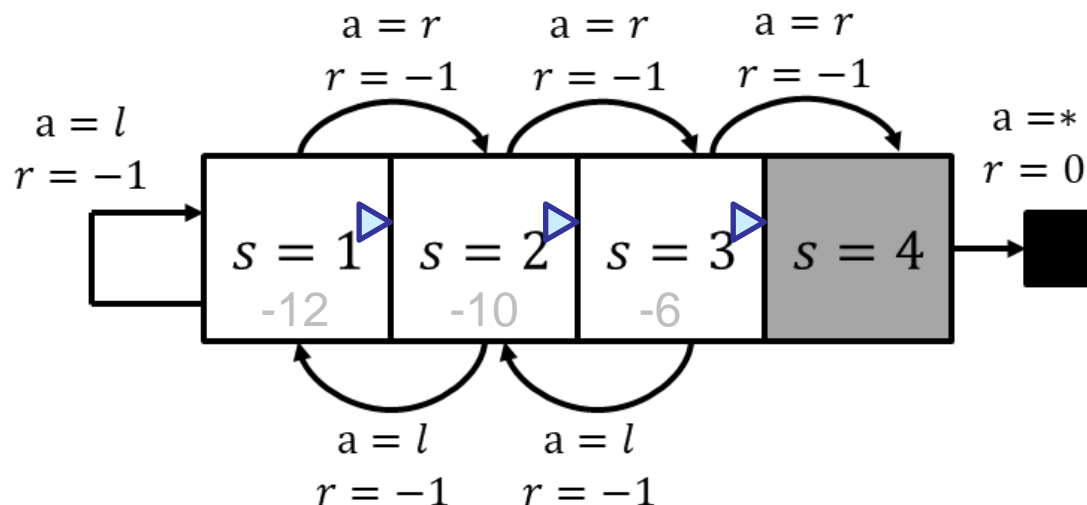
- Bellman Exp Equation:  $v_{\pi}(s) = \sum_a \pi(a|s) q_{\pi}(s, a)$ ;  $q_{\pi}(s, a) = R_s^a + \gamma v_{\pi}(s')$
- $v_{\pi}(1) = .5[q_{\pi}(1, l) + q_{\pi}(1, r)] = -1 + .5[v_{\pi}(1) + v_{\pi}(2)]$ 
  - $q_{\pi}(1, l) = -1 + v_{\pi}(1), q_{\pi}(1, r) = -1 + v_{\pi}(2)$
- $v_{\pi}(2) = .5[q_{\pi}(2, l) + q_{\pi}(2, r)] = -1 + .5[v_{\pi}(1) + v_{\pi}(3)]$ 
  - $q_{\pi}(2, l) = -1 + v_{\pi}(1), q_{\pi}(2, r) = -1 + v_{\pi}(3)$
- $v_{\pi}(3) = .5[q_{\pi}(3, l) + q_{\pi}(3, r)] = -1 + .5 v_{\pi}(2)$ 
  - $q_{\pi}(3, l) = -1 + v_{\pi}(2), q_{\pi}(3, r) = -1 + v_{\pi}(4) = -1$
- Solution:  $v_{\pi}(1) = -12, v_{\pi}(2) = -10, v_{\pi}(3) = -6$



	$V_{\pi}(1)$	$V_{\pi}(2)$	$V_{\pi}(3)$
Iter1	-12	-10	-6
Iter2	4	3	4

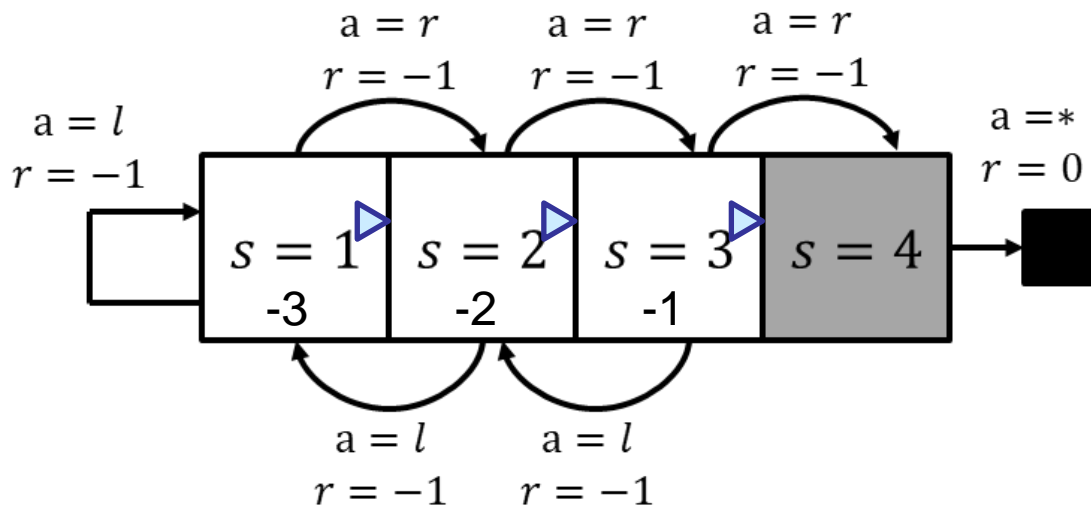
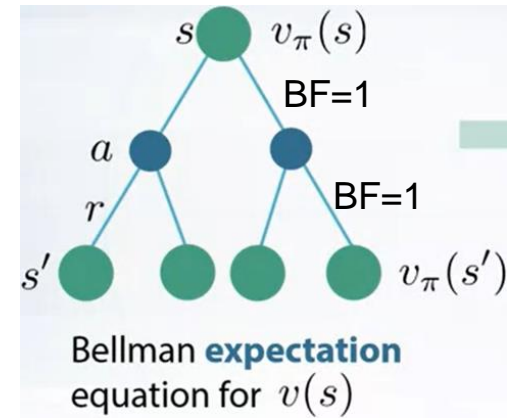
# 1.2 Policy Improvement

- Plug in values from PE to get new policy
- $\pi'(1) = \operatorname{argmax}_a (q_\pi(1, l), q_\pi(1, r)) = r$ 
  - $q_\pi(1, l) = -1 + v_\pi(1) = -13$ ,  $q_\pi(1, r) = -1 + v_\pi(2) = -11$ ,
- $\pi'(2) = \operatorname{argmax}_a (q_\pi(2, l), q_\pi(2, r)) = r$ 
  - $q_\pi(2, l) = -1 + v_\pi(1) = -13$ ,  $q_\pi(2, r) = -1 + v_\pi(3) = -7$
- $\pi'(3) = \operatorname{argmax}_a (q_\pi(3, l), q_\pi(3, r)) = r$ 
  - $q_\pi(3, l) = -1 + v_\pi(2) = -11$ ,  $q_\pi(3, r) = -1$



# 2.1 Policy Evaluation of Det Policy

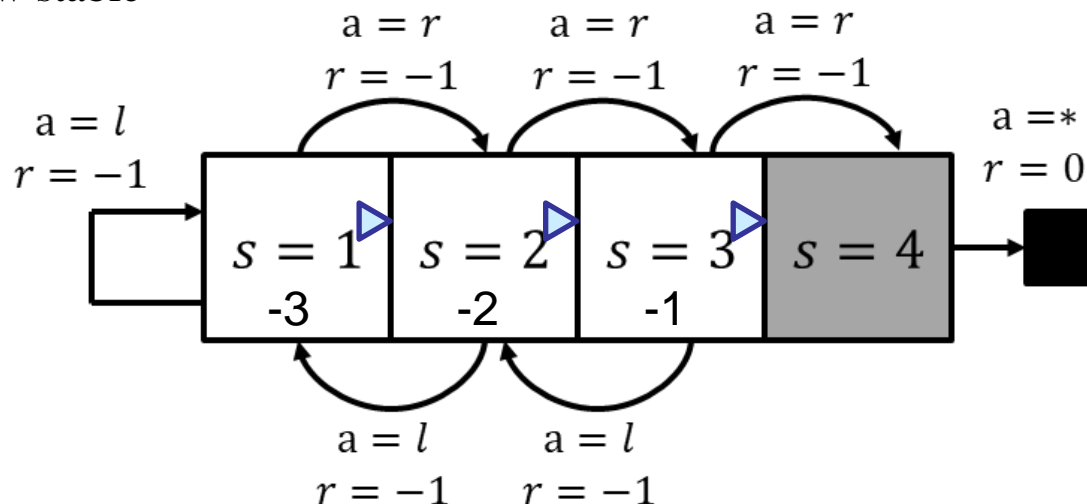
- $v_{\pi}(1) = 1.0q_{\pi}(1, r) = -1 + v_{\pi}(2)$ 
  - $q_{\pi}(1, r) = -1 + v_{\pi}(2)$
- $v_{\pi}(2) = 1.0q_{\pi}(2, r) = -1 + v_{\pi}(3)$ 
  - $q_{\pi}(2, r) = -1 + v_{\pi}(3)$
- $v_{\pi}(3) = 1.0q_{\pi}(3, r) = -1$ 
  - $q_{\pi}(3, r) = -1$
- Solution:  $v_{\pi}(1) = -3, v_{\pi}(2) = -2, v_{\pi}(3) = -1$



	$V_{\pi}(1)$	$V_{\pi}(2)$	$V_{\pi}(3)$
Iter1	-12	-10	-6
Iter2	-3	-2	-1

## 2.2 Policy Improvement

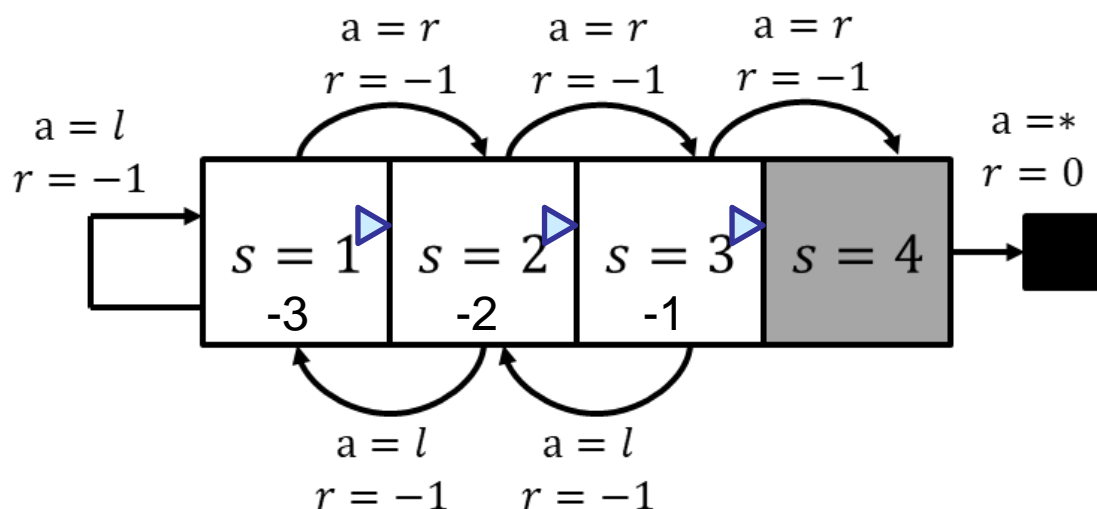
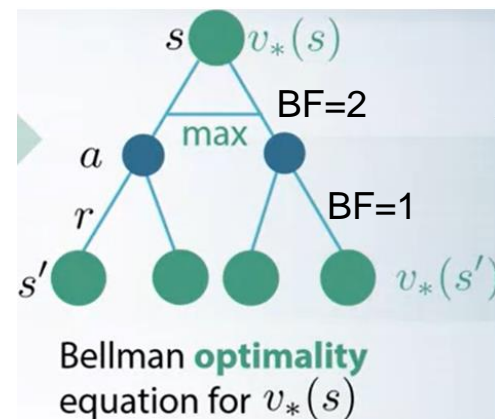
- Plug in values from PE to get new policy
- $\pi'(1) = \operatorname{argmax}_a(q_\pi(1, l), q_\pi(1, r)) = r$ 
  - $q_\pi(1, l) = -1 - 3 = -4, q_\pi(1, r) = -1 - 2 = -3$
- $\pi'(2) = \operatorname{argmax}_a(q_\pi(2, l), q_\pi(2, r)) = r$ 
  - $q_\pi(2, l) = -1 - 3 = -4, q_\pi(2, r) = -1 - 1 = -2$
- $\pi'(3) = \operatorname{argmax}_a(q_\pi(3, l), q_\pi(3, r)) = r$ 
  - $q_\pi(3, l) = -1 - 2 = -3, q_\pi(3, r) = -1$
- Policy is now stable



# Value Iteration

# Value Iteration

- Bellman Opt Equation:  $v_*(s) = \max_a q_*(s, a); q_*(s, a) = R_s^a + \gamma v_*(s')$
- $v_*(1) = \max_a [q_*(1, l), q_*(1, r)] = \max_a [-1 + v_*(1), -1 + v_*(2)]$ 
  - $q_*(1, l) = -1 + v_*(1), q_*(1, r) = -1 + v_*(2)$
- $v_*(2) = \max_a [q_*(2, l), q_*(2, r)] = \max_a [-1 + v_*(1), -1 + v_*(3)]$ 
  - $q_*(2, l) = -1 + v_*(1), q_*(2, r) = -1 + v_*(3)$
- $v_*(3) = \max_a [q_*(3, l), q_*(3, r)] = \max_a [-1 + v_*(2), -1 + v_*(4)] = \max_a [-1 + v_*(2), -1]$ 
  - $q_*(3, l) = -1 + v_*(2), q_*(3, r) = -1 + v_*(4) = -1$
- We use Value Iteration to solve it. Table shows the iteration process until convergence (not using in-place updates for clarity). Solution:  $v_*(1) = -3, v_*(2) = -2, v_*(3) = -1$
- Optimal policy:  $\pi_*(1) = \operatorname{argmax}_a q_*(1, a) = r; \pi_*(2) = \operatorname{argmax}_a q_*(2, a) = r; \pi_*(3) = \operatorname{argmax}_a q_*(3, a) = r$



(The  $V_*(4)$  column is omitted since it is always 0)

	$V_*(1)$	$V_*(2)$	$V_*(3)$
Init	0	0	0
Iter1	-1	-1	-1
Iter2	-2	-2	-1
Iter3	-3	-2	-1
Iter4	-3	-2	-1

# MC, TD, Sarsa, QL (Simple)

- B. Assume learning rate  $\alpha = 0.5$ . Consider an episode in the form of  $(s,a,r)$  :

EP1:  $(3, l, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

- Derive the following:
  1. State value functions  $V(s)$  after MC learning.
  2. State value functions  $V(s)$  after TD learning.
  3. State-action value functions  $Q(s, a)$  after Sarsa, and the resulting policy.
  4. State-action value functions  $Q(s, a)$  after Q learning, and the resulting policy.



# MC EP1

- MC update equation:  $V(S_t) \leftarrow V(S_t) + \alpha(G(S_t) - V(S_t))$
- $V(4) \equiv 0$ . Initialize  $V(1) = V(2) = V(3) = 0$
- EP1:  $(3, l, -1), (2, r, -1), (3, r, -1), (4, r, 0)$
- MC (every-visit w. EP1  $3' \rightarrow 2 \rightarrow 3 \rightarrow 4$ ):

- Update  $G(s)$  backward:

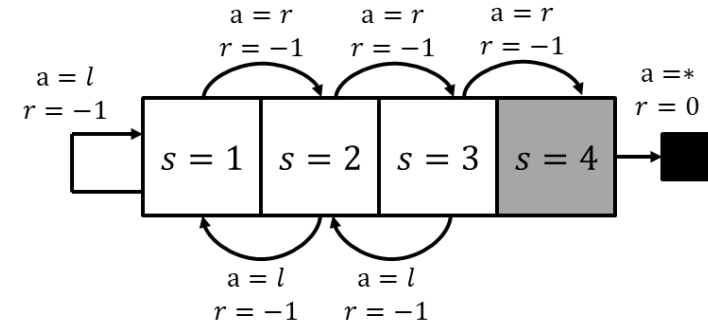
1.  $G(3) \leftarrow -1$
2.  $G(2) \leftarrow -1 + \gamma G(3) = -1 - 1 = -2$
3.  $G(3') \leftarrow -1 + \gamma G(2) = -1 - 2 = -3,$

- Update  $V(s)$  forward:

1.  $V(3) \leftarrow V(3) + \alpha(G(3') - V(3)) = 0 + .5(-3 - 0) = -1.5$
2.  $V(2) \leftarrow V(2) + \alpha(G(2) - V(2)) = 0 + .5(-2 - 0) = -1$
3.  $V(3) \leftarrow V(3) + \alpha(G(3) - V(3)) = -1.5 + .5(-1 + 1.5) = -1.25$

- $G(3') = -3$  is misleading: based on EP1  $3' \rightarrow 2 \rightarrow 3 \rightarrow 4$ , the agent needs 3 steps to get to the terminal state by moving left in the 1<sup>st</sup> visit to state 3, but in fact it only needs 1 step by moving right in the 2<sup>nd</sup> visit to state 3. That is why “more recent visits are given more weight”. In the extreme case, if learning rate  $\alpha = 1$ , then each  $V(S)$  is completely overwritten in each update, and we have a more correct estimate of  $V(3)$ :

1.  $V(3) \leftarrow V(3) + \alpha(G(3') - V(3)) = 0 + 1(-3 - 0) = -3$
2.  $V(2) \leftarrow V(2) + \alpha(G(2) - V(2)) = 0 + 1(-2 - 0) = -2$
3.  $V(3) \leftarrow V(3) + \alpha(G(3) - V(3)) = -3 + 1(-1 + 3) = -1$



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-1.25	-1	-1.5

- TD update equation:  $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$
- $V(4) \equiv 0$ . Initialize  $V(1) = V(2) = V(3) = 0$ ,
- EP1:  $(3, l, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

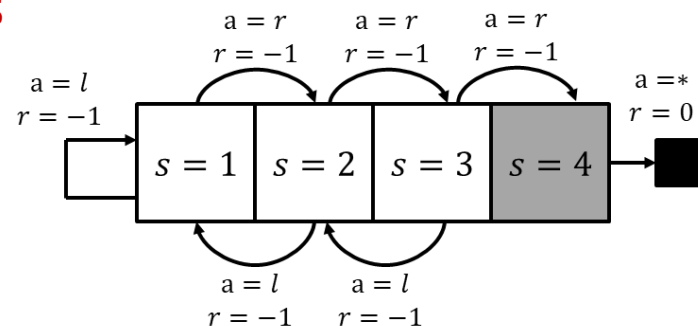
$$1. \quad V(3) \leftarrow V(3) + \alpha(R + \gamma V(2) - V(3)) = 0 + .5(-1 + 0 - 0) = -0.5$$

$$2. \quad V(2) \leftarrow V(2) + \alpha(R + \gamma V(3) - V(2)) = 0 + .5(-1 - .5 - 0) = -0.725$$

$$3. \quad V(3) \leftarrow V(3) + \alpha(R + \gamma V(4) - V(3)) = -.5 + .5(-1 + 0 + .5) = -0.75$$

- Arrows denote bootstrap dependencies, e.g.,  $V(3)$  bootstraps off  $V(2)$ ,  $V(2)$  bootstraps off  $V(3)$ ,  $V(3)$  bootstraps off  $V(4)$ . They also denote direction of information flow during learning, e.g.,  $V(4) \equiv 0$  is the external learning signal, and info flows  $V(4) \rightarrow V(3)$ .

# TD EP1



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	0	-0.725	-0.5

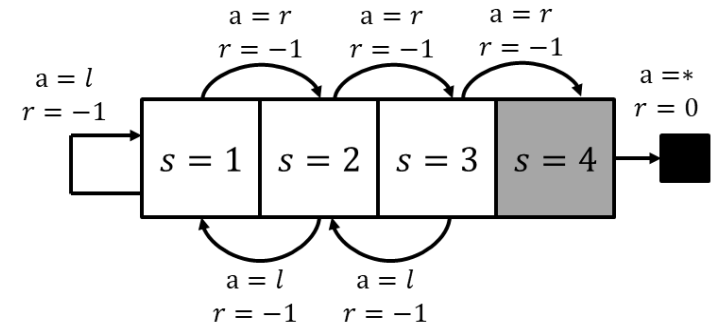
Arrows indicate bootstrap dependencies: from  $V(2)$  to  $V(3)$ , from  $V(3)$  to  $V(2)$ , and from  $V(4)$  (represented by -0.75) to  $V(3)$ .

- Sarsa update equation:  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$
- $Q(4, a) \equiv 0$ . Initialize  $Q(1, *) = Q(2, *) = Q(3, *) = 0$
- EP1:

$(3, l, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

- $Q(3, l) \leftarrow Q(3, l) + \alpha(R + \gamma Q(2, r) - Q(3, l)) = 0 + .5(-1 + 0 - 0) = -0.5$
- $Q(2, r) \leftarrow Q(2, r) + \alpha(R + \gamma Q(3, r) - Q(2, r)) = 0 + .5(-1 + 0 - 0) = -0.5$
- $Q(3, r) \leftarrow Q(3, r) + \alpha(R + \gamma Q(4, r) - Q(3, r)) = 0 + .5(-1 + 0 - 0) = -0.5$

# Sarsa EP1



Sarsa	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	0	0	-0.5	-0.5	-0.5

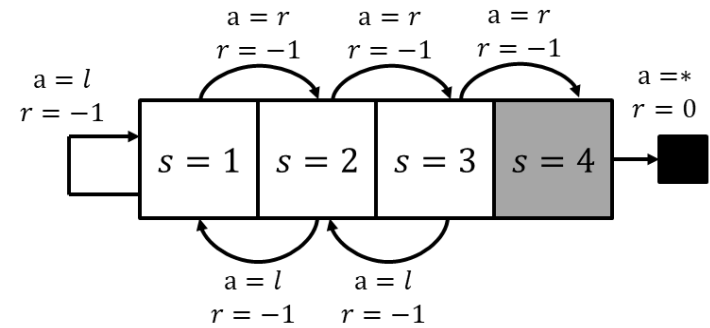
- QL update equation:  $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t))$
- $Q(4, a) \equiv 0$ . Initialize  $Q(1,*) = Q(2,*) = Q(3,*) = 0$
- EP1: (3,l, -1), (2,r, -1), (3,r, -1), (4,r, 0)

$$1. \quad Q(3, l) \leftarrow Q(3, l) + \alpha \left( R + \gamma \max_{a'} Q(2, a') - Q(3, l) \right) = 0 + .5(-1 + \max(0, 0) - 0) = -0.5$$

$$2. \quad Q(2, r) \leftarrow Q(2, r) + \alpha \left( R + \gamma \max_{a'} Q(3, a') - Q(2, r) \right) = 0 + .5(-1 + \max(-0.5, 0) - 0) = -0.5$$

$$3. \quad Q(3, r) \leftarrow Q(3, r) + \alpha \left( R + \gamma \max_{a'} Q(4, a') - Q(3, r) \right) = 0 + .5(-1 + 0 - 0) = -0.5$$

# QL EP1



Sarsa	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	0	0	-0.5	-0.5	-0.5

# MC, TD, Sarsa, QL (Complex)

- C. Assume learning rate  $\alpha = 1$ . Consider 8 given consecutive episodes in the form of (s,a,r) (we do not consider  $\epsilon$ -greedy exploration here):
  1. EP1: (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  2. EP2: (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  3. EP3: (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  4. EP4: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  5. EP5: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  6. EP6: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  7. EP7: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
  8. EP8: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)
- Derive the following:
  1. State value functions  $V(s)$  after MC learning.
  2. State value functions  $V(s)$  after TD learning.
  3. State-action value functions  $Q(s, a)$  after Sarsa, and the resulting policy.
  4. State-action value functions  $Q(s, a)$  after Q learning, and the resulting policy.

MC

- MC update equation:  $V(S_t) \leftarrow G_t$
- $V(4) \equiv 0$ . Initialize  $V(1) = V(2) = V(3) = 0$
- EP1:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

- MC (every-visit w. EP1  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ):

- Update  $G(s)$  backward:

1.  $G(3) \leftarrow -1$

2.  $G(2) \leftarrow -1 + \gamma G(3) = -2$

3.  $G(1) \leftarrow -1 + \gamma G(2) = -3,$

- Update  $V(s)$  forward:

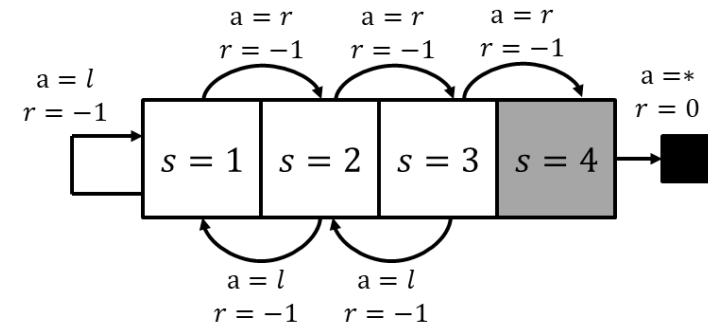
1.  $V(1) \leftarrow G(1) = -3$

2.  $V(2) \leftarrow G(2) = -2$

3.  $V(3) \leftarrow G(3) = -1$

- EP2-3: same as EP1

# MC EP1-3



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-3	-2	-1
After EP2	-3	-2	-1
After EP3	-3	-2	-1
After EP4	-3	-2	-1
After EP5	-3	-2	-1
After EP6	-3	-2	-1
After EP7	-3	-2	-1
After EP8	-3	-2	-1

- MC update equation:  $V(S_t) \leftarrow G_t$
- EP4:  
(3,l, -1), (2,l, -1), (1,l, -1), (1,r, -1), (2,r, -1), (3,r, -1), (4,r, 0)
- MC (every-visit w. EP4  $3' \rightarrow 2' \rightarrow 1' \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ):
- Update  $G(s)$  backward:

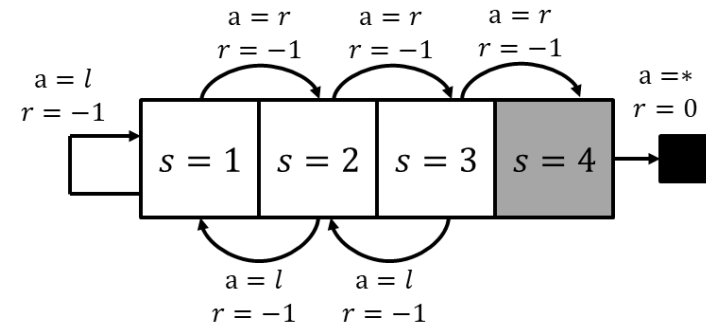
- $G(3) \leftarrow -1 + \gamma G(4) = -1$  (2<sup>nd</sup> visit)
- $G(2) \leftarrow -1 + \gamma G(3) = -2$  (2<sup>nd</sup> visit)
- $G(1) \leftarrow -1 + \gamma G(2) = -3$  (2<sup>nd</sup> visit)
- $G(1') \leftarrow -1 + \gamma G(1) = -4$  (1<sup>st</sup> visit)
- $G(2') \leftarrow -1 + \gamma G(1') = -5$  (1<sup>st</sup> visit)
- $G(3') \leftarrow -1 + \gamma G(2') = -6$  (1<sup>st</sup> visit)

- Update  $V(s)$  forward:

- $V(3) = G(3') = -6$
- $V(2) = G(2') = -5$
- $V(1) = G(1') = -4$
- $V(1) = G(1) = -3$
- $V(2) = G(2) = -2$
- $V(3) = G(3) = -1$

- EP5-8: same as EP4

# MC EP4-8



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-3	-2	-1
After EP2	-3	-2	-1
After EP3	-3	-2	-1
After EP4	-3	-2	-1
After EP5	-3	-2	-1
After EP6	-3	-2	-1
After EP7	-3	-2	-1
After EP8	-3	-2	-1



TD

- TD update equation:  $V(S_t) \leftarrow R_{t+1} + V(S_{t+1})$
- $V(4) \equiv 0$ . Initialize  $V(1) = V(2) = V(3) = 0$ ,
- EP1:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

1.  $V(1) \leftarrow -1 + V(2) = -1 + 0 = -1$
2.  $V(2) \leftarrow -1 + V(3) = -1 + 0 = -1$
3.  $V(3) \leftarrow -1 + V(4) = -1 + 0 = -1$

- EP2:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

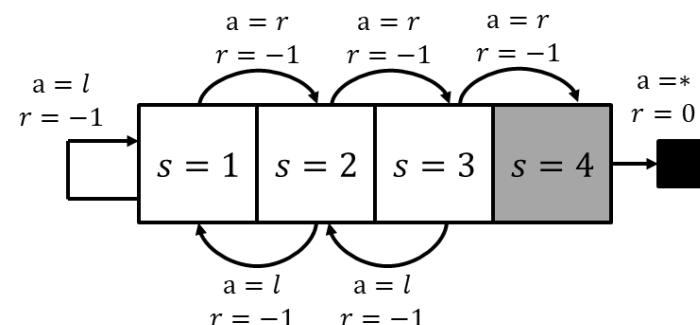
1.  $V(1) \leftarrow -1 + V(2) = -1 - 1 = -2$
2.  $V(2) \leftarrow -1 + V(3) = -1 - 1 = -2$
3.  $V(3) \leftarrow -1 + V(4) = -1 + 0 = -1$

- EP3:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

1.  $V(1) \leftarrow -1 + V(2) = -1 - 2 = -3$
2.  $V(2) \leftarrow -1 + V(3) = -1 - 1 = -2$
3.  $V(3) \leftarrow -1 + V(4) = -1 + 0 = -1$

- Arrows denote bootstrap dependencies, e.g.,  $V(1)$  bootstraps off  $V(2)$ ,  $V(2)$  bootstraps off  $V(3)$ ,  $V(3)$  bootstraps off  $V(4)$ . They also denote direction of information flow during learning, e.g.,  $V(4) \equiv 0$  is the external learning signal, and info flows  $V(4) \rightarrow V(3) \rightarrow V(2) \rightarrow V(1)$ .

# TD EP1-3



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-1	-1	-1
After EP2	-2	-2	-1
After EP3	-3	-2	-1
After EP4	-5	-4	-1
After EP5	-7	-6	-1
After EP6	-9	-8	-1
After EP7	-11	-10	-1
After EP8	-13	-12	-1

- TD update equation:  $V(S_t) \leftarrow R_{t+1} + V(S_{t+1})$

1. EP4:

$(3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

2.  $V(3) \leftarrow -1 + V(2) = -1 - 2 = -3$

3.  $V(2) \leftarrow -1 + V(1) = -1 - 3 = -4$

4.  $V(1) \leftarrow -1 + V(1) = -1 - 3 = -4$

5.  $V(1) \leftarrow -1 + V(2) = -1 - 4 = -5$

6.  $V(2) \leftarrow -1 + V(3) = -1 - 3 = -4$

7.  $V(3) \leftarrow -1 + V(4) = -1 + 0 = -1$

• EP5:

$(3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

1.  $V(3) \leftarrow -1 + V(2) = -1 - 4 = -5$

2.  $V(2) \leftarrow -1 + V(1) = -1 - 5 = -6$

3.  $V(1) \leftarrow -1 + V(1) = -1 - 5 = -6$

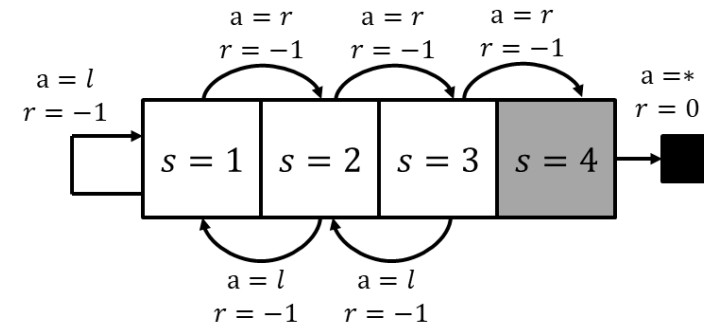
4.  $V(1) \leftarrow -1 + V(2) = -1 - 6 = -7$

5.  $V(2) \leftarrow -1 + V(3) = -1 - 5 = -6$

6.  $V(3) \leftarrow -1 + V(4) = -1 + 0 = -1$

• EP6-8 omitted.

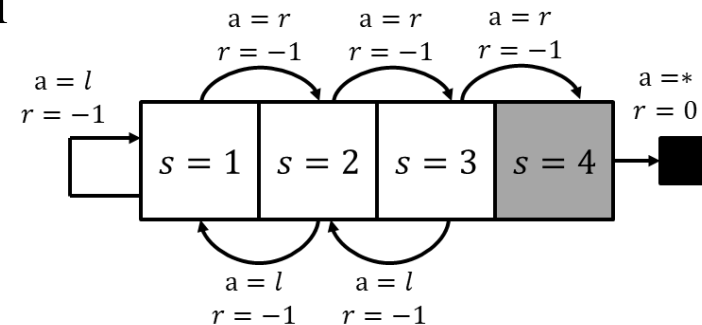
# TD EP4-8



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-1	-1	-1
After EP2	-2	-2	-1
After EP3	-3	-2	-1
After EP4	-4	-4	-3
After EP4	-5	-4	-1
After EP4	-6	-6	-5
After EP4	-7	-6	-1
After EP6	-9	-8	-1
After EP7	-11	-10	-1
After EP8	-13	-12	-1

# TD Failed to Converge

- TD failed to converge for this set of episodes, all value functions grow increasingly negative.
- The reason is that  $V(1)$  and  $V(2)$  bootstrap off each other and form a bootstrap dependency cycle  $V(2) \leftarrow V(1) \leftarrow V(2) \dots$ , i.e., a cycle of TD updates:  $V(2) = -1 + V(1)$ ,  $V(1) = -1 + V(2)$ , ...
  - An analogy: 2 students  $V(1)$  and  $V(2)$  are copying from each other, but they never get any true reward feedback from the external teacher ( $V(4) \equiv 0$ )
- $V(3)$  is bootstrapped off  $V(2)$  when moving left, and is bootstrapped off  $V(4) \equiv 0$  when moving right. Even though  $V(3)$  is updated to the correct  $V(3) = -1 + V(4) = -1$  when it moves right to state 4, the episode ends immediately afterwards, so  $V(1)$  and  $V(2)$  do not have a chance to bootstrap off  $V(3) = -1$ .
- If the episode does not end immediately, but the agent moves left again, then  $V(1)$  and  $V(2)$  will have a chance to bootstrap off the new  $V(3)$ , and they may converge to the correct values.



TD	$V(1)$	$V(2)$	$V(3)$
Init	0	0	0
After EP1	-1	-1	-1
After EP2	-2	-2	-1
After EP3	-3	-2	-1
After EP4	-4	-4	-3
After EP4	-5	-4	-1
After EP4	-6	-6	-5
After EP4	-7	-6	-1
After EP6	-9	-8	-1
After EP7	-11	-10	-1
After EP8	-13	-12	-1

# Sarsa

- Sarsa update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + Q(S_{t+1}, A_{t+1})$
- $Q(4, a) \equiv 0$ . Initialize  $Q(1, *) = Q(2, *) = Q(3, *) = 0$
- EP1:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. Q(1, r) \leftarrow -1 + Q(2, r) = -1 + 0 = -1$$

$$2. Q(2, r) \leftarrow -1 + Q(3, r) = -1 + 0 = -1$$

$$3. Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

- EP2:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 1 = -2$$

$$2. Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$3. Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

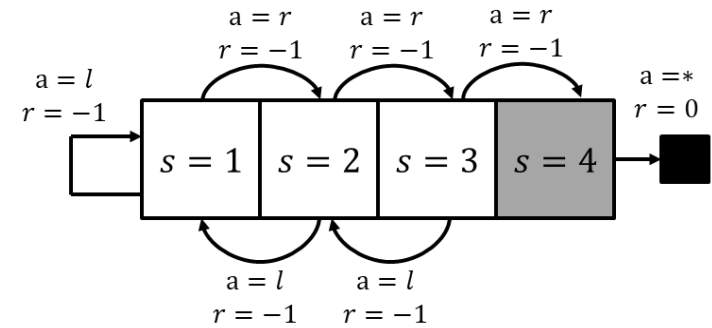
- EP3:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$$

$$2. Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$3. Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

# Sarsa EP1-3



Sarsa	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-2	0	-2	0	-1
After EP3	0	-3	0	-2	0	-1
After EP4	-4	-3	-1	-2	-1	-1
After EP5	-4	-3	-5	-2	-2	-1
After EP6	-4	-3	-5	-2	-6	-1
After EP7	-4	-3	-5	-2	-6	-1
After EP8	-4	-3	-5	-2	-6	-1

- Sarsa update equation:  $Q(S_t, A_t) \leftarrow R_{t+1} + Q(S_{t+1}, A_{t+1})$
- EP4: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)

$$1. \quad Q(3, l) \leftarrow -1 + Q(2, l) = -1 + 0 = -1$$

$$2. \quad Q(2, l) \leftarrow -1 + Q(1, l) = -1 + 0 = -1$$

$$3. \quad Q(1, l) \leftarrow -1 + Q(1, r) = -1 - 3 = -4$$

$$4. \quad Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$$

$$5. \quad Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$6. \quad Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

- EP5: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)

$$1. \quad Q(3, l) \leftarrow -1 + Q(2, l) = -1 - 1 = -2$$

$$2. \quad Q(2, l) \leftarrow -1 + Q(1, l) = -1 - 4 = -5$$

$$3. \quad Q(1, l) \leftarrow -1 + Q(1, r) = -1 - 3 = -4$$

$$4. \quad Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$$

$$5. \quad Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$6. \quad Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

- EP6: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)

$$1. \quad Q(3, l) \leftarrow -1 + Q(2, l) = -1 - 5 = -6$$

$$2. \quad Q(2, l) \leftarrow -1 + Q(1, l) = -1 - 4 = -5$$

$$3. \quad Q(1, l) \leftarrow -1 + Q(1, r) = -1 - 3 = -4$$

$$4. \quad Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$$

$$5. \quad Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$6. \quad Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$$

- EP7: (3, l, -1), (2, l, -1), (1, l, -1), (1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)

$$1. \quad Q(3, l) \leftarrow -1 + Q(2, l) = -1 - 5 = -6$$

$$2. \quad Q(2, l) \leftarrow -1 + Q(1, l) = -1 - 4 = -5$$

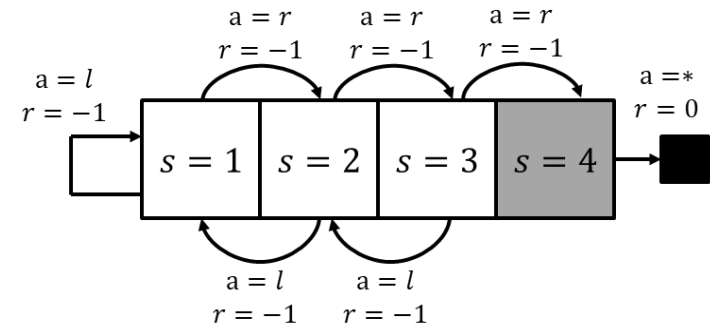
$$3. \quad Q(1, l) \leftarrow -1 + Q(1, r) = -1 - 3 = -4$$

$$4. \quad Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$$

$$5. \quad Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$$

$$6. \quad Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1 \text{ (EP8 omitted)}$$

# Sarsa EP4-8

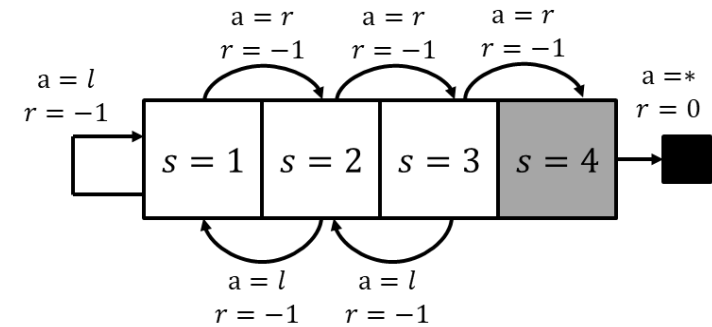


Sarsa	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-2	0	-2	0	-1
After EP3	0	-3	0	-2	0	-1
After EP4	-4	-3	-1	-2	-1	-1
After EP5	-4	-3	-5	-2	-2	-1
After EP6	-4	-3	-5	-2	-6	-1
After EP7	-4	-3	-5	-2	-6	-1
After EP8	-4	-3	-5	-2	-6	-1

Q values have converged at EP6. Bootstrap dependency arrows are omitted for EP7-8, since they are the same as EP6. **Red arrows** denote the stable set of dependencies that keep the Q values stable after EP6.

# Comments on Sarsa

- State-action value functions for moving right look reasonable:  $Q(1,r) = -3, Q(2,r) = -2, Q(3,r) = -1$ .
- State-action value functions for moving left look unreasonable:  $Q(1,l) = -4, Q(2,l) = -5, Q(3,l) = -6$ .  
This is because the only episodes with move left actions are  $3 \rightarrow 2 \rightarrow 1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ , the Q values are updated based on only this episode (on-policy),  $Q(3,l)$  bootstraps off  $Q(2,r)$  to get  $Q(3,l) = -1 + Q(2,l) = -6$ . Only if agent had experienced additional trajectories like  $3 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ,  $Q(3,l)$  would bootstrap off  $Q(2,r)$  to learn the correct value of  $Q(3,l) = -1 + Q(2,r) = -3$ .
- Even though the Q values for left actions are inaccurate, the greedy policy is still optimal (policy stable before value functions converge.)
- $\pi_*(1) = \operatorname{argmax}_a(Q(1,l), Q(1,r)) = r$ ;  $\pi_*(2) = \operatorname{argmax}_a(Q(2,l), Q(2,r)) = r$ ;  $\pi_*(3) = \operatorname{argmax}_a(Q(3,l), Q(3,r)) = r$



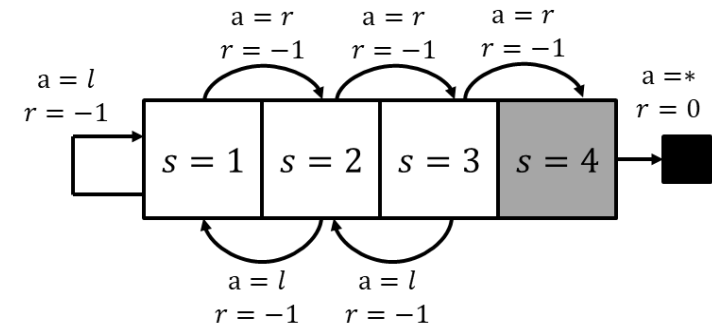
Sarsa	$Q(1,l)$	$Q(1,r)$	$Q(2,l)$	$Q(2,r)$	$Q(3,l)$	$Q(3,r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-2	0	-2	0	-1
After EP3	0	-3	0	-2	0	-1
After EP4	-4	-3	-1	-2	-1	-1
After EP5	-4	-3	-5	-2	-2	-1
After EP6	-4	-3	-5	-2	-6	-1
After EP7	-4	-3	-5	-2	-6	-1
After EP8	-4	-3	-5	-2	-6	-1



# Why Sarsa Converges

- When agent moves left from state  $s$ ,  $Q(s, l)$  is updated; when agent moves right,  $Q(s, r)$  is updated. The bootstrap dependency chain is  $Q(3, l) \leftarrow Q(2, l) \leftarrow Q(1, l) \leftarrow Q(1, r) \leftarrow Q(2, r) \leftarrow Q(3, r) \leftarrow Q(4, r)$ . So there is no bootstrap dependency cycle like TD ( $V(2) \leftarrow V(1) \leftarrow V(2) \dots$ ). The bootstrap dependency chain determines the stable values:

- $Q(3, l) \leftarrow -1 + Q(2, l) = -1 - 5 = -6$
- $Q(2, l) \leftarrow -1 + Q(1, l) = -1 - 4 = -5$
- $Q(1, l) \leftarrow -1 + Q(1, r) = -1 - 3 = -4$
- $Q(1, r) \leftarrow -1 + Q(2, r) = -1 - 2 = -3$
- $Q(2, r) \leftarrow -1 + Q(3, r) = -1 - 1 = -2$
- $Q(3, r) \leftarrow -1 + Q(4, r) = -1 + 0 = -1$



Sarsa	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-2	0	-2	0	-1
After EP3	0	-3	0	-2	0	-1
After EP4	-4	-3	-1	-2	-1	-1
After EP5	-4	-3	-5	-2	-2	-1
After EP6	-4	-3	-5	-2	-6	-1
After EP7	-4	-3	-5	-2	-6	-1
After EP8	-4	-3	-5	-2	-6	-1

# Q Learning

- QL update equation:  $(S_t, A_t) \leftarrow R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a')$

- EP1:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. \quad Q(1, r) \leftarrow -1 + \max_{a'} Q(2, a') = -1 + \max(0, 0) = -1$$

$$2. \quad Q(2, r) \leftarrow -1 + \max_{a'} Q(3, a') = -1 + \max(0, 0) = -1$$

$$3. \quad Q(3, r) \leftarrow -1 + \max_{a'} Q(4, a') = -1 + 0 = -1$$

- EP2:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. \quad Q(1, r) \leftarrow -1 + \max_{a'} Q(2, a') = -1 + \max(-1, 0) = -1$$

$$2. \quad Q(2, r) \leftarrow -1 + \max_{a'} Q(3, a') = -1 + \max(-1, 0) = -1$$

$$3. \quad Q(3, r) \leftarrow -1 + \max_{a'} Q(4, a') = -1 + 0 = -1$$

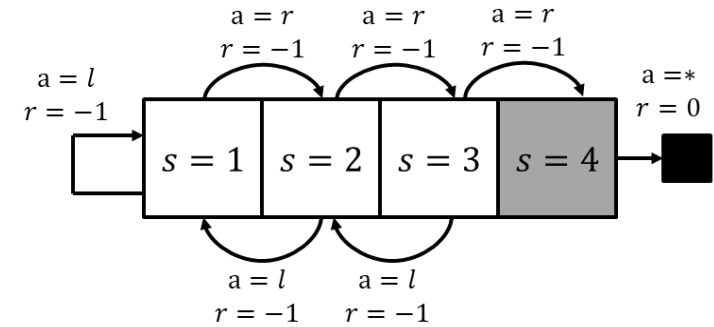
- EP3:  $(1, r, -1), (2, r, -1), (3, r, -1), (4, r, 0)$

$$1. \quad Q(1, r) \leftarrow -1 + \max_{a'} Q(2, a') = -1 + \max(-1, 0) = -1$$

$$2. \quad Q(2, r) \leftarrow -1 + \max_{a'} Q(3, a') = -1 + \max(-1, 0) = -1$$

$$3. \quad Q(3, r) \leftarrow -1 + \max_{a'} Q(4, a') = -1 + 0 = -1$$

# QL EP1-3



QL	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-1	0	-1	0	-1
After EP3	0	-1	0	-1	0	-1
After EP4	-1	-2	-1	-2	-1	-1
After EP5	-2	-3	-2	-2	-2	-1
After EP6	-3	-3	-3	-2	-3	-1
After EP7	-4	-3	-4	-2	-3	-1
After EP8	-4	-3	-4	-2	-3	-1

- EP4: (3,l,-1), (2,l,-1), (1,l,-1), (1,r,-1), (2,r,-1), (3,r,-1), (4,r,0)

1.  $Q(3,l) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(0, -1) = -1$
2.  $Q(2,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(0, -1) = -1$
3.  $Q(1,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(0, -1) = -1$
4.  $Q(1,r) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-1, -1) = -2$
5.  $Q(2,r) \leftarrow -1 + \max_{a'} Q(3,a') = -1 + \max(-1, -1) = -2$
6.  $Q(3,r) \leftarrow -1 + \max_{a'} Q(4,a') = -1 + 0 = -1$

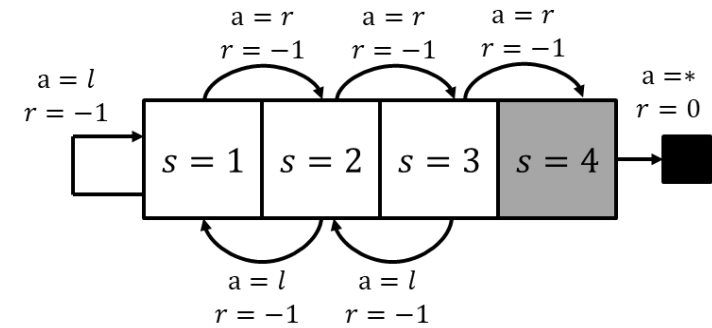
- EP5: (3,l,-1), (2,l,-1), (1,l,-1), (1,r,-1), (2,r,-1), (3,r,-1), (4,r,0)

1.  $Q(3,l) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-1, -2) = -2$
2.  $Q(2,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-1, -2) = -2$
3.  $Q(1,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-1, -2) = -2$
4.  $Q(1,r) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-2, -2) = -3$
5.  $Q(2,r) \leftarrow -1 + \max_{a'} Q(3,a') = -1 + \max(-2, -1) = -2$
6.  $Q(3,r) \leftarrow -1 + \max_{a'} Q(4,a') = -1 + 0 = -1$

- EP6: (3,l,-1), (2,l,-1), (1,l,-1), (1,r,-1), (2,r,-1), (3,r,-1), (4,r,0)

1.  $Q(3,l) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-2, -2) = -3$
2.  $Q(2,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-2, -3) = -3$
3.  $Q(1,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-2, -3) = -3$
4.  $Q(1,r) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-3, -2) = -3$
5.  $Q(2,r) \leftarrow -1 + \max_{a'} Q(3,a') = -1 + \max(-3, -1) = -2$
6.  $Q(3,r) \leftarrow -1 + \max_{a'} Q(4,a') = -1 + 0 = -1$

# QL EP4-6



QL	$Q(1,l)$	$Q(1,r)$	$Q(2,l)$	$Q(2,r)$	$Q(3,l)$	$Q(3,r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-1	0	-1	0	-1
After EP3	0	-1	0	-1	0	-1
After EP4	-1	-2	-1	-2	-1	-1
After EP5	-2	-3	-2	-2	-2	-1
After EP6	-3	-3	-3	-2	-3	-1
After EP7	-4	-3	-4	-2	-3	-1
After EP8	-4	-3	-4	-2	-3	-1

- EP7: (3,l,-1), (2,l,-1), (1,l,-1), (1,r,-1), (2,r,-1), (3,r,-1), (4,r,0)

$$1. \quad Q(3,l) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-3,-2) = -3$$

$$2. \quad Q(2,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-3,-3) = -4$$

$$3. \quad Q(1,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-3,-3) = -4$$

$$4. \quad Q(1,r) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-4,-2) = -3$$

$$5. \quad Q(2,r) \leftarrow -1 + \max_{a'} Q(3,a') = -1 + \max(-3,-1) = -2$$

$$6. \quad Q(3,r) \leftarrow -1 + \max_{a'} Q(4,a') = -1 + 0 = -1$$

- EP8: (3,l,-1), (2,l,-1), (1,l,-1), (1,r,-1), (2,r,-1), (3,r,-1), (4,r,0)

$$1. \quad Q(3,l) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-4,-2) = -3$$

$$2. \quad Q(2,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-4,-3) = -4$$

$$3. \quad Q(1,l) \leftarrow -1 + \max_{a'} Q(1,a') = -1 + \max(-4,-3) = -4$$

$$4. \quad Q(1,r) \leftarrow -1 + \max_{a'} Q(2,a') = -1 + \max(-4,-2) = -3$$

$$5. \quad Q(2,r) \leftarrow -1 + \max_{a'} Q(3,a') = -1 + \max(-3,-1) = -2$$

$$6. \quad Q(3,r) \leftarrow -1 + \max_{a'} Q(4,a') = -1 + 0 = -1$$

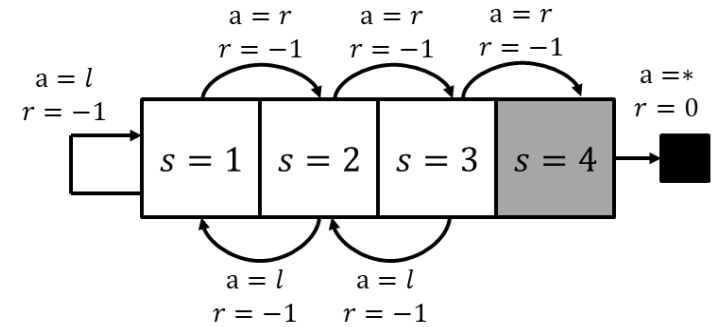
- Q values have converged at EP7. Red arrows denote the stable set of dependencies that keep the Q values stable after EP7.

- Q values learned by QL are accurate, and the greedy policy is optimal:

$$\pi_*(1) = \operatorname{argmax}_a (Q(1,l), Q(1,r)) = r; \pi_*(2) =$$

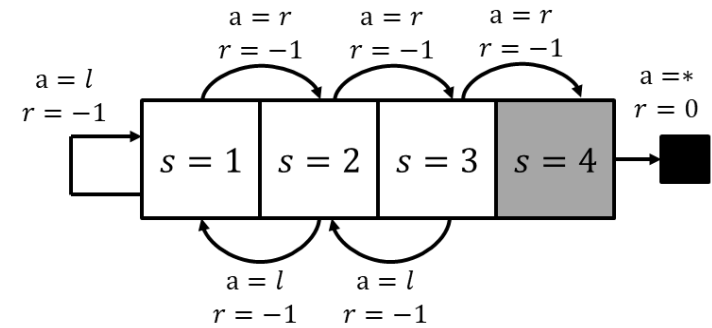
$$\operatorname{argmax}_a (Q(2,l), Q(2,r)) = r; \pi_*(3) = \operatorname{argmax}_a (Q(3,l), Q(3,r)) = r$$

# QL EP7-8



QL	$Q(1,l)$	$Q(1,r)$	$Q(2,l)$	$Q(2,r)$	$Q(3,l)$	$Q(3,r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-1	0	-1	0	-1
After EP3	0	-1	0	-1	0	-1
After EP4	-1	-2	-1	-2	-1	-1
After EP5	-2	-3	-2	-2	-2	-1
After EP6	-3	-3	-3	-2	-3	-1
After EP7	-4	-3	-4	-2	-3	-1
After EP8	-4	-3	-4	-2	-3	-1

# Comments on QL



QL	$Q(1, l)$	$Q(1, r)$	$Q(2, l)$	$Q(2, r)$	$Q(3, l)$	$Q(3, r)$
Init	0	0	0	0	0	0
After EP1	0	-1	0	-1	0	-1
After EP2	0	-1	0	-1	0	-1
After EP3	0	-1	0	-1	0	-1
After EP4	-1	-2	-1	-2	-1	-1
After EP5	-2	-3	-2	-2	-2	-1
After EP6	-3	-3	-3	-2	-3	-1
After EP7	-4	-3	-4	-2	-3	-1
After EP8	-4	-3	-4	-2	-3	-1

- QL converges. All state-action value functions look reasonable.
- $Q(1, r) = -3, Q(2, r) = -2, Q(3, r) = -1$ . The optimal path can be derived from bootstrap dependencies, e.g., dependency chain  $Q(1, r) \leftarrow Q(2, r) \leftarrow Q(3, r)$  corresponds to the optimal path  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ .
- $Q(1, l) = -4$ : If agent moves left in state 1, dependency chain  $Q(1, l) \leftarrow Q(1, r) \leftarrow Q(2, r) \leftarrow Q(3, r)$  corresponds to the optimal path  $1 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$  w. 4 steps to reach goal state 4.
- $Q(2, l) = -4$ : If agent moves left in state 2, dependency chain  $Q(2, l) \leftarrow Q(1, r) \leftarrow Q(2, r) \leftarrow Q(3, r)$  corresponds to the optimal path  $2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$  w. 4 steps to reach goal state 4.
- $Q(3, l) = -3$ : If agent moves left in state 3, dependency chain  $Q(3, l) \leftarrow Q(2, r) \leftarrow Q(3, r)$  corresponds to the optimal path  $3 \rightarrow 2 \rightarrow 3 \rightarrow 4$  w. 3 steps to reach goal state 4.
- QL is smarter than Sarsa: since it is off-policy, agent can learn the correct Q value functions that correspond to trajectories that it has never experienced.
- Bootstrap dependencies change during learning:  $Q(3, l)$  initially bootstraps off  $Q(2, l)$  based on the initialized Q values, but as  $Q(2, l)$  decreases gradually to below  $Q(2, r)$  after EP6,  $Q(3, l)$  switches to bootstrap off  $Q(2, r)$  to learn the correct value of  $Q(3, l) = -1 + Q(2, r) = -3$ , even though it has never experienced the trajectory  $3 \rightarrow 2 \rightarrow 3 \rightarrow 4$  (contrast this to Sarsa). Similarly, both  $Q(2, l)$  and  $Q(1, l)$  switch from bootstrapping off  $Q(1, l)$  to  $Q(1, r)$  after EP7.
- The intermediate Q values before convergence may not correspond to a valid policy, e.g., before EP7,  $\arg\max_a Q(1, a) = l$ , so the agent would be stuck in state 1 trying to go left forever.