Though questions

- Objective:
  - To show to me you have read the textbook and thought about the material you have read
  - To practice thinking of original questions and answers
  - To practice writing
- Thought question #1 is worth 1%
- Thought question #2 is worth 2%
- Thought question #3 is worth 2%
- One page maximum, submitted via Gradescope
  - Assignment late policy applies

Though Questions Nuts and Bolts

- Pose one question about the readings up to and including chapter 3
  - This question should not be answered in the textbook, lecture or assignments
  - If I can say: “the answer is on page xx”, thats bad!
- Provide on potential answer to your question:
  - There might be many possible answers
  - The question might be hard, so your answer simply outlines how to think about the question
- If you ask a simple question, then we expect your answer to be of higher quality

Though questions marking

- This is intended to be opened and free form!
- You will be marked on the originality of your question/ answer
- You will be marked on the quality and clarity of your writing:
  - Write in complete sentences (no bullet point lists)
  - Write paragraphs with topic sentences
  - Proof read for grammar and spelling issues

Thought question #1 is a learning experience, that is why its only worth 1%

Bad Though Questions

- Trivial questions: e.g., is RL a type of machine learning?
- Assignment-like questions: e.g., what is the Bellman equation for action values?
- Questions covered in the textbook: e.g.,
  - How is TD related to Neuroscience?
  - How can we deal with infinite or continuous state spaces?
- Commentary on course material:
  - e.g., Chapter 3 is not practical, I think we should skip it
- These are all **bad questions!**

Good Though Questions

- Show me you have read, and thought about the material
  - Go beyond what is covered in the textbook

- E.g., Natural learning systems like young animals and babies seem to learn a lot about the world through what we might call curiosity and play. How might we encode such mechanisms in an intelligence system?

Good Though Questions

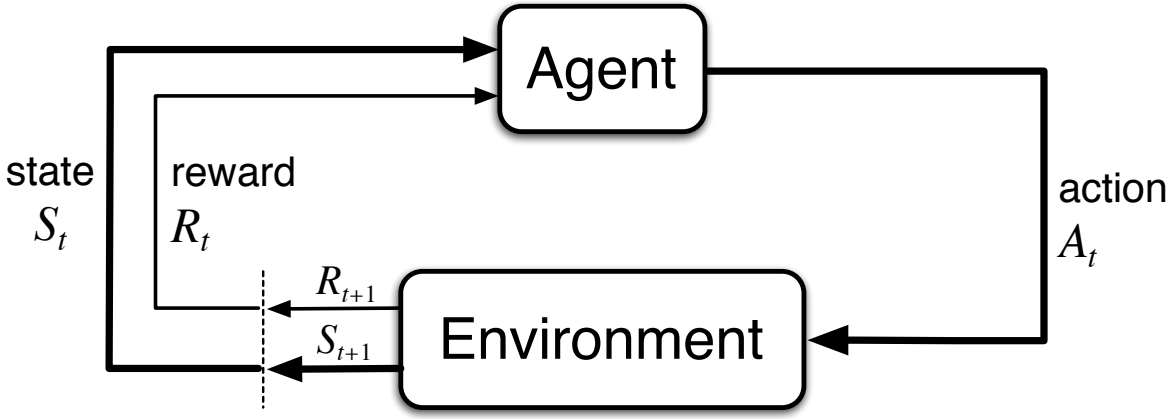
- Deciding what to learn about seems like a key thing for intelligent systems. Even if we assume some scalar external reward, a learning system will likely sometimes what to do other things, such as achieve subgoals or learn a skill. These things might be useful later. For example, learning the fastest route out of building might not generate much reward now, but would be useful in the event of a fire. How can we formalize the notions of subgoals and skills in the reinforcement learning framework? And how might a learning system decide which of the infinite possible subgoals and skills to learn about?

Chapter 3: The Reinforcement Learning Problem (Markov Decision Processes, or MDPs)

Objectives of this chapter:

- present Markov decision processes—an idealized form of the AI problem for which we have precise theoretical results
- introduce key components of the mathematics: value functions and Bellman equations

The Agent-Environment Interface



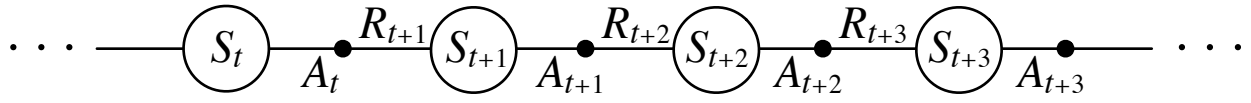
Agent and environment interact at discrete time steps: $t = 0, 1, 2, 3, \ldots$

Agent observes state at step t: $S_t \in \mathcal{S}$

produces action at step t : $A_t \in \mathcal{A}(S_t)$

gets resulting reward: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$

and resulting next state: $S_{t+1} \in \mathcal{S}^+$



Markov Decision Processes

- If a reinforcement learning task has the Markov Property, it is basically a **Markov Decision Process (MDP)**.
- If state and action sets are finite, it is a **finite MDP**.
- To define a finite MDP, you need to give:
  - **state and action sets**
  - one-step "dynamics"

$$p(s', r|s, a) = \mathbf{Pr}\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\}$$

Markov Decision Processes

- If a reinforcement learning task has the Markov Property, it is basically a **Markov Decision Process (MDP)**.
- If state and action sets are finite, it is a **finite MDP**.
- To define a finite MDP, you need to give:
  - **state and action sets**
  - one-step "dynamics"

$$p(s', r|s, a) = \mathbf{Pr}\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\}$$

  - there is also:

Markov Decision Processes

- If a reinforcement learning task has the Markov Property, it is basically a **Markov Decision Process (MDP)**.
- If state and action sets are finite, it is a **finite MDP**.
- To define a finite MDP, you need to give:
  - **state and action sets**
  - one-step "dynamics"

$$p(s', r|s, a) = \mathbf{Pr}\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\}$$

  - there is also:

$$p(s'|s, a) \doteq \Pr\{S_{t+1} = s' \mid S_t = s, A_t = a\} = \sum_{r \in \mathcal{R}} p(s', r|s, a)$$

Markov Decision Processes

- If a reinforcement learning task has the Markov Property, it is basically a **Markov Decision Process (MDP)**.
- If state and action sets are finite, it is a **finite MDP**.
- To define a finite MDP, you need to give:
  - **state and action sets**
  - one-step "dynamics"

$$p(s', r|s, a) = \mathbf{Pr}\{S_{t+1}=s', R_{t+1}=r \mid S_t=s, A_t=a\}$$

  - there is also:

$$p(s'|s, a) \doteq \Pr\{S_{t+1}=s' \mid S_t=s, A_t=a\} = \sum_{r\in\mathcal{R}} p(s', r|s, a)$$

$$r(s, a) \doteq \mathbb{E}[R_{t+1} \mid S_t=s, A_t=a] = \sum_{r\in\mathcal{R}} r \sum_{s'\in\mathcal{S}} p(s', r|s, a)$$

The Agent Learns a Policy

Policy at step t = π_t =

a mapping from states to action probabilities

$\pi_t(a \mid s)$ = probability that $A_t = a$ when $S_t = s$

Special case - *deterministic policies*:

$\pi_t(s)$ = the action taken with prob=1 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.

The Meaning of Life
(goals, rewards, and returns)

Return

Suppose the sequence of rewards after step t is:

$$R_{t+1}, R_{t+2}, R_{t+3}, \ldots$$

What do we want to maximize?

At least three cases, but in all of them,

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

- Total reward, G_t = sum of all future reward in the episode
- Discounted reward, G_t = sum of all future *discounted* reward
- Average reward, G_t = average reward per time step

Episodic Tasks

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

In episodic tasks, we almost always use simple *total reward*:

$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T ,$$

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

Continuing Tasks

Continuing tasks: interaction does not have natural episodes, but just goes on and on...

In this class, for continuing tasks we will always use *discounted return*:

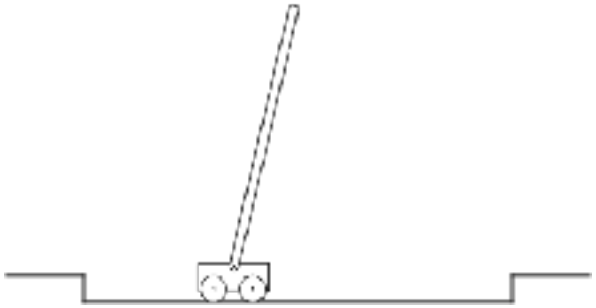
$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

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

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

Typically, $\gamma = 0.9$

An Example: Pole Balancing



Avoid **failure:** the pole falling beyond a critical angle or the cart hitting end of track

As an **episodic task** where episode ends upon failure:

reward = +1 for each step before failure

$\Rightarrow$ return = number of steps before failure

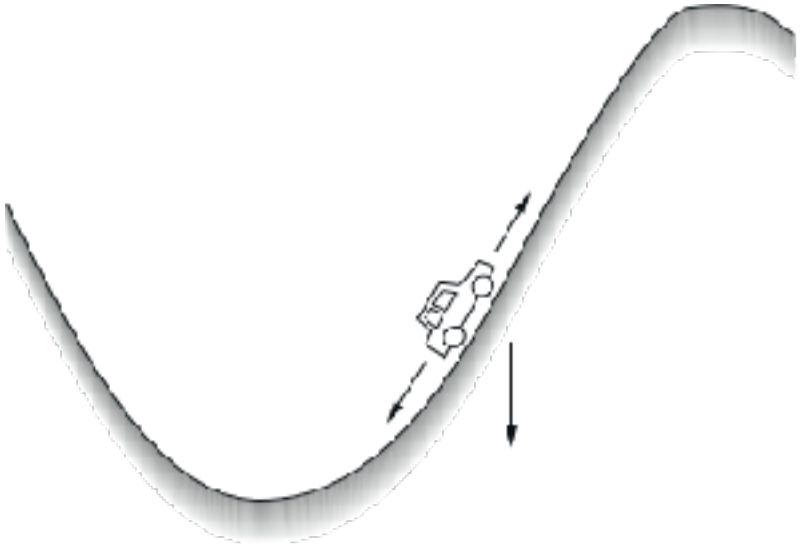
As a **continuing task** with discounted return:

reward = −1 upon failure; 0 otherwise

$\Rightarrow$ return = $-\gamma^k$, for k steps before failure

In either case, return is maximized by avoiding failure for as long as possible.

Another Example: Mountain Car



Get to the top of the hill
as quickly as possible.

reward = –1 for each step where **not** at top of hill

$\Rightarrow$ return = – number of steps before reaching top of hill

Return is maximized by minimizing
number of steps to reach the top of the hill.

An Abstract and Flexible Framework

Actions could be:

- Low-level voltages applied to motors
- High-level decisions, like whether to go to school
- Could be what the agent chooses to think about

>>decisions we learn to make

States could be:

- Low-level sensor readings
- Symbolic descriptions of objects in a room
- Might include memory, or encode mental states

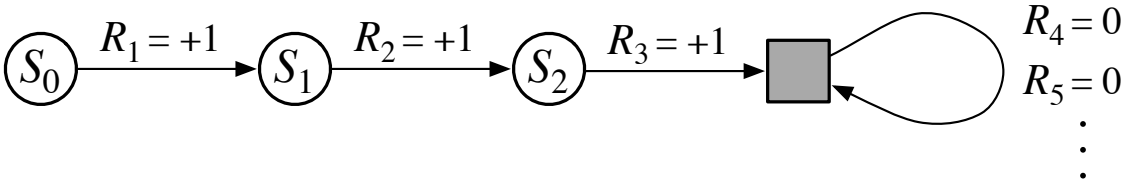
>> anything we might know that is useful for making decisions

Agent-environment Boundary

- What is part of the agent, what is part of the environment?
- Motors, mechanical linkages, sensors are usually considered part of the environment
- Consider a person or animal:
  - Muscles, skeleton, sense organs are part of the env.
  - The rewards computed inside your body are outside the agent
- An RL agent is not necessarily a *whole* animal or robot.
- Anything that cannot be changed arbitrary by the agent is considered outside and thus part of the environment
- The environment is not necessarily unknown to the agent, only incompletely controllable.

A Trick to Unify Notation for Returns

- In episodic tasks, we number the time steps of each episode starting anew from zero.
- We usually do not have to distinguish between episodes, so instead of writing $S_{t,j}$ for states in episode j, we write just S_t
- Think of each episode as ending in an absorbing state that always produces reward of zero:



- We can cover <u>all</u> cases by writing $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$,

where γ can be 1 only if a zero reward absorbing state is always reached.

Goals and Rewards

- Is a scalar reward signal an adequate notion of a goal?—maybe not, but it is surprisingly flexible.
- A goal should specify **what** we want to achieve, not **how** we want to achieve it.
- A goal must be outside the agent's direct control—thus outside the agent.
- The agent must be able to measure success:
  - explicitly;
  - frequently during its lifespan.

The reward hypothesis

- That all of what we mean by goals and purposes can be well thought of as the maximization of the cumulative sum of a received scalar signal (reward)

- A sort of *null hypothesis*.
  - Probably ultimately wrong, but so simple we have to disprove it before considering anything more complicated

Rewards and returns

- The objective in RL is to maximize long-term future reward
- That is, to choose A_t so as to maximize $R_{t+1}, R_{t+2}, R_{t+3}, \ldots$
- But what exactly should be maximized?
- The discounted *return* at time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

the *discount rate*

$$\gamma \in [0, 1)$$

γ	Reward sequence	Return
0.5(or any)	1 0 0 0…	
0.5	0 0 2 0 0 0…	
0.9	0 0 2 0 0 0…	
0.5	-1 2 6 3 2 0 0 0…	

Rewards and returns

- The objective in RL is to maximize long-term future reward
- That is, to choose A_t so as to maximize $R_{t+1}, R_{t+2}, R_{t+3}, \ldots$
- But what exactly should be maximized?
- The discounted *return* at time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

the *discount rate*

$$\gamma \in [0, 1)$$

γ	Reward sequence	Return
0.5(or any)	1 0 0 0…	1
0.5	0 0 2 0 0 0…	
0.9	0 0 2 0 0 0…	
0.5	-1 2 6 3 2 0 0 0…	

Rewards and returns

- The objective in RL is to maximize long-term future reward
- That is, to choose A_t so as to maximize $R_{t+1}, R_{t+2}, R_{t+3}, \ldots$
- But what exactly should be maximized?
- The discounted *return* at time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

the *discount rate*

$$\gamma \in [0, 1)$$

γ	Reward sequence	Return
0.5(or any)	1 0 0 0…	1
0.5	0 0 2 0 0 0…	0.5
0.9	0 0 2 0 0 0…	
0.5	-1 2 6 3 2 0 0 0…	

Rewards and returns

- The objective in RL is to maximize long-term future reward
- That is, to choose A_t so as to maximize $R_{t+1}, R_{t+2}, R_{t+3}, \ldots$
- But what exactly should be maximized?
- The discounted *return* at time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

the *discount rate*

$$\gamma \in [0, 1)$$

γ	Reward sequence	Return
0.5(or any)	1 0 0 0…	1
0.5	0 0 2 0 0 0…	0.5
0.9	0 0 2 0 0 0…	1.62
0.5	-1 2 6 3 2 0 0 0…	

Rewards and returns

- The objective in RL is to maximize long-term future reward
- That is, to choose A_t so as to maximize $R_{t+1}, R_{t+2}, R_{t+3}, \ldots$
- But what exactly should be maximized?
- The discounted *return* at time t:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

the *discount rate*

$$\gamma \in [0, 1)$$

γ	Reward sequence	Return
0.5(or any)	1 0 0 0…	1
0.5	0 0 2 0 0 0…	0.5
0.9	0 0 2 0 0 0…	1.62
0.5	-1 2 6 3 2 0 0 0…	2

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 =$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 = 0$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 =$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 =$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 =$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 =$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $G =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

$$G = \frac{1}{1 - \gamma}$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

$$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

$$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

$$G = \frac{1}{1 - \gamma} = 2$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13, \text{ and so on, all 13s}$$

  $$G_2 = 26$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 = 26 \quad G_1 =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 = 26 \quad G_1 = 26$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 = 26 \quad G_1 = 26 \quad G_0 =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13, \text{ and so on, all 13s}$$

  $$G_2 = 26 \quad G_1 = 26 \quad G_0 = 14$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 = 26 \quad G_1 = 26 \quad G_0 = 14$

- And if $\gamma = 0.9$?

  $G_1 =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13, \text{ and so on, all 13s}$$

  $$G_2 = 26 \quad G_1 = 26 \quad G_0 = 14$$

- And if $\gamma = 0.9$?

  $$G_1 = 130$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16,$ then zeros for R_5 and later

- What are the following returns?

  $G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13,$ and so on, all 13s

  $G_2 = 26 \quad G_1 = 26 \quad G_0 = 14$

- And if $\gamma = 0.9$?

  $G_1 = 130 \quad G_0 =$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \qquad \gamma \in [0, 1)$$

- Suppose $\gamma = 0.5$ and the reward sequence is

  $$R_1 = 1, R_2 = 6, R_3 = -12, R_4 = 16, \text{ then zeros for } R_5 \text{ and later}$$

- What are the following returns?

  $$G_4 = 0 \quad G_3 = 16 \quad G_2 = -4 \quad G_1 = 4 \quad G_0 = 3$$

- Suppose $\gamma = 0.5$ and the reward sequence is all 1s.

  $$G = \frac{1}{1-\gamma} = 2$$

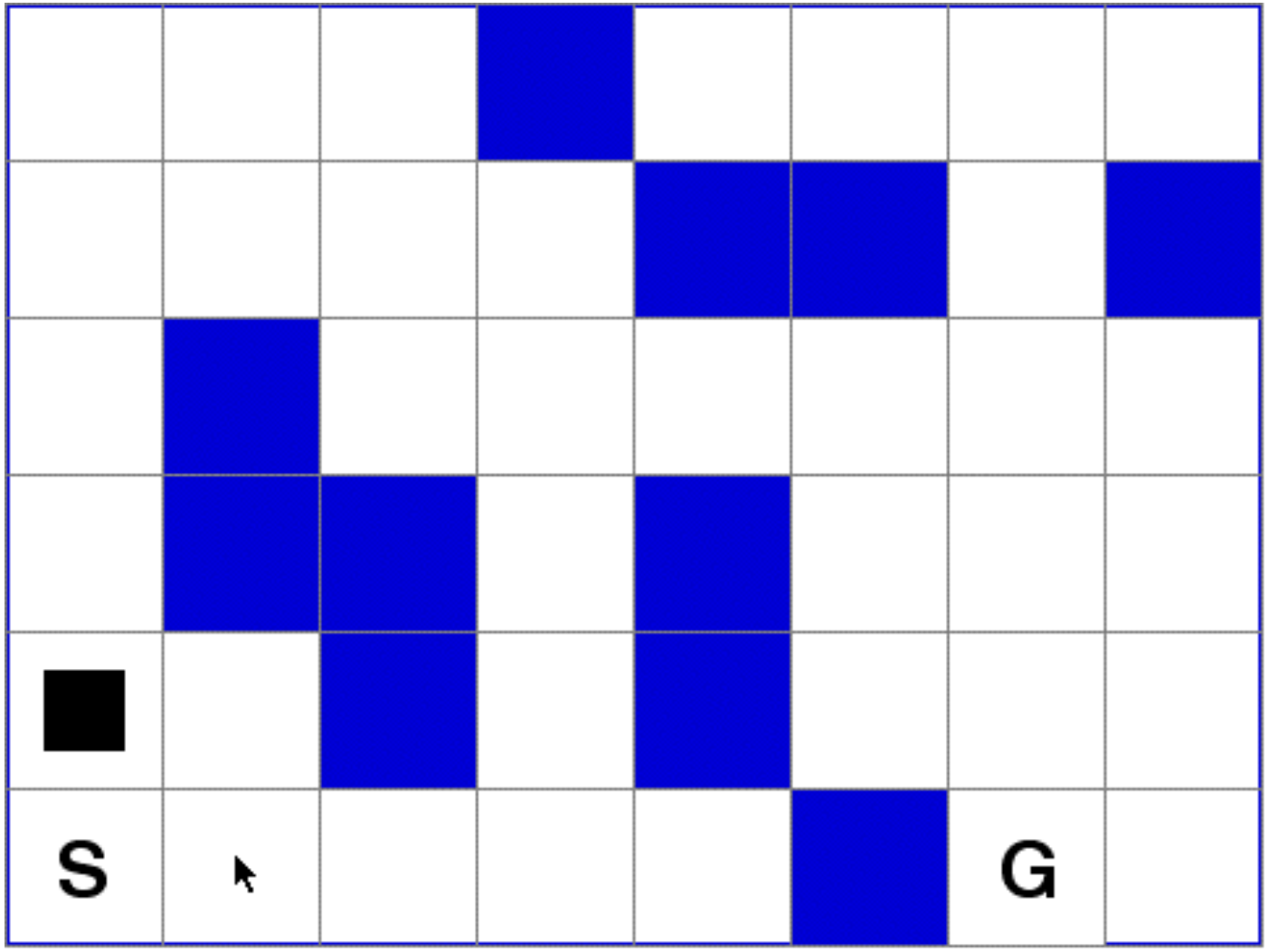
- Suppose $\gamma = 0.5$ and the reward sequence is

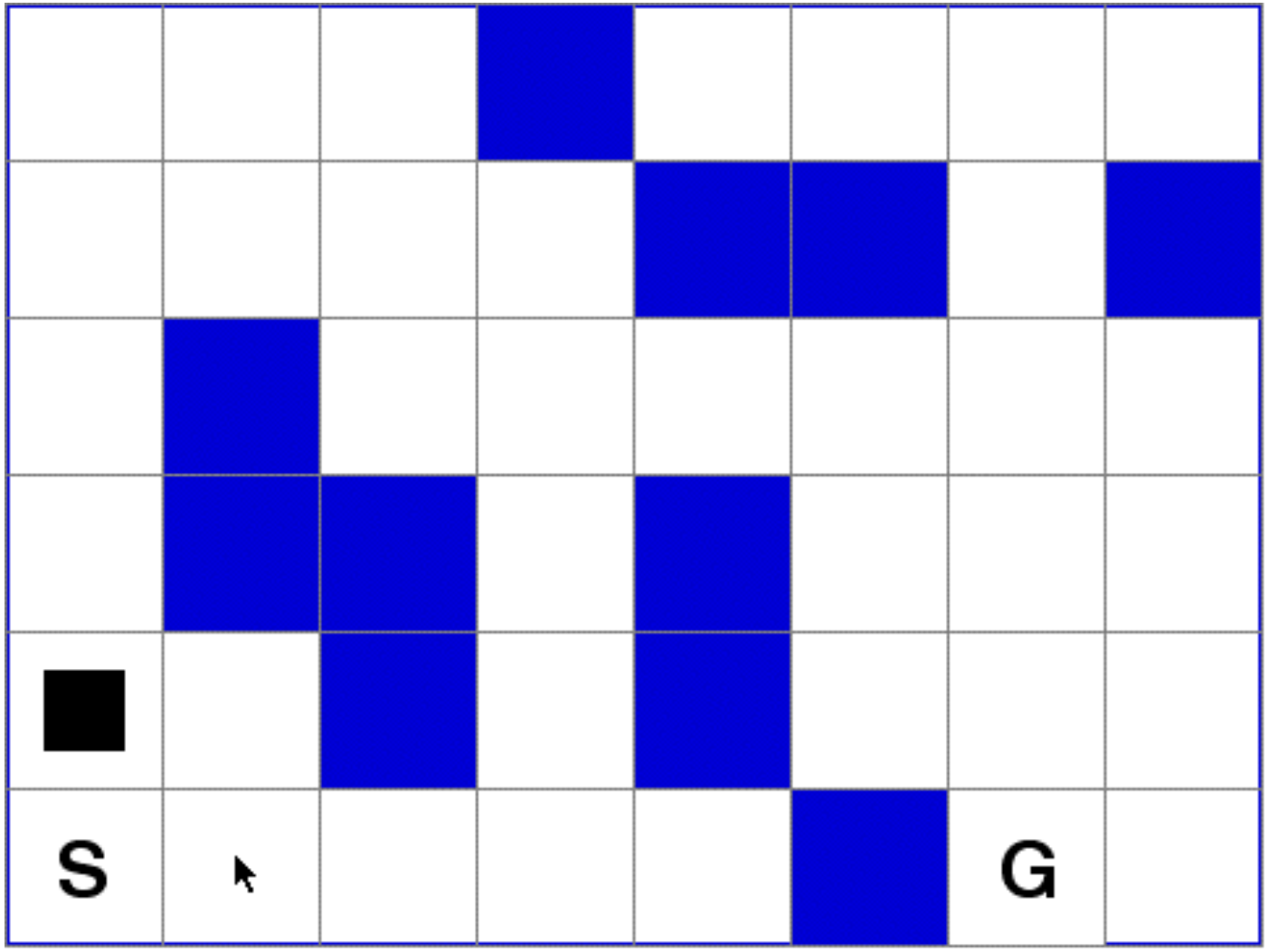
  $$R_1 = 1, R_2 = 13, R_3 = 13, R_4 = 13, \text{ and so on, all 13s}$$

  $$G_2 = 26 \quad G_1 = 26 \quad G_0 = 14$$

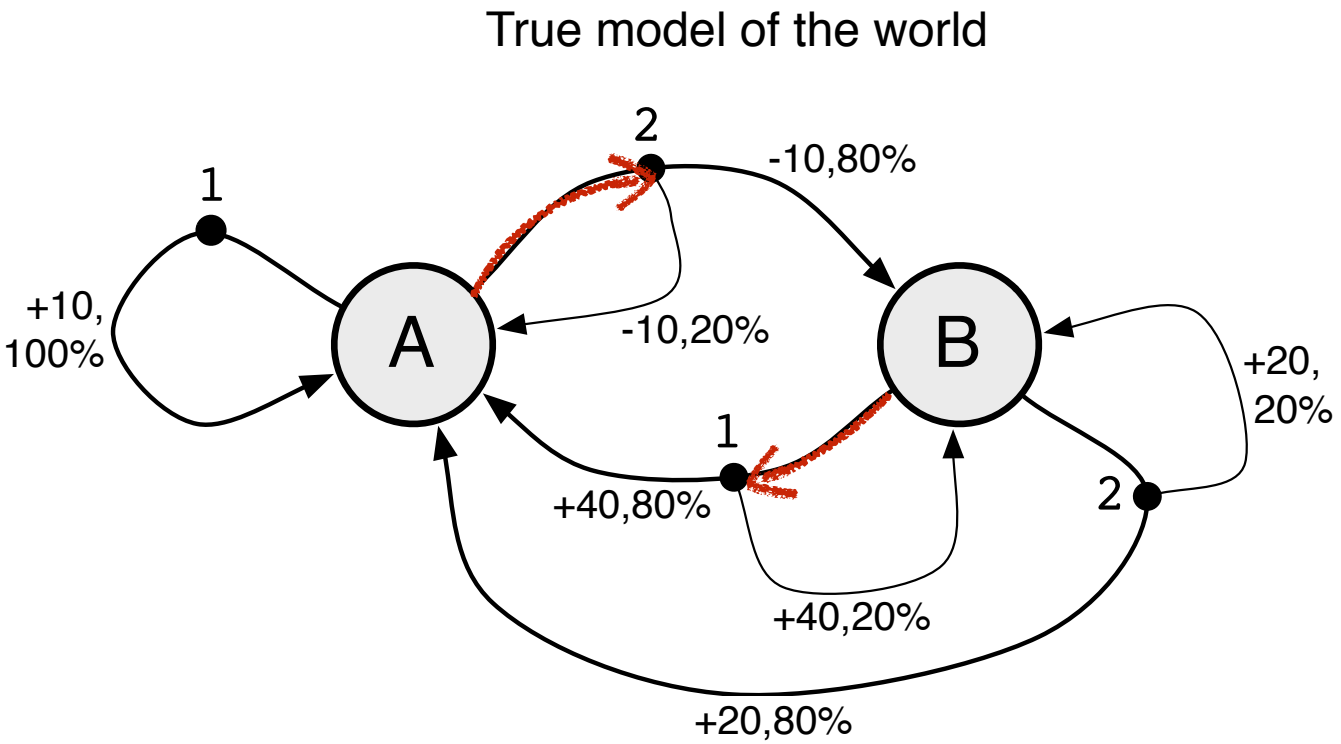
- And if $\gamma = 0.9$?

  $$G_1 = 130 \quad G_0 = 118$$

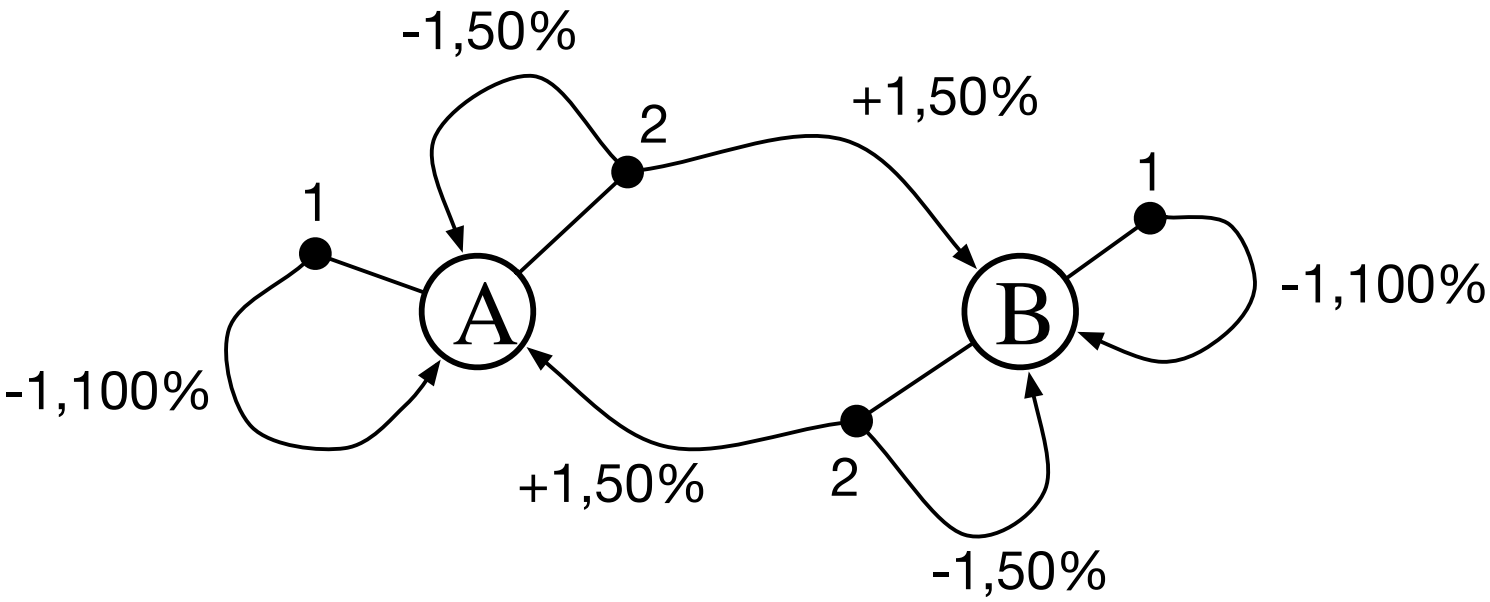

S
G
Go
Step
Policy
Values
Faster
Slower
0|1|0


S
G
Go
Step
Policy
Values
Faster
Slower
0|1|0

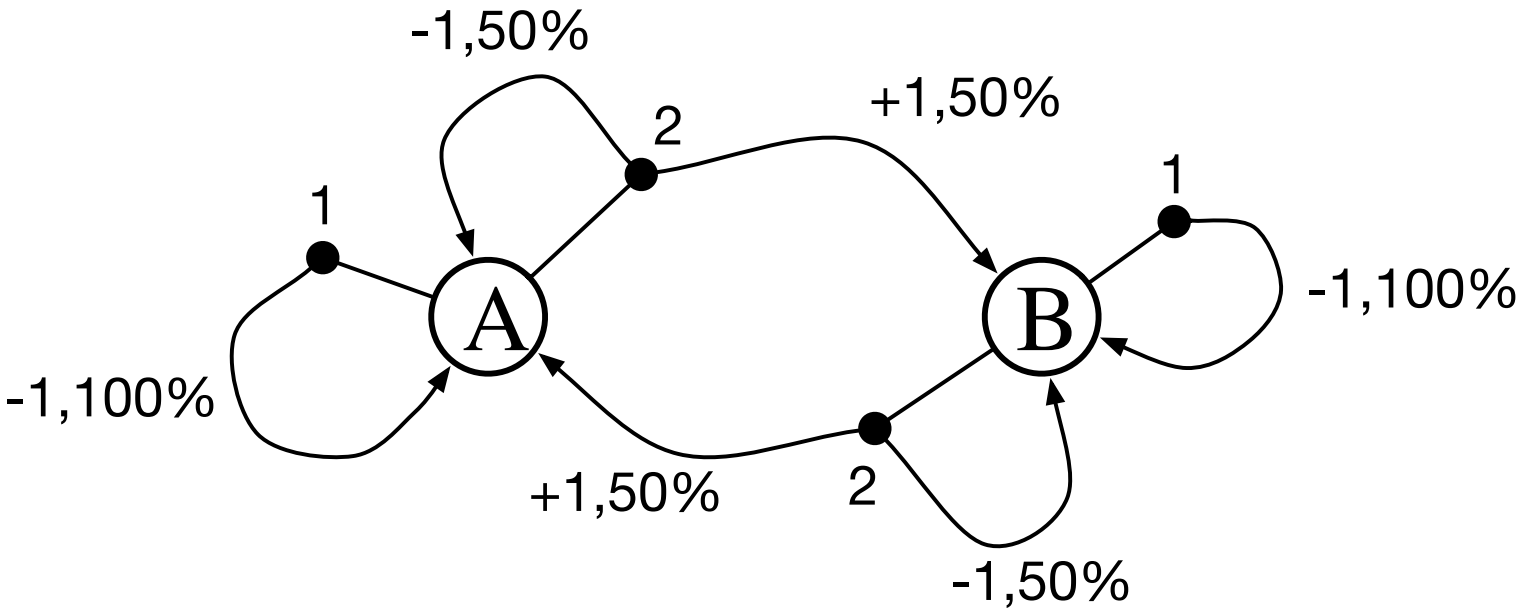
Revisiting our Simple MDP



An even simpler MDP



Exercise: write out the table of p(s',r | s,a), p(s' | s,a), and r(s,a)



Values are *expected* returns

Values are *expected* returns

- The value of a state, given a policy:

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

$$v_*(s) = \max_\pi v_\pi(s) \qquad v_* : \mathcal{S} \to \Re$$

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

$$v_*(s) = \max_\pi v_\pi(s) \qquad v_* : \mathcal{S} \to \Re$$

- The optimal value of a state-action pair:

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

$$v_*(s) = \max_\pi v_\pi(s) \qquad v_* : \mathcal{S} \to \Re$$

- The optimal value of a state-action pair:

$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad q_* : \mathcal{S} \times \mathcal{A} \to \Re$$

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

$$v_*(s) = \max_\pi v_\pi(s) \qquad v_* : \mathcal{S} \to \Re$$

- The optimal value of a state-action pair:

$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad q_* : \mathcal{S} \times \mathcal{A} \to \Re$$

- Optimal policy: π_* is an optimal policy if and only if

Values are *expected* returns

- The value of a state, given a policy:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s]$$

- The value of a state-action pair, given a policy:

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

- The optimal value of a state:

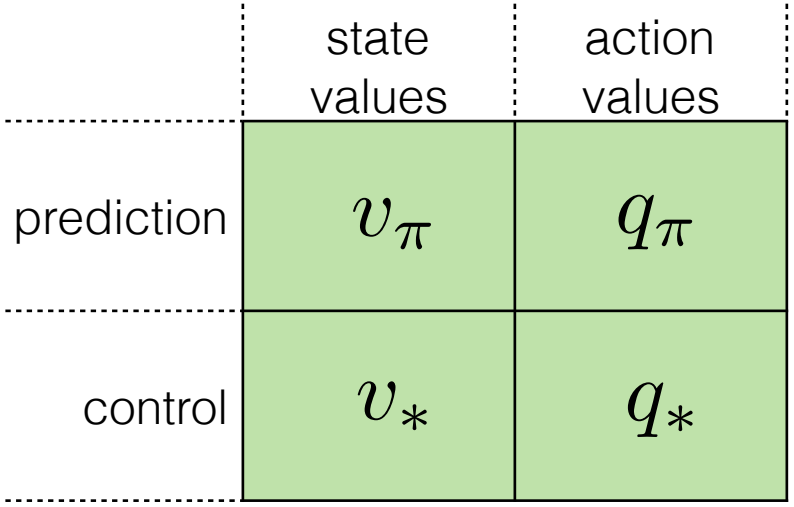
$$v_*(s) = \max_\pi v_\pi(s) \qquad v_* : \mathcal{S} \to \Re$$

- The optimal value of a state-action pair:

$$q_*(s,a) = \max_\pi q_\pi(s,a) \qquad q_* : \mathcal{S} \times \mathcal{A} \to \Re$$

- Optimal policy: π_* is an optimal policy if and only if
  - it is *greedy* wrt q_*

4 value functions



- All theoretical objects, mathematical ideals (expected values)

- Distinct from their estimates:

$$V_t(s) \qquad Q_t(s,a)$$

What we learned so far

- Finite Markov decision processes!
  - States, actions, and rewards
  - And returns
  - And time, discrete time
  - They capture essential elements of life — state, causality
- The goal is to optimize expected returns
  - returns are *discounted sums* of *future* rewards
- Thus we are interested in *values* — expected returns
- There are four value *functions*
  - state vs state-action values
  - values for a policy vs values for the optimal policy

Value Functions

- The **value of a state** is the expected return starting from that state; depends on the agent's policy:

State - value function for policy π :

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

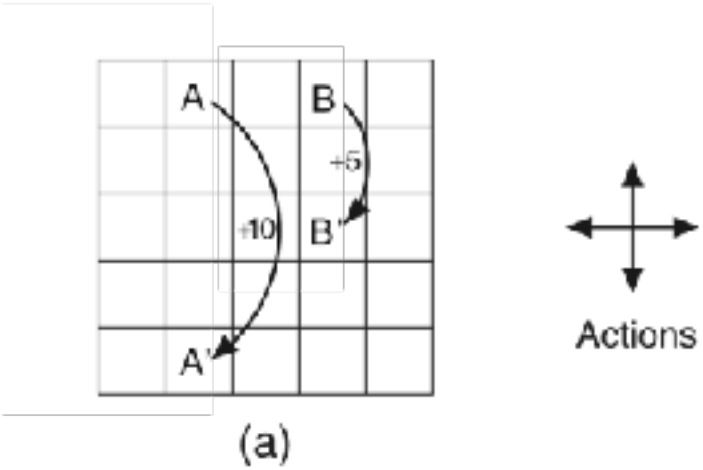
- The **value of an action (in a state)** is the expected return starting after taking that action from that state; depends on the agent's policy:

Action - value function for policy π :

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

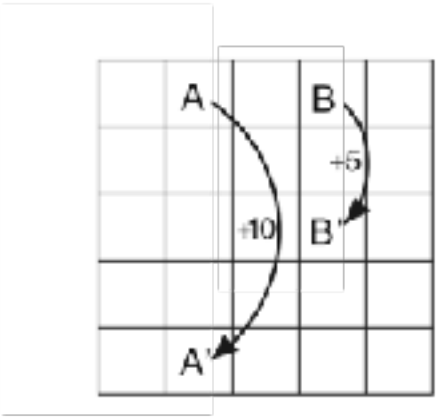
Gridworld

- Actions: `north`, `south`, `east`, `west`; deterministic.
- If would take agent off the grid: no move but reward = –1
- Other actions produce reward = 0, except actions that move agent out of special states A and B as shown.



Gridworld

- Actions: `north`, `south`, `east`, `west`; deterministic.
- If would take agent off the grid: no move but reward = –1
- Other actions produce reward = 0, except actions that move agent out of special states A and B as shown.



(a)

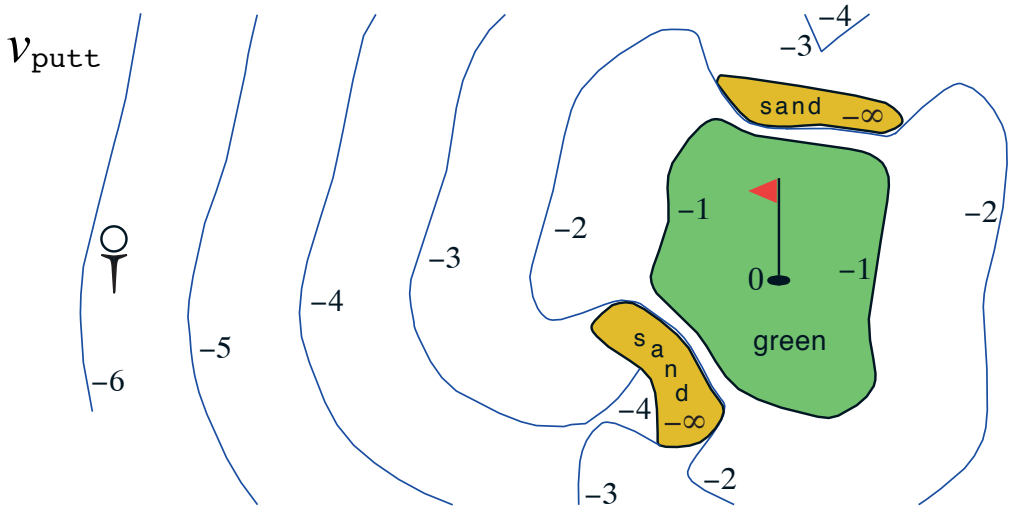
3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

State-value function for equiprobable random policy;
$\gamma = 0.9$

Golf

- State is ball location
- Reward of –1 for each stroke until the ball is in the hole
- Value of a state?
- Actions:
  - `putt` (use putter)
  - `driver` (use driver)
- `putt` succeeds anywhere on the green



Bellman Equation for a Policy π

- Value functions satisfy a recursive relationship similar to the return
- The Bellman Equation describes an interesting consistency condition that holds between the value of a state and the value of it's possible successor states
  - How $v_\pi(s)$ relates to $v_\pi(s')$
  - True for all π
- $v_\pi(s)$ is the unique solution to the Bellman Equation
- **Why they matter:** we will use the Bellman equations to design algorithms to compute and estimate value functions

Bellman Equation for a Policy π

The basic idea:

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

So:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \end{aligned}$$

Or, without the expectation operator:

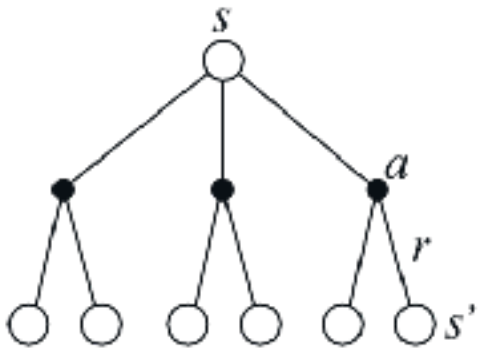
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big]$$

More on the Bellman Equation

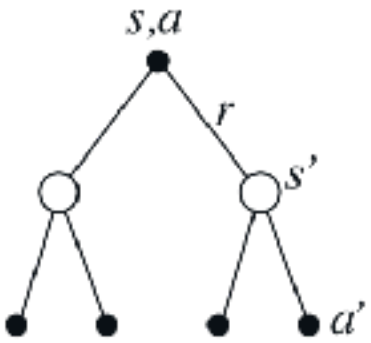
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \Big[r + \gamma v_\pi(s') \Big]$$

This is a set of equations (in fact, linear), one for each state.
The value function for π is its unique solution.

Backup diagrams:



for v_π



for q_π

Value Functions x 4

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

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

Bellman Equations x 4

Bellman Equations x 4

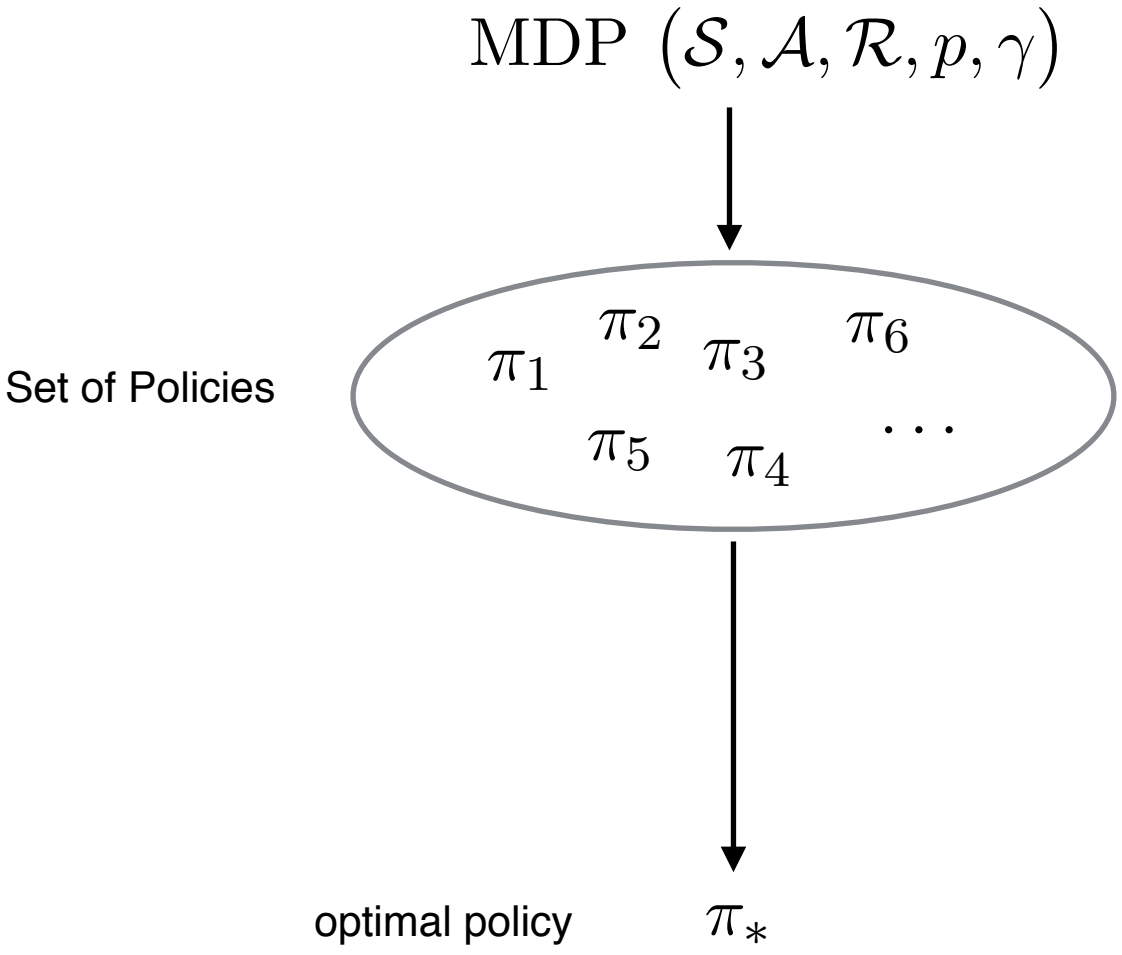
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

Bellman Equations x 4

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

$$q_\pi(s, a)$$

Overall Goal



Optimal Policy: Example 1

Tim and Bert are both given the option to bet on the outcome of a dice roll. The two betting options they are given are:

a) if the outcome of the dice roll is even, you win 10$

b) if the outcome of the dice roll is 1, you win 10$

Tim chooses option a; Bert chooses option b.

Who made the best decision?

Optimal Policy: Example 1

Tim and Bert are both given the option to bet on the outcome of a dice roll. The two betting options they are given are:

a) if the outcome of the dice roll is even, you win 10$

b) if the outcome of the dice roll is 1, you win 10$

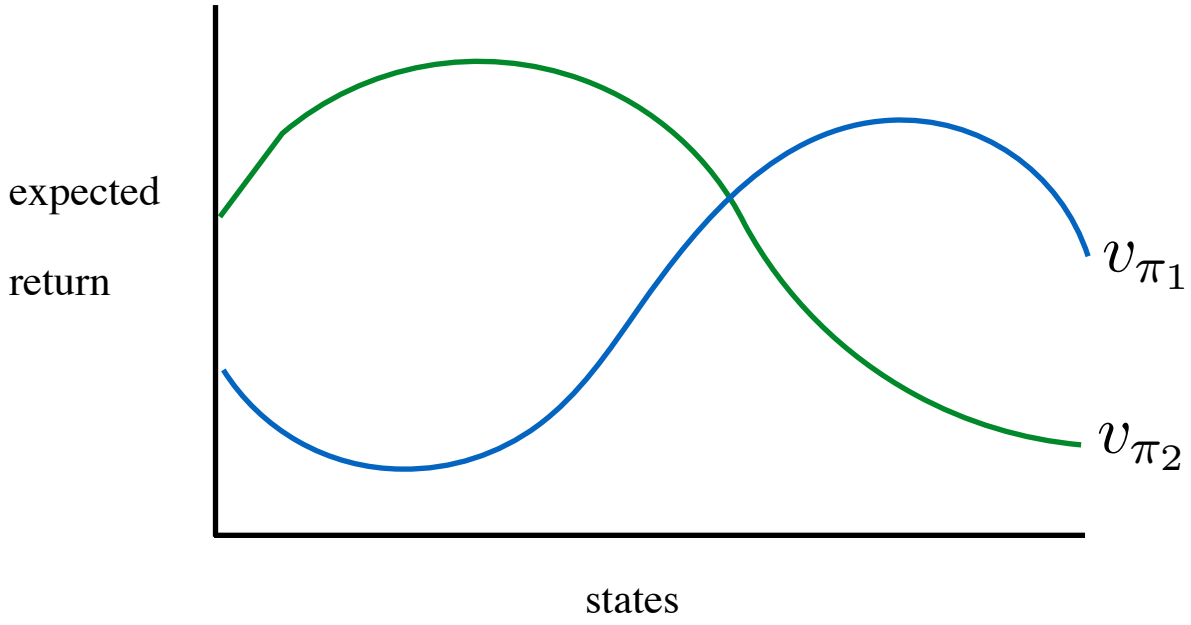
Tim chooses option a; Bert chooses option b.

Who made the best decision?

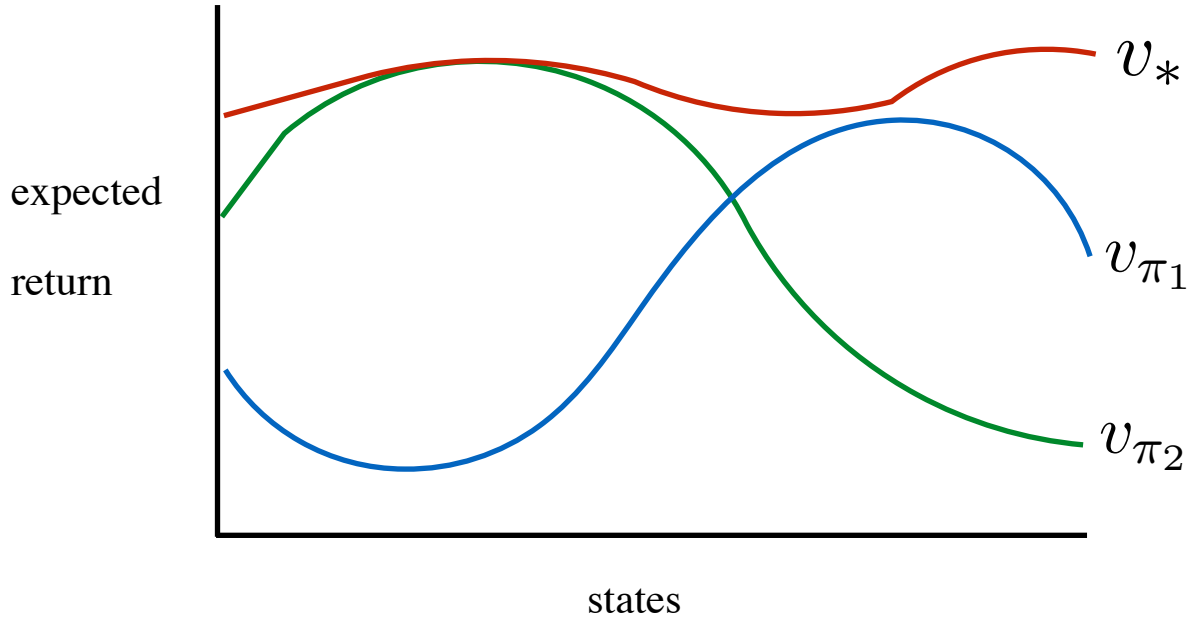
The dice is rolled and the outcome is 1.

In hindsight, who made the best decision?

- A policy does not fully control whether it receives a good return or not, it can only maximize its chances on a good return.
- The optimal policy will not always produce the highest return, but it has the highest return in expectation.
- When an experiment is run many times, the optimal policy produces the highest average return.



π_*



For an MDP with full observability there always exists a policy that is at least as good as all other policies for each state. This is called the optimal policy π_* .

Optimal Value Functions

- For finite MDPs, policies can be **partially ordered**:

$$\pi \geq \pi' \quad \text{if and only if} \quad v_\pi(s) \geq v_{\pi'}(s) \quad \text{for all } s \in \mathcal{S}$$

- There are always one or more policies that are better than or equal to all the others. These are the **optimal policies**. We denote them all π_*.

- Optimal policies share the same **optimal state-value function**:

$$v_*(s) = \max_\pi v_\pi(s) \quad \text{for all} \quad s \in \mathcal{S}$$

- Optimal policies also share the same **optimal action-value function**:

$$q_*(s,a) = \max_\pi q_\pi(s,a) \quad \text{for all} \quad s \in \mathcal{S} \text{ and } a \in \mathcal{A}$$

This is the expected return for taking action a in state s and thereafter following an optimal policy.

Value Functions x 4

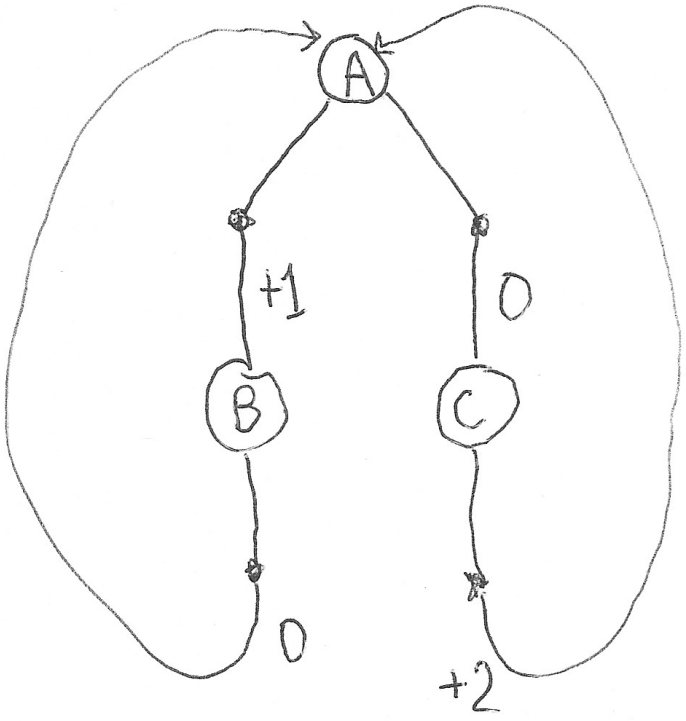
$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

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

$$v_*(s) \doteq \max_\pi v_\pi(s)$$

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a)$$

optimal policy example



What policy is optimal?

A: left

B: Right C: other

If $\gamma = 0$?

If $\gamma = .99$

If $\gamma = \frac{1}{2}$?

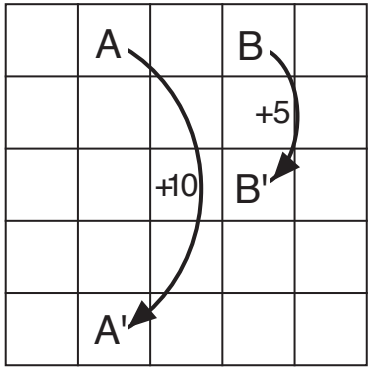
- How can we prove that both are optimal under $\gamma = 0.5$?

Why Optimal State-Value Functions are Useful

Any policy that is greedy with respect to v_* is an optimal policy.

Therefore, given v_*, one-step-ahead search produces the long-term optimal actions.

E.g., back to the gridworld:



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

b) v_*

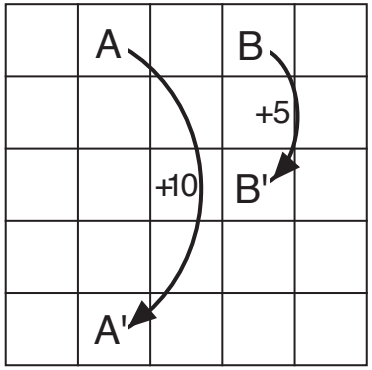
c) π_*

Why Optimal State-Value Functions are Useful

Any policy that is greedy with respect to v_* is an optimal policy.

Therefore, given v_*, one-step-ahead search produces the long-term optimal actions.

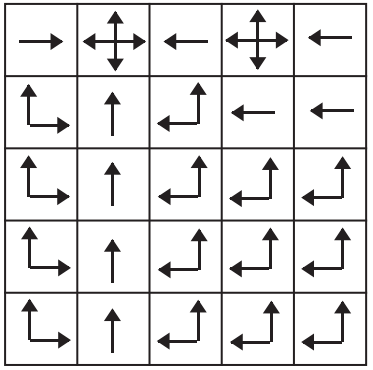
E.g., back to the gridworld:



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

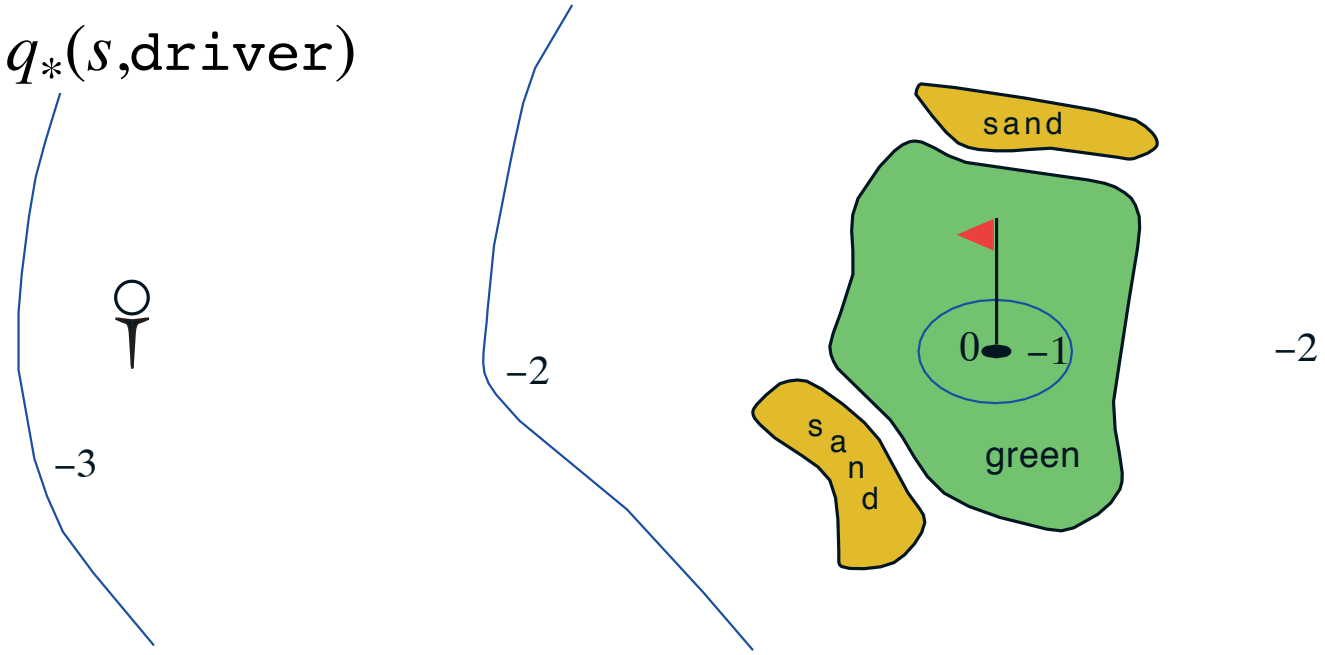
b) v_*



c) π_*

Optimal Value Function for Golf

- We can hit the ball farther with `driver` than with `putter`, but with less accuracy
- q_* (s,`driver`) gives the value or using `driver` first, then using whichever actions are best



What About Optimal Action-Value Functions?

Given q_*, the agent does not even have to do a one-step-ahead search:

$$\pi_*(s) = \arg\max_a q_*(s,a)$$

Bellman Optimality Equation for v_*

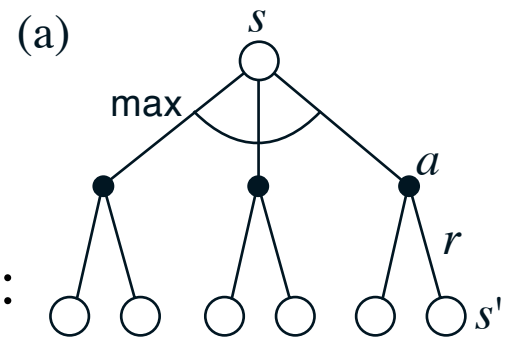
- Because v_* is the value function for a policy it must satisfy the self-consistency condition given by the Bellman equation
  - How $v_*(s)$ relates to $v_*(s')$
- v_* is special and its Bellman equation can be written without any reference to any specific policy
- **Why they matter:** we will use the Bellman Optimality equations to design algorithms to compute and estimate value functions

Bellman Optimality Equation for v_*

The value of a state under an optimal policy must equal the expected return for the best action from that state:

$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].
\end{aligned}
$$

The relevant backup diagram:

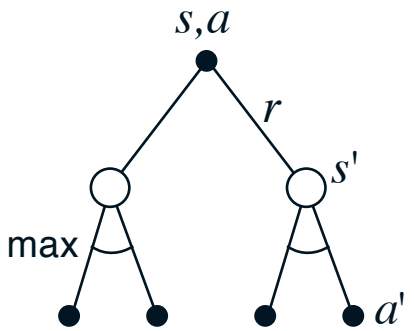


v_* is the unique solution of this system of nonlinear equations.

Bellman Optimality Equation for q_*

$$
\begin{aligned}
q_*(s,a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \;\middle|\; S_t = s, A_t = a\right] \\
&= \sum_{s',r} p(s', r|s, a)\left[r + \gamma \max_{a'} q_*(s', a')\right].
\end{aligned}
$$

The relevant backup diagram:



q_* is the unique solution of this system of nonlinear equations.

Bellman Equations x 4

Bellman Equations x 4

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

Bellman Equations x 4

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

$$q_\pi(s, a)$$

Bellman Equations x 4

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

$$q_\pi(s, a)$$

$$v_*(s) = \max_a \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma v_*(s') \big]$$

Bellman Equations x 4

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

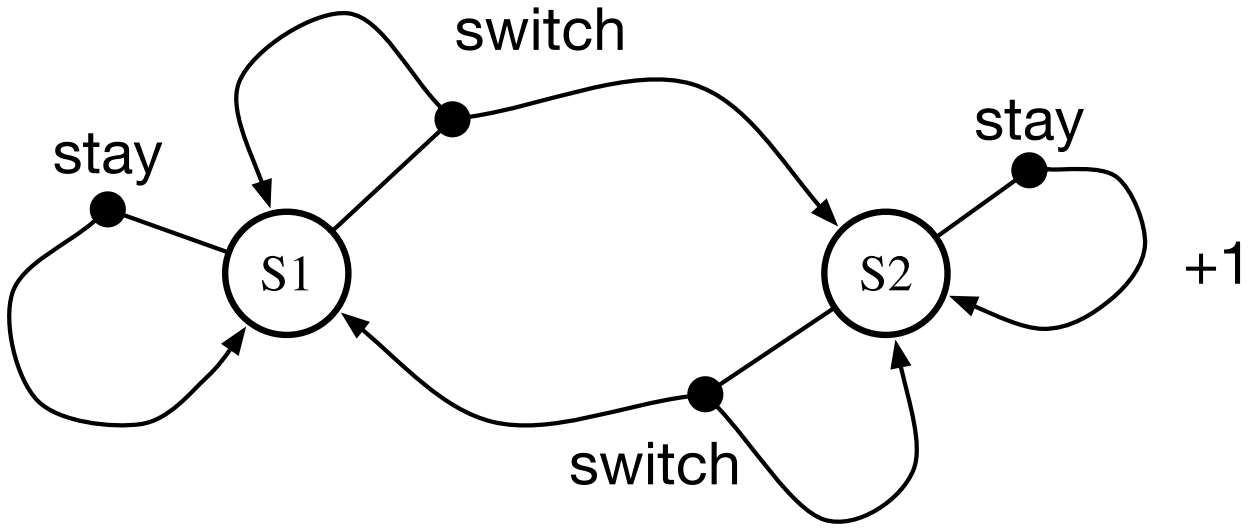
$$q_\pi(s, a)$$

$$v_*(s) = \max_a \sum_{s',r} p(s', r \,|\, s, a) \big[r + \gamma v_*(s') \big]$$

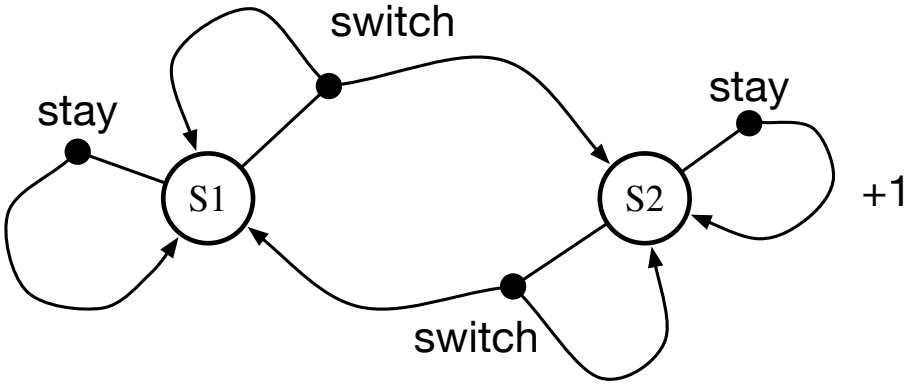
$$q_*(s, a) = \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma \max_{a'} q_*(s', a') \Big]$$

Exercise: computing value functions

Consider the MDP in the figure below. There are two states, *S1* and *S2*, and two actions, *switch* and *stay*. The *switch* action takes the agent to the other state with probability 0.8 and stays in the same state with probability 0.2. The *stay* action keeps the agent in the same state with probability 1. The reward for action *stay* in state *S2* is 1. All other rewards are 0. The discount factor is $\gamma = \frac{1}{2}$.

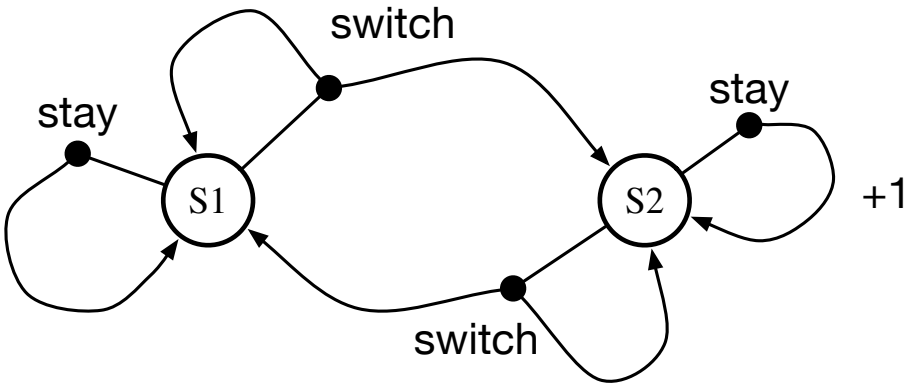


Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

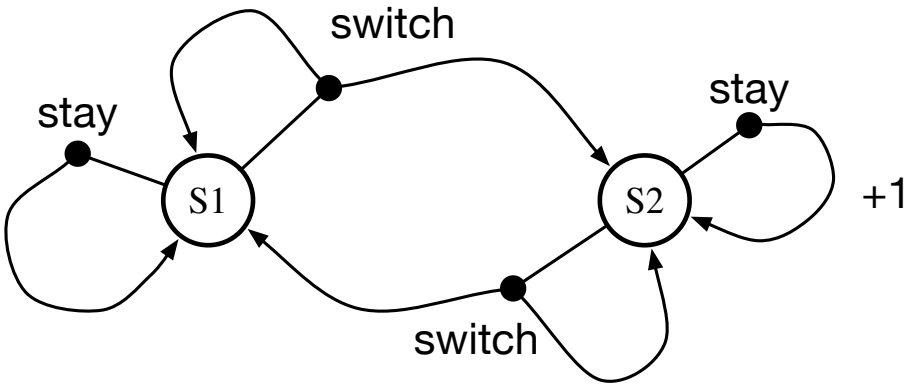
Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

What is the optimal policy?

Exercise: computing value functions

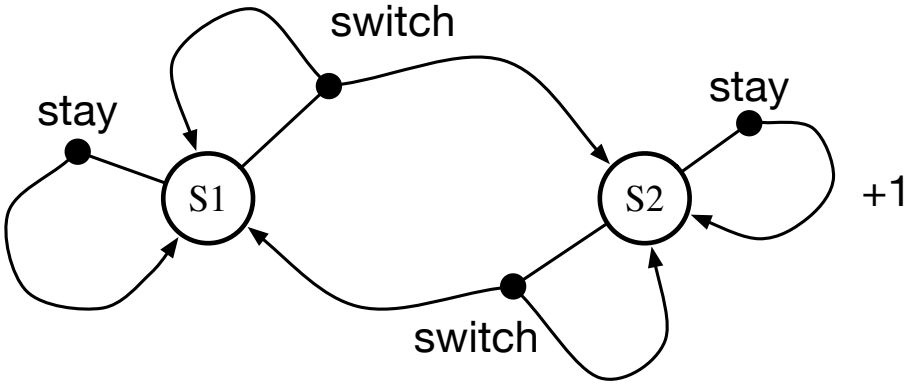


- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

What is the optimal policy?

- S2 —> stay

Exercise: computing value functions

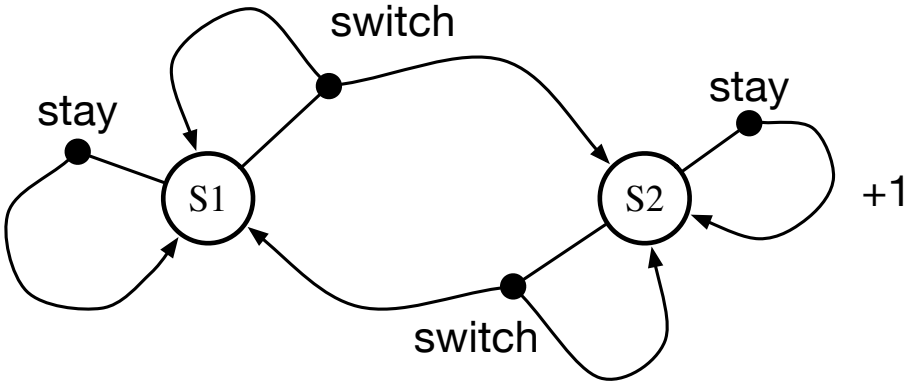


- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

What is the optimal policy?

- S2 —> stay
- S1 —> switch

Exercise: computing value functions



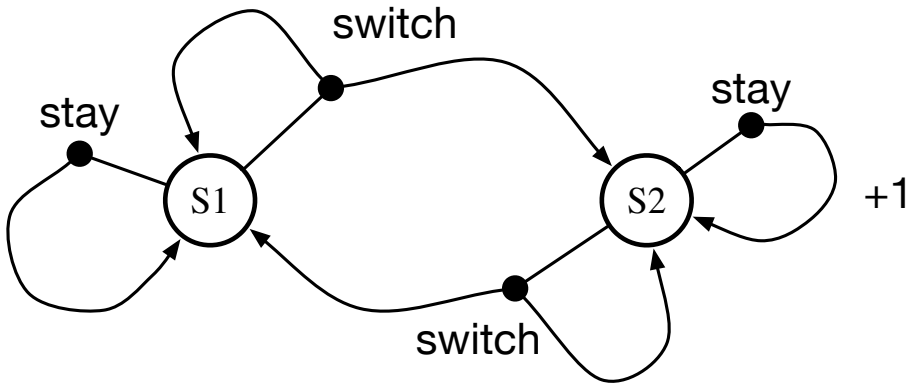
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

What is the optimal policy?

- S2 —> stay
- S1 —> switch

Compute the value function for the policy above:

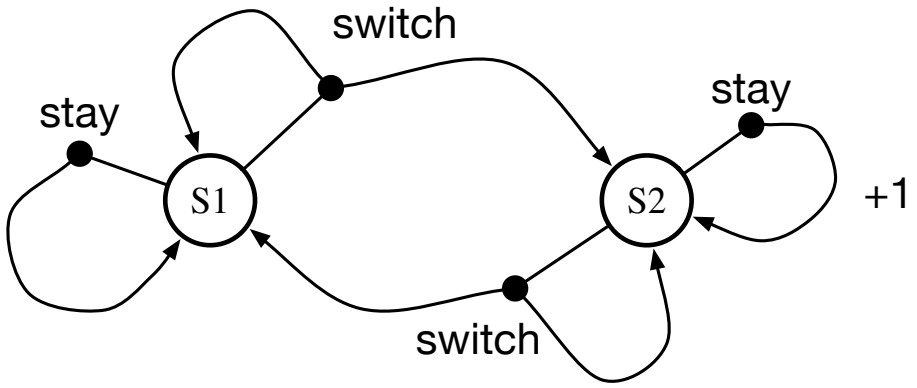
Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Recall:

Exercise: computing value functions

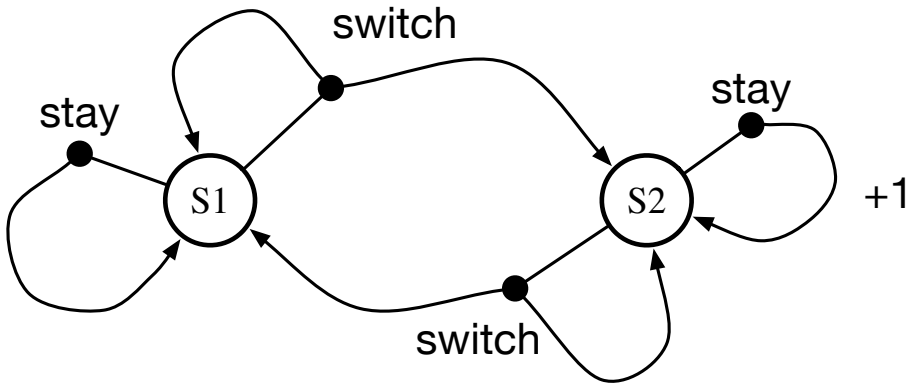


- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Recall:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

Exercise: computing value functions



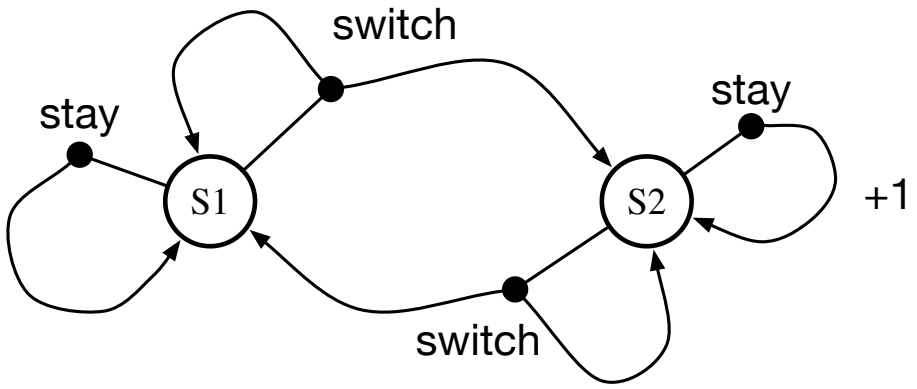
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Recall:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

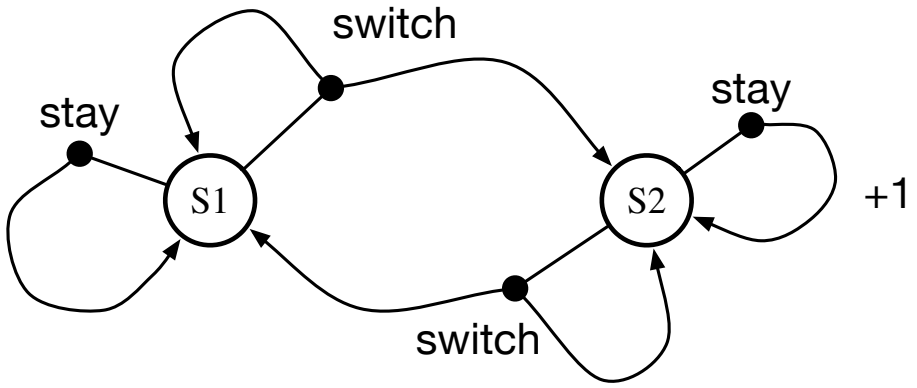
Compute the value function for the policy above:

Hint:



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Hint:

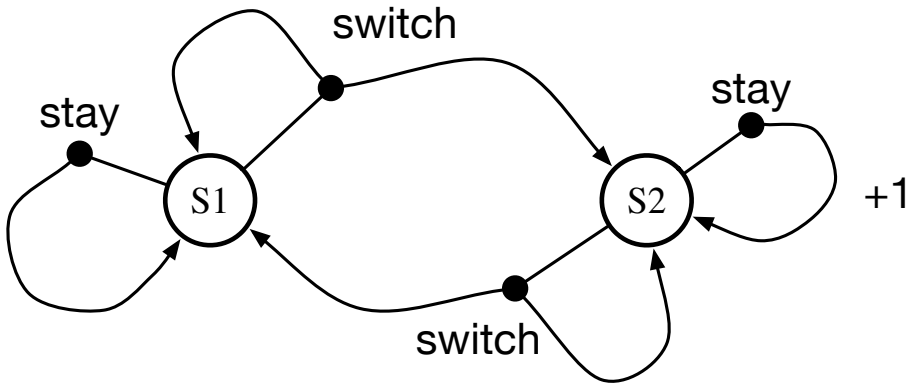


$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

Hint:



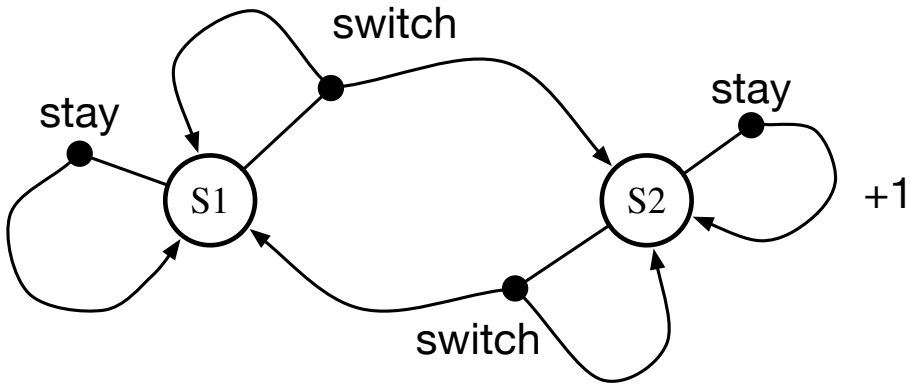
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

$v_\pi(S2)$

Hint:



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

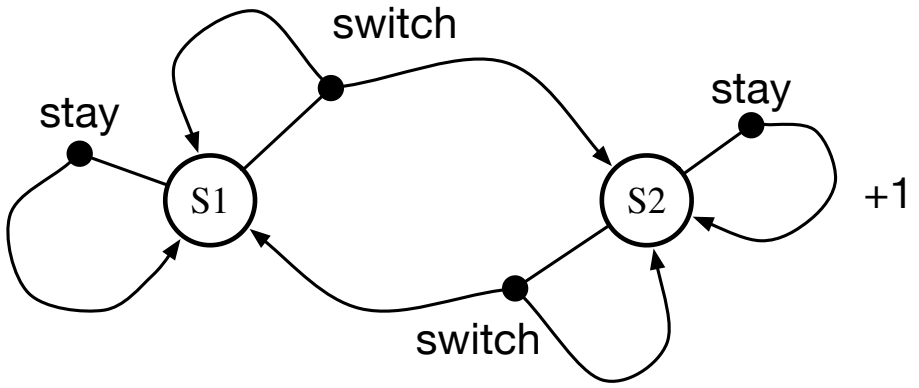
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

$v_\pi(S2)$

$= (1 + \gamma v_\pi(S2))$

Hint:



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

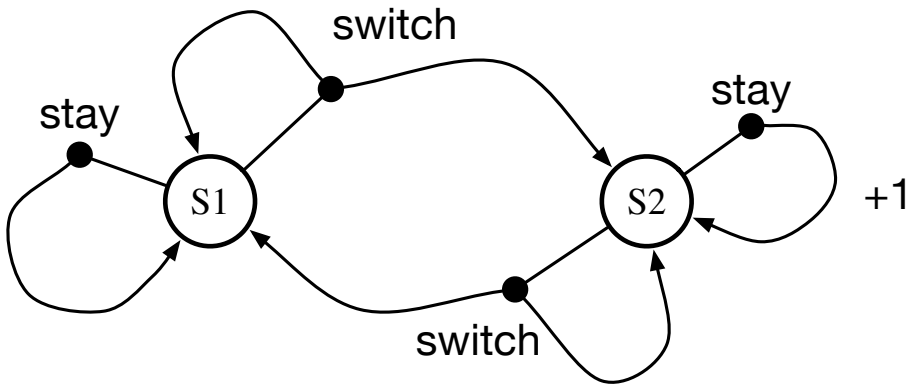
Compute the value function for the policy above:

v_π(S1)

v_π(S2)

= (1 + γv_π(S2))

Hint:



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

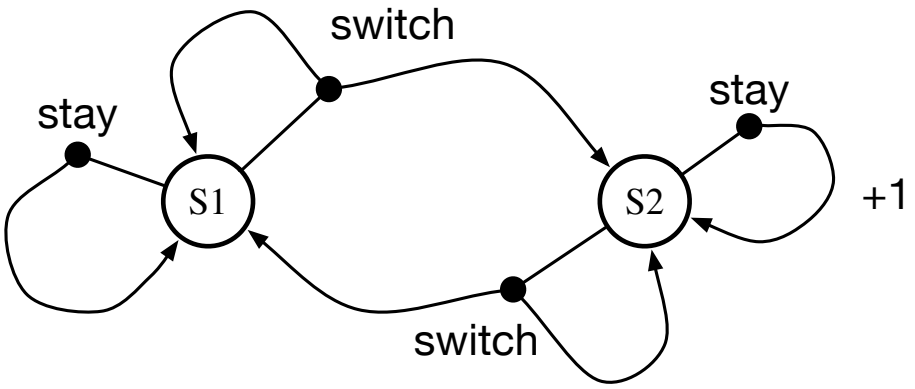
v_π(S1)

$= 0.2^* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

v_π(S2)

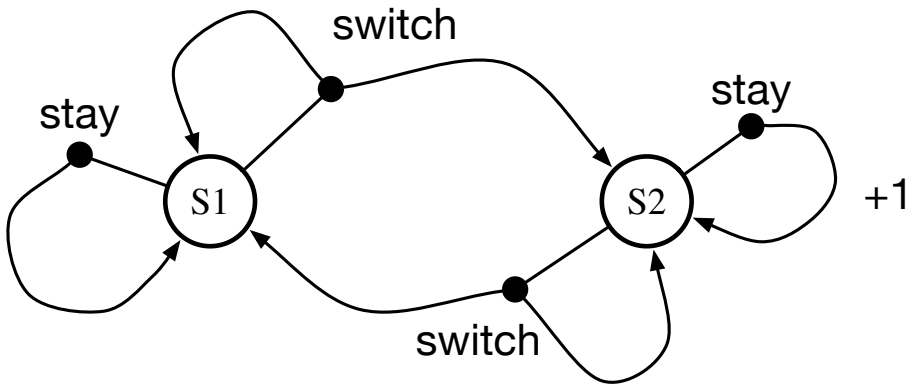
$= (1 + \gamma v_\pi(S2))$

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

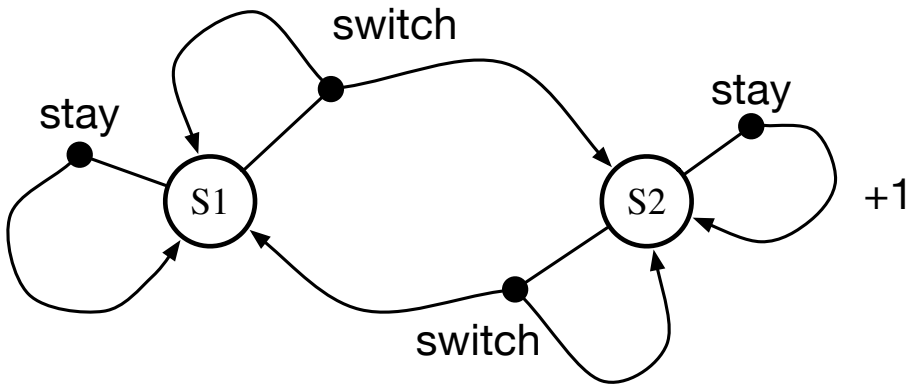
$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

Compute the value function for the policy above:

$v_\pi(S1) = 0.2^* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

$v_\pi(S2) = (1 + \gamma v_\pi(S2))$

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

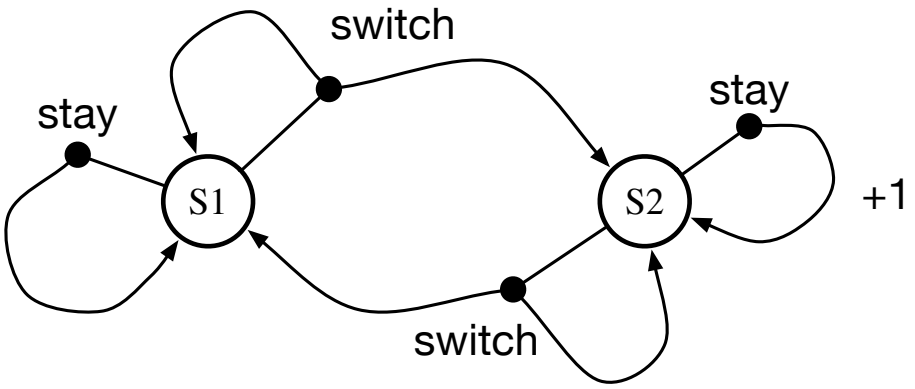
Compute the value function for the policy above:

$v_\pi(S1) = 0.2* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

$v_\pi(S2) = (1 + \gamma v_\pi(S2))$

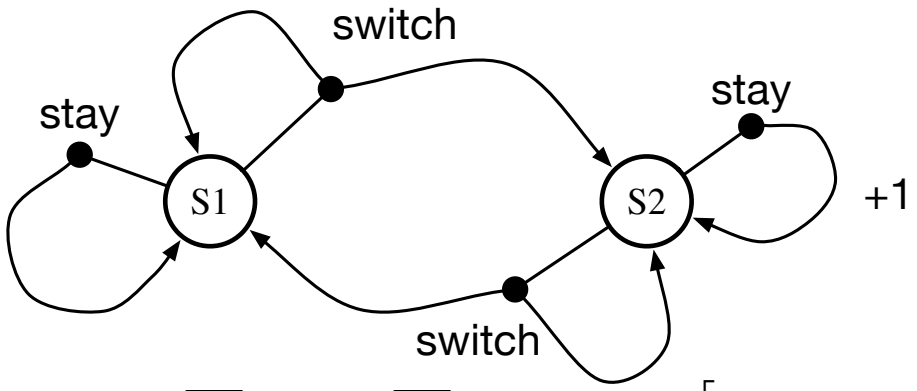
$= 1/(1-\gamma) = 2.0$

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Exercise: computing value functions



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

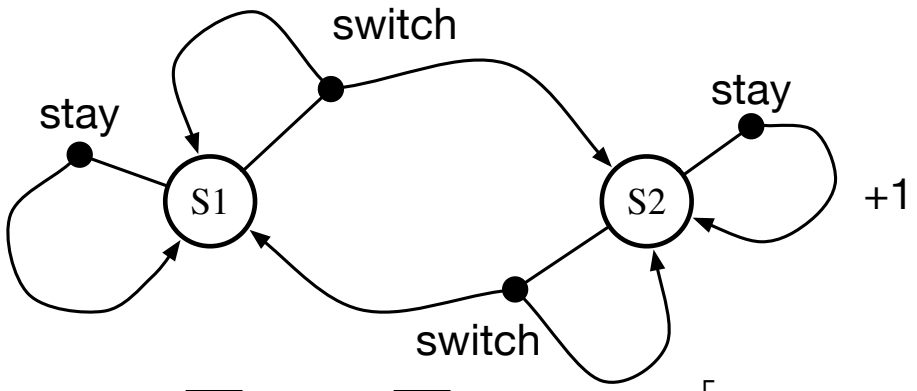
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

$v_\pi(S1) = 0.2^* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

$v_\pi(S2) = 2$

Exercise: computing value functions



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

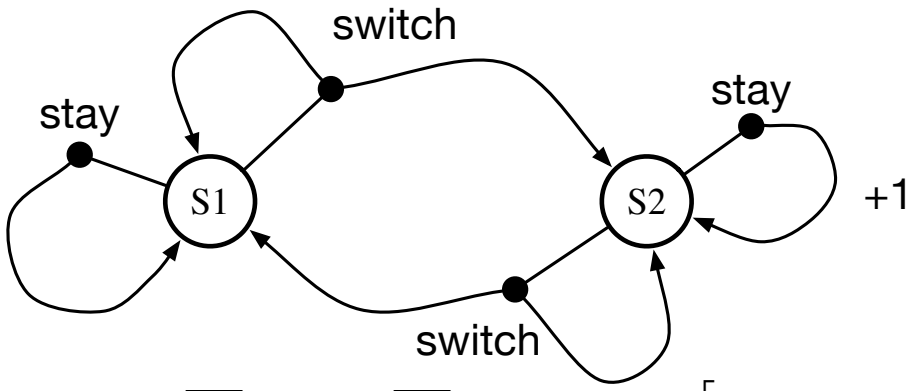
Compute the value function for the policy above:

$v_\pi(S1) = 0.2* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

$= 0.1*v_\pi(S1) + 0.4*2$

$v_\pi(S2) = 2$

Exercise: computing value functions



$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a) \Big[r + \gamma v_\pi(s') \Big]$$

- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**
- Optimal policy:
  - S2 —> stay
  - S1 —> switch

Compute the value function for the policy above:

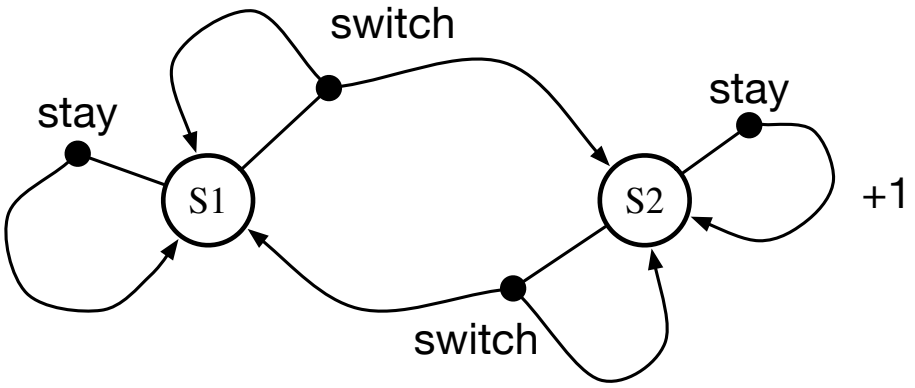
$v_\pi(S1) = 0.2^* (0 + \gamma v_\pi(S1)) + 0.8(0 + \gamma v_\pi(S2))$

$= 0.1^* v_\pi(S1) + 0.4^* 2$

$= 0.8/0.9 = 0.8888\ldots$

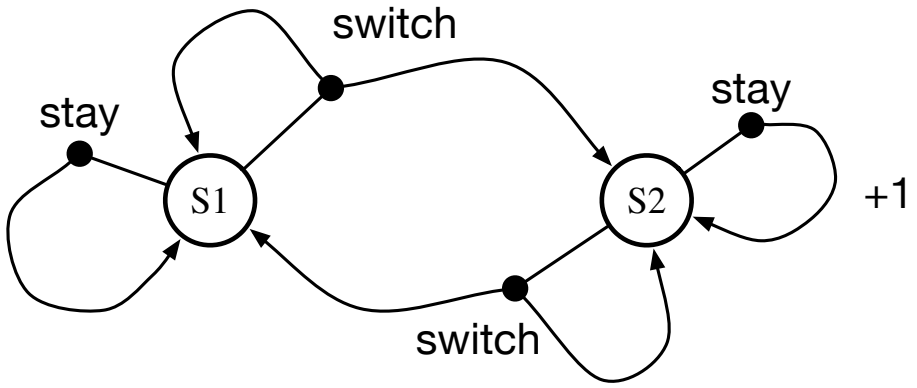
$v_\pi(S2) = 2$

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

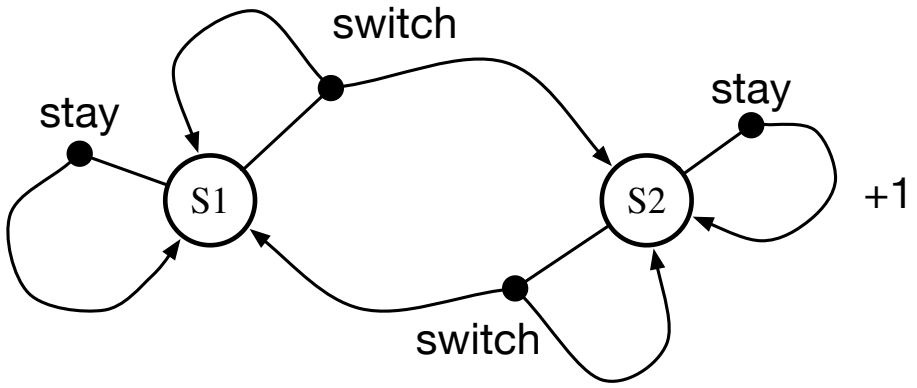
Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

Compute the value function for the **optimal** policy:

Exercise: computing value functions



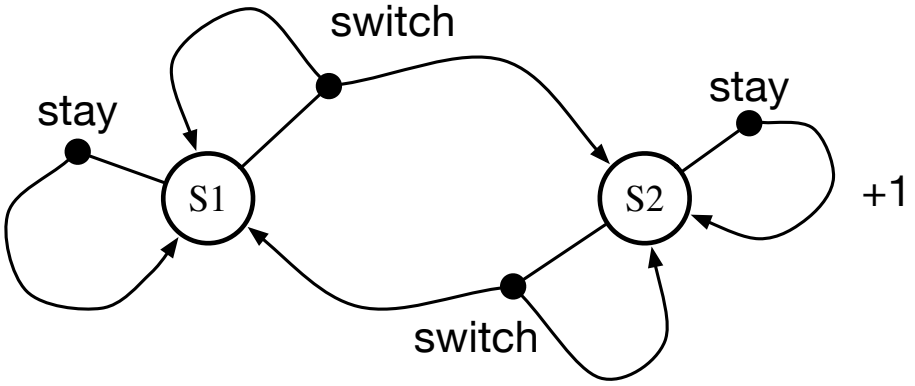
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

Compute the value function for the **optimal** policy:

$v_*(S1) = 0.888\ldots$

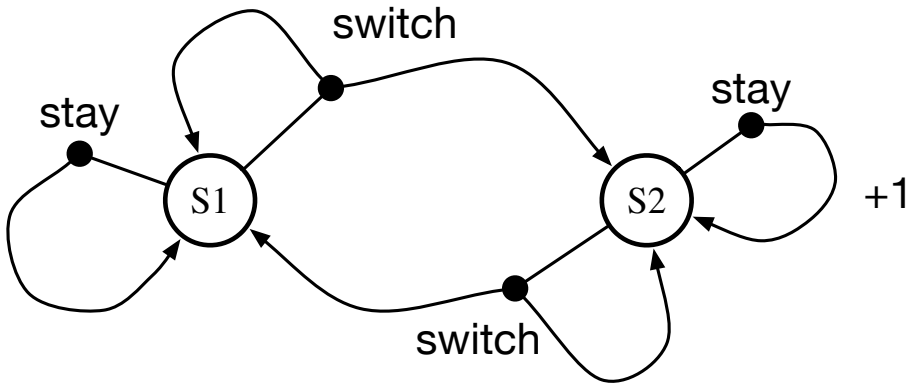
$v_*(S2) = 2.0$

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

Exercise: computing value functions



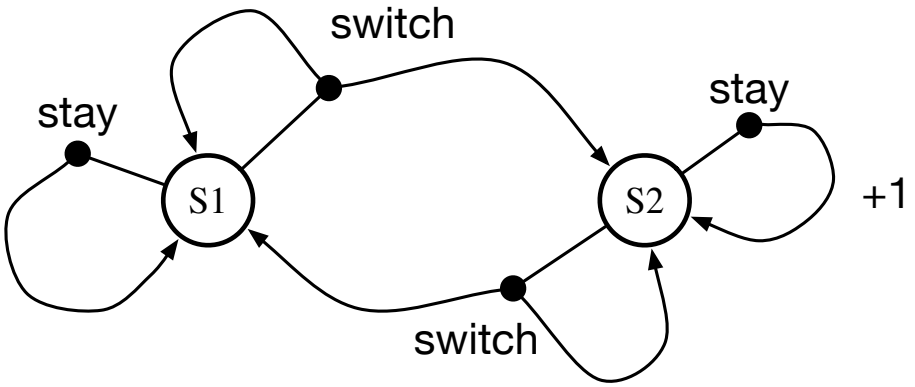
- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

Compute the value function for the following policy:

S1 —> switch

S2 —> switch

Exercise: computing value functions



- γ=0.5
- all rewards are **zero**, except in S2
- **stay** action always succeeds
- **switch** action changes state with prob **0.8**

Compute the value function for the following policy:

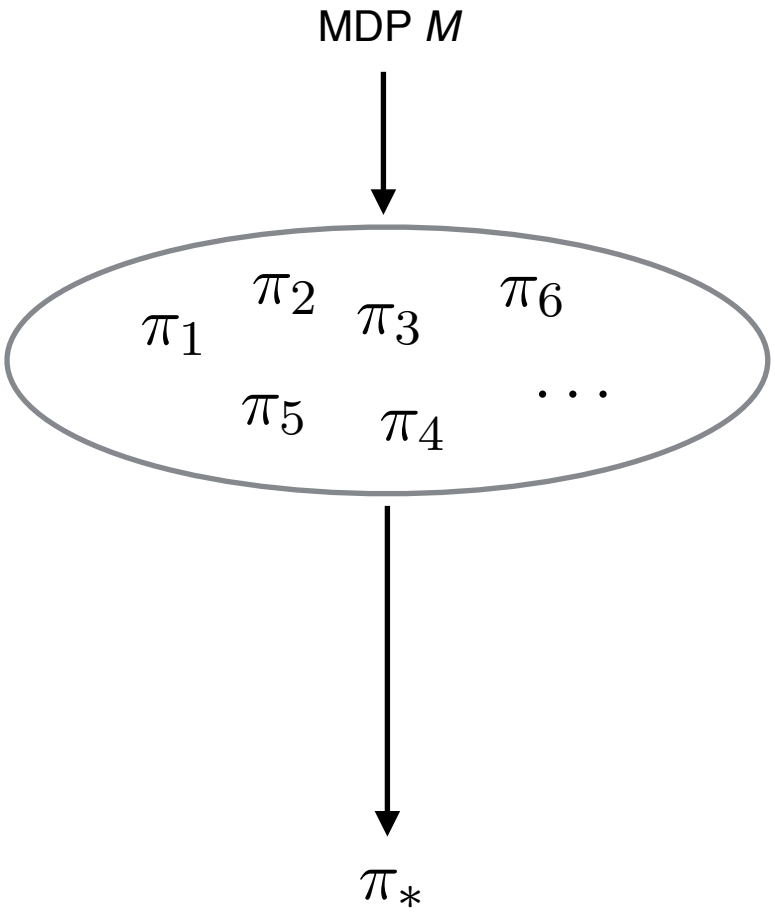
S1 —> switch

S2 —> switch

$v_\pi(S1) = 0$

$v_\pi(S2) = 0$

Naive Strategy for finding π_*



1) For each policy in the policy set, estimate v_π

2) Determine policy with the overall highest estimate

Solving the Bellman Optimality Equation

- Finding an optimal policy by solving the Bellman Optimality Equation requires the following:
  - accurate knowledge of environment dynamics;
  - we have enough space and time to do the computation;
  - the Markov Property.
- How much space and time do we need?
  - polynomial in number of states (via dynamic programming methods; Chapter 4),
  - BUT, number of states is often huge (e.g., backgammon has about 10^{20} states).
- We usually have to settle for approximations.
- Many RL methods can be understood as approximately solving the Bellman Optimality Equation.

Summary

- Agent-environment interaction
  - States
  - Actions
  - Rewards
- Policy: stochastic rule for selecting actions
- Return: the function of future rewards agent tries to maximize
- Episodic and continuing tasks
- Markov Property
- Markov Decision Process
  - Transition probabilities
- Value functions
  - State-value function for a policy
  - Action-value function for a policy
  - Optimal state-value function
  - Optimal action-value function
- Optimal value functions
- Optimal policies
- Bellman Equations
- The need for approximation