Q-Learning

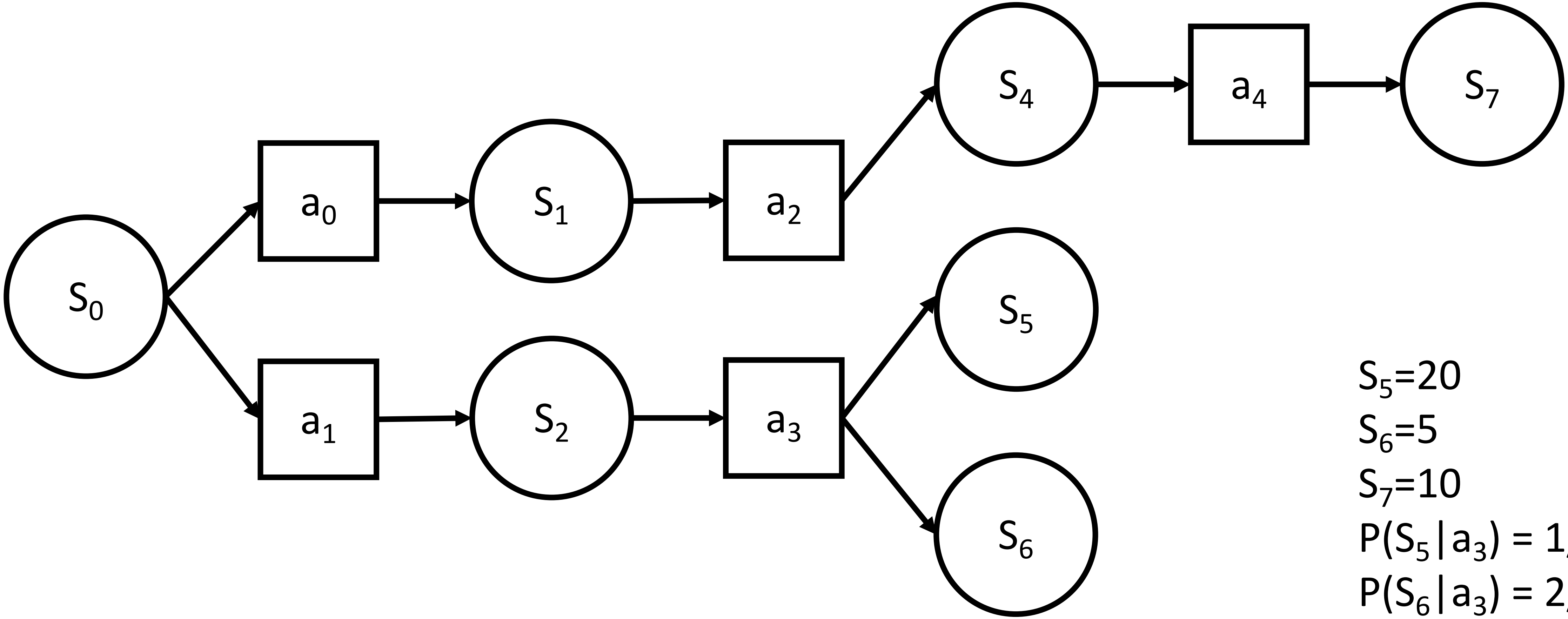
Week 7

Adapted from CS188 Berkeley

Recap

- Trees
  - Deterministic
  - Fully Observable
  - Search
- Min Max Trees
  - Deterministic
  - Fully Observable
  - Expected Value
- Markov Decision Processes
  - Probabilistic
  - Fully Observable
  - Expected Value
- POMDPs
  - Probabilistic
  - Partially Observable
  - Expected Value

Rewards in MDPs



S_5=20
S_6=5
S_7=10
$P(S_5|a_3) = 1/3$
$P(S_6|a_3) = 2/3$

Rewards in MDPs

- What should we prefer?
  - [1,3,5] or [1,2,3]
  - [1,0,0] or [0,0,1]
- Discount future rewards the further they are.
  - Discount factor γ

$$E(s_i) = r_0 + \gamma r_1 + \gamma^2 r_2 \dots \gamma^n r_n = \sum_{k=0}^{n} \gamma^k r_k$$

Bellman Equation

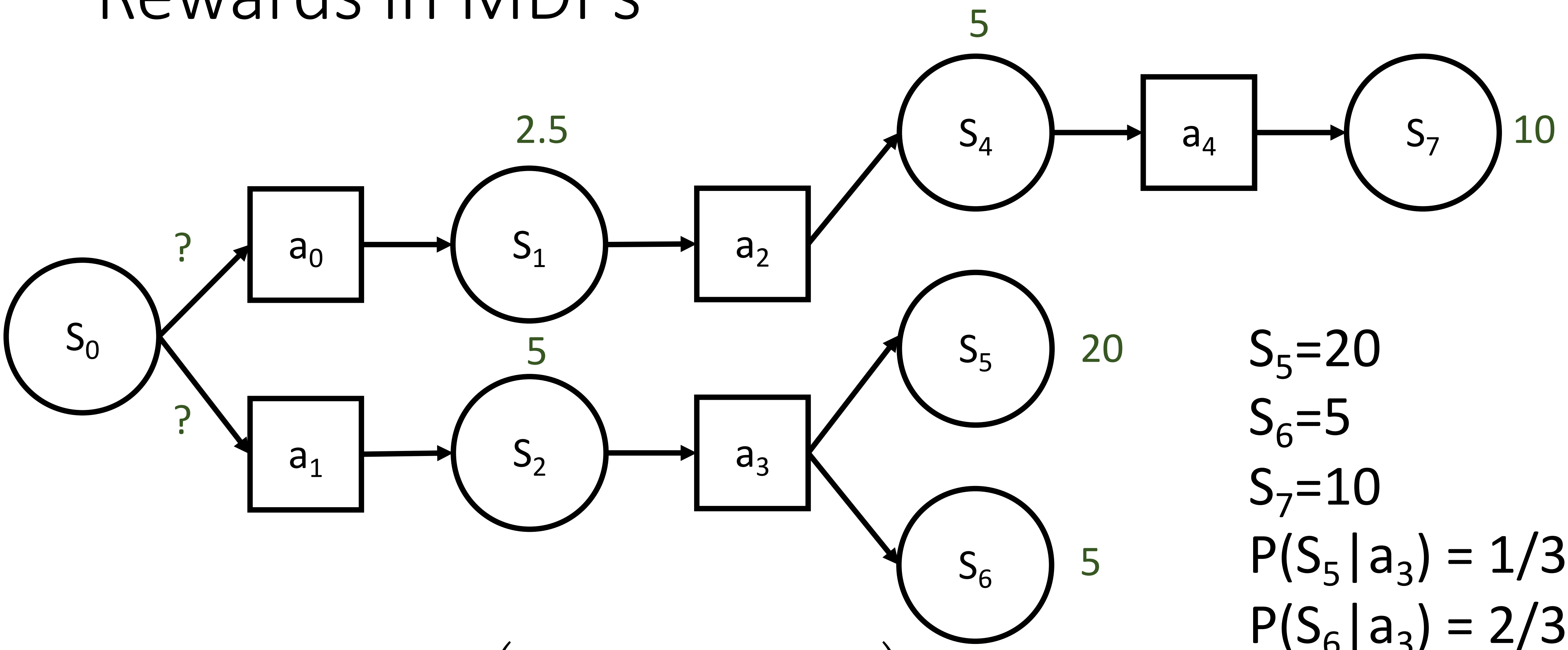
- Optimal Control in Markov Decision Processes

$$V(s_0) = R(s_0) + \gamma P(s_1|a_o)R(s_1)$$

$$V(s_0) = R(s_0) + \gamma P(s_1|a_o)\big(\gamma P(s_2|a_1)R(s_2)\big)$$

$$V(s) = R(s) + max_a\left(\gamma \sum_{s'} P(s'|s,a)V(s')\right)$$

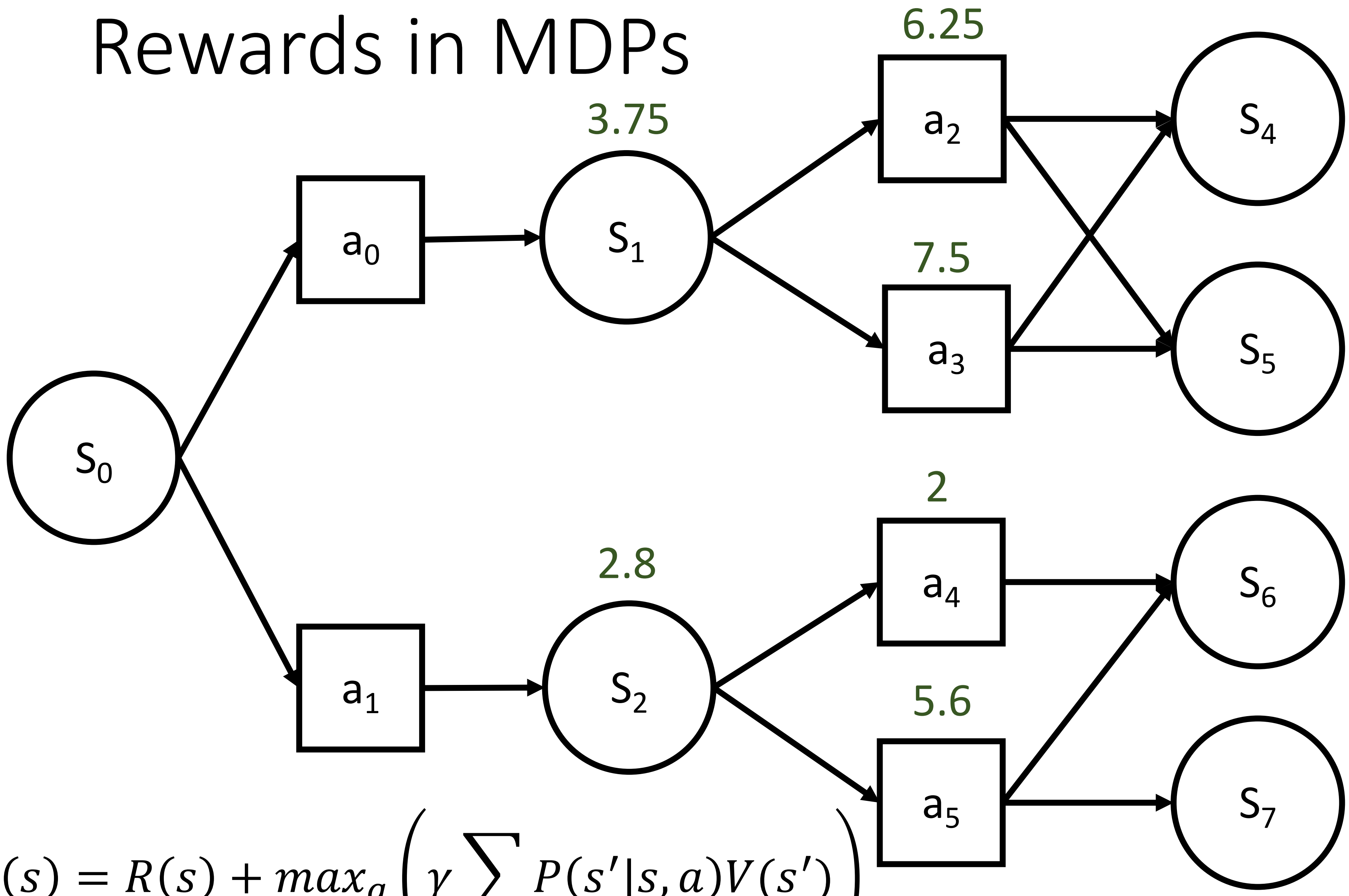
Rewards in MDPs



$$V(s) = R(s) + max_a\left(\gamma \sum_{s'} P(s'|s,a)V(s')\right)$$

S_5=20
S_6=5
S_7=10
$P(S_5|a_3) = 1/3$
$P(S_6|a_3) = 2/3$
$\gamma = 0.5$

Rewards in MDPs



S_4=10
S_5=5
S_6=2
S_7=20
$P(S_4|a_2) = 1/4$
$P(S_5|a_2) = 3/4$
$P(S_4|a_3) = 1/2$
$P(S_5|a_3) = 1/2$
$P(S_6|a_5) = 4/5$
$P(S_7|a_5) = 1/5$
$\gamma = 0.5$

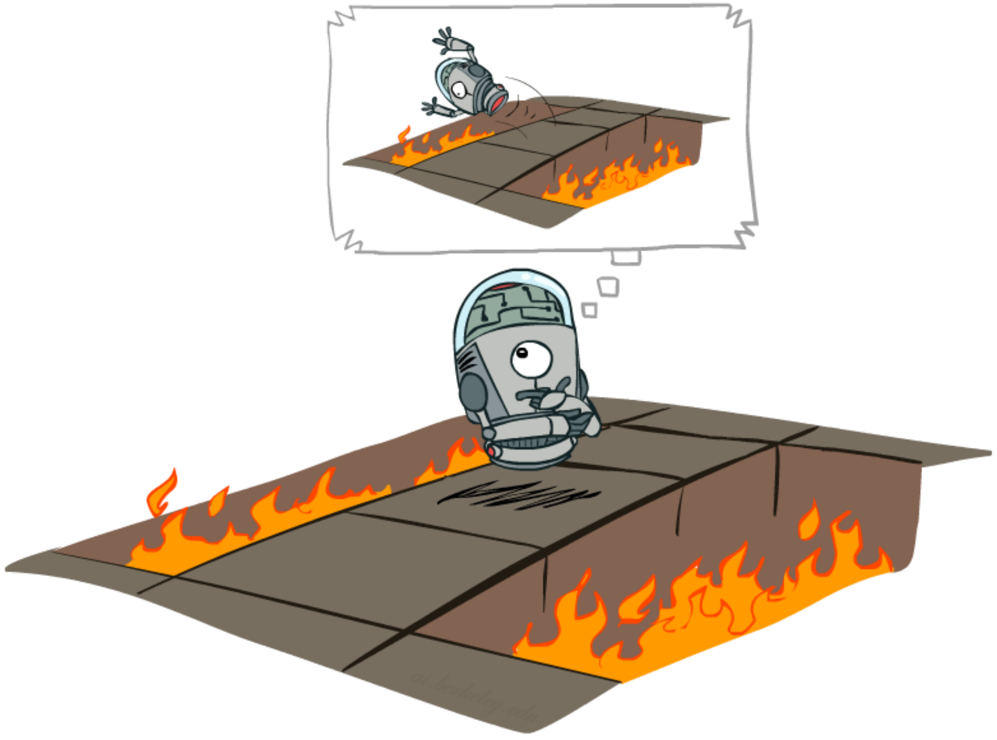
$$V(s) = R(s) + max_a\left(\gamma \sum_{s'} P(s'|s,a)V(s')\right)$$

Problems with MDPs

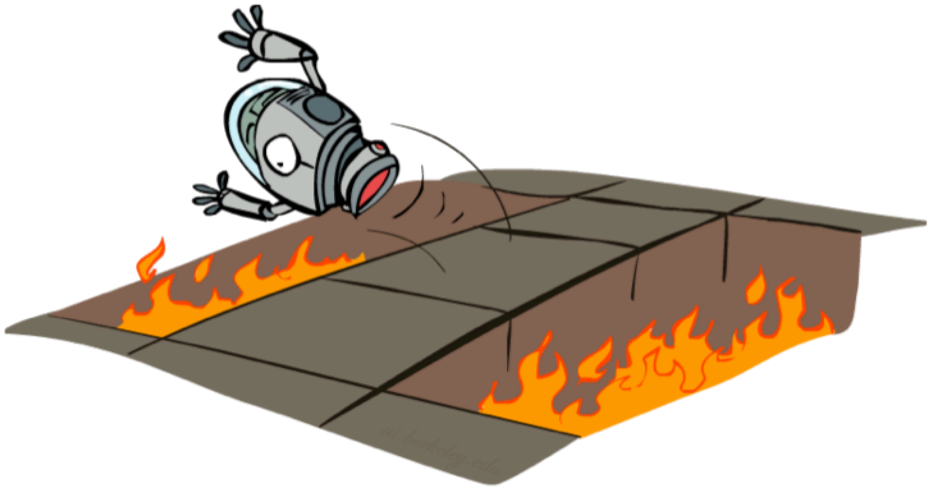
$$V(s) = R(s) + max_a \left(\gamma \sum_{s'} P(s'|s,a)V(s') \right)$$

- Do we need transition probabilities?
- Do we need terminal state values?

Reinforcement Learning



Offline Solution



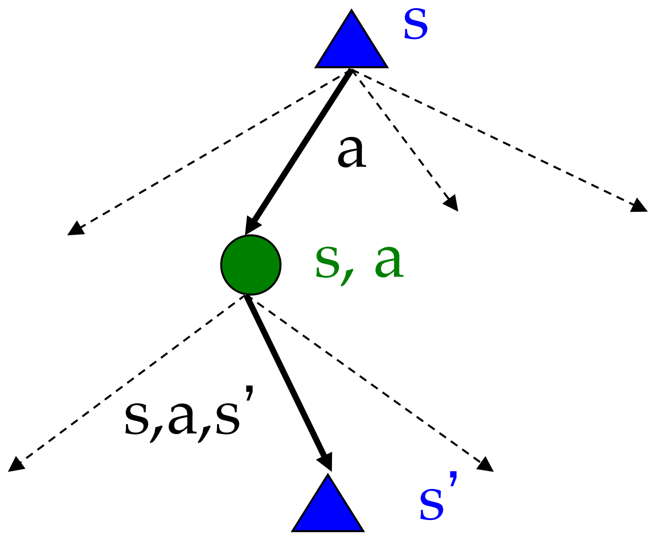
Online Learning

Salvaging MDPs

- MDPs are for planning, not learning.
- Missing Transition Probabilities
  - Learn them?

The “Learning” in RL

- Observations of Real Data
- Each Observation of a Path is an Episode



Types of RL Agents

- Utility Based (Learn State Values)
- Q-Learning (Learn Action Values)
- Reflex (Learn Actions)

Model Based Learning

- Model-Based Idea:
  - Learn an approximate model based on experiences
  - Solve for values as if the learned model were correct

- Step 1: Learn empirical MDP model
  - Count outcomes s' for each s, a
  - Normalize to give an estimate of $\widehat{T}(s, a, s')$

- Step 2: Solve the learned MDP

Expected Age

Goal: Compute expected age of BIME 591 students

Known P(A)

$$E[A] = \sum_a P(a) \cdot a \qquad = 0.35 \times 20 + \ldots$$

Without P(A), instead collect samples [a_1, a_2, … a_N]

Unknown P(A): "Model Based"

$$\hat{P}(a) = \frac{\text{num}(a)}{N}$$

$$E[A] \approx \sum_a \hat{P}(a) \cdot a$$

Why does this work? Because eventually you learn the right model.

Unknown P(A): "Model Free"

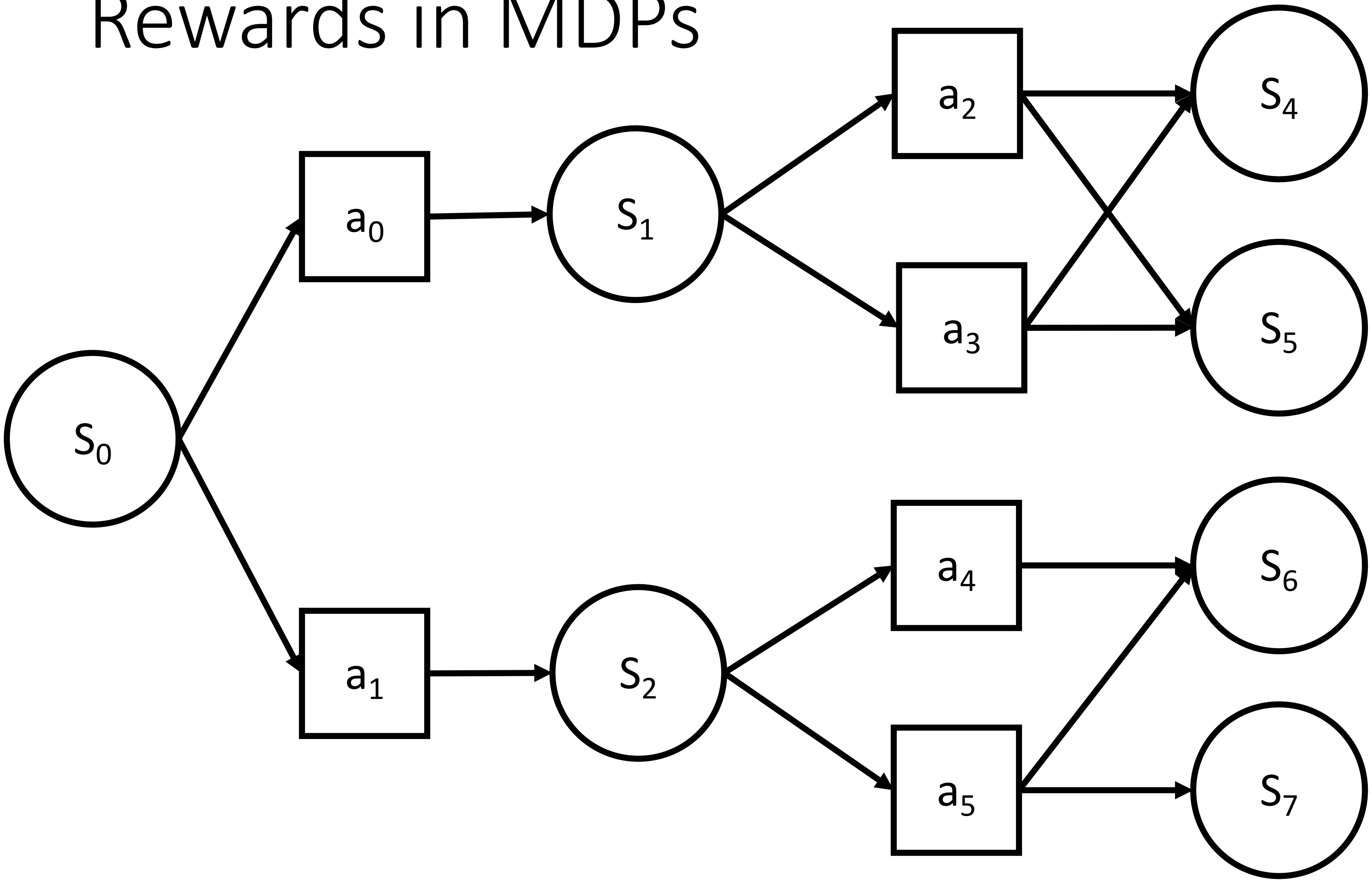
$$E[A] \approx \frac{1}{N} \sum_i a_i$$

Why does this work? Because samples appear with the right frequencies.

Direct Evaluation

- Model-Free Idea: Average together observed sample values
  - Watch episodes
  - Every time you visit a state, write down what the sum of discounted rewards turned out to be
  - Average those samples

Rewards in MDPs



S_4=10
S_5=5
S_6=2
S_7=20
γ = 0.5

Problems with Direct Evaluation

- Just Supervised Learning!
- No Decisions by Agent
- Probabilistic Actions still need Transitions
- Very Slow, No Information about Connections

Q-Learning

- We need to choose actions not states

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

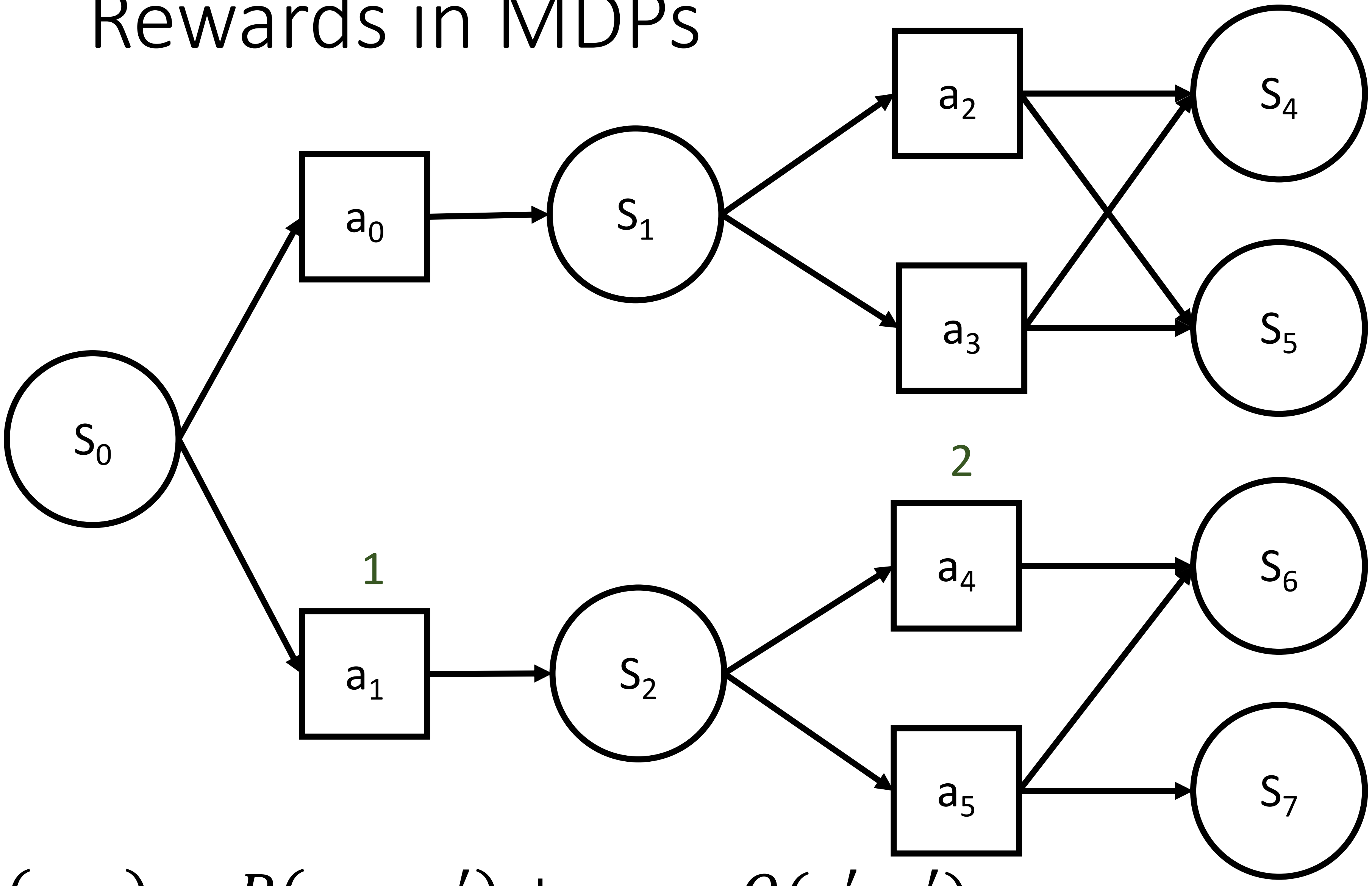
- Direct Evaluation of Actions

$$Q(s,a) = R(s,a,s') + \gamma max Q(s',a')$$

Tabular Q-Learning

- Keep a running tab on every possible action at each state
- Online learning
  - Explore

Rewards in MDPs



S_4=10
S_5=5
S_6=2
S_7=20
γ = 0.5

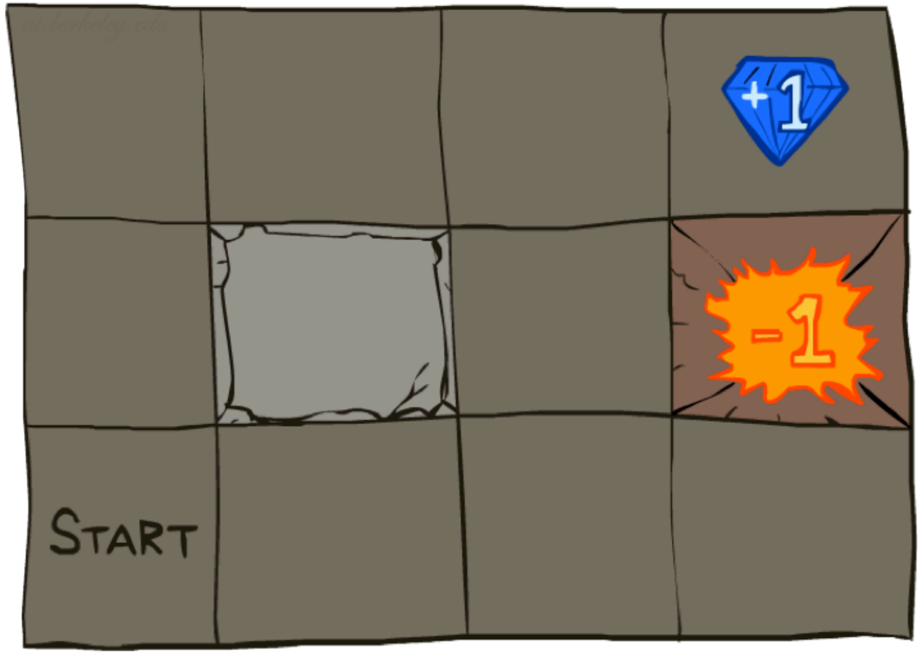
$$Q(s,a) = R(s,a,s') + \gamma max Q(s',a')$$

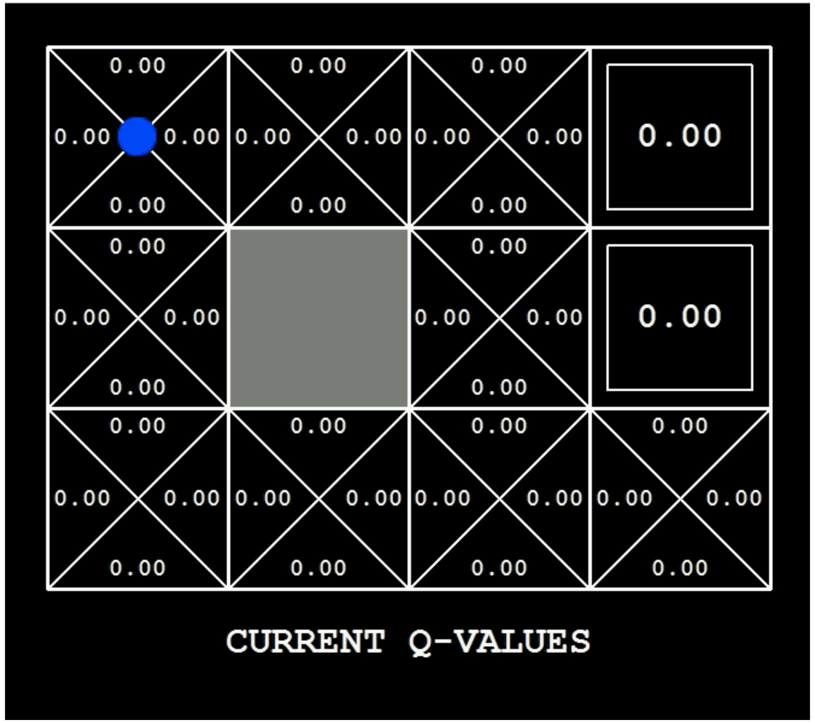
Tabular Q-Learning

- How to update Q-Values?

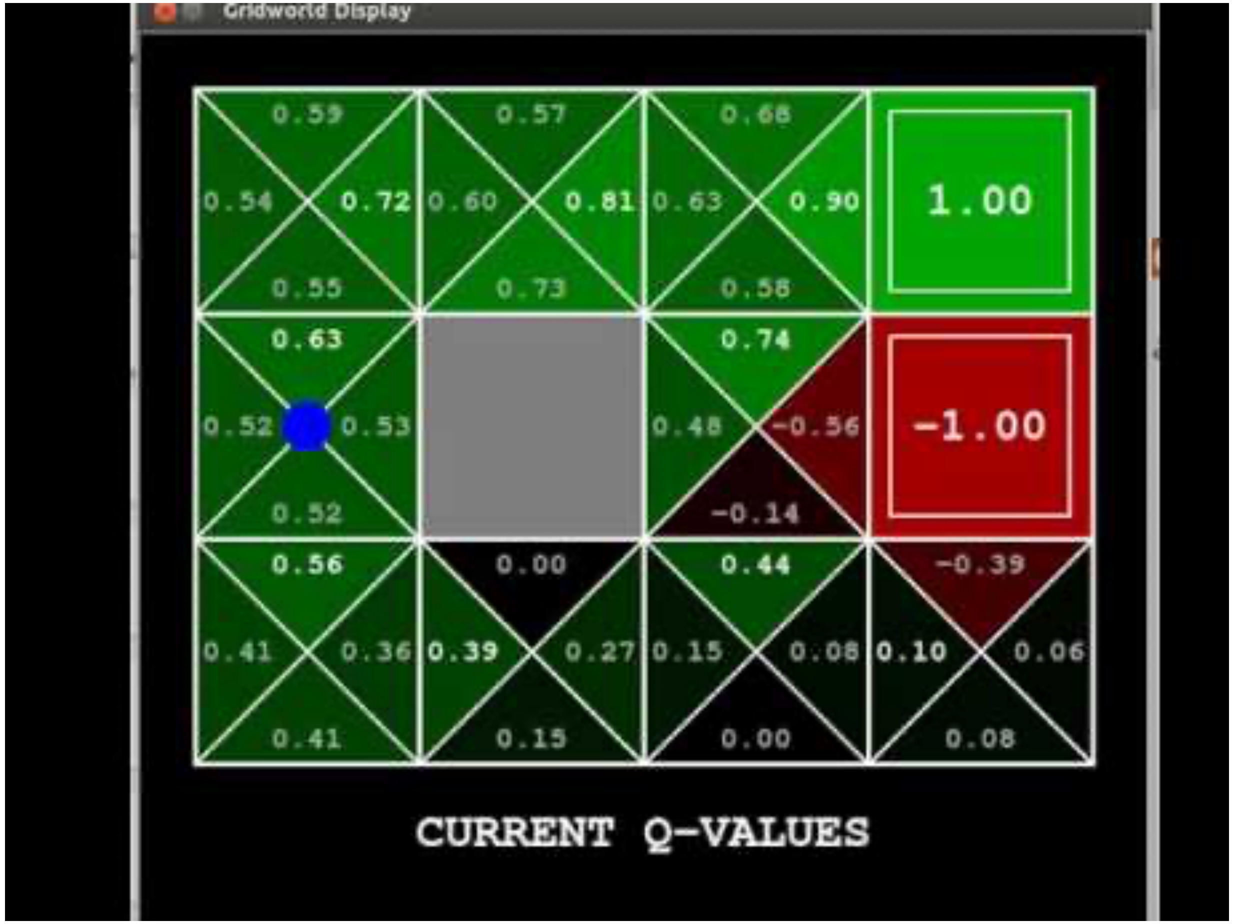
$$Q(s,a)_{new} = (1-\alpha)Q(s,a)_{old} + \alpha\gamma maxQ(s',a')$$

Q-Learning Training

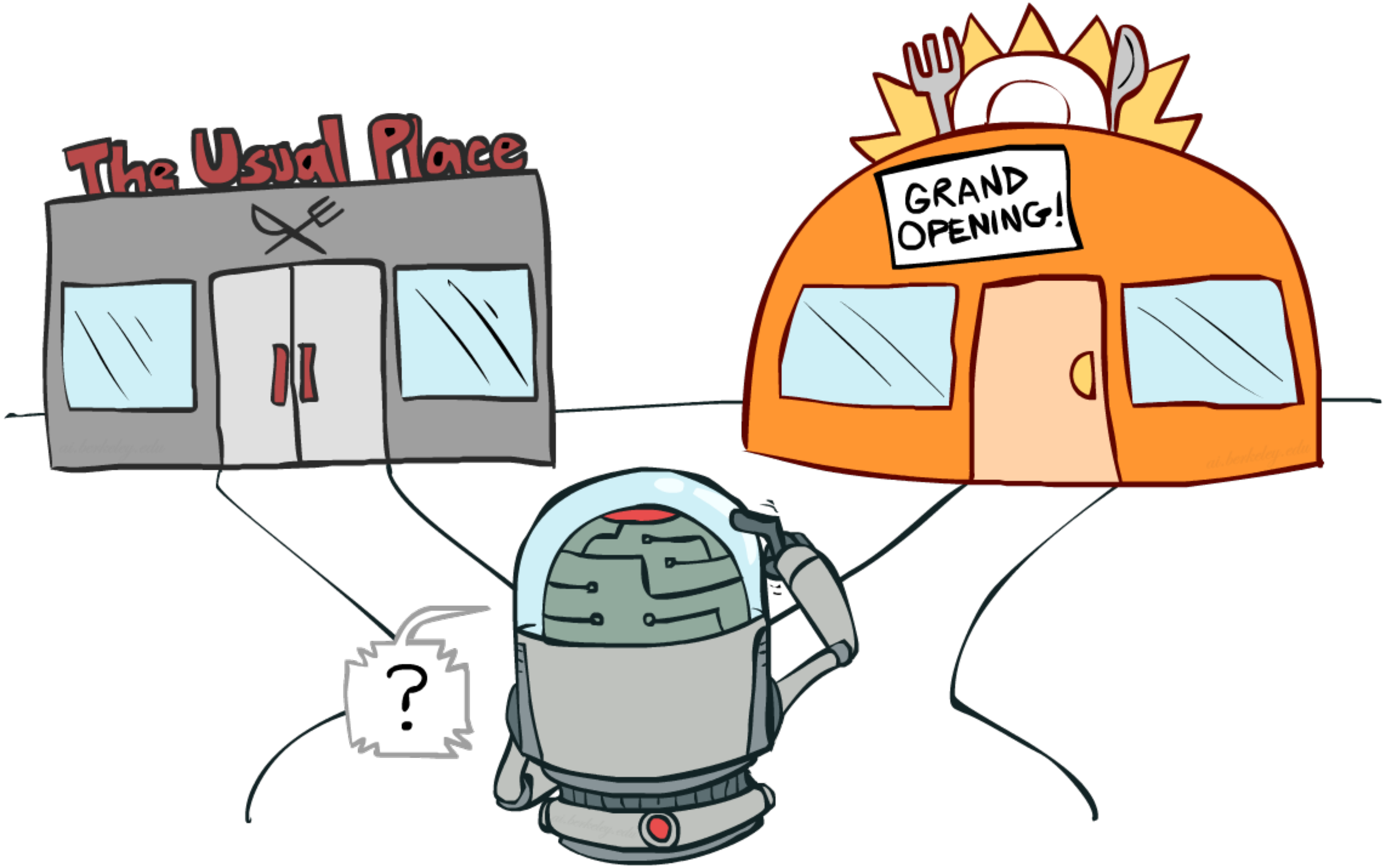




Tabular Q-Learning



Exploration vs. Exploitation

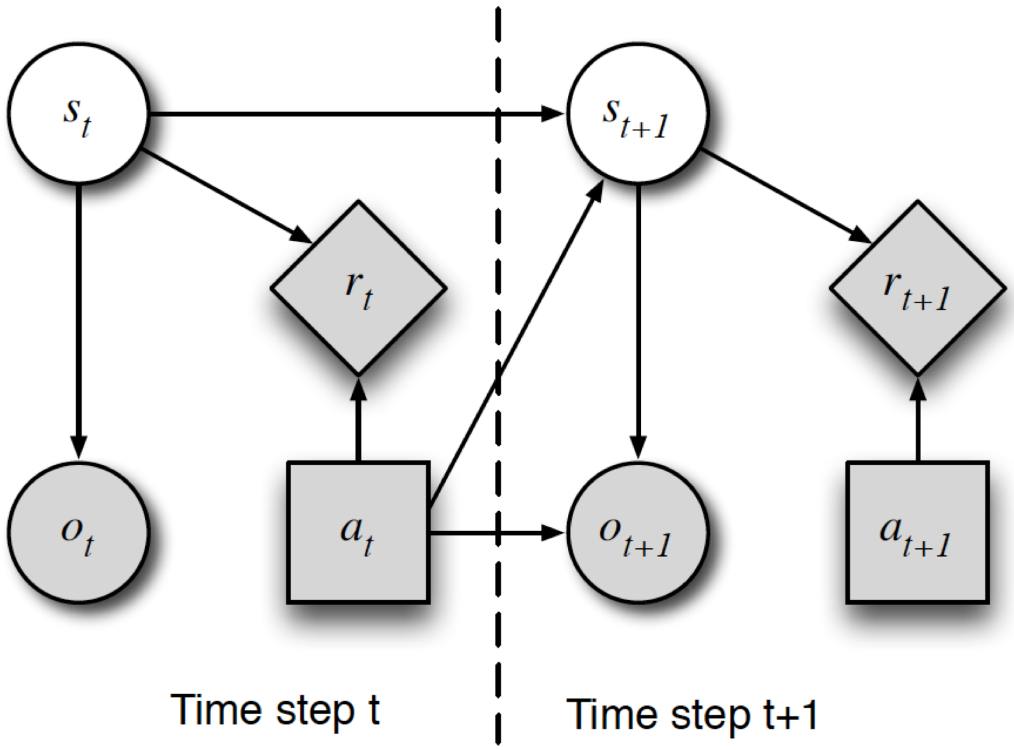


Q-Learning Epsilon-Greedy Search

- Simplest: random actions (ε-greedy)
  - Every time step, flip a coin
  - With (small) probability ε, act randomly
  - With (large) probability 1-ε, act on current policy
- Problems with random actions?
  - Should we keep exploring?
  - One solution: lower ε over time
  - Another solution: exploration functions

Applications for Dialogue Management

- POMDPs for modeling belief states in conversations.
- Lots of unknown transition probabilities.
- Implement Q-Learning



Problems with Q-Learning

- Slow Convergence
  - Cannot guess values of intermediate states
- Discrete States
- Discrete Actions