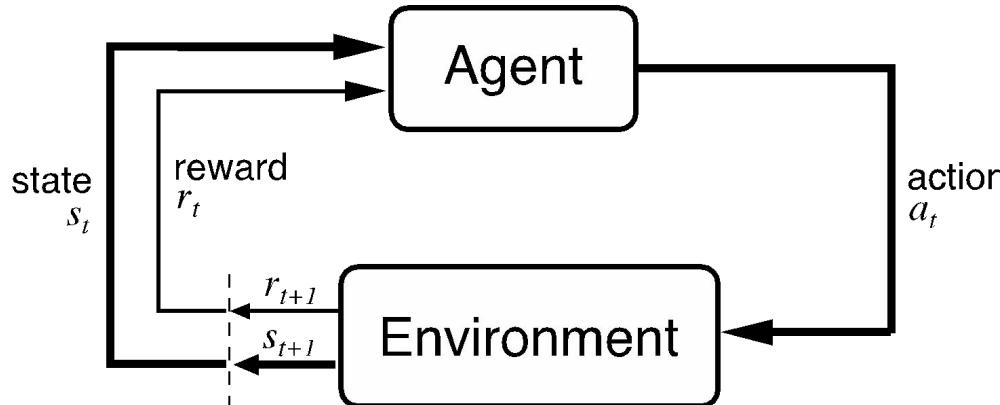


# MDPs, Returns, Value functions, Q-function

B. Ravindran

# The Agent-Environment Interface



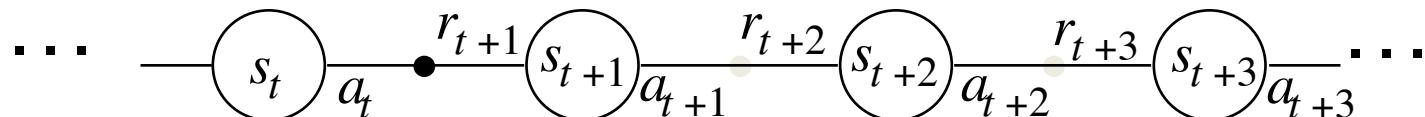
Agent and environment interact at discrete time steps:  $t = 0, 1, 2, \dots$

Agent observes state at step  $t$ :  $s_t \in S$

produces action at step  $t$  :  $a_t \in A(s_t)$

gets resulting reward:  $r_{t+1} \in \mathcal{R}$

and resulting next state:  $s_{t+1}$

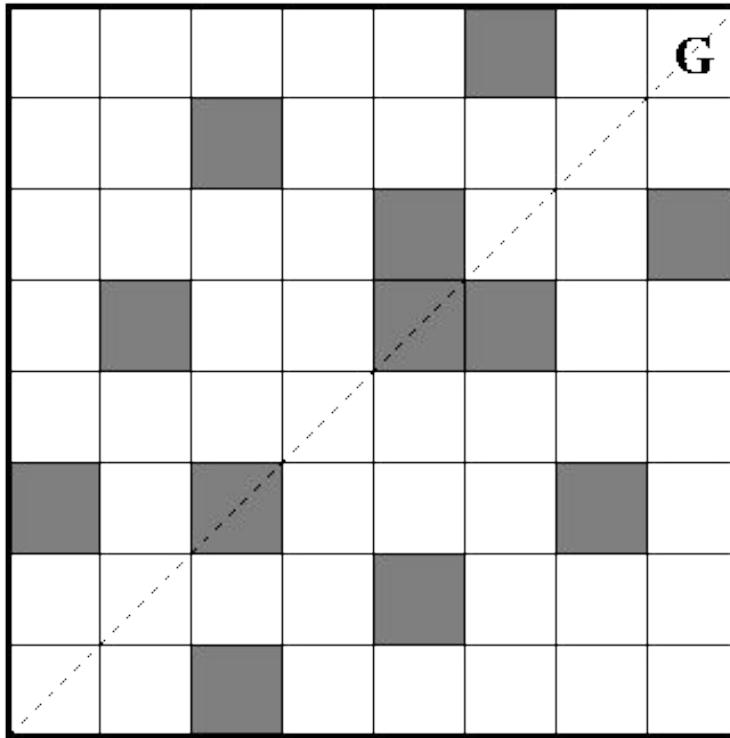


# Markov Decision Processes

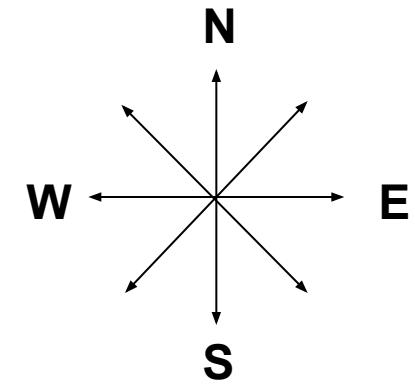
- MDP,  $M$ , is the tuple:  $M = \langle S, A, p, r \rangle$ 
  - $S$  : set of states.
  - $A$  : set of actions.
  - $p : S \times A \times S \rightarrow [0, 1]$  : probability of transition.
  - $r : S \times A \times S \rightarrow \mathbb{R}$  : expected reward.
- Policy:  $\pi : S \times A \rightarrow [0, 1]$  (can be deterministic)
- Maximize total expected reward
- Learn an *optimal* policy

# Example

## 2-D workspace



$$M = \langle S, A, p, r \rangle$$



# Robot Control

- **Input** consists of the reading of the sonars, the bump sensors, the camera, the arm position, and wheel encoder

- **State** is typically a short history of the sensor readings

- **Actions** are the torques to the motors

- **Positive rewards** on achieving the goal; **Negative rewards** for bumping into obstacles



# The Agent Learns a Policy

**Policy** at step  $t$ ,  $\pi_t$  :

a mapping from states to action probabilities

$\pi_t(s, a)$  = probability that  $a_t = a$  when  $s_t = s$

- Reinforcement learning methods specify how the agent changes its policy as a result of experience.
- Roughly, the agent's goal is to get as much reward as it can over the long run.

# Returns

Suppose the sequence of rewards after step  $t$  is :

$$r_{t+1}, r_{t+2}, r_{t+3}, \dots$$

What do we want to maximize?

We want to maximize the **return**,  $G_t$ , for each step  $t$ .

**Episodic tasks:** interaction breaks naturally into episodes, e.g., plays of a game, trips through a maze.

$$G_t = r_{t+1} + r_{t+2} + \dots + r_T,$$

where  $T$  is a final time step at which a **terminal state** is reached, ending an episode.

# Returns for Continuing Tasks

**Continuing tasks:** interaction does not have natural episodes.

**Discounted return:**

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

where  $\gamma, 0 \leq \gamma \leq 1$ , is the **discount rate**.

shortsighted  $0 \leftarrow \gamma \rightarrow 1$  farsighted

In general,

we want to maximize the **expected return**,  $E\{G_t\}$ , for each step  $t$ .

# Rewards

X	O X	O X   X	O   X   X	X   O   X	X   O   X	X   O   X	1
							-1

• 0  
• -1  
• 0  
• 0  
• 0

X	O X	O X   X	O   X   X	X   O   X	X   O   X	X   O   X	1
							0

# Value Functions

- Expected future rewards starting from a state (or state-action pair) and following policy  $\pi$

**State - value function for policy  $\pi$  :**

$$v_{\pi}(s) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

**Action - value function for policy  $\pi$  :**

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

# Value Functions

- Expected future rewards starting from a state (or state-action pair) and following policy  $\pi$

**State - value function for policy  $\pi$  :**

$$v_{\pi}(s) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

**Action - value function for policy  $\pi$  :**

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

$$v_{\pi}(s) = \sum_a \pi(a|s) q_{\pi}(s, a)$$

# Solving RL problems

- Learn an optimal *policy* – a mapping from states to actions such that no other policy has a higher long term reward

# Solving RL problems

- Learn an optimal *policy* – a mapping from states to actions such that no other policy has a higher long term reward
- Can learn such a policy directly
- Or through estimating an optimal *value* function
- Optimal Value function: The estimated long term reward that you would get starting from a state and behaving optimally

# Why Action Value functions?

- Let  $q_*(s, a)$  be the expected value of starting in state  $s$  and doing action  $a$  and behaving optimally thereafter.
- Given the optimal value function one can recover the optimal policy easily

$$\pi_*(s) = \arg \max_{a \in \mathcal{A}(s)} q_*(s, a)$$