

# Markov Decision Processes

# Infinite-horizon discounted MDPs

An MDP  $M = (S, A, P, R, \gamma)$

- State space  $S$ .
- Action space  $A$ .
- Transition function  $P : S \times A \rightarrow \Delta(S)$ .  $\Delta(S)$  is the probability simplex over  $S$ , i.e., all non-negative vectors of length  $|S|$  that sums up to 1
- Reward function  $R: S \times A \rightarrow \mathbb{R}$ . (deterministic reward function)
- Discount factor  $\gamma \in [0,1)$
- The agent starts in some state  $s_1$ , takes action  $a_1$ , receives reward  $r_1 = R(s_1, a_1)$ , transitions to  $s_2 \sim P(s_1, a_1)$ , takes action  $a_2$ , so on so forth — the process continues **forever**
- Objective: (discounted) expected total reward
  - other terms used: return, value, utility, long-term reward, etc

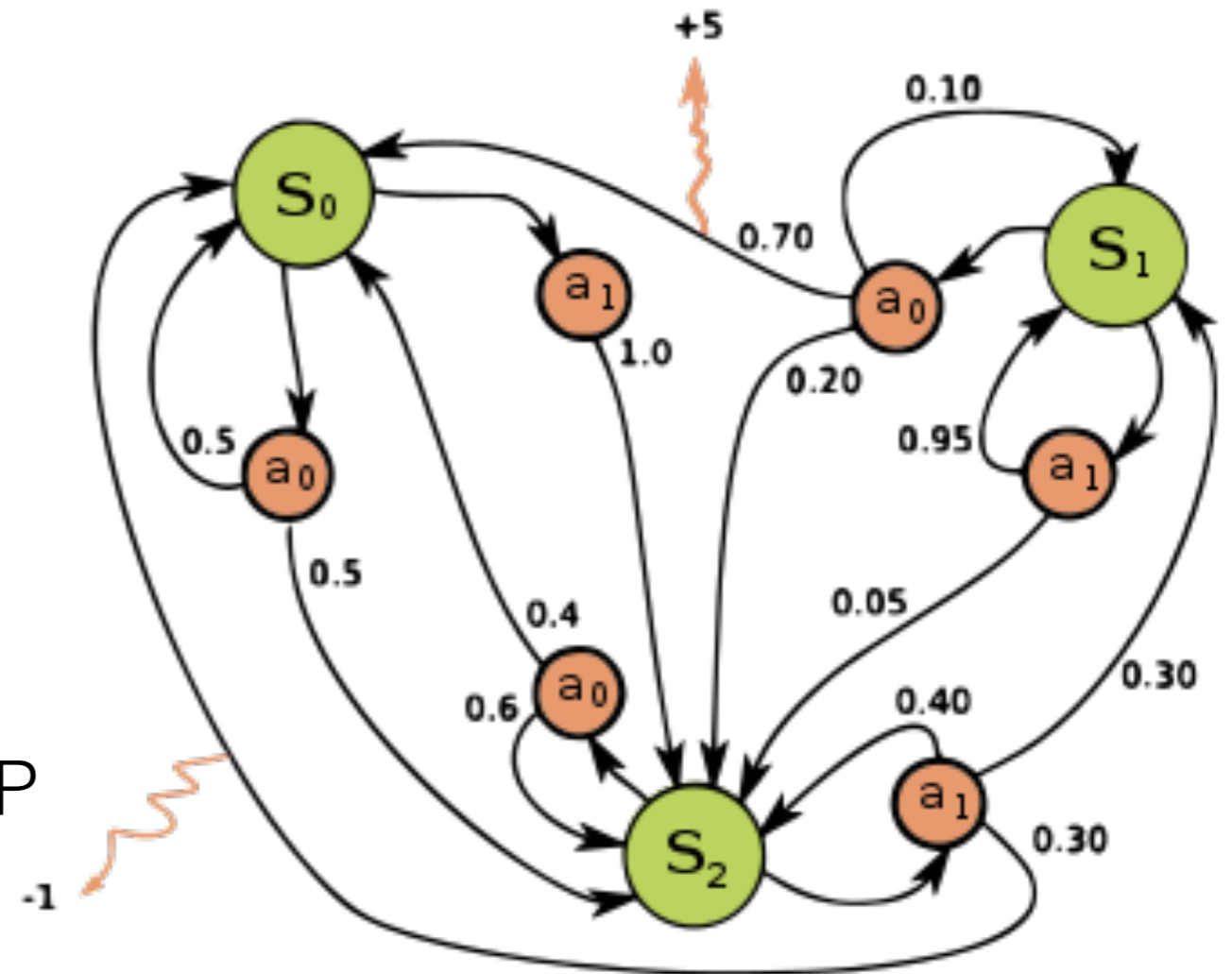
We will only consider discrete and finite spaces in this course (w/o losing much).

## Additional/alternative notations

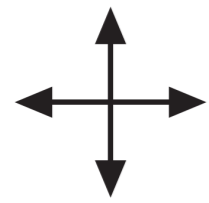
- The probability of transitioning to a particular state:  $P(s' | s, a)$
- Sometimes reward is random and/or depends on the next-state, e.g.,  $R(s, a, s')$ , or  $R(s, a)$  is a random variable
  - The most general case: given  $(s, a)$ ,  $(r, s')$  is drawn from some joint distribution (Sutton & Barto 2nd ed; see link on course website)
  - When we consider random rewards we will assume a simplified case:  $r$  and  $s'$  are independent conditioned on  $(s, a)$
  - Such differences usually don't matter—we will revisit later
- Sometimes the available actions depend on the state,  $A_s$ 
  - Again, the theory/algs developed for fixed action spaces usually extend to state-dependent action spaces

# Toy Example

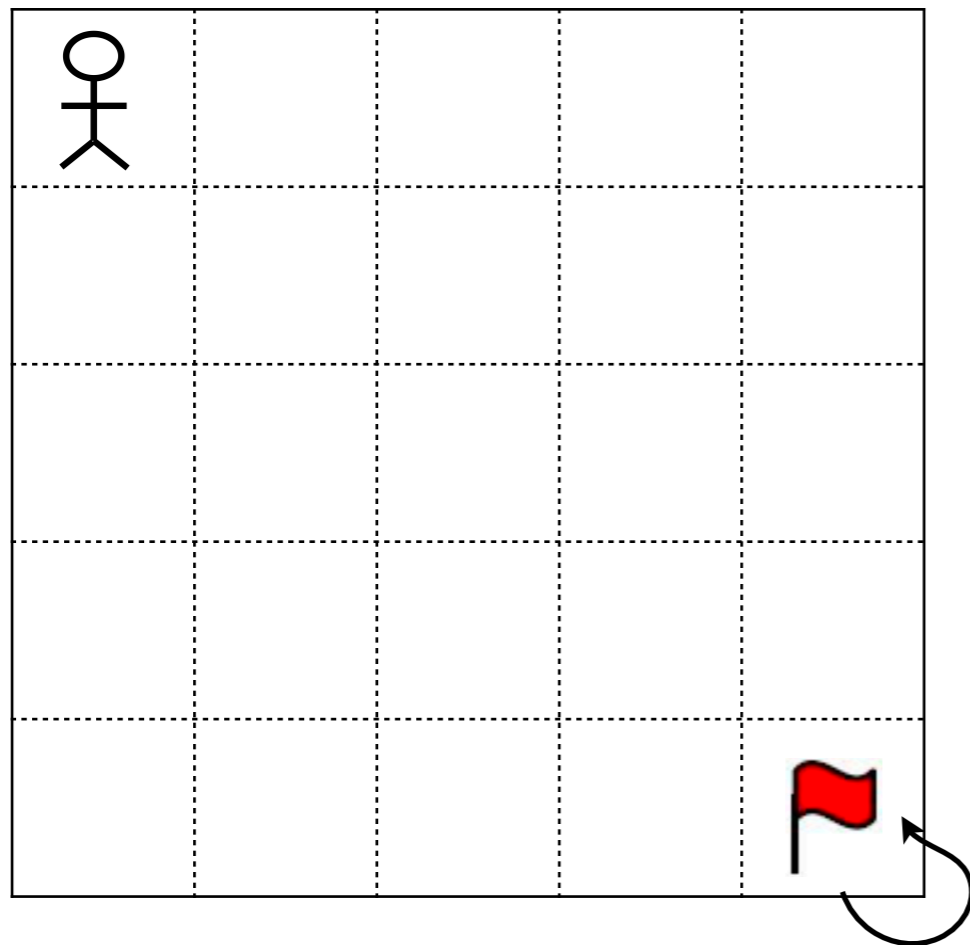
- $P(s_0 | s_2, a_0) = 0.4$
- $P(s_1 | s_2, a_0) = 0$
- $P(s_2 | s_2, a_0) = 0.6$
- ...
- If there is only 1 action, the MDP becomes a Markov chain (associated with a reward function)
- As another special case, if there are multiple actions but transition is deterministic, the MDP becomes a directed graph



# Example: Gridworld



Actions



“Absorbing state”: emulate the termination of a task in infinite-horizon MDPs

- State: grid No. (or integer  $x, y$  coordinates)
- Action: N, S, E, W
- Dynamics:
  - Deterministic transitions to the adjacent grid in the direction of action in most cases.
  - Keep in the current state if moving towards wall or having reached goal
- Reward: 0 in the goal state and -1 everywhere else
- Discount factor  $\gamma$ : 0.99

# Why discounting?

- When defining the MDP (esp. the reward function & the discount factor/horizon), you should make sure that the total expected reward of a policy precisely reflects how you like that policy. **Don't worry about learning yet.**
  - e.g., in the grid world example, if there is no discounting ( $\gamma = 1$ ), the total expected reward is the negative expected number of steps before reaching the goal
  - After you've learned a policy, **you should evaluate the policy using this reward function / horizon**
- So why discounting?
  - In the previous example,  $\gamma = 1$  allows some strategies to obtain  $-\infty$  expected return—we don't like infinities

## Why discounting? (cont.)

As a mathematical convenience...

- We introduce a discount factor close to (but smaller than) 1, so that even the agent moves in circles, the expected total reward is still finite.
- On the other hand, for most reasonable policies (esp. the near-optimal ones), the total reward is still approximately the negative total number of steps—**our training objective** closely tracks **what we really care about**
- Reason 2: discounting + infinite horizon = stationary optimal policy & value functions, again a mathematical convenience
- Reason 3: heavy discounting (small  $\gamma$ ) yields faster planning / learning (we will see)
- Discounting does have economic interpretations, but they are seldom relevant in RL

# Value and policy

- Want to take actions in a way that maximizes value (or return):

$$\mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \right]$$

- This value depends on where you start and how you act
- Often assume boundedness of rewards:  $r_t \in [0, R_{\max}]$ 
  - What's the range of  $\mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \right]$  ?  $\left[ 0, \frac{R_{\max}}{1-\gamma} \right]$
- A (deterministic) policy  $\pi: S \rightarrow A$  describes how the agent acts: at state  $s_t$ , chooses action  $a_t = \pi(s_t)$ .
- More generally, the agent may choose actions randomly ( $\pi: S \rightarrow \Delta(A)$ ), or even in a way that varies across time steps (“non-stationary policies”)
- Define 
$$V^{\pi}(s) = \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi \right]$$



# Bellman equation for policy evaluation

$$\begin{aligned}
 V^\pi(s) &= \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi \right] \\
 &= \mathbb{E} \left[ r_1 + \sum_{t=2}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi \right] \\
 &= R(s, \pi(s)) + \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) \mathbb{E} \left[ \gamma \sum_{t=2}^{\infty} \gamma^{t-2} r_t \mid s_1 = s, s_2 = s', \pi \right] \\
 &= R(s, \pi(s)) + \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) \mathbb{E} \left[ \gamma \sum_{t=2}^{\infty} \gamma^{t-2} r_t \mid s_2 = s', \pi \right] \\
 &= R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s', \pi \right] \\
 &= R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s' | s, \pi(s)) V^\pi(s') \\
 &= R(s, \pi(s)) + \gamma \langle P(\cdot | s, \pi(s)), V^\pi(\cdot) \rangle
 \end{aligned}$$

## Bellman equation for policy evaluation

$$V^\pi(s) = R(s, \pi(s)) + \gamma \langle P(\cdot | s, \pi(s)), V^\pi(\cdot) \rangle$$

Matrix form: define

- $V^\pi$  as the  $|S| \times 1$  vector  $[V^\pi(s)]_{s \in S}$
- $R^\pi$  as the vector  $[R(s, \pi(s))]_{s \in S}$
- $P^\pi$  as the matrix  $[P(s' | s, \pi(s))]_{s \in S, s' \in S}$

$$V^\pi = R^\pi + \gamma P^\pi V^\pi$$

$$(I - \gamma P^\pi) V^\pi = R^\pi$$

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

This is always invertible (Proof?), i.e., solution to Bellman equation is always unique

## Generalize to stochastic policies

- If  $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{a \sim \pi(\cdot|s)} [R(s, a)] + \gamma \sum_{a \in \mathcal{A}, s' \in \mathcal{S}} \pi(a|s) P(s'|s, a) V^\pi(s') \\ &= \mathbb{E}_{a \sim \pi(\cdot|s), s' \sim P(\cdot|s, a)} [R(s, a) + \gamma V^\pi(s')] \end{aligned}$$

- Matrix form  $V^\pi = R^\pi + \gamma P^\pi V^\pi$  still holds with

$$R^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} [R(s, a)] \quad \text{Shorthand: } R(s, \pi)$$

$$P^\pi(s'|s) = \sum_{a \in \mathcal{A}} \pi(a|s) P(s'|s, a) \quad \text{Shorthand: } P(s' | s, \pi)$$

- Convention: “(s)” after  $\pi$  dropped & integration over action implicit

## Some remarks about rewards

- For more general formulations of rewards (e.g., random rewards, rewards that depend on  $s'$ , etc)

$$V^\pi(s) = \mathbb{E}_{r,s'|s,\pi(s)}[r + \gamma V^\pi(s')]$$

- Can also be translated to an equivalent problem where rewards deterministically depends on  $(s,a)$ :  $R(s, a) := \mathbb{E}_{r|s,a}[r]$
- In infinite-horizon discounted MDPs, shifting the reward function by a constant for every state-action pair does not change anything
  - If we increase the reward function by  $c$ , you are just earning an extra  $c$  units of “background” rewards however you behave
  - All value functions increase by  $c/(1-\gamma)$

## Interpretation of $(I - \gamma P^\pi)^{-1}$

Each row of this matrix (indexed by  $s$ ) is the unnormalized **discounted state occupancy**, whose  $(s')$ -th entry is  $d_s^\pi(s')/(1 - \gamma)$ , where

$$d_s^\pi(s') := (1 - \gamma) \cdot \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{I}[s_t = s'] \mid s_1 = s, \pi \right]$$

- $\mathbb{I}[\cdot]$  is the indicator function: =1 if (.) is true, and =0 if false.
- Each row of  $(I - \gamma P^\pi)^{-1}$  is like a distribution vector—except that the entries sum up to  $1/(1-\gamma)$ . Multiplying by  $1-\gamma$  normalizes it
- Can also be interpreted as the value function of indicator reward function:  $d_s^\pi(s') = (1 - \gamma)V^\pi(s)$ , where  $V^\pi$  is defined wrt the reward function that is equal to 1 at state  $s'$  and 0 elsewhere
- Can similarly define state-action occupancy

## More on State occupancy

- Alternative way of defining state occupancy: let  $d_{s,t}^\pi$  be a (distribution) vector such that  $d_{s,t}^\pi(s') = \Pr[s_t = s' | s_1 = s, \pi]$ 
  - It's the  $t$ -th step state distribution induced by starting at state  $s$  and following policy  $\pi$
- The discounted occupancy is  $d_s^\pi = (1 - \gamma) \sum_{t=1}^{\infty} \gamma^{t-1} d_{s,t}^\pi$
- The  $s$ -th row of  $(I - \gamma P^\pi)^{-1}$  is  $d_s^\pi / (1 - \gamma)$ , because:
  - Let  $e_s = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^\top$ , where the 1 is at the  $s$ -th entry
  - $s$ -th row of  $(I - \gamma P^\pi)^{-1}$  is

$$\begin{aligned}
 e_s^\top (I - \gamma P^\pi)^{-1} &= e_s^\top \left( \sum_{t=1}^{\infty} (\gamma P^\pi)^{t-1} \right) \\
 &= \sum_{t=1}^{\infty} \gamma^{t-1} e_s^\top (P^\pi)^{t-1} = \sum_{t=1}^{\infty} \gamma^{t-1} (d_{s,t}^\pi)^\top
 \end{aligned}$$

## Alternative formula for value function

- Some further exercise that helps understanding

- $$V^\pi(s) = e_s^\top (I - \gamma P^\pi)^{-1} R^\pi$$
$$= e_s^\top \left( \sum_{t=1}^{\infty} \gamma^{t-1} (P^\pi)^{t-1} \right) R^\pi$$
$$= \sum_{t=1}^{\infty} \gamma^{t-1} \underbrace{\langle d_{s,t}^\pi, R^\pi \rangle}_{\mathbb{E}[r_t | s_1 = s, \pi]}$$

# Optimality

- For infinite-horizon discounted MDPs, there always exists a stationary and deterministic policy that is optimal for all starting states simultaneously
  - Proof: Puterman'94, Thm 6.2.7 (reference due to Shipra Agrawal)

- Let  $\pi^*$  denote this optimal policy, and  $V^* := V^{\pi^*}$

- Bellman Optimality Equation:

$$V^*(s) = \max_{a \in A} \left( R(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} [V^*(s')] \right)$$

- If we know  $V^*$ , how to get  $\pi^*$  ?
- Easier to work with Q-values:  $Q^*(s, a)$ , as  $\pi^*(s) = \arg \max_{a \in A} Q^*(s, a)$

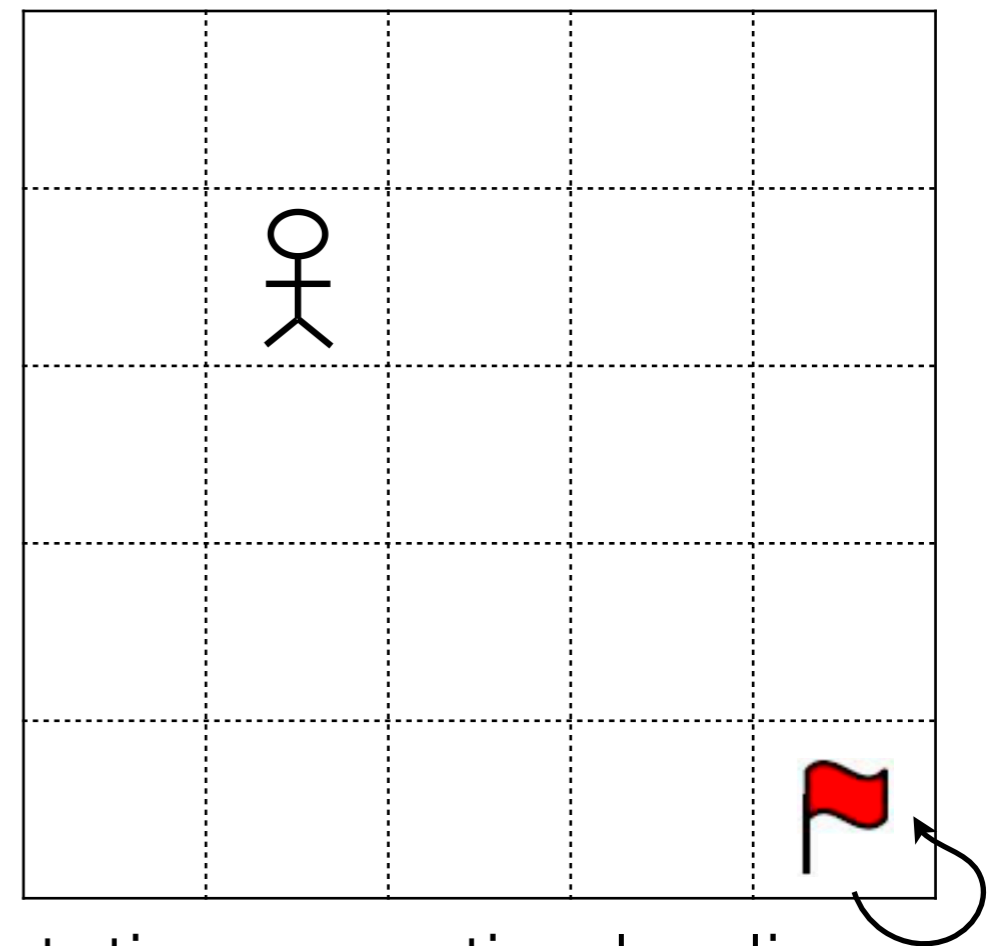
$$Q^*(s, a) = R(s, a) + \gamma \mathbb{E}_{s' \sim P(s, a)} \left[ \max_{a' \in A} Q^*(s', a') \right]$$

- Note that  $V^*(s) = \max_{a \in A} Q^*(s, a) = Q^*(s, \pi^*(s))$



# Optimality

- Similar to the Bellman equations for policy evaluation, the Bellman optimality equations also have unique solutions  $V^*$  and  $Q^*$
- But  $\pi^*$  may not be unique
- e.g., in the state shown in the figure, the optimal policy can take any of the following
  - Go right
  - Go down
  - Any probability distribution over right and down (stochastic policy)
- Remark: the fact that a deterministic & stationary optimal policy exists does NOT mean that all optimal policies have to be this way (they can be stochastic and/or non-stationary)



## (Policy-specific) Q-functions

$$Q^\pi(s, a) := \mathbb{E} \left[ \sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, a_1 = a; \pi \right]$$

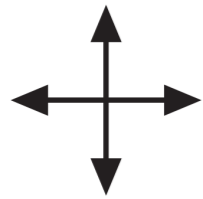
- $V^\pi(s) = Q^\pi(s, \pi(s))$ 
  - When  $\pi$  is stochastic, RHS becomes  $\mathbb{E}_{a \sim \pi(\cdot | s)} [Q^\pi(s, a)]$
  - We will often abbreviate  $Q^\pi(s, \pi(s))$  as  $Q^\pi(s, \pi)$
- Also satisfies a similar Bellman equation
$$Q^\pi(s, a) = R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} [Q^\pi(s', \pi)]$$
- Compare to the optimality equation
$$Q^*(s, a) = R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot | s, a)} \left[ \max_{a' \in A} Q^*(s', a') \right]$$
  - The only difference is in the action-selection operator at the next-state  $s'$ : if you choose  $a'$  greedily (max operator), you get  $Q^*$ ; if you choose  $a'$  according to  $\pi$ , you get  $Q^\pi$

# Clarification on definition of value functions

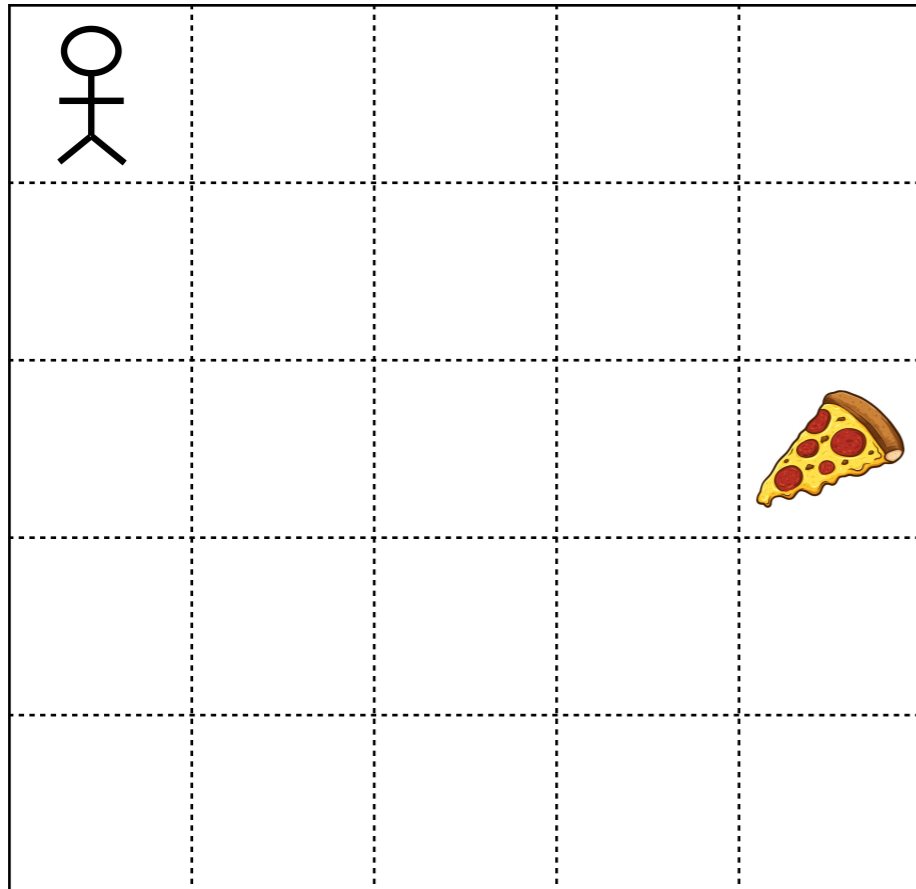
- $V^\pi(s), Q^\pi(s, a), V^*(s), Q^*(s, a)$
- they are the expected return when starting from  $s_1=s$  and following the behaviors below:
  - $V^\pi(s): a_1 \sim \pi(s_1), a_2 \sim \pi(s_2), a_3 \sim \pi(s_3), \dots$
  - $Q^\pi(s, a): a_1 = a, a_2 \sim \pi(s_2), a_3 \sim \pi(s_3), \dots$
  - $V^*(s) = V^{\pi^*}(s): a_1 \sim \pi^*(s_1), a_2 \sim \pi^*(s_2), a_3 \sim \pi^*(s_3), \dots$
  - $Q^*(s, a) = Q^{\pi^*}(s, a): a_1 = a, a_2 \sim \pi^*(s_2), a_3 \sim \pi^*(s_3), \dots$
- The policy (or  $*$ ) on the superscript denotes the long-term behavior
- For Q values, the second argument of the function specifies the first action; all future actions are according to the policy on the superscript

# Some Subtleties in Formulating Problems as MDPs

# Defining the state (in toy problems)



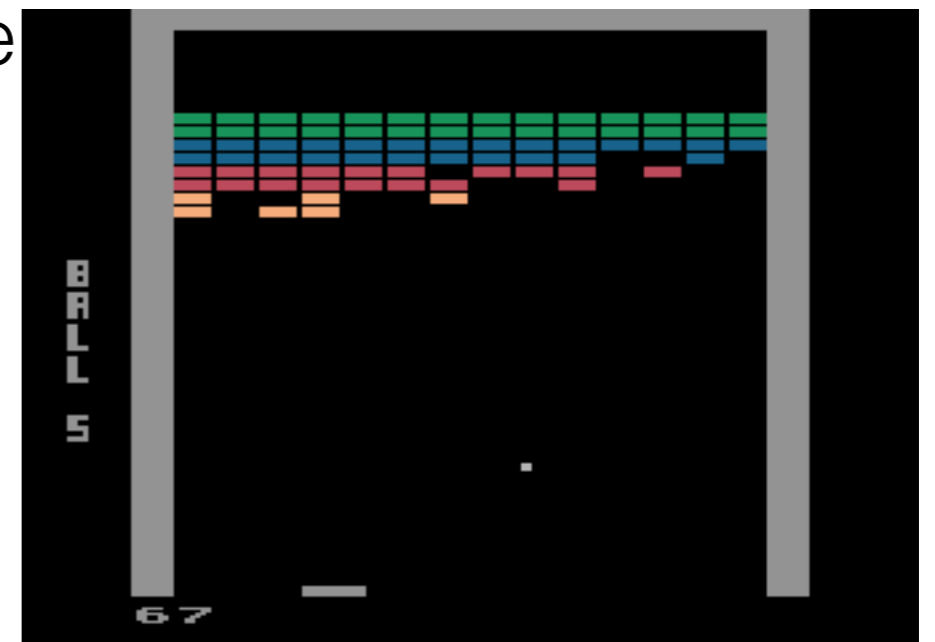
Actions



- The agent can move around just as in the previous example
  - Reward: +1 when reaching food for the first time, and 0 otherwise
    - i.e., food is consumable
  - How to define state?
    - The status of the food (consumed or not) also needs to be part of the state representation
- 
- In general, make sure that you can fully determine reward and (the distribution of) next-state using  $(s, a)$  alone without other info
  - The tricky part of this example is that dynamics are Markovian in the coordinates, but reward function is not

# Defining state beyond toy problems

- In toy problems, we can give any information about the system to the agent, so that the agent always has a (Markov) state
- In reality, agent receives limited sensory inputs every time step
- i.e., a trajectory looks like  $o_1, a_1, r_1, o_2, a_2, r_2, o_3...$
- We say that the problem is Markov (in  $o$ ), if the future observations and rewards are independent of the past conditioned on the current observation (and  $o$  is state)
- But this is generally not true...
- Need to construct an (approximate) state from **history**
- Will come back to this when we discuss POMDPs later in this course



## Alternative notions of horizons

- It is often more natural to model the problem as **indefinite-horizon** undiscounted MDPs
  - Some states are considered terminal
  - Trajectory does not go on forever—it stops at terminal states
  - The length of a trajectory is not fixed—it's random and/or may depend on the policy
  - e.g., navigation task: goal state is terminal, -1 per step before you terminate
  - Natural to model *episodic* tasks in this way
- Bellman equations remain the same for non-terminal states
- For terminal states,  $V(s) = 0$  (regardless of  $V^\pi$  or  $V^*$ )

# Fixed-horizon MDPs

- Specified by  $(S, A, R, P, H)$
- All trajectories end in precisely  $H$  steps
- No terminal states; termination is enforced “externally”
- Optimal policy may additionally depend on the time step
  - When selling goods, may price things differently near the end of the season, even if we assume the demand doesn’t change
  - In the navigation example, when you are too far away from goal to reach it within the remaining steps, any action is optimal (there is nothing you can do)
- So do (policy-specific/optimal) value functions:  $V_{H+1}^\pi(s) \equiv 0$ 
$$V_h^\pi(s) = R(s, \pi(s)) + \mathbb{E}_{s' \sim P(s,a)}[V_{h+1}^\pi(s')]$$
- Similar to the discounted setting, shifting rewards by constant doesn’t change the relative preference over policies (all value functions increase by  $cH$ )



## Translating between indefinite-horizon and fixed horizon MDPs

- “=>”: If trajectory length has an upper bound, set  $H$  to be this upper bound and add a dummy absorbing state; if a trajectory terminates early, loop in the dummy state for the remaining steps
- “<=“: Augment state space: include time step as part of the state representation, and define all states at time  $H+1$  as terminal, i.e., If  $P$  is the dynamics of the fixed horizon MDP, then define a new transition function

$$P'((s', h') | (s, h), a) = \mathbb{1}[h' = h + 1] \cdot P(s' | s, a)$$

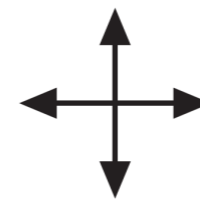
- State augmentation is a common trick when we convert between different formulations in RL (e.g., non-Markov problems become Markov if we view history as state)

# Translating between different formulations

- For discounted & fixed horizon, shifting rewards don't matter
- What about indefinite-horizon?
  - Think about what happens with the navigation example
  - Shift -1 to 1: you are rewarded if you don't reach the goal!
- Why is that?
  - For indefinite-horizon problems, since trajectory length is a variable, the positivity of reward actually carries a meaning: “+” means “live longer”, and “-“ means “finish faster”
  - For the other two formulations, such meaning doesn't exist
  - In-class exercise: formulate the navigation problem as a fixed horizon problem, then shift the reward. Explain why shifting rewards is fine in this case.

# The navigation example

- Original formulation: indef-horizon
  - -1 reward per step everywhere
  - Goal is terminal
  - Cannot shift reward
- Translate into fixed-horizon
  - Make goal state absorbing
  - 0 reward in goal and -1 everywhere else
  - Terminate in H steps
- Shift reward?
  - Have to shift everywhere, **including the goal**
  - e.g., if shift by 1, then +1 in goal and 0 everywhere else
  - No penalty, but being rewarded when looping in goal



Actions

