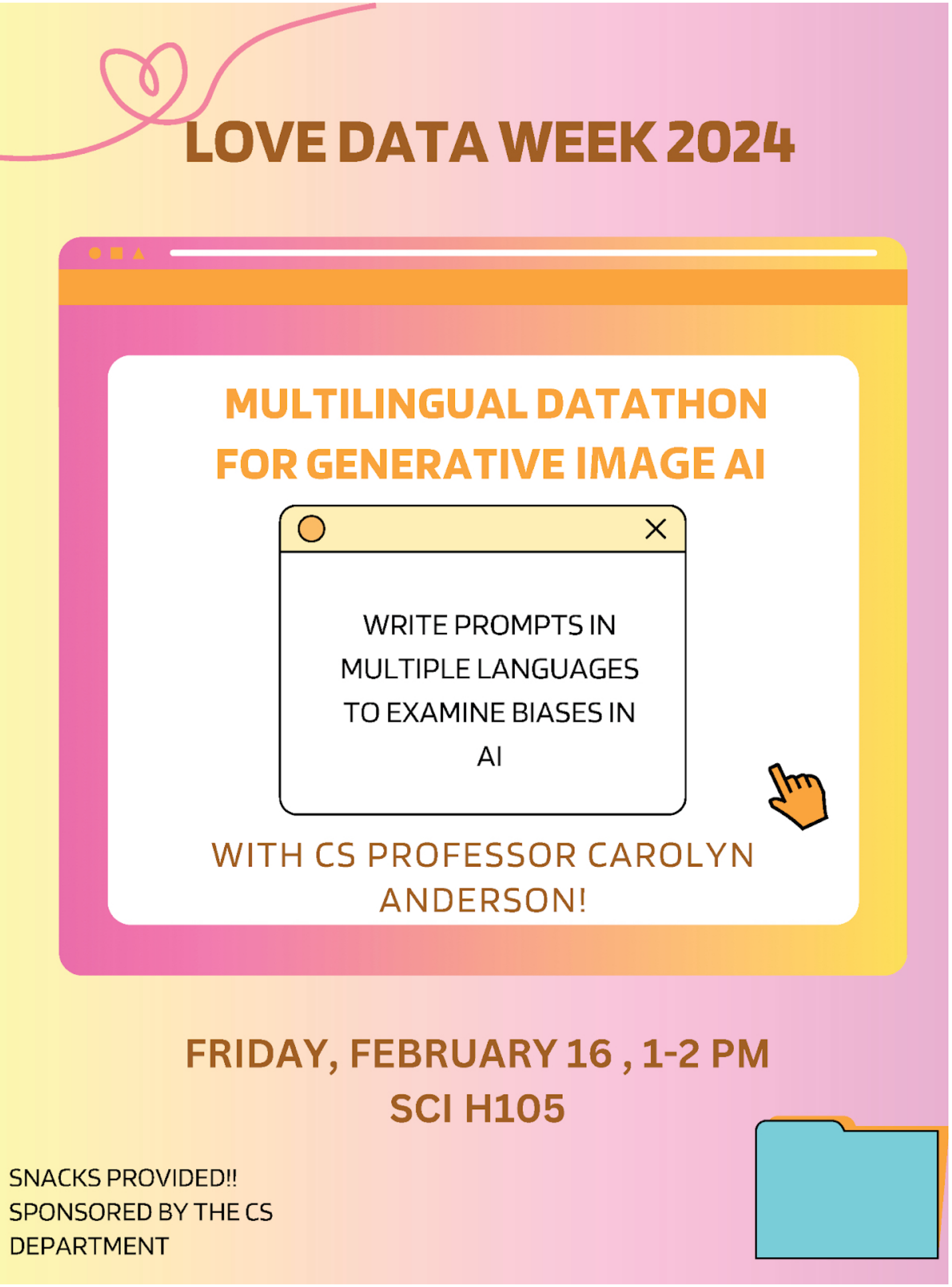
CS 232: Artificial Intelligence

Spring 2024

Prof. Carolyn Anderson
Wellesley College

Reminders

- I have help hours today from 3:30-4:30 in W422
- Monday is a holiday, but I will still have help hours from 4-5:15
- Midterm is two weeks away; no new HW next week


LOVE DATA WEEK 2024
MULTILINGUAL DATATHON
FOR GENERATIVE IMAGE AI
WRITE PROMPTS IN MULTIPLE LANGUAGES TO EXAMINE BIASES IN AI
WITH CS PROFESSOR CAROLYN ANDERSON!
FRIDAY, FEBRUARY 16 , 1-2 PM
SCI H105
SNACKS PROVIDED!!
SPONSORED BY THE CS DEPARTMENT


From CS Inclusive Excellence Advocates
Bugs don't want you to know this one simple trick...
CS 001: MISSING SEMESTER
Workshop #2: Shell
Terminal? Command line? Shell? Come demystify the power of simple commands to up your hacking game!
RSVP here:
Lunch provided!
in SCI H402
Monday 2/19 @ 1 pm
contact: sh102, th105
accessibility: accessibility@wellesley.edu
exp: 2/20

Uncertain Outcomes

Error: we try to do a thing & it doesn't work

Non-rational players: we can't predict what others do

Chance: there's randomness built in

Dynamic Environments: something changed!

Stochastic Transition Model

So far we have considered search problems where the outcomes of our actions was known.

Our transition function took our current state and an action, and told us which state we would end up in next:

Deterministic:

$T(s,a) \rightarrow s'$

Stochastic:

$T(s,a,s') \longrightarrow p(s' \mid s,a)$

But what if we don't know what will happen next?

Assumptions About Transitions

When the transition function is known and deterministic:

DFS, BFS, UCS, A*, MiniMax

When the transition function is known but stochastic:

Markov Decision Processes

ExpectiMax

When the transition function is unknown:

Reinforcement Learning

Stochastic Transitions

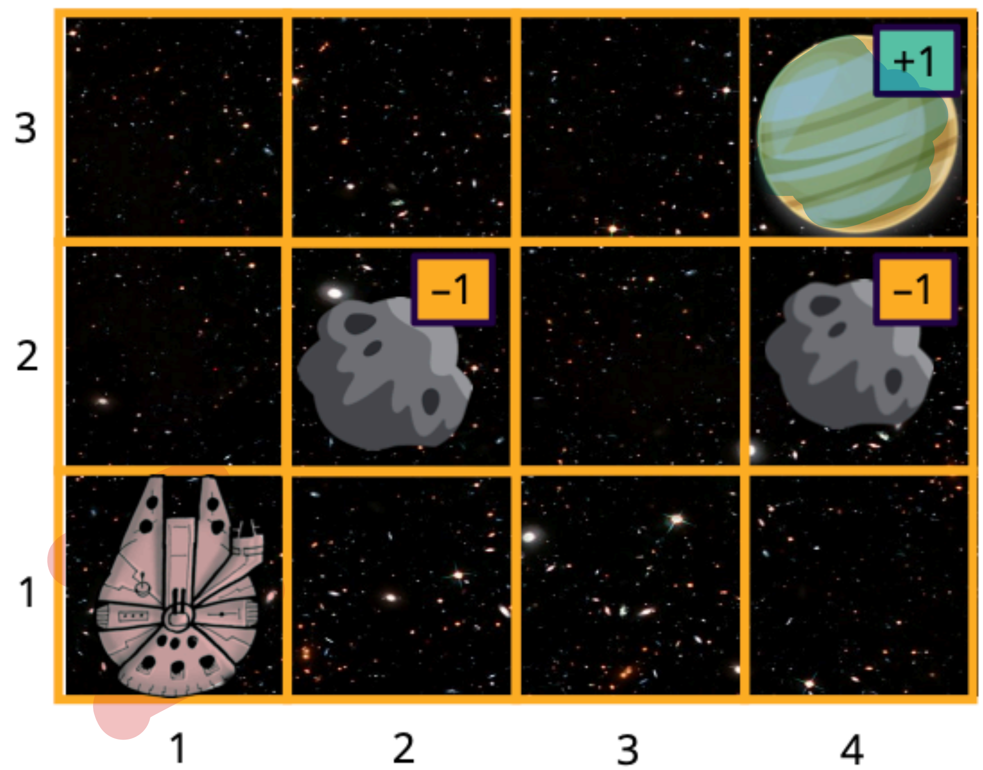
Navigating an Asteroid Field

Suppose we have a **fully-observable** 4x3 environment with goal states.

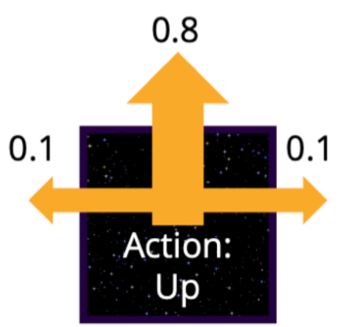
The millennium falcon begins in the start state and **picks an action at each time step**.

Actions: *Up, Down, Left, Right*

The game **terminates when it reaches a goal state** (+1 or -1).



Transition Model:



0.0

Example borrowed from Chris Callison-Burch

Markov Decision Processes

- Known state space
- Known transition function
- Transition function is stochastic:

successor state

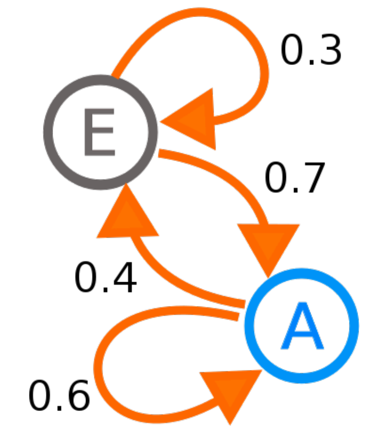
action to take in s

$$T(s,a,s') \longrightarrow p(s' \mid s,a)$$

current state

- Reward function that tells us the value of being in each state

Markovian transitions depend only on the current state

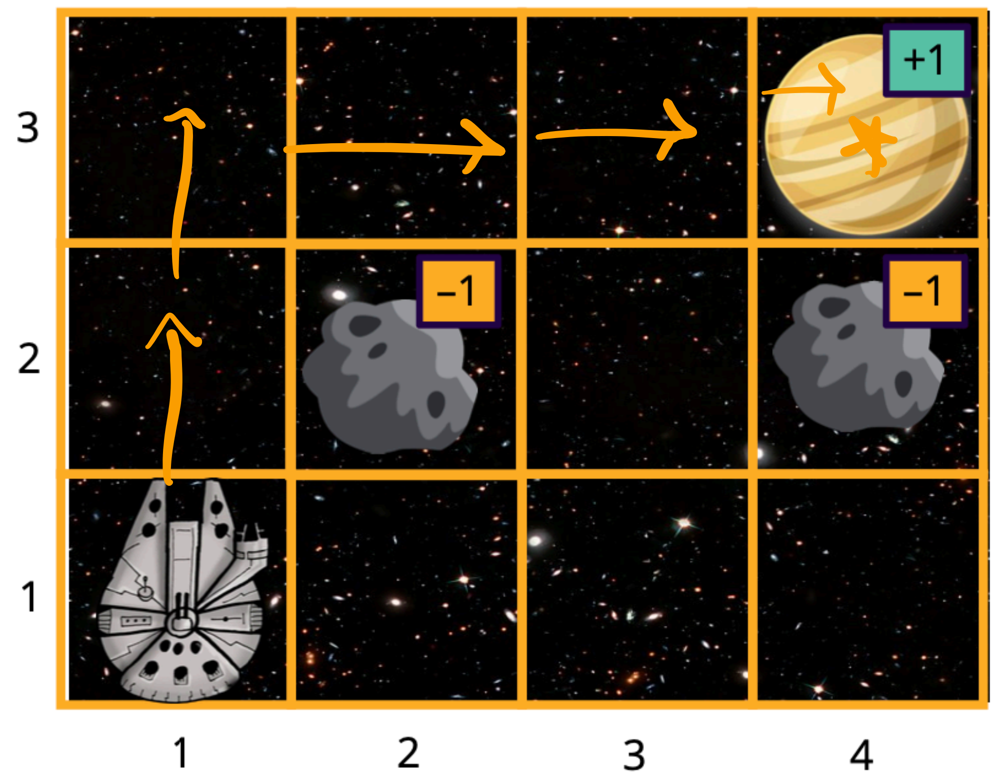


Navigating an Asteroid Field

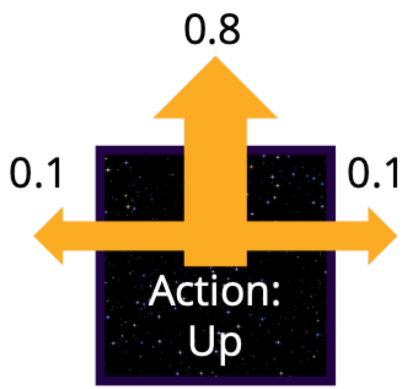
For action sequence

[Up, Up, Right, Right, Right],

what's the probability that the millennium falcon reaches the intended goal?



Transition Model:



$0.8 \cdot 0.8 \cdot 0.8 \cdot 0.8 \cdot 0.8$

Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

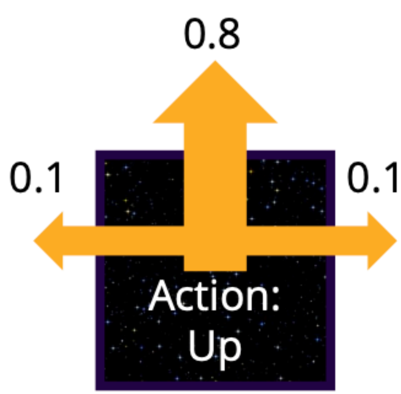
For action sequence

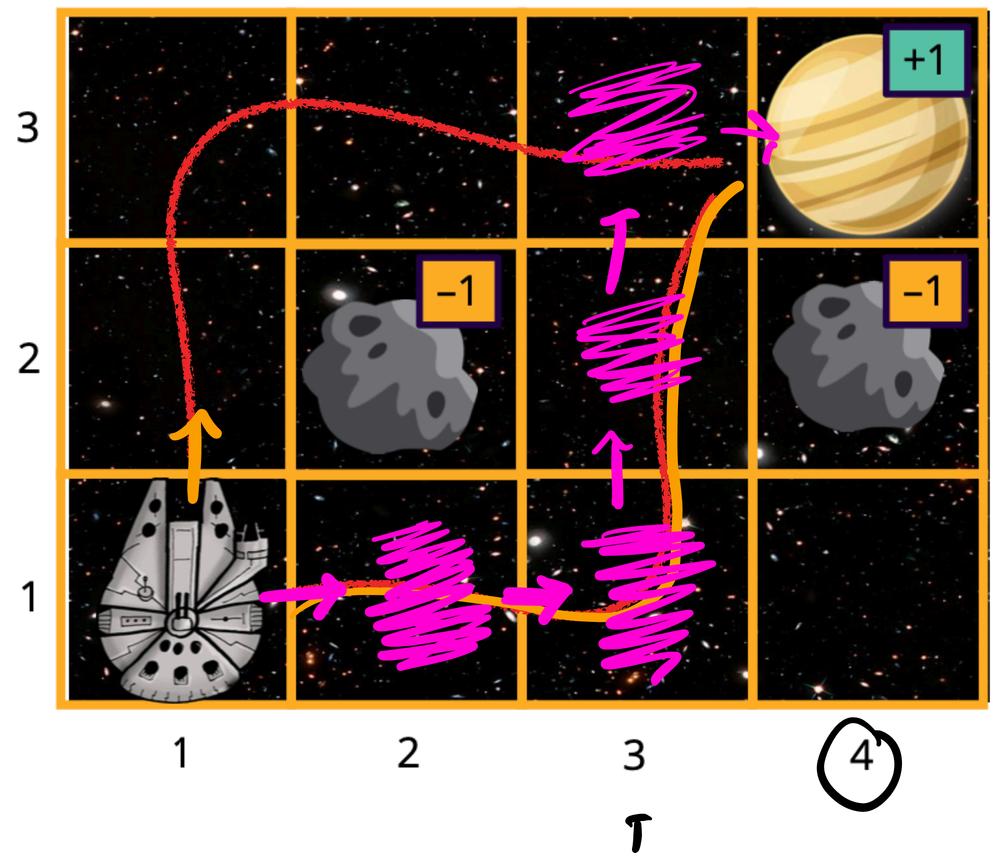
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

What are the routes to the goal?

Transition Model:





$0.1 \cdot 0.1 \cdot 0.1 \cdot 0.1 \cdot 0.8$

Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

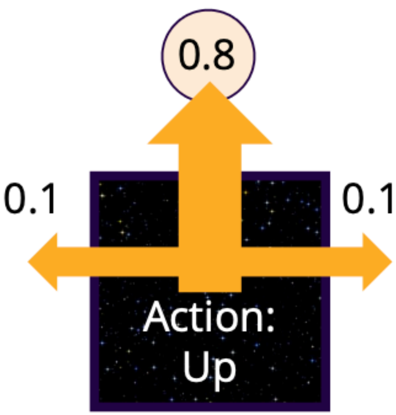
For action sequence

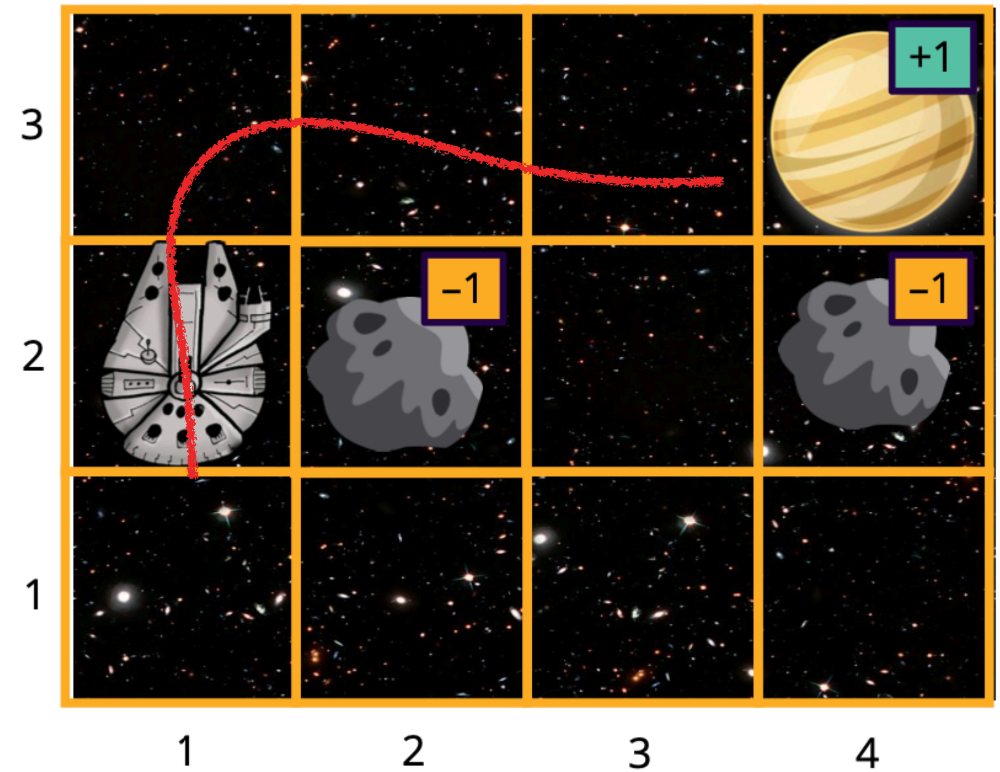
[Up, Up, Right, Right, Right],

what's the probability that the millennium falcon reaches the intended goal?

0.8

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

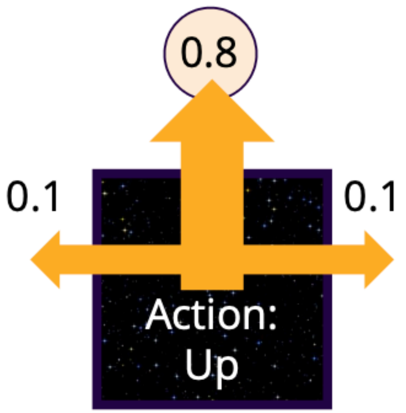
For action sequence

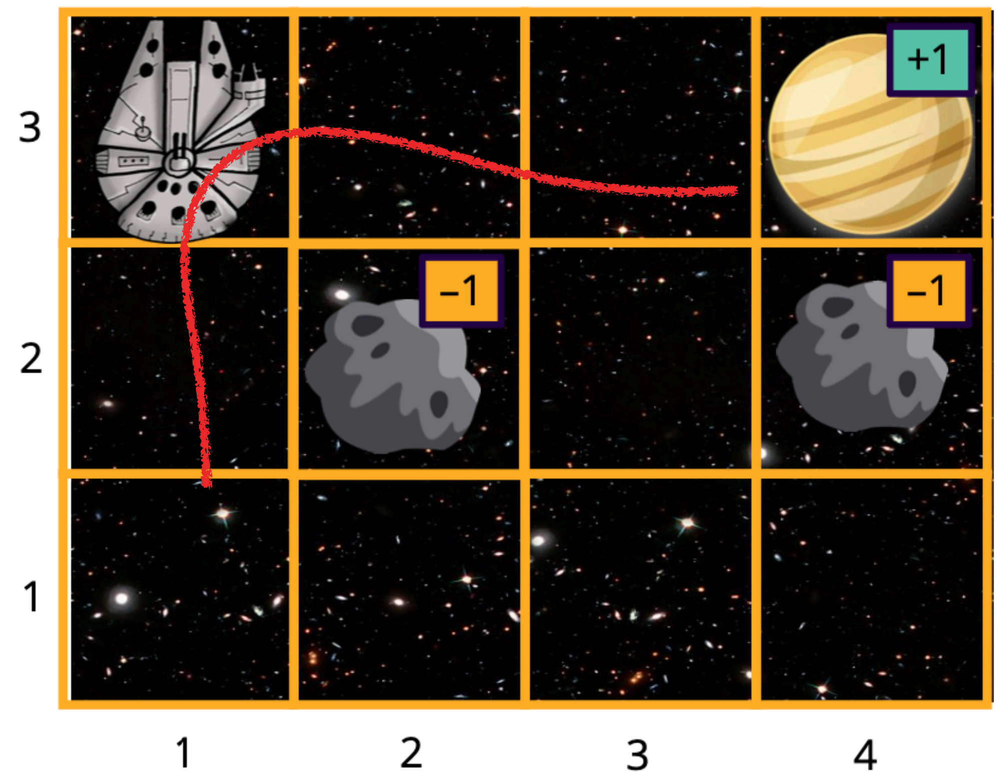
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.8 * 0.8

Transition Model:





Example borrowed from Chris Callison-Burch

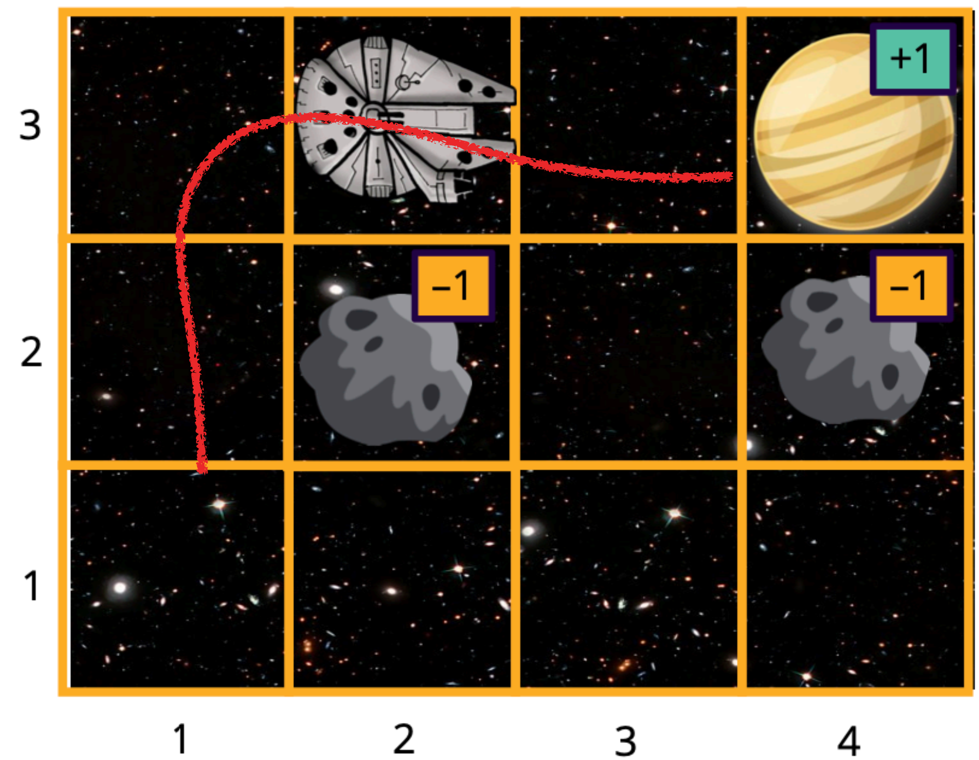
Navigating an Asteroid Field

For action sequence

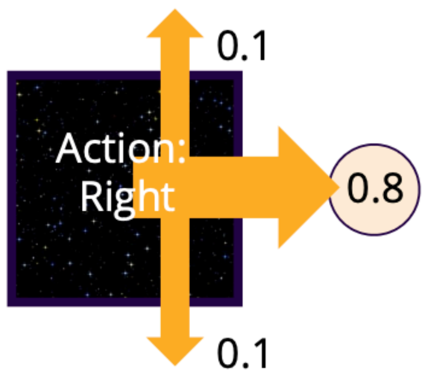
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.8 * 0.8 * 0.8



Transition Model:



Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

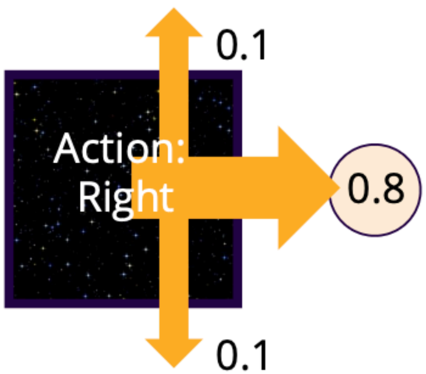
For action sequence

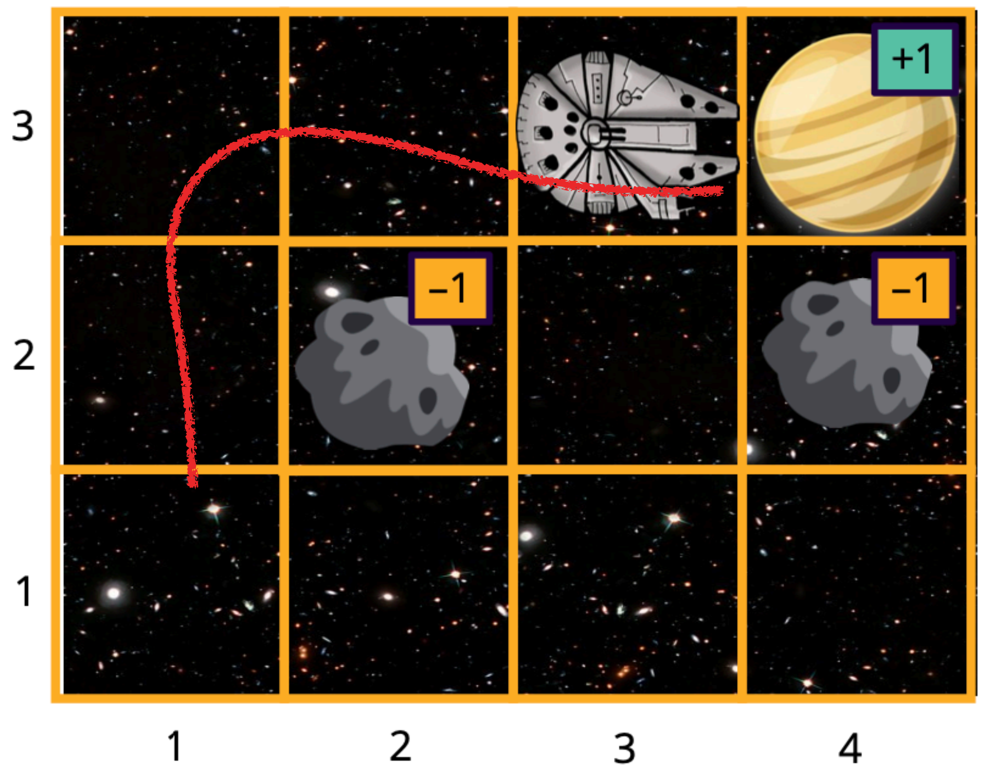
[*Up*, *Up*, *Right*, *Right*, *Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.8 * 0.8 * 0.8 * 0.8

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

For action sequence

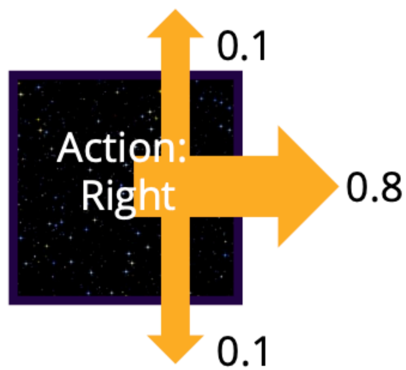
[*Up, Up, Right, Right, Right*],

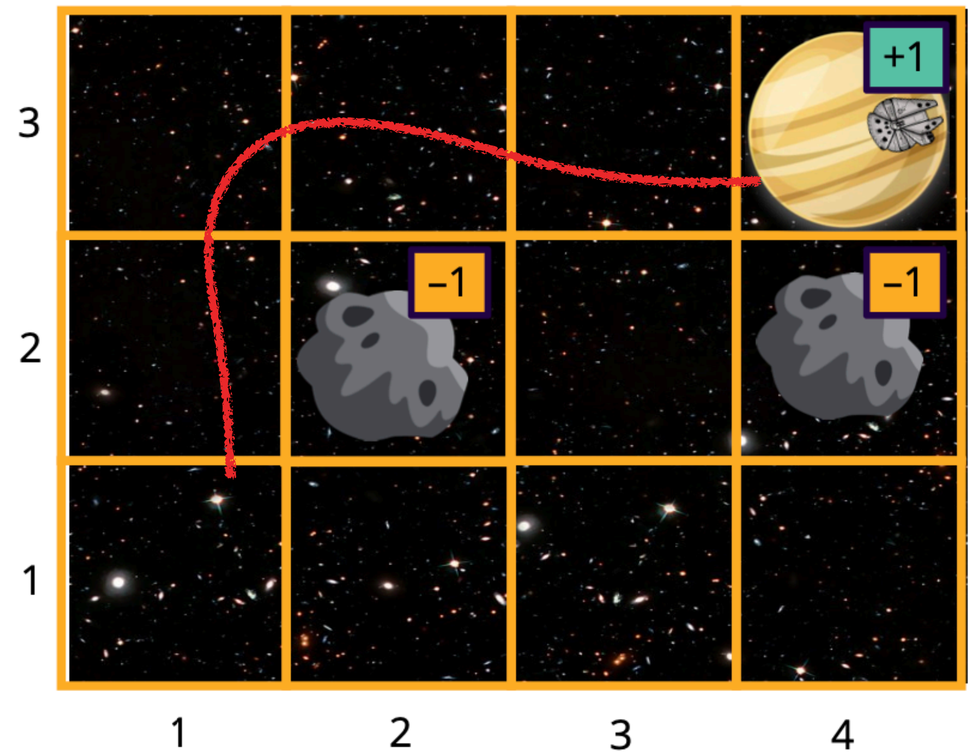
what's the probability that the millennium falcon reaches the intended goal?

0.8 * 0.8 * 0.8 * 0.8 * 0.8

= 0.32768

Transition Model:





Example borrowed from Chris Callison-Burch

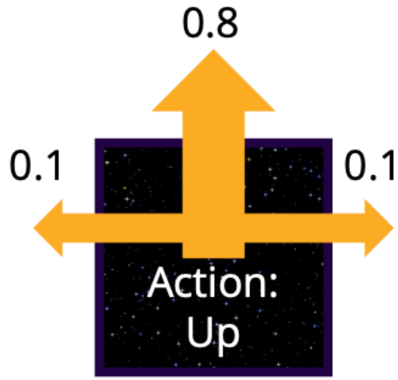
Navigating an Asteroid Field

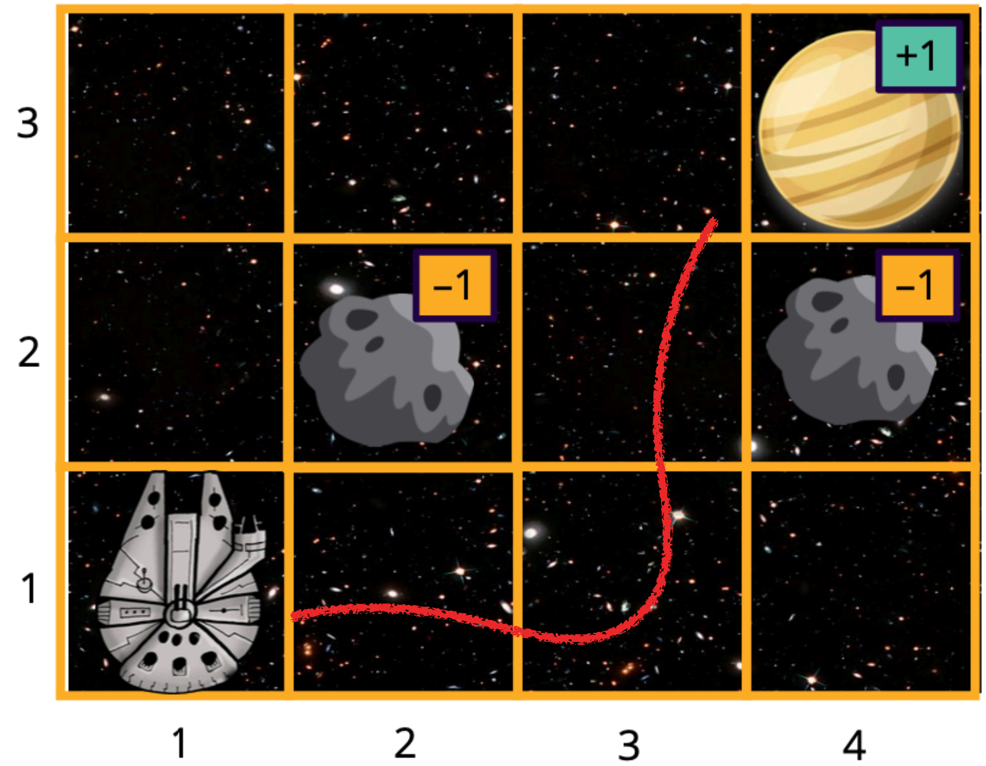
For action sequence

[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

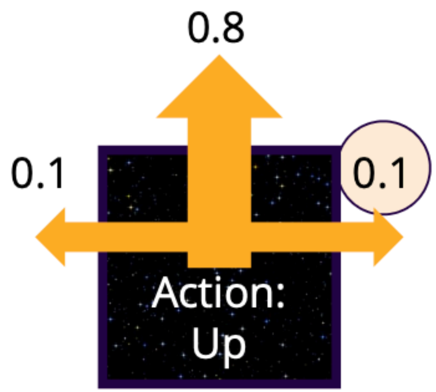
For action sequence

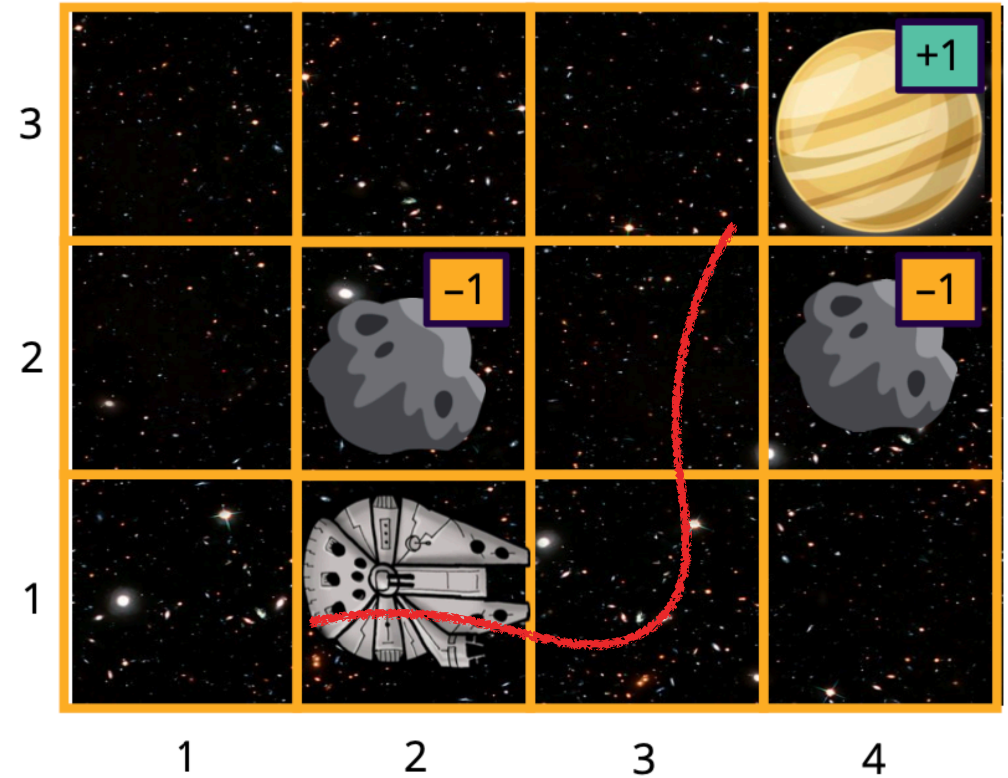
[Up, Up, Right, Right, Right],

what's the probability that the millennium falcon reaches the intended goal?

0.1

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

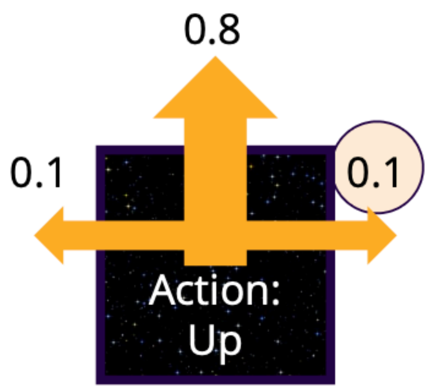
For action sequence

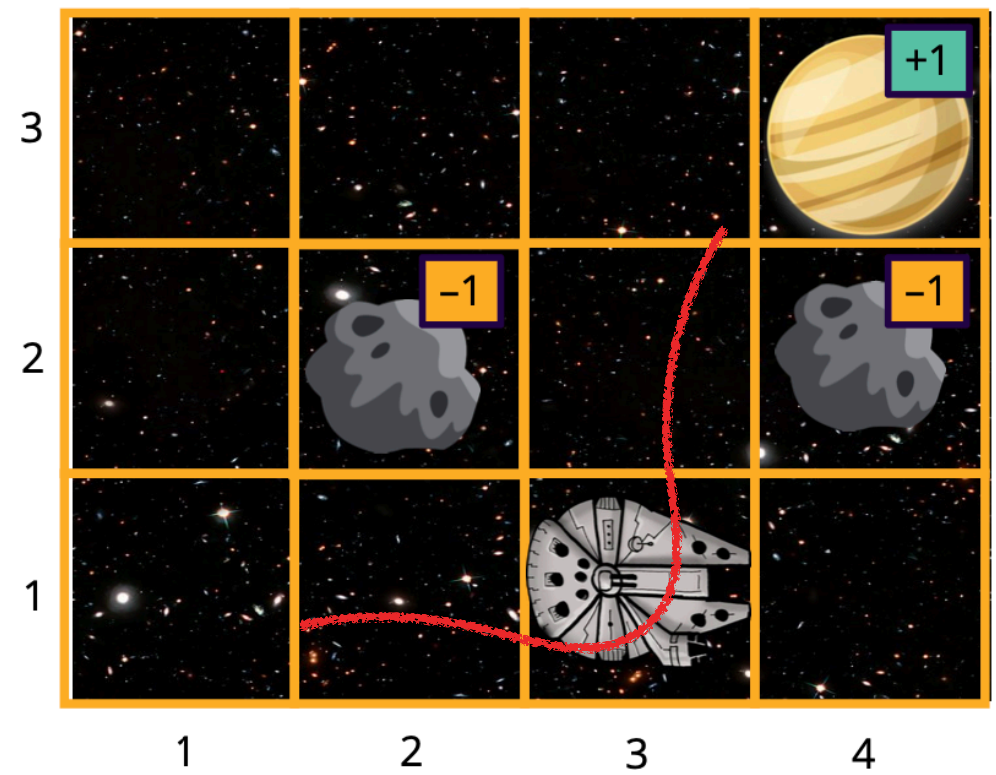
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.1 * 0.1

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

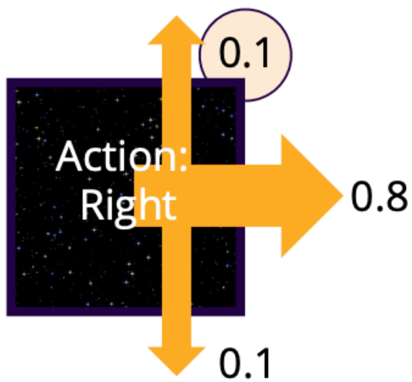
For action sequence

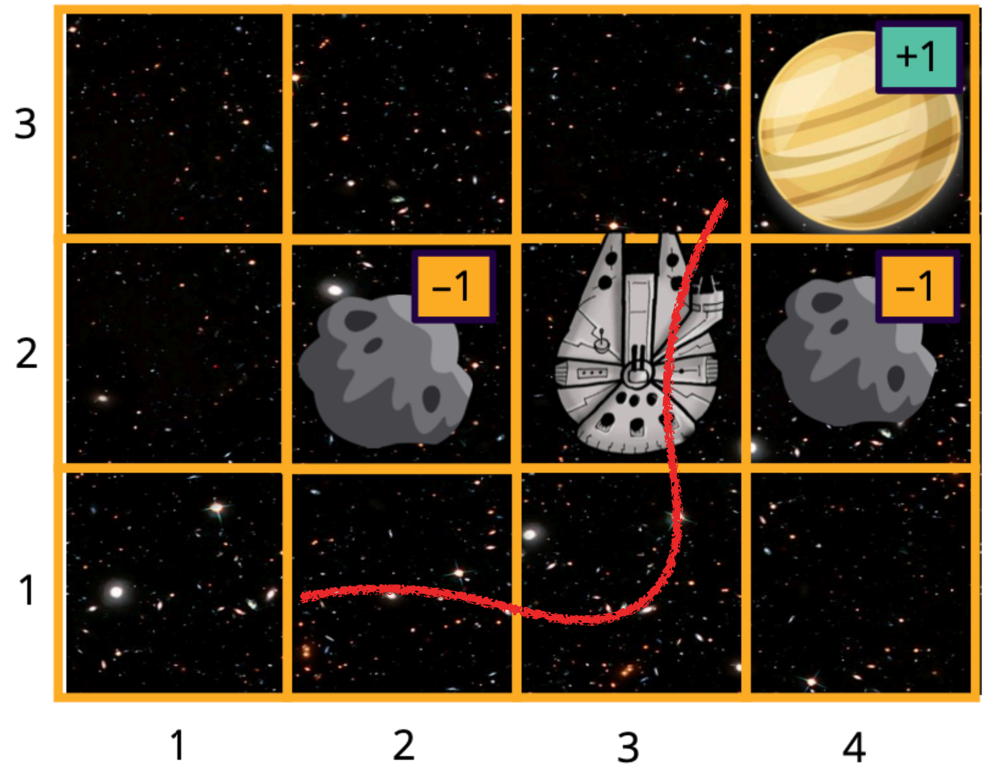
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.1 * 0.1 * 0.1

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

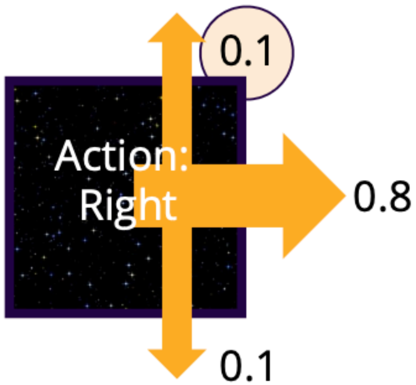
For action sequence

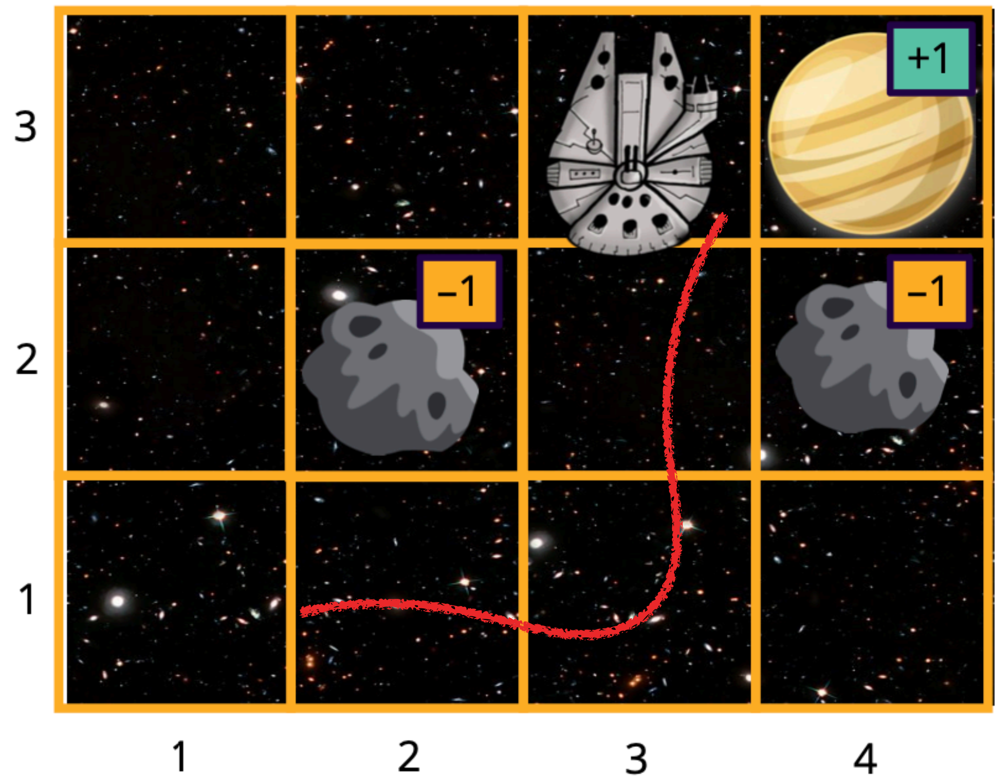
[*Up, Up, Right, Right, Right*],

what's the probability that the millennium falcon reaches the intended goal?

0.1 * 0.1 * 0.1 * 0.1

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

For action sequence

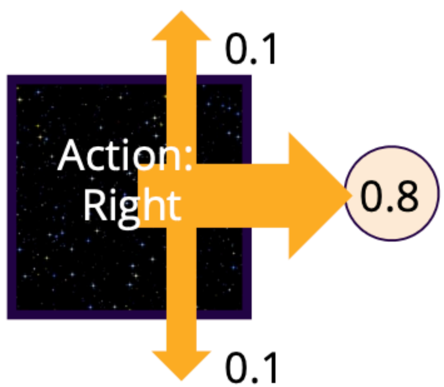
[*Up, Up, Right, Right, Right*],

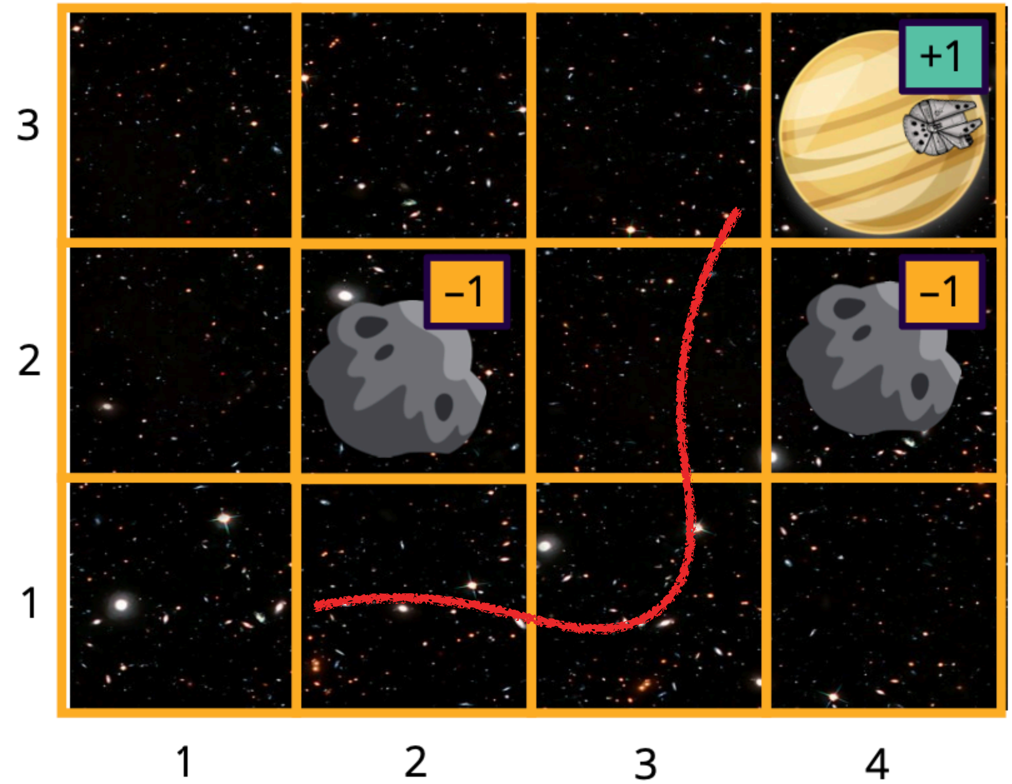
what's the probability that the millennium falcon reaches the intended goal?

$$0.1 * 0.1 * 0.1 * 0.1 * 0.8$$

$$= 0.00008$$

Transition Model:





Example borrowed from Chris Callison-Burch

Navigating an Asteroid Field

For action sequence

[*Up, Up, Right, Right, Right*],

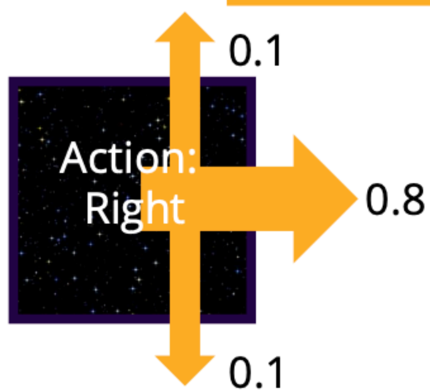
what's the probability that the millennium falcon reaches the intended goal?

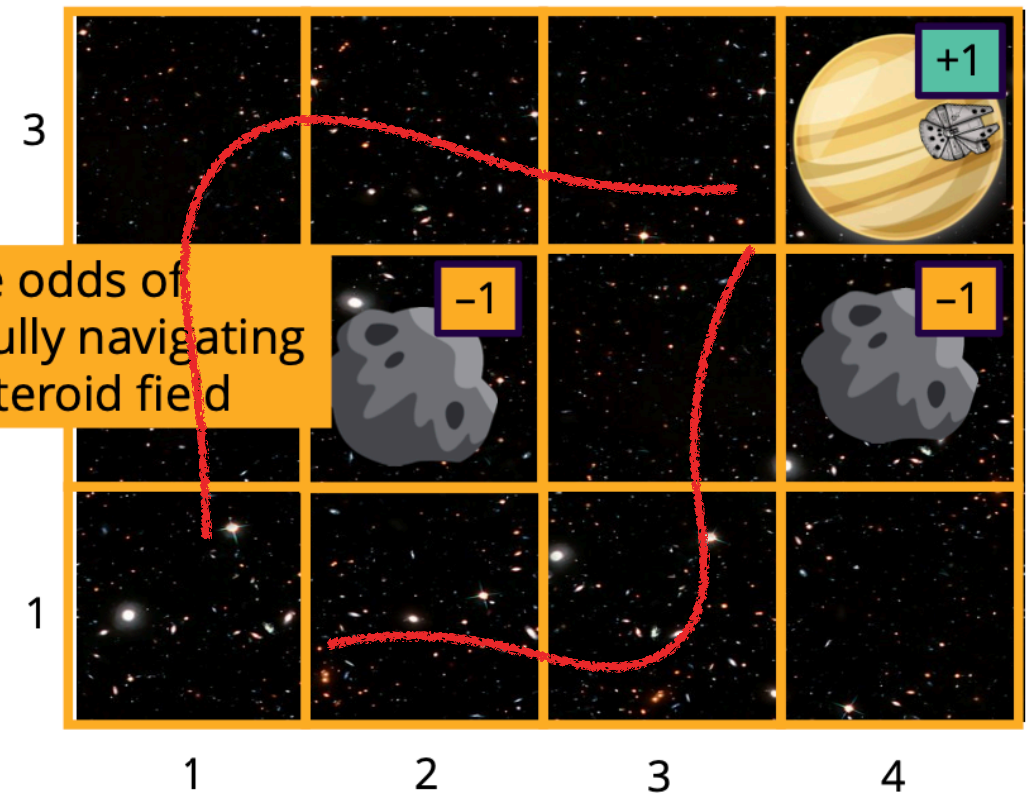
0.32768 + 0.00008

= 0.32776

The odds of successfully navigating an asteroid field

Transition Model:





Example borrowed from Chris Callison-Burch

Solution → Policy

It's harder to plan now that our transition function is stochastic. Instead, we should come up with a **policy**: a function that tells us what to do in every situation.

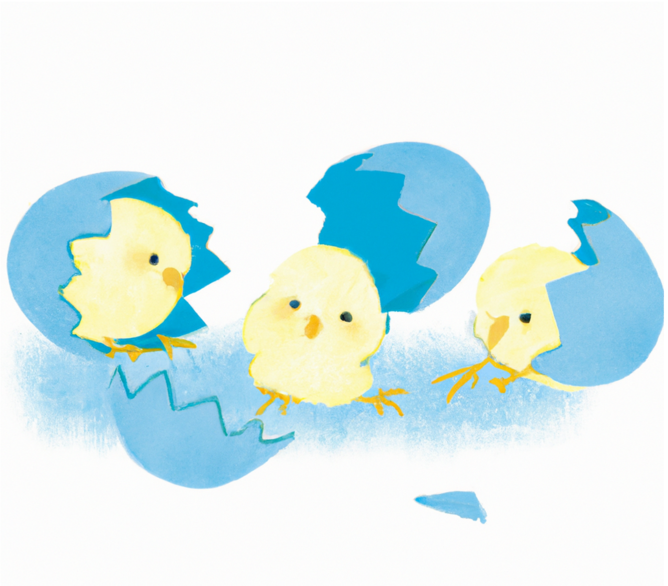
$$\pi(s) \rightarrow a$$

If the same policy is enacted multiple times, it may lead to different rewards. We care about the **expected utility of a policy**.

Rewards Are Uncertain

When outcomes are uncertain, future rewards are uncertain too. After all, we may never reach them!

For this reason, we typically **discount future rewards**: we set a **weight decay** that decreases the value of rewards based on how far out they are in the future.



Reinforcement Learning

Transitions Are Unknown

In Markov Decision Processes, the outcomes were uncertain, but not completely unknown.

In Reinforcement Learning, we have to **discover** the transition function and the rewards associated with each state.

Reinforcement Learning

- We still assume the following components:

A set of states $s \in S$

A set of actions per state : A

$T(s, a, s')$

$R(s, a, s')$

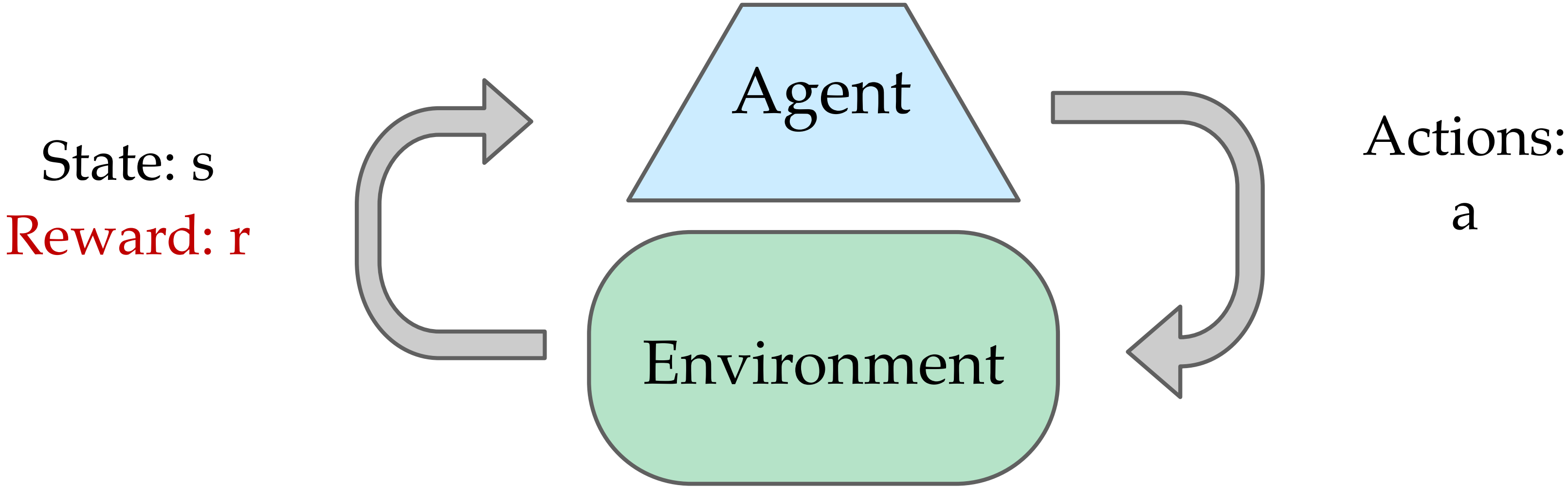
} unknown initially

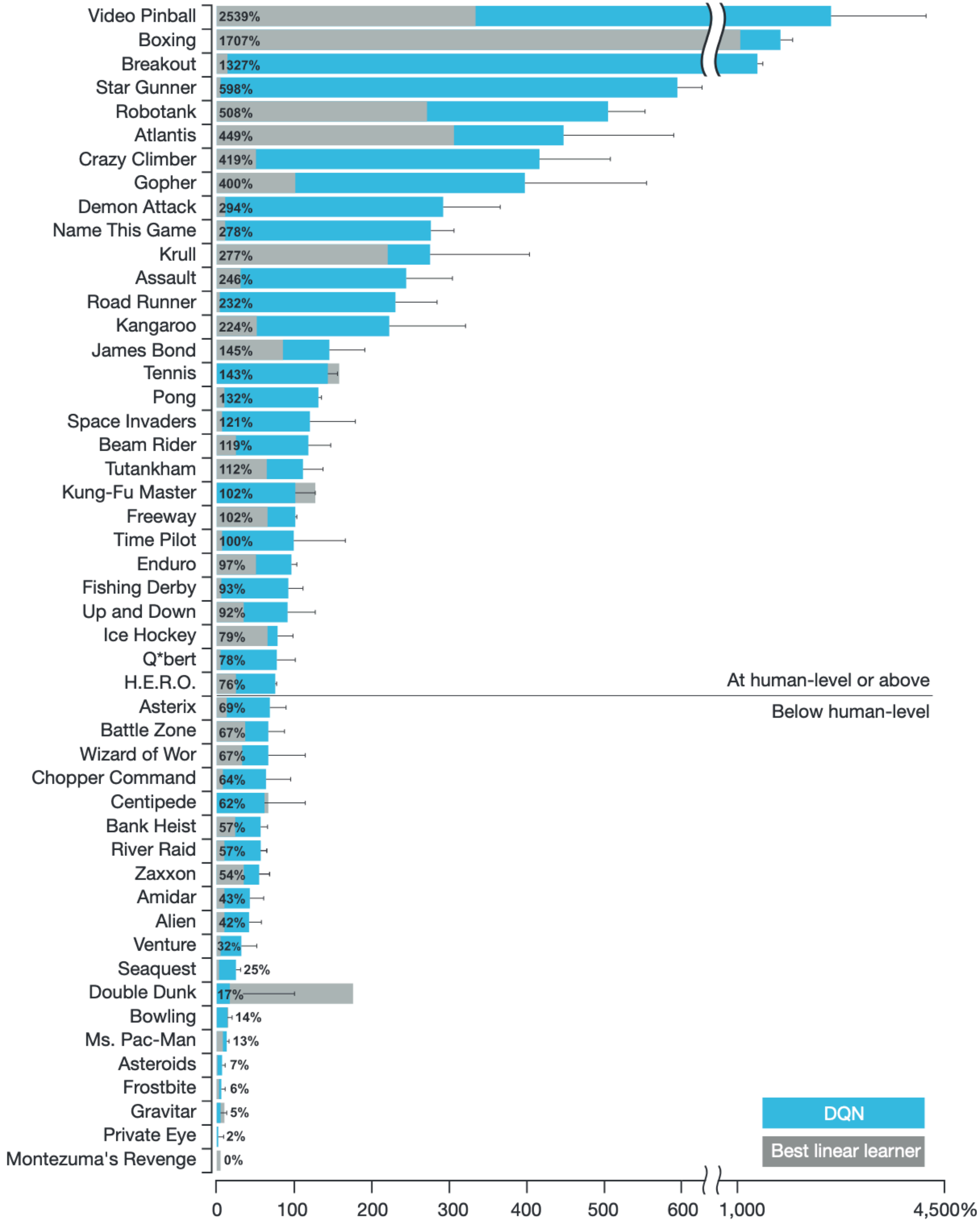
we have to estimate them

Reinforcement Learning

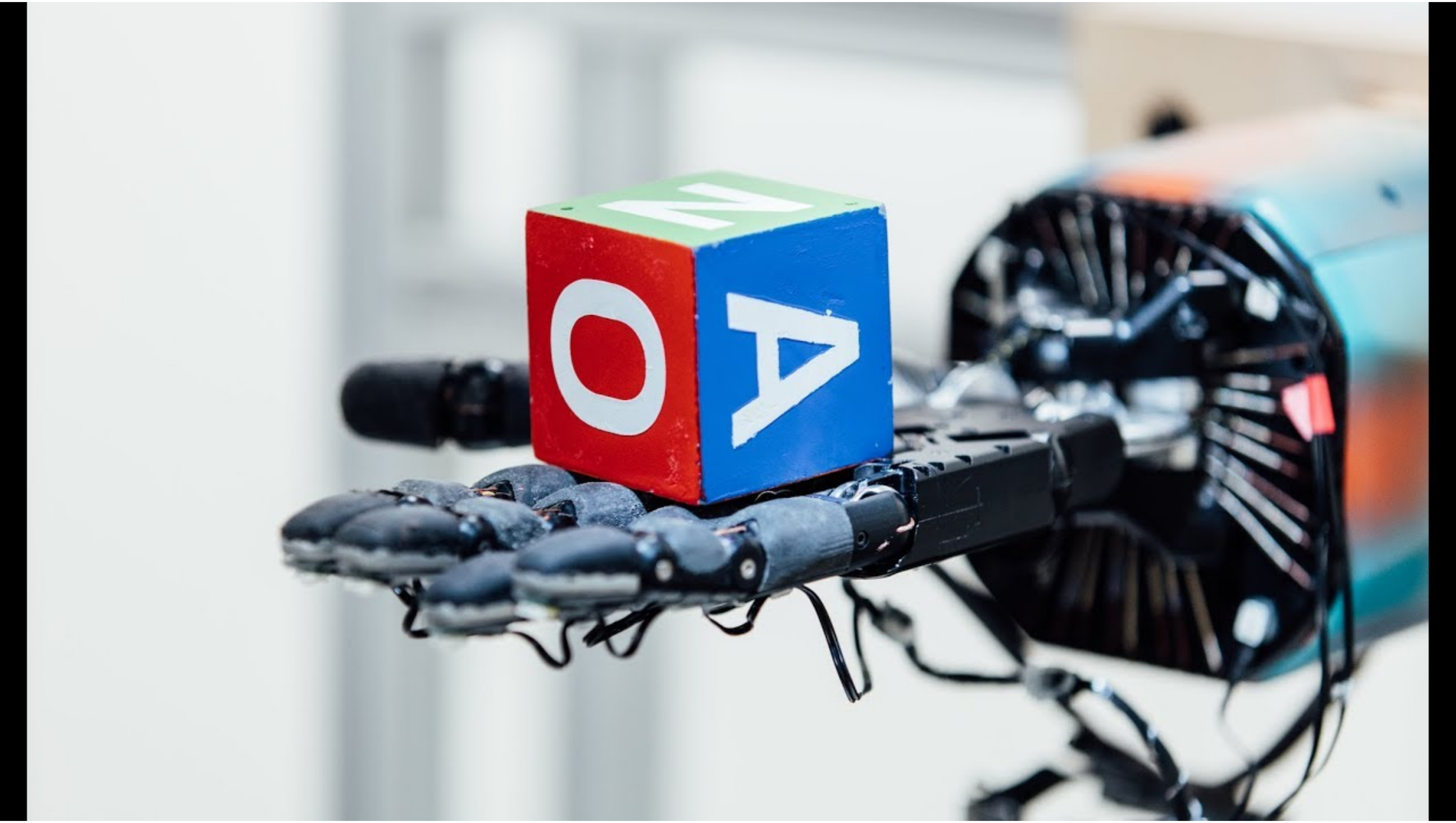
Basic idea:

- Agent receives feedback in the form of rewards
- Agent tries to learn to act in order to maximize expected rewards
- Agent learns from samples of observed outcomes (AKA: try something out and see what happens!)





Mnih et al. 2015



Exploration Versus Exploitation

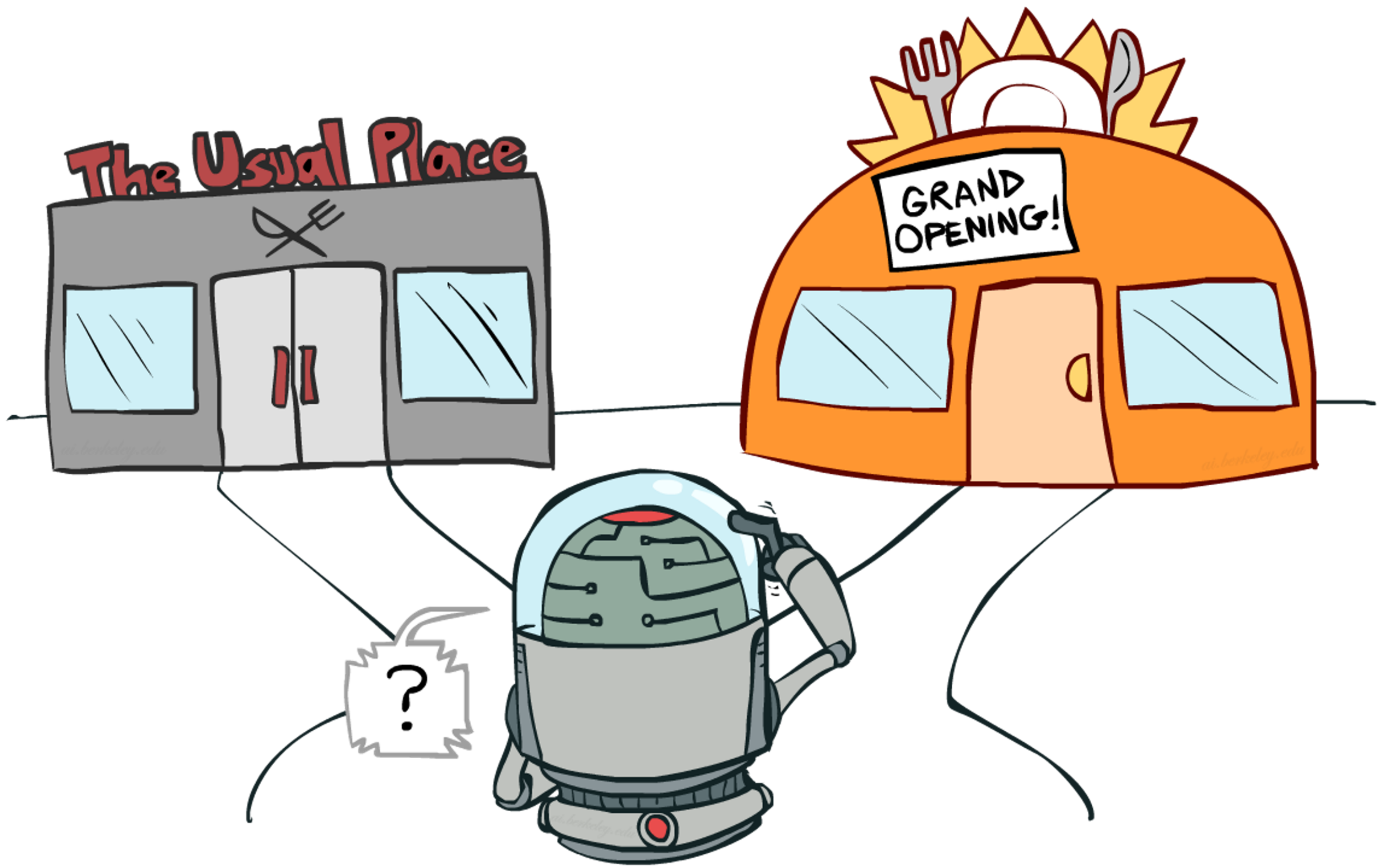


Image from CS188 Intro to AI at UC Berkeley

How To Explore

Passive Reinforcement Learning

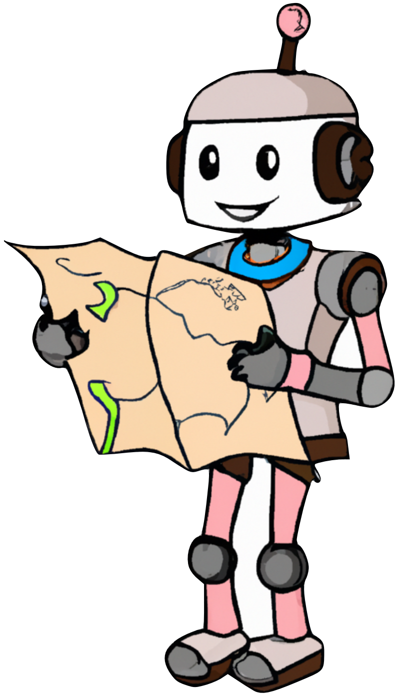
- Take a fixed policy and follow it

Random Exploration

- Pick actions randomly

Balancing Exploration and Exploitation

- Pick actions randomly initially, but gradually switch to mostly doing actions that you have already found to be valuable



How To Evaluate

In Direction Evaluation, we repeatedly follow the same policy and observe its rewards.

Direct Evaluation:

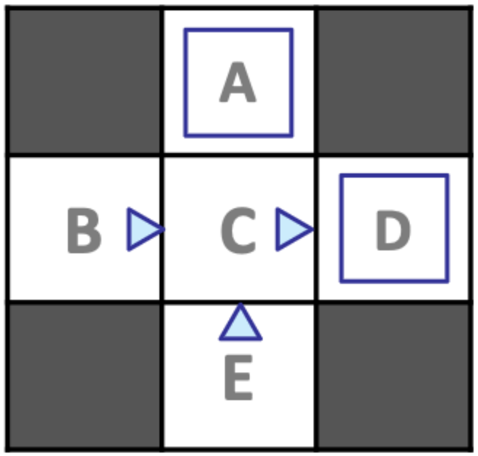
Act according to policy π

In each state, we write down the sum of discounted rewards

Average samples

Example: Direct Evaluation

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

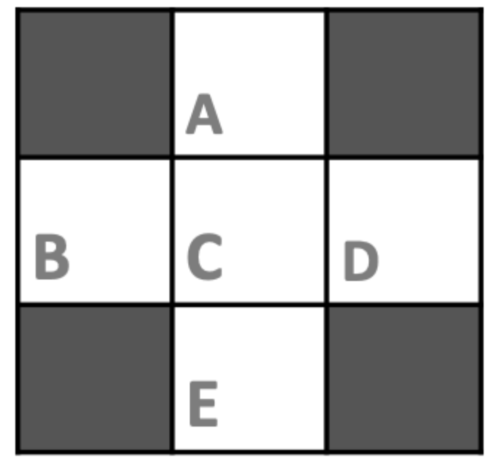
Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values

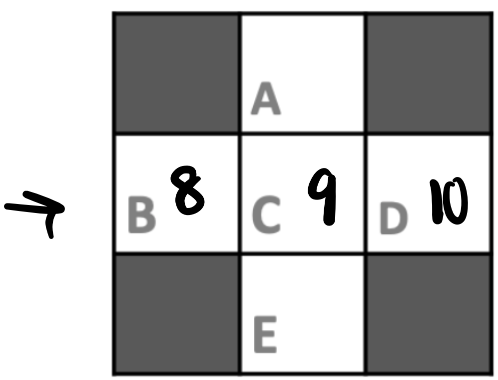


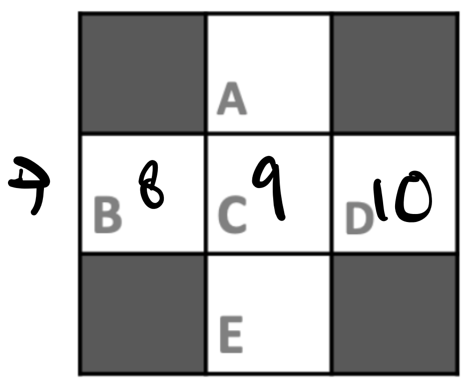
Example from CS188 Intro to AI at UC Berkeley

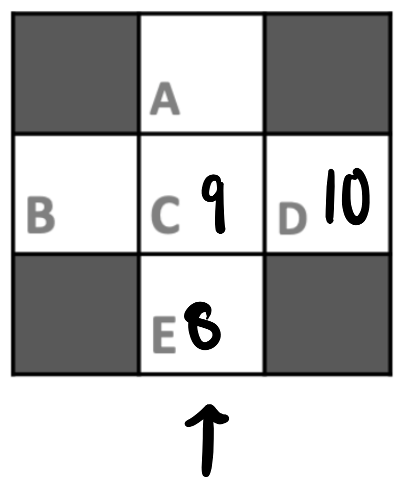
Example: Direct Evaluation

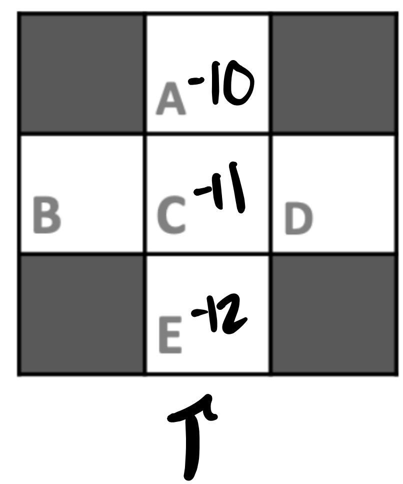
Episode 1	Episode 2	Episode 3	Episode 4
B, east, C, -1	B, east, C, -1	E, north, C, -1	E, north, C, -1
C, east, D, -1	C, east, D, -1	C, east, D, -1	C, east, A, -1
D, exit, x, +10	D, exit, x, +10	D, exit, x, +10	A, exit, x, -10

γ : discount future rewards $\quad \gamma = 1$



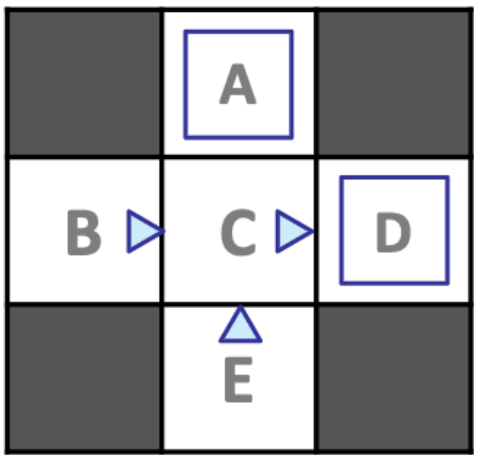






Example: Direct Evaluation

Input Policy π



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

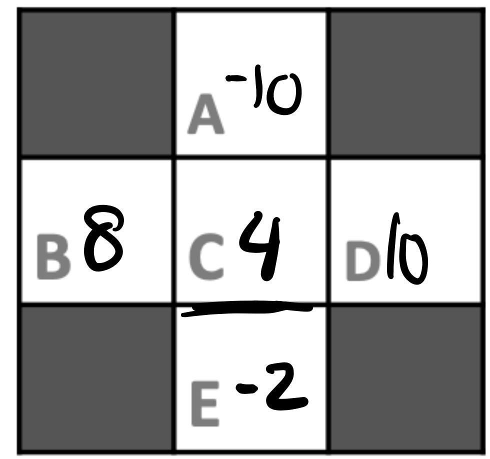
Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values



Example from CS188 Intro to AI at UC Berkeley

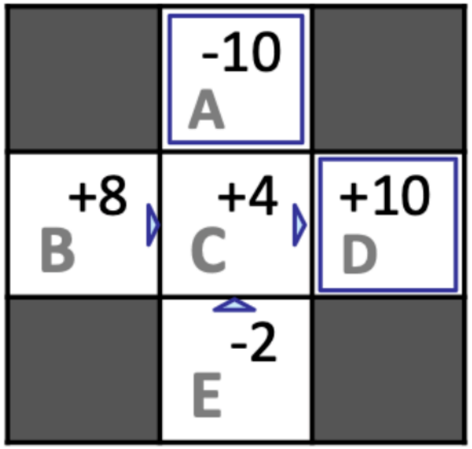
Weaknesses of Direct Evaluation

Direct Evaluation works.

But, we had to learn each state separately.

That isn't very efficient!

Output Values



If B and E both go to C under this policy, how can their values be different?

Temporal Difference Learning

Temporal Difference Learning

Main Idea: learn from every experience

TD Learning

- Update V(s) every time we make a transition T(s,a,s',r)
- Most likely outcomes (s') will contribute more updates

Sample of V(s): $sample = R(s, \pi(s), s') + \gamma V^{\pi}(s')$

Update to V(s): $V^{\pi}(s) \leftarrow (1 - \alpha)V^{\pi}(s) + (\alpha)sample$

Same update: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$

TD: A Moving Average

In TD Learning, we average our observations in a way that weights more recent samples highly.

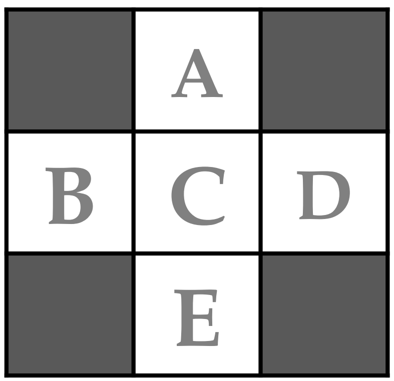
Over time, we forget our initial estimates (which probably weren't very good!)

$$\bar{x}_n = \frac{x_n + (1-\alpha) \cdot x_{n-1} + (1-\alpha)^2 \cdot x_{n-2} + \dots}{1 + (1-\alpha) + (1-\alpha)^2 + \dots}$$

We decrease the **learning rate α** over time.

Example: Temporal Difference Learning

States



Assume:

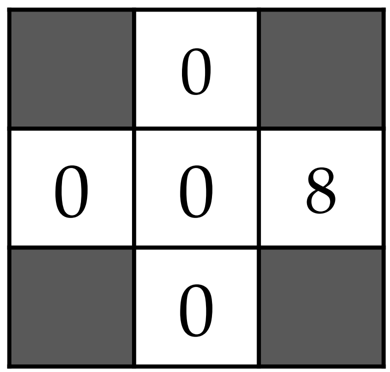
$\gamma = 1,$

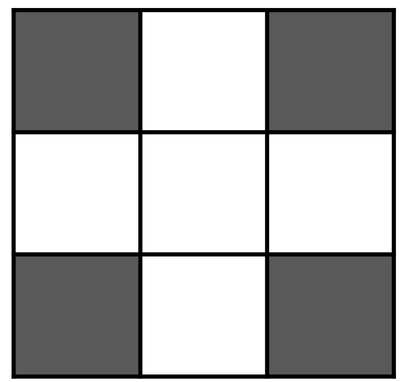
$\alpha = 1/2$

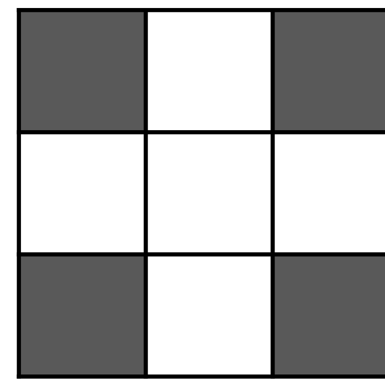
Observed Transitions

B, east, C, -2

C, east, D, -2







$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha \left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

Example from CS188 Intro to AI at UC Berkeley

Example: Temporal Difference Learning

States

	A	
B	C	D
	E	

Assume:

$\gamma = 1,$

$\alpha = 1/2$

Observed Transition:

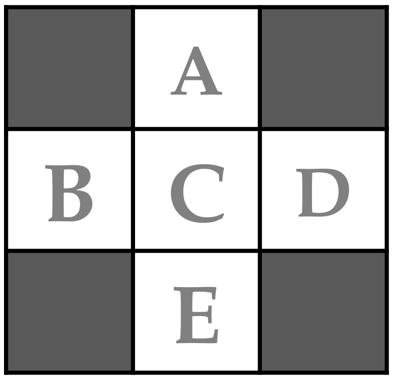
B, east, C, -2

	0	
0	0	8
	0	

$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

Example: Temporal Difference Learning

States



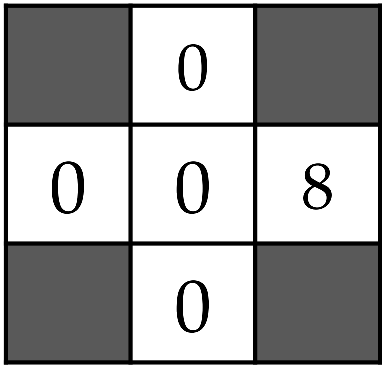
Assume:

$\gamma = 1,$

$\alpha = 1/2$

Observed Transition:

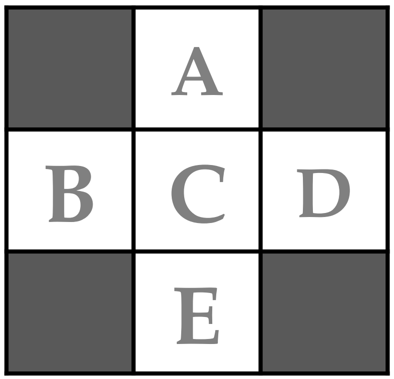
C, east, D, -2



$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

Example: Temporal Difference Learning

States



Assume:

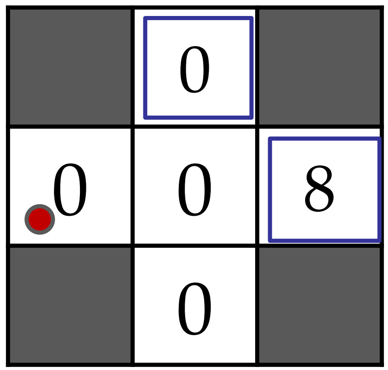
$\gamma = 1,$

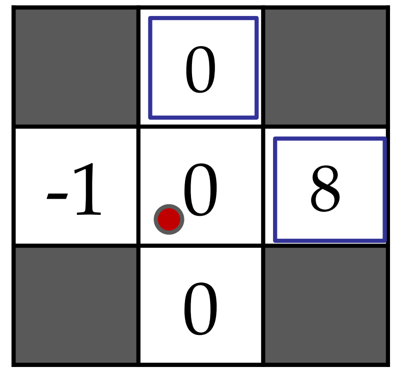
$\alpha = 1/2$

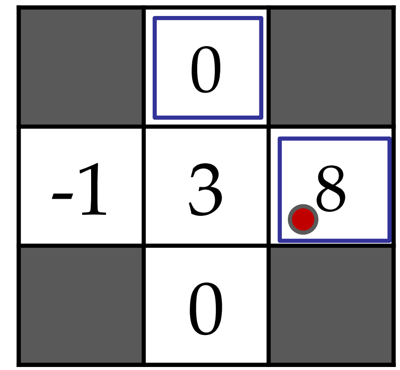
Observed Transitions

B, east, C, -2

C, east, D, -2





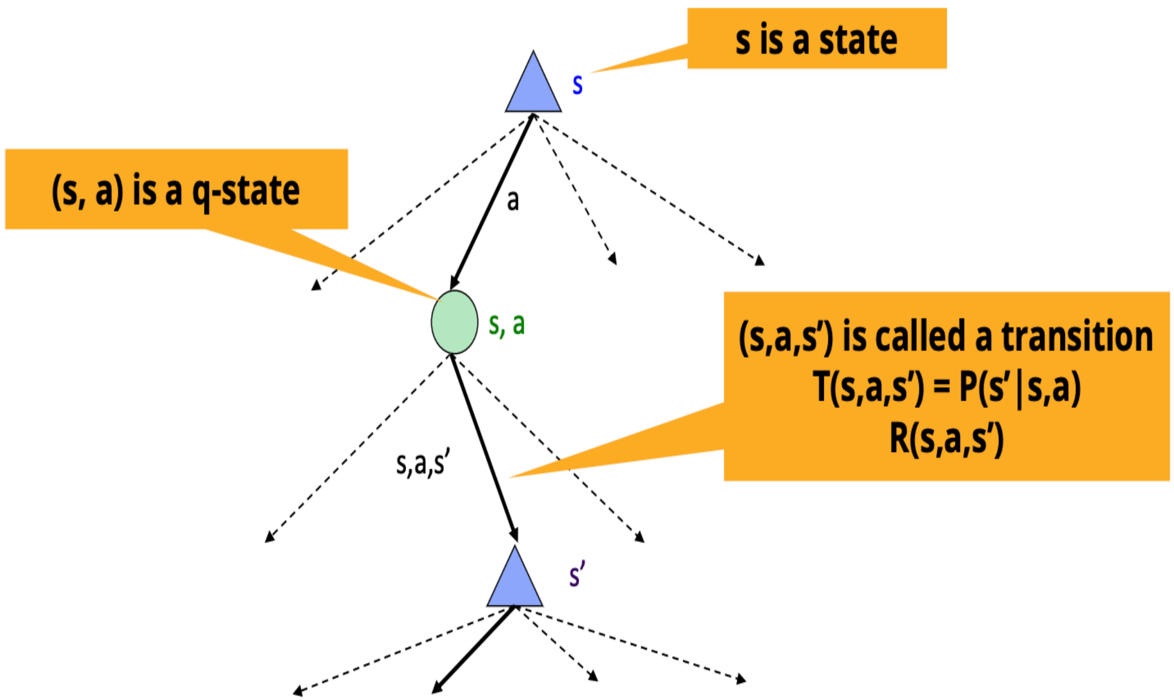


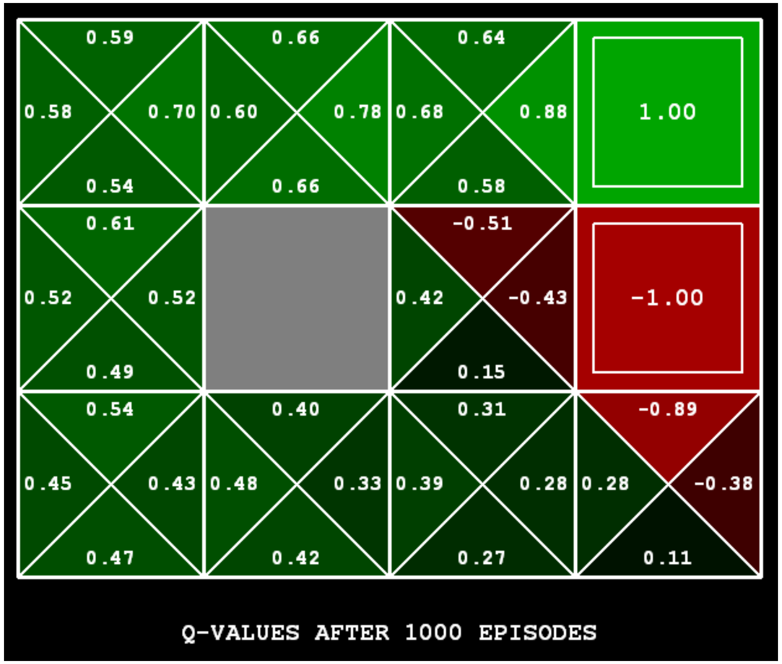
$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

Q-Learning

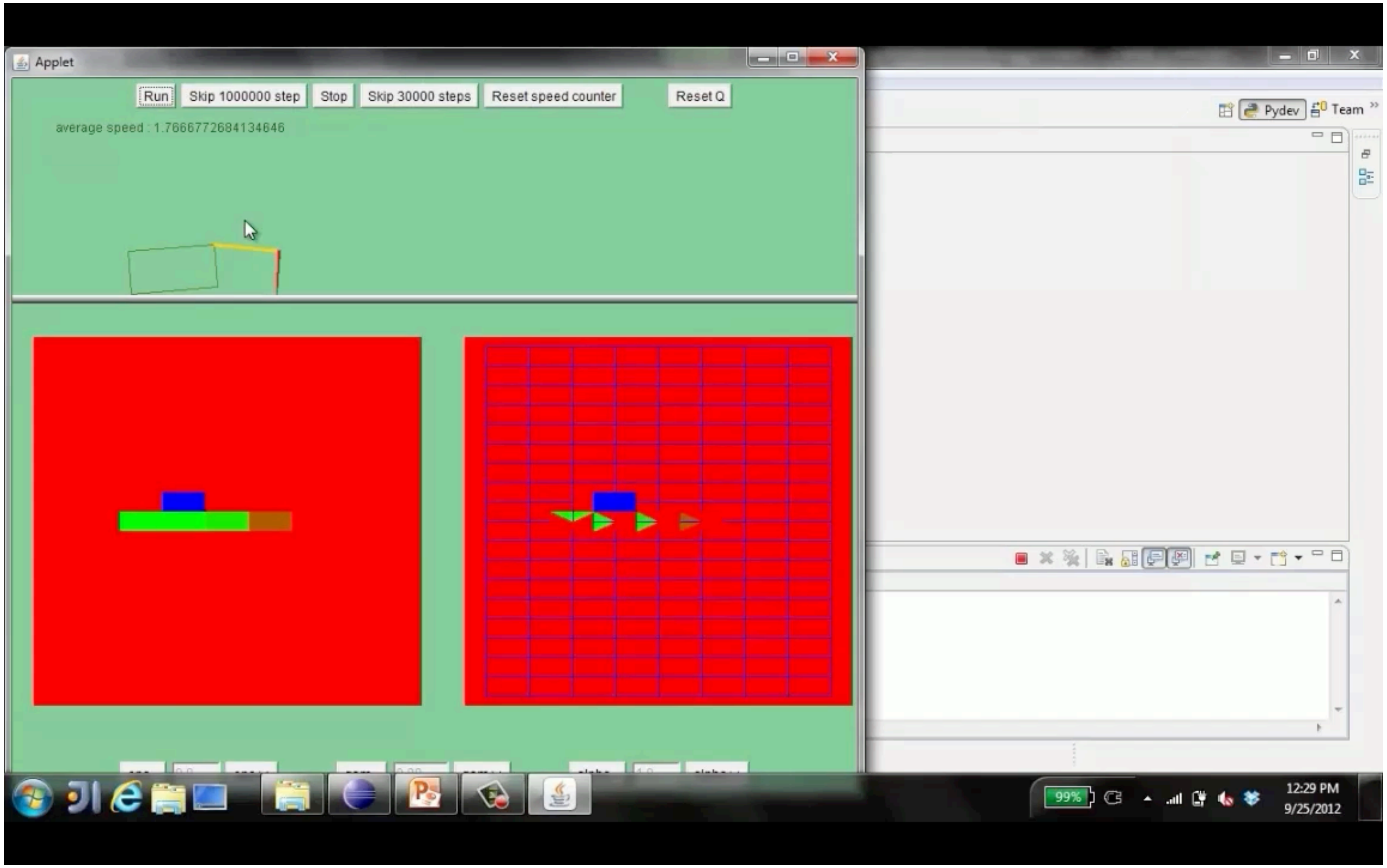
In practice, we want to know the value of state-action pairs, not just states.

This version of TD Learning is called **Q-Learning**, because the value of a state-action pair is called a **q-value**.





Demo of Q-Learning



Regret

Even if our Reinforcement Learning agent learns an optimal policy eventually, it will still take sub-par actions along the way.

Regret is the total cost of all of those mistakes: the differences between optimal expected rewards and the agent's actual rewards.

The best RL agent is one that **minimizes regret**: learns optimally!