Warm-up as You Walk In

Given

- Set `actions` (persistent/static)
- Set `states` (persistent/static)
- Function `T(s,a,s_prime)`

Write the pseudo code for:

- `function V(s) return value`

that implements:

$$V(s) = \max_{a \in actions} \sum_{s' \in states} T(s,a,s')V(s')$$

Announcements

Assignments:

- P3: Optimization; Due 10/17 Thu, 10 pm
- HW6 (online) 10/15 Tue, 10 pm
- HW7 (online) 10/22 Tue, 10 pm; Will be released today

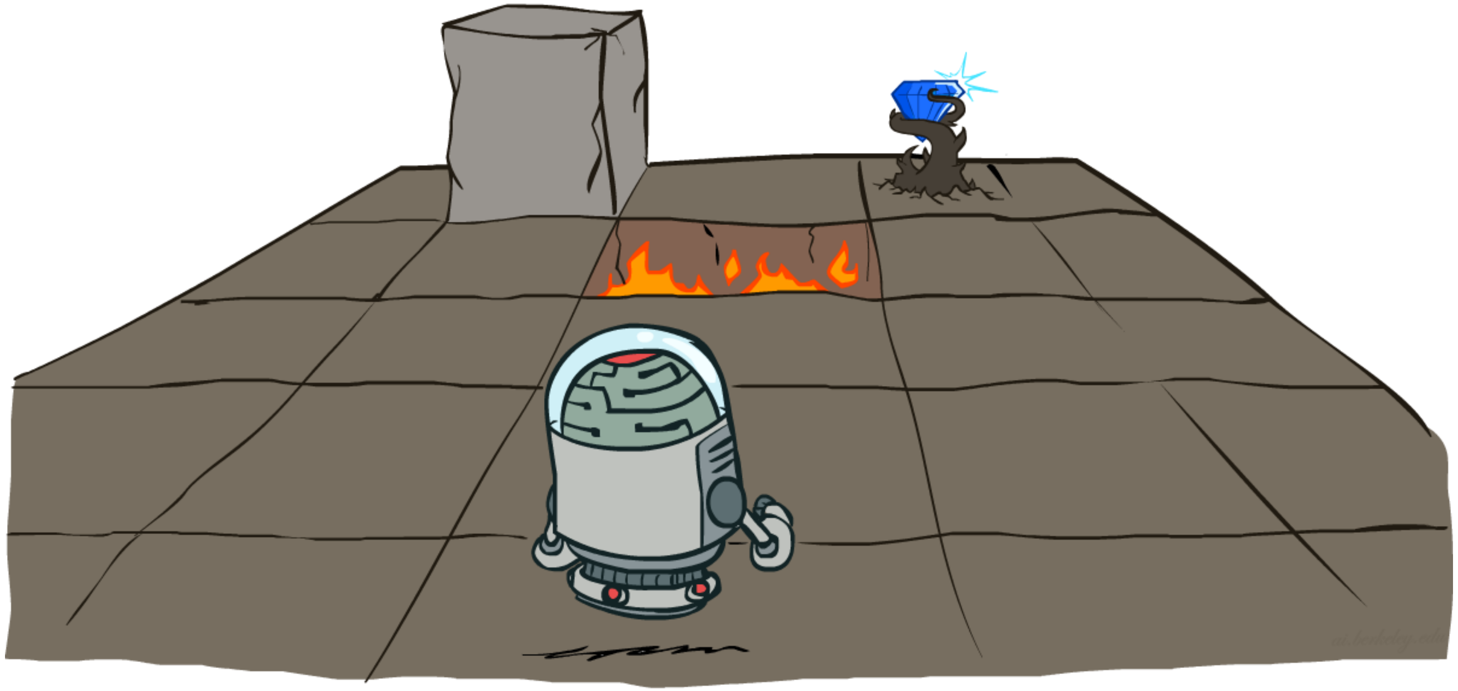
Lectures: 4 lectures by Dr. Fei Fang on MDP/RL, followed by 4 lectures by Dr. Pat Virtue on Bayes' Nets

Recitations canceled on October 18 (Mid-Semester Break) and October 25 (Day for Community Engagement). Recitation worksheet available (reference for midterm/final)

New: Piazza post for In-class Questions

AI: Representation and Problem Solving

Markov Decision Processes



Instructors: Fei Fang & Pat Virtue

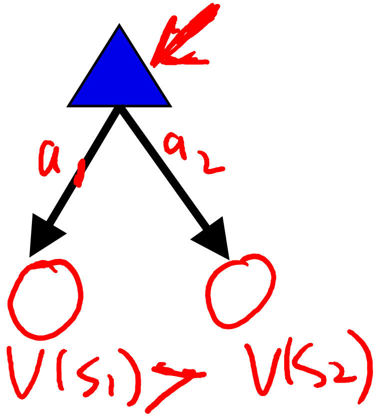
Slide credits: CMU AI and http://ai.berkeley.edu

Learning Objectives

- Describe the definition of Markov Decision Process
- Compute utility of a reward sequence given discount factor
- Define policy and optimal policy of an MDP
- Define state-value and (true) state value of an MDP
- Define Q-value and (true) Q value of an MDP
- Derive optimal policy from (true) state value or (true) Q-values

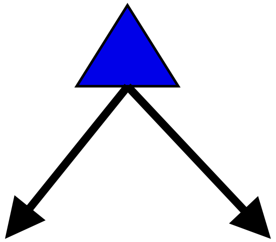
Next Lecture

Recall: Minimax Notation



$$V(s) = \max_a V(s'), = V(s_1)$$

where $s' = result(s, a)$



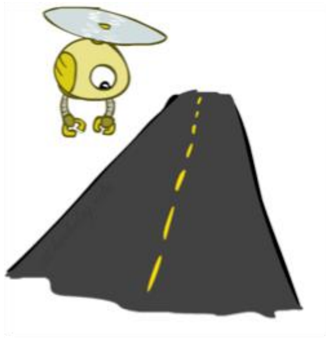
$$\hat{a} = \operatorname*{argmax}_a V(s'),$$

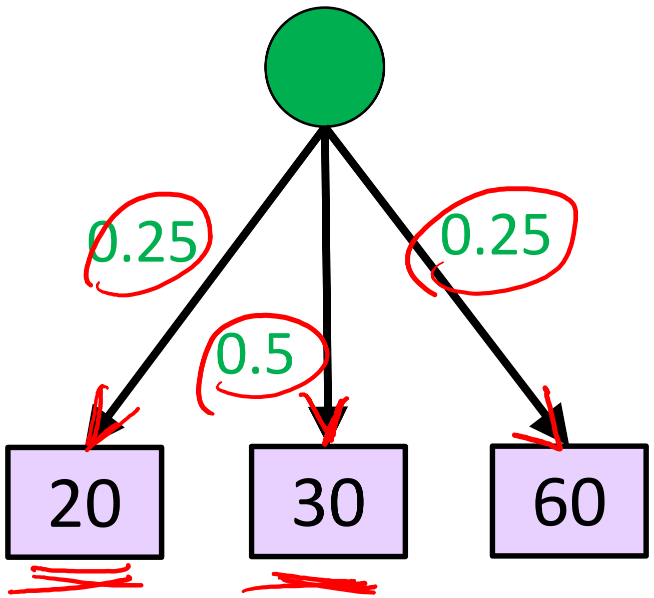
$$\hat{a} = \operatorname*{argmax}_a V(s') = a_1$$

where $s' = result(s, a)$

Recall: Expectations

Time:	20 min		30 min		60 min
	x	+	x	+	x
Probability:	0.25		0.50		0.25





Max node notation

$$V(s) = \max_a V(s'),$$

where $s' = result(s, a)$

Chance node notation

$$V(s) = \sum_{s'} P(s')\, V(s')$$

Piazza Poll 1

argmax → right

max → eight

$\frac{1}{3} \times 12 + \frac{2}{3} \times 6 = 8$

Expectimax tree search:

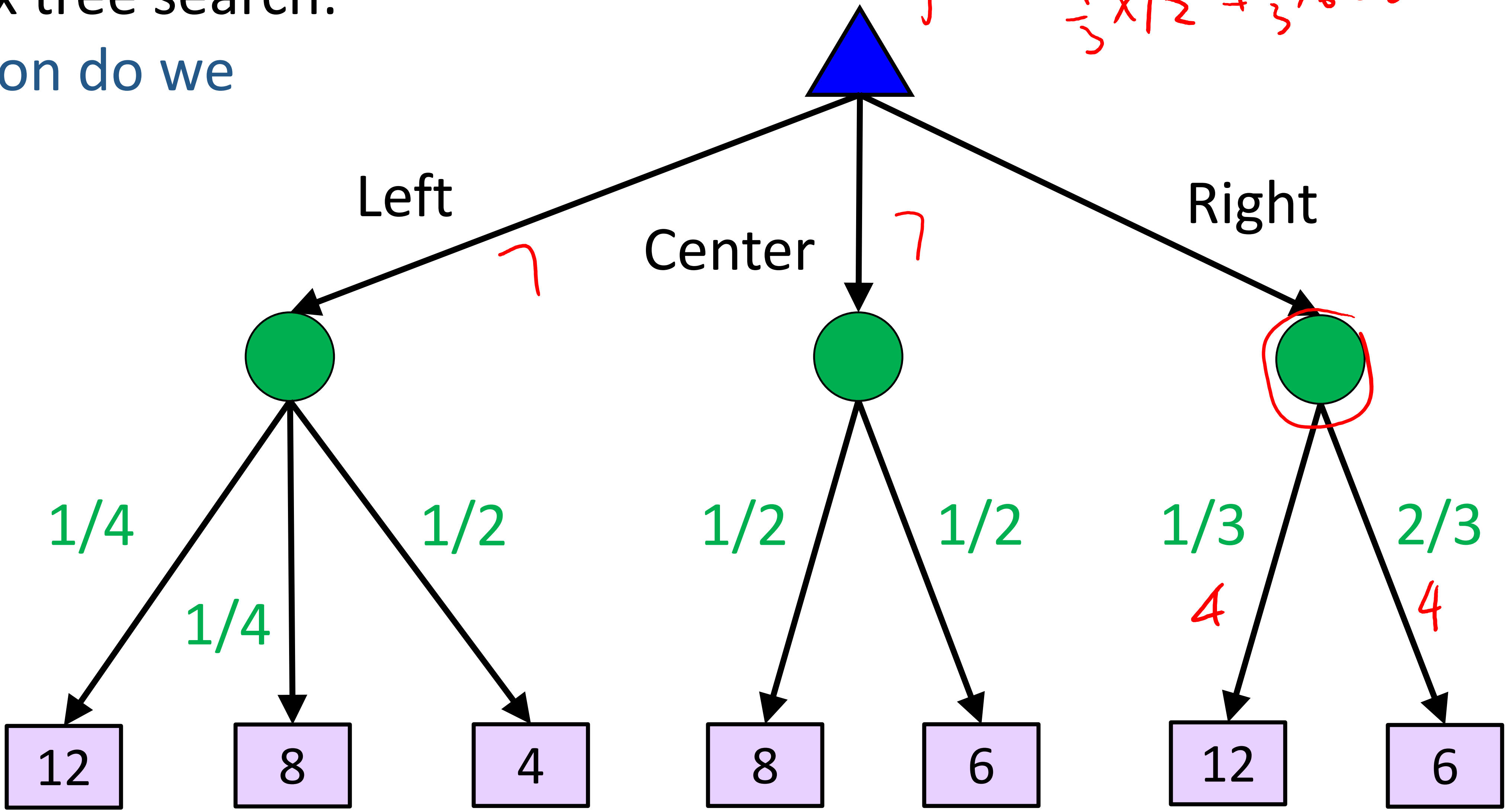
Which action do we choose?

A) Left

B) Center

C) Right

D) Eight



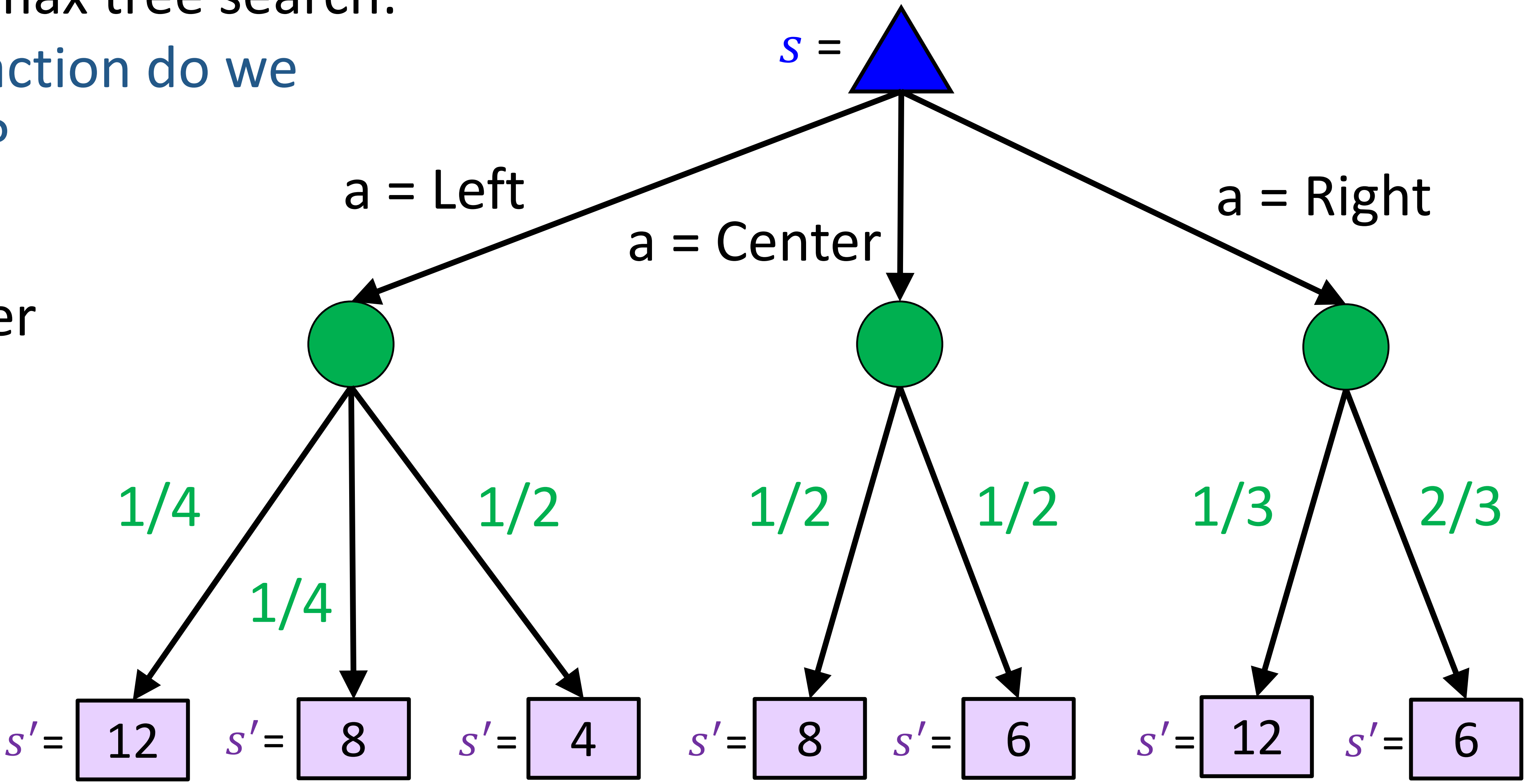
Piazza Poll 1

$$\hat{a} = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)\, V(s')$$

Expectimax tree search:

Which action do we choose?

A) Left
B) Center
C) Right
D) Eight



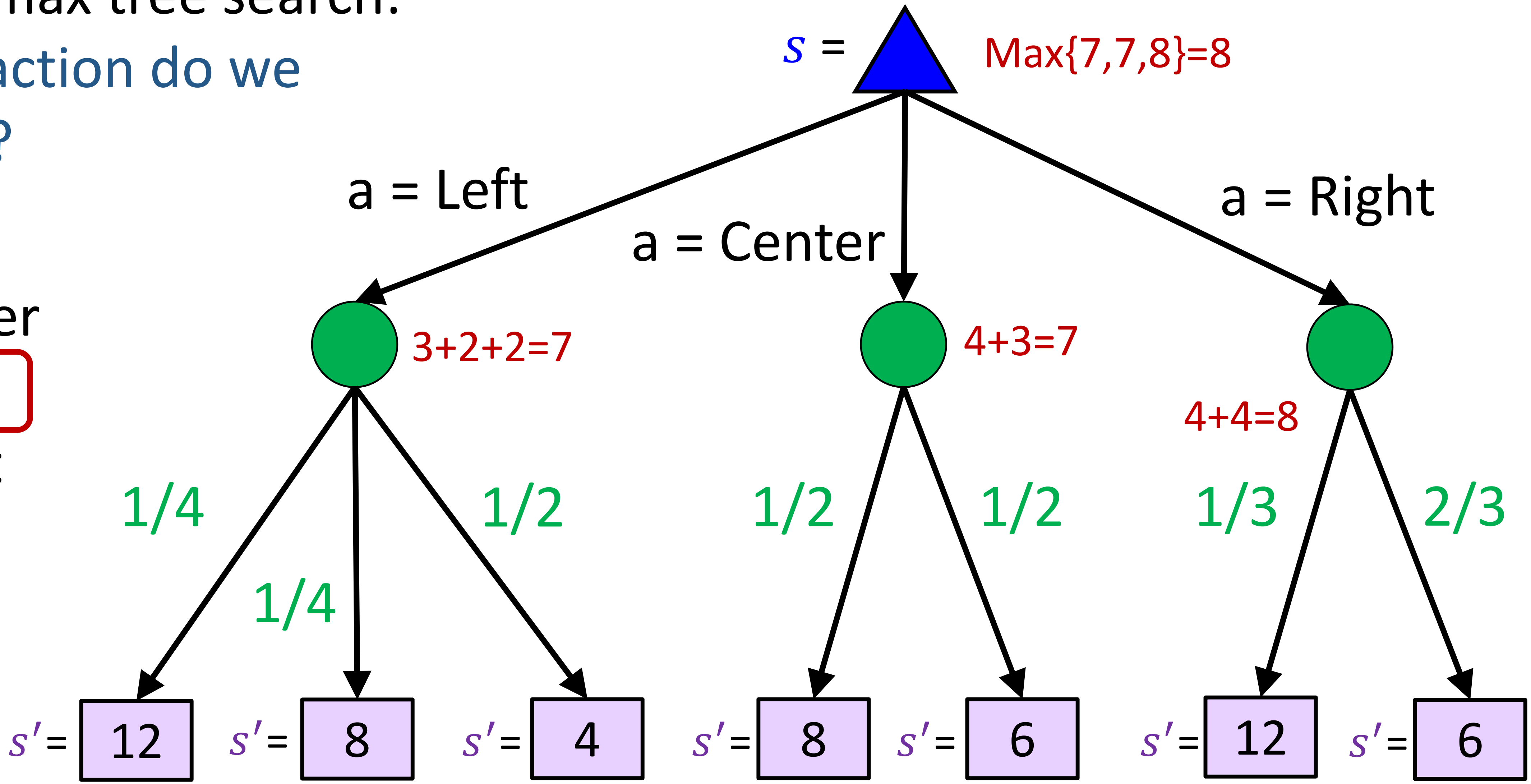
Piazza Poll 1

$$\hat{a} = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)\, V(s')$$

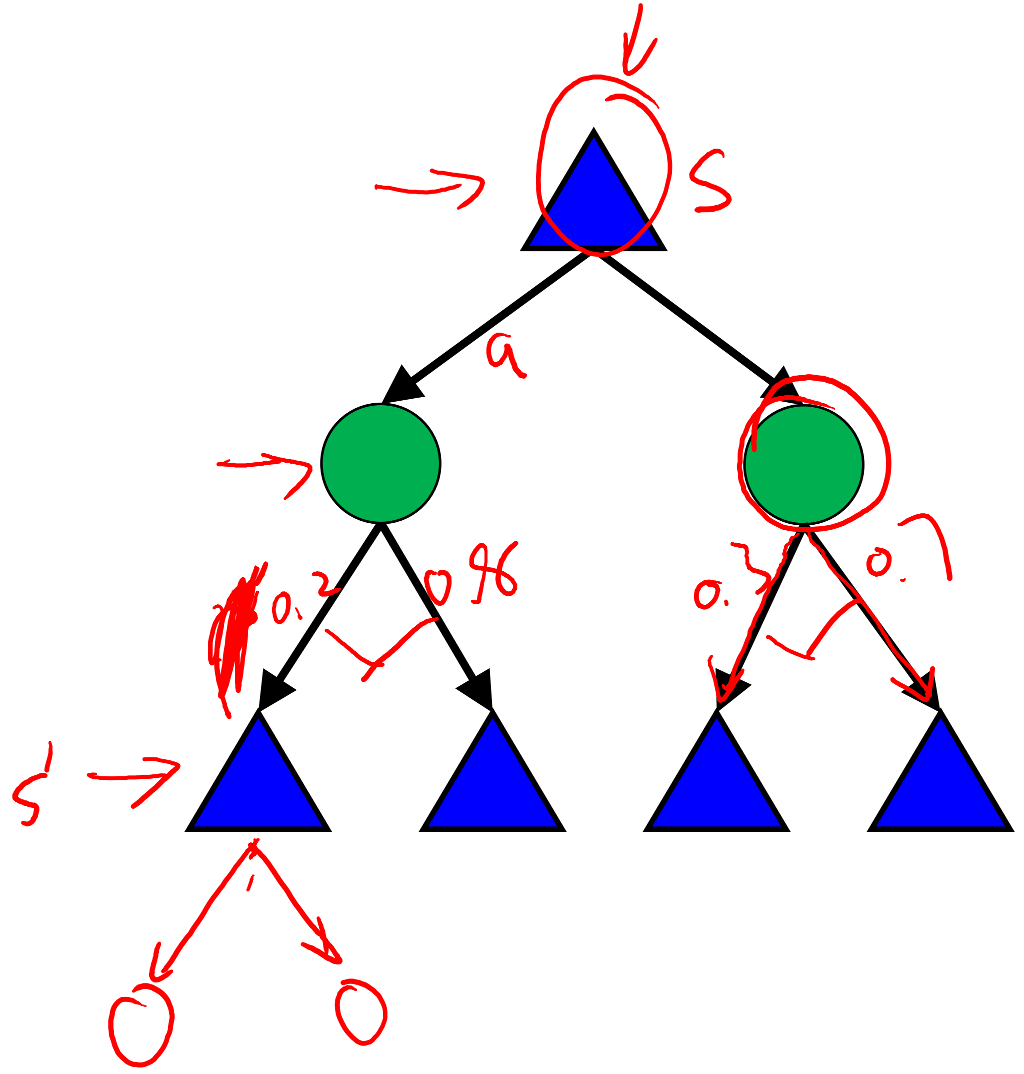
Expectimax tree search:

Which action do we choose?

A) Left
B) Center
C) Right
D) Eight



Expectimax Notation



$$V(s) = \max_{a} \sum_{s'} \boxed{P(s'|s,a)}\, V(s')$$

Warm-up as You Walk In

Given

- Set `actions` (persistent/static)
- Set `states` (persistent/static)
- Function `T(s,a,s_prime)`

Write the pseudo code for:

- `function V(s) return value`

that implements:

$$V(s) = \max_{a \in actions} \sum_{s' \in states} T(s, a, s')V(s')$$

function $V(s)$

bestvalue $\leftarrow -\infty$

for $a \in$ actions

sumvalue $\leftarrow 0$

for $s' \in$ states

sumvalue $+= T(s,a,s')\boxed{V(s')}$

bestvalue $\leftarrow$ max(bestvalue, sumvalue)

MDP Notation

Standard expectimax: $V(s) = \max_a \sum_{s'} P(s'|s,a)V(s')$

Bellman equations: $V(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V(s')]$

Value iteration: $V_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')], \quad \forall\, s$

Q-iteration: $Q_{k+1}(s,a) = \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')], \quad \forall\, s,a$

Policy extraction: $\pi_V(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V(s')], \quad \forall\, s$

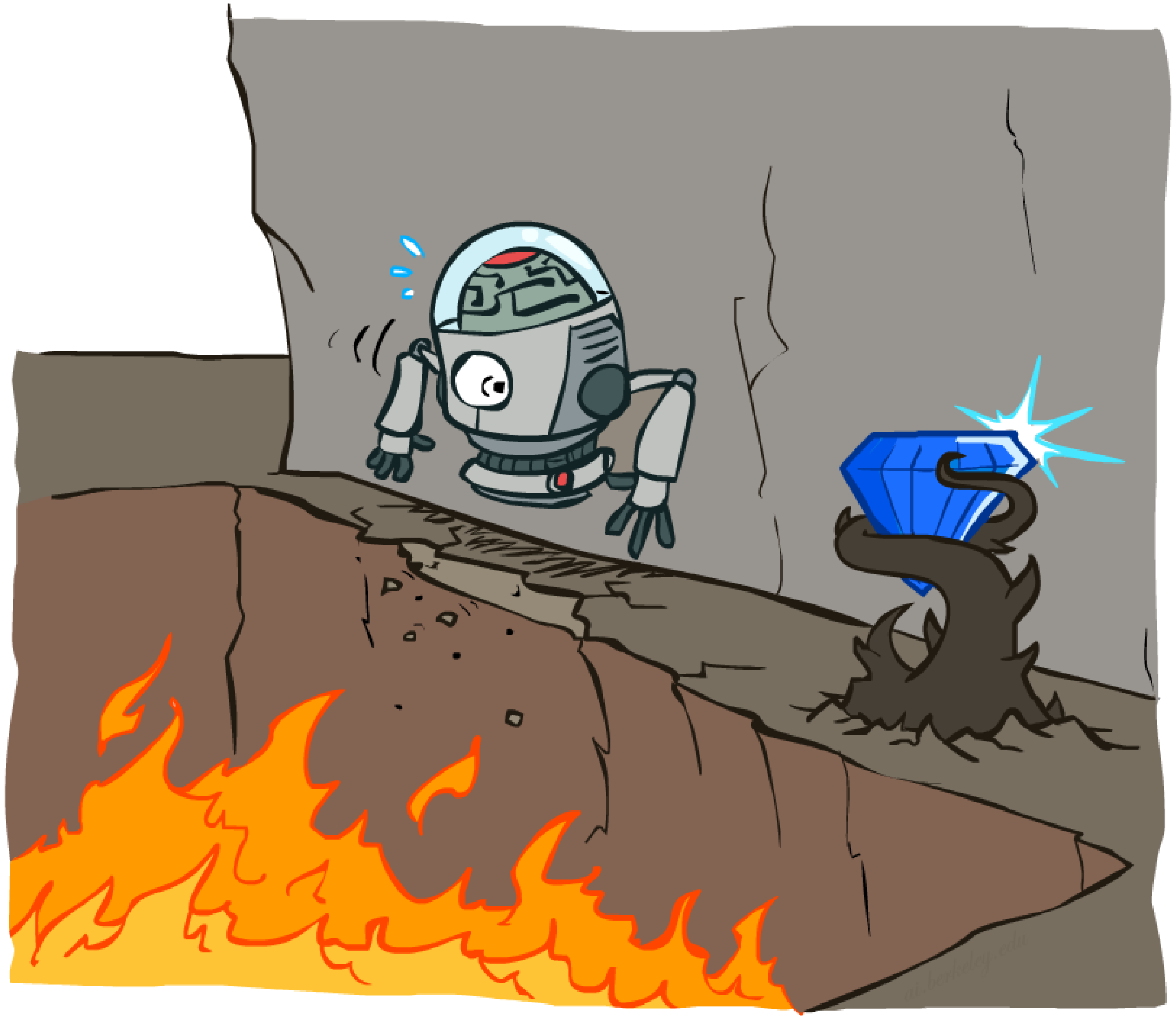
Policy evaluation: $V^{\pi}_{k+1}(s) = \sum_{s'} P(s'|s,\pi(s))[R(s,\pi(s),s') + \gamma V^{\pi}_k(s')], \quad \forall\, s$

Policy improvement: $\pi_{new}(s) = \operatorname{argmax}_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^{\pi_{old}}(s')], \quad \forall\, s$

MDP Notation

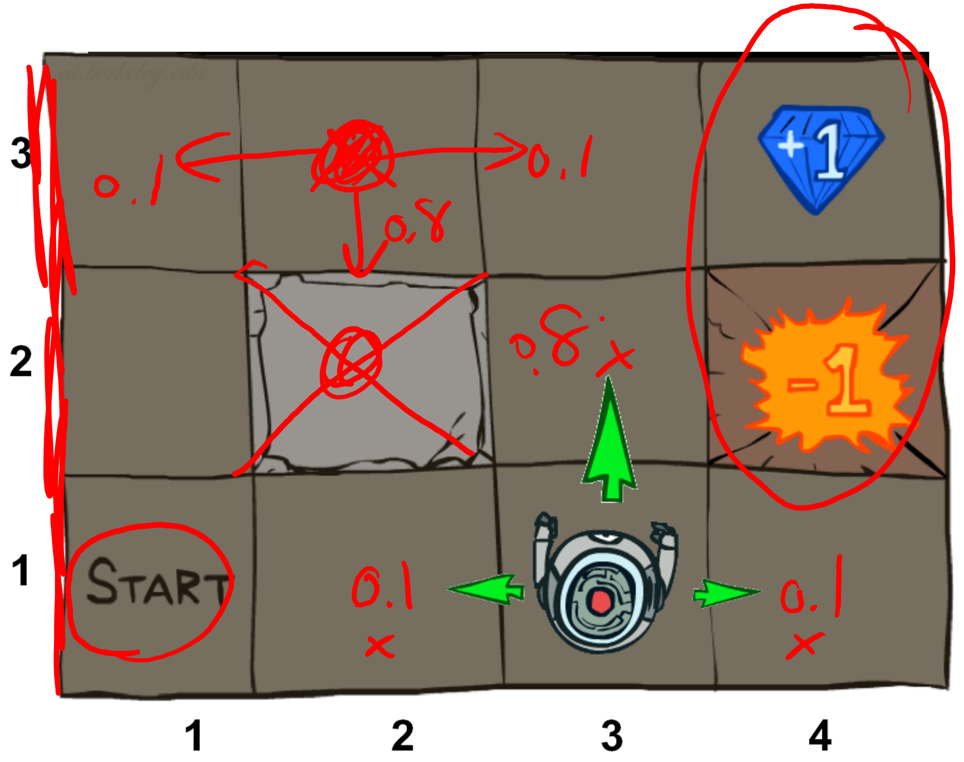
Standard expectimax:	$V(s) = \max_a \sum_{s'} P(s'\|s,a)V(s')$
Bellman equations:	$V(s) = \max_a \sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V(s')]$
Value iteration:	$V_{k+1}(s) = \max_a \sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V_k(s')], \quad \forall\, s$
Q-iteration:	$Q_{k+1}(s,a) = \sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')], \quad \forall\, s,a$
Policy extraction:	$\pi_V(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V(s')], \quad \forall\, s$
Policy evaluation:	$V^{\pi}_{k+1}(s) = \sum_{s'} P(s'\|s,\pi(s))[R(s,\pi(s),s') + \gamma V^{\pi}_k(s')], \quad \forall\, s$
Policy improvement:	$\pi_{new}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V^{\pi_{old}}(s')], \quad \forall\, s$

Non-Deterministic Search



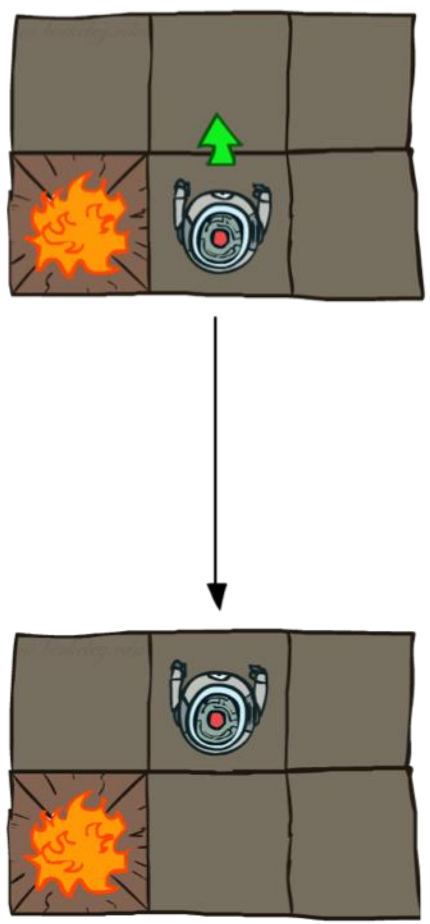
Example: Grid World

- A maze-like problem
  - The agent lives in a grid
  - Walls block the agent's path
- Noisy movement: actions do not always go as planned
  - If agent takes action North
    - 80% of the time: Get to the cell on the North (if there is no wall there)
    - 10%: West; 10%: East
  - If path after roll dice blocked by wall, stays put
- The agent receives rewards each time step
  - "Living" reward (can be negative)
  - Additional reward at pit or target (good or bad) and will exit the grid world afterward
- Goal: maximize sum of rewards

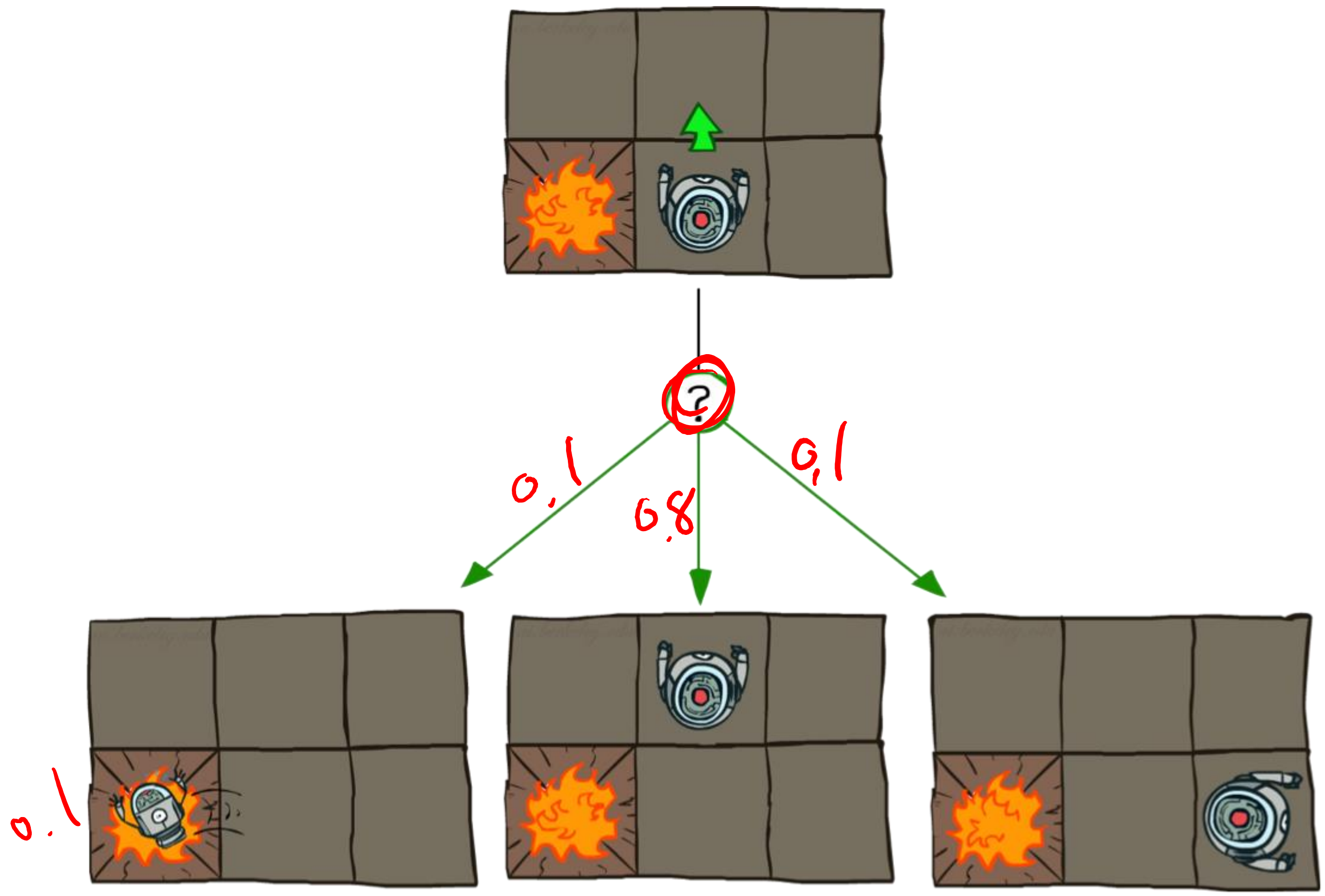


Grid World Actions

Deterministic Grid World



Stochastic Grid World



$T(S_{31}, N, S_{32}) = 0.8$ $T(S_{31}, N, S_{21}) = 0.1$

Markov Decision Processes

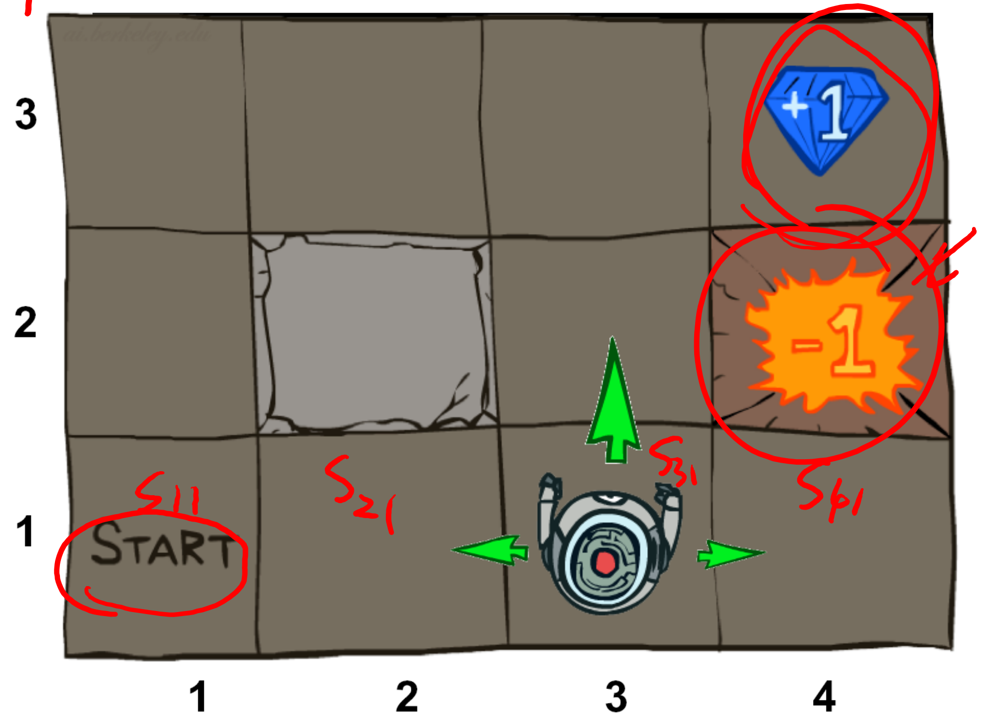
$T(S_{31}, W, S_{21}) = 0.8$

An MDP is defined by a tuple (S,A,T,R):

- S: a set of states
- A: a set of actions
- T: a transition function
  - T(s, a, s') where s ∈ S, a ∈ A, s' ∈ S is P(s' | s, a)
- R: a reward function
  - $\boxed{\text{R(s, a, s')}}$ is reward at this time step
  - Sometimes just R(s) or R(s')   R(s, a)

$R(S_{42}) = -1$, $R(S_{43}) = 1$

- Sometimes also have
  - γ: discount factor (introduced later)
  - μ: distribution of initial state (or just a start state s_0)
  - Terminal states: processes end after reaching these states



The Grid World problem as an MDP

$\boxed{R(s_{4,2}, exit, s_{virtual\_terminal}) = -1}$

N, W, E, exit, 0.1

~~R($s_{4,2}$)=-1, no virtual terminal state~~

R(s, a, s') = -1, ∀ s, a, s' where s ≠ S_{42} or S_{43}

R(S_{42}, exit, terminal state) = -1

How to define the terminal states and reward function for the Grid World problem?

Demo of Gridworld

[Demo – gridworld manual intro (L8D1)]

Markov Decision Processes

An MDP is defined by a tuple (S,A,T,R)

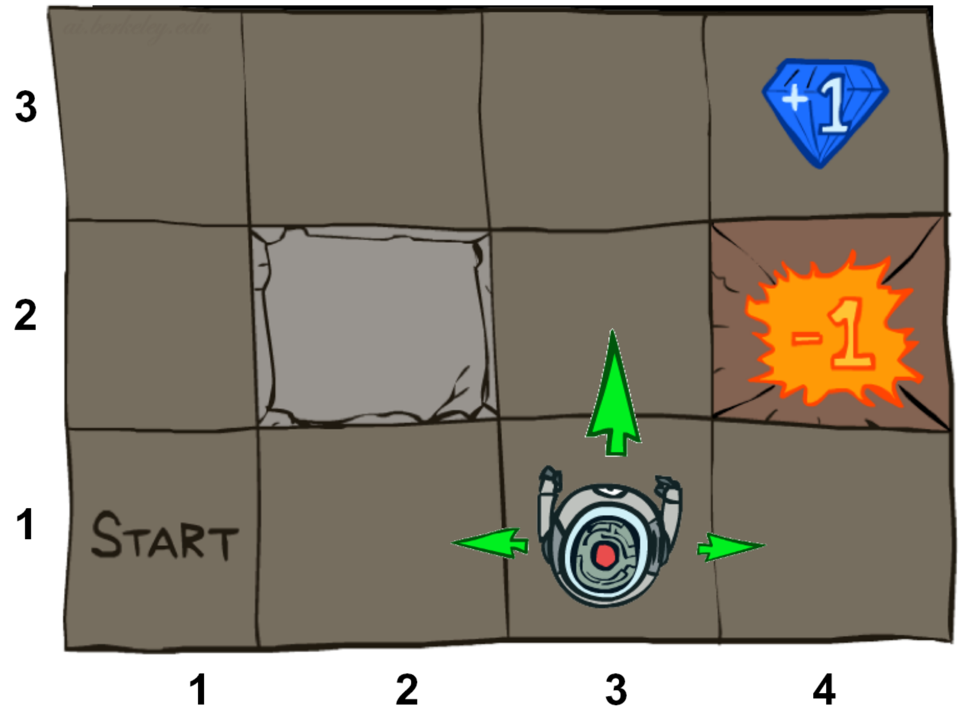
Why is it called Markov Decision Process?

agent

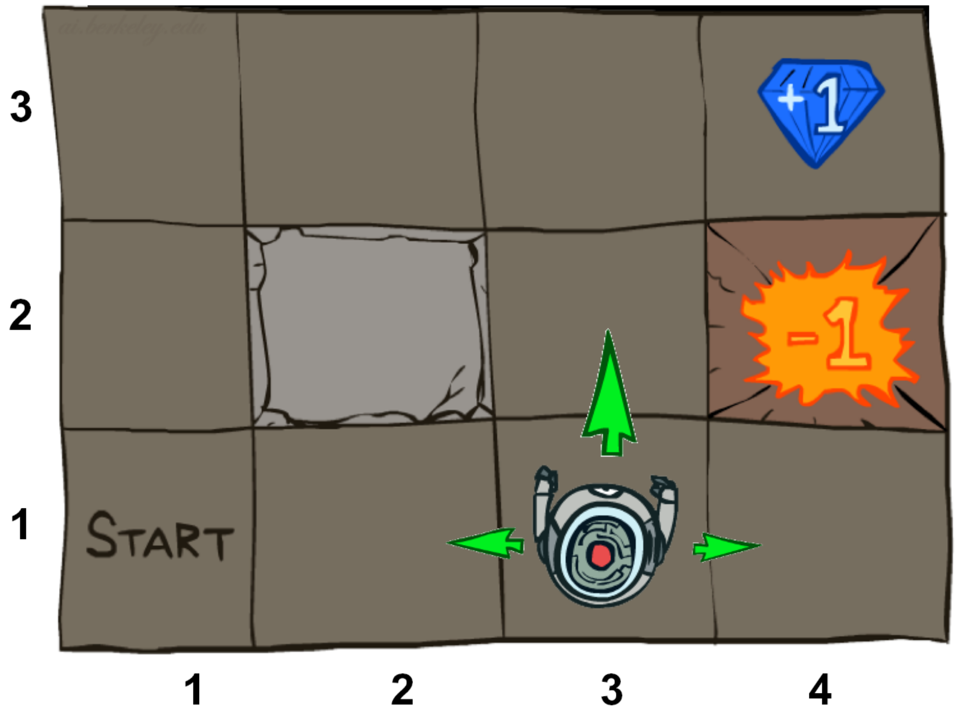
Decision: choose an agent

Process: dynamic system, system changes over time

(envirnment & agent)



Markov Decision Processes



An MDP is defined by a tuple (S,A,T,R)

Why is it called Markov Decision Process?

Decision:

Agent decides what action to take in each time step

Process:

The system (environment + agent) is changing over time

What is Markov about MDPs?

Markov property: Conditional on the present state, the future and the past are independent

In MDP, it means outcome of an action depend only on current state

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \ldots S_0 = s_0)$$

$$=$$

$$P(S_{t+1} = s' | S_t = s_t, A_t = a_t)$$

Recall in search, successor function only depends on current state (not the history)

Andrey Markov
(1856-1922)
Russian mathematician

Policies

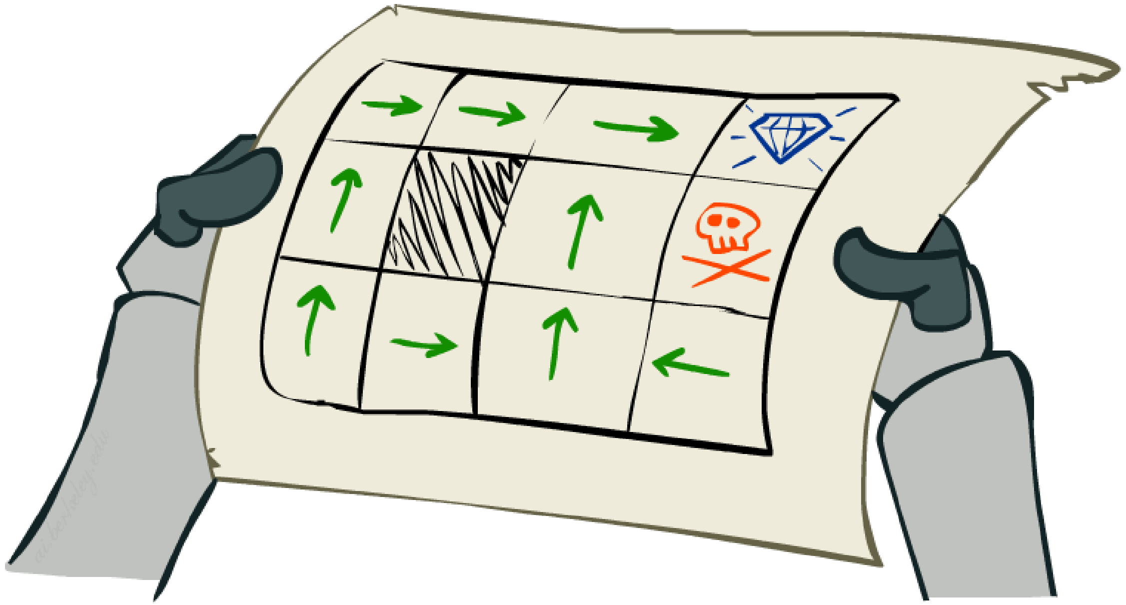
In deterministic single-agent search problems, we wanted an optimal plan, or sequence of actions, from start to a goal

For MDPs, we focus on policies

- Policy = map of states to actions
- $\pi(s)$ gives an action for state s

We want an optimal policy $\pi^*: S \rightarrow A$

- An optimal policy is one that maximizes expected utility if followed



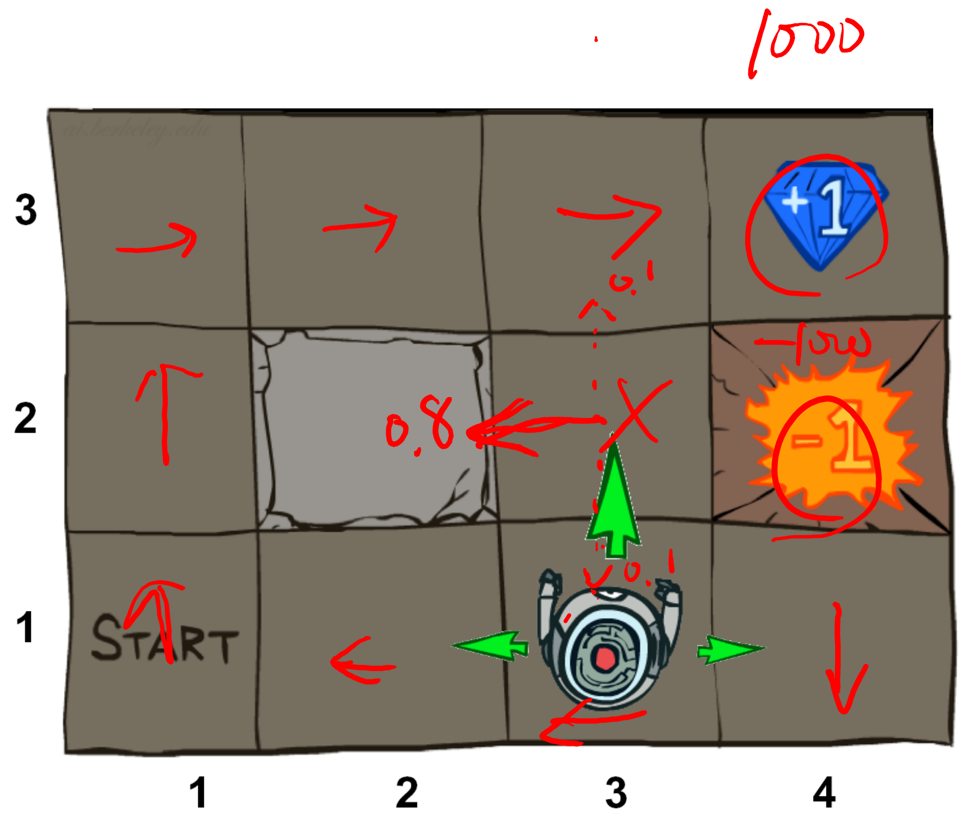
Policies

Recall: An MDP is defined S,A,T,R

Keep S,A,T fixed, optimal policy may vary given different R

What is the optimal policy if R(s,a,s')=-1000 for all states other than pit and target?

What is the optimal policy if R(s,a,s')=0 for all states other than pit and target, and reward=1000 and -1000 at pit and target respectively?



Piazza Poll 2

Which sequence of optimal policies matches the following sequence of living rewards:

{-0.01, -0.03, -0.04, -2.0}

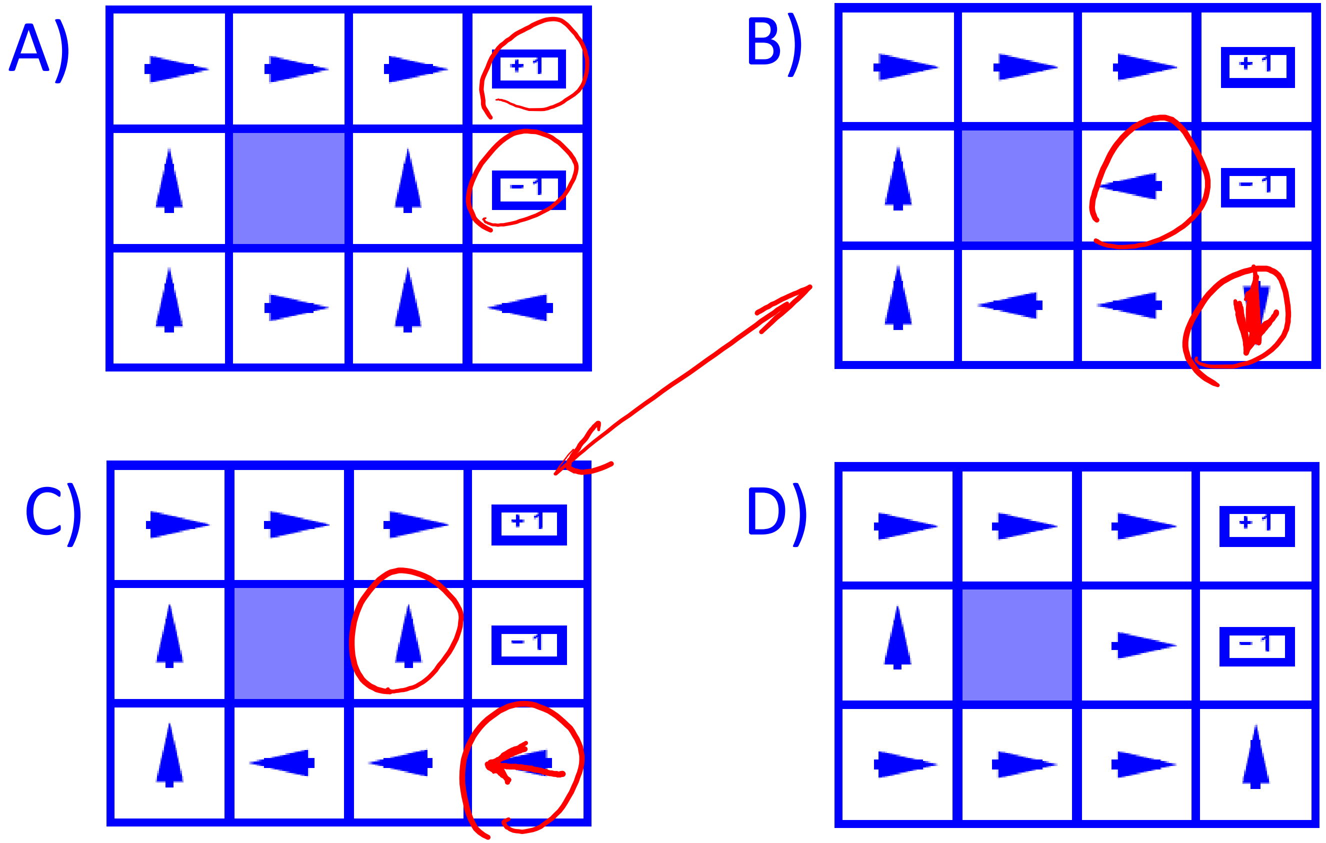
I. {B, A, C, D}

70% II. {B, C, A, D}

20% III. {C, B, A, D}

IV. {D, A, C, B}

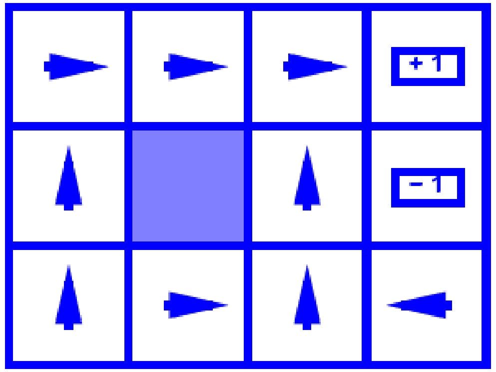
conservative

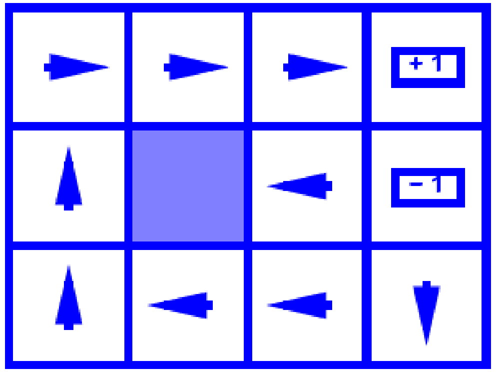


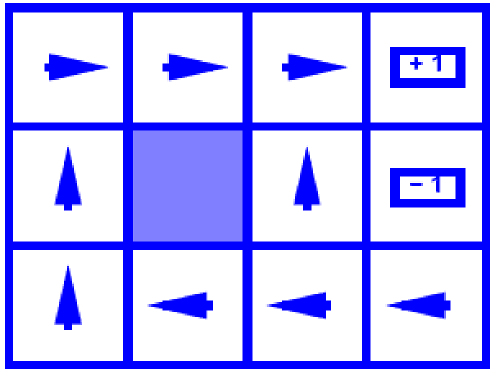
Piazza Poll 2

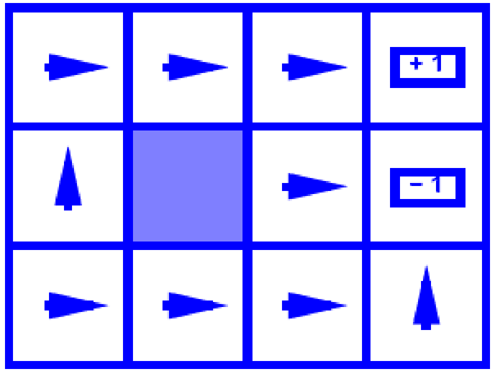
Which sequence of optimal policies matches the following sequence of living rewards:
{-0.01, -0.03, -0.04, -2.0}

I. {B, A, C, D}
II. {B, C, A, D}
III. {C, B, A, D}
IV. {D, A, C, B}

A)


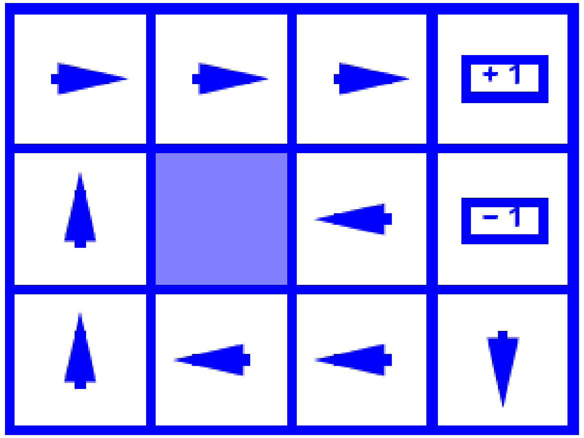
B)


C)


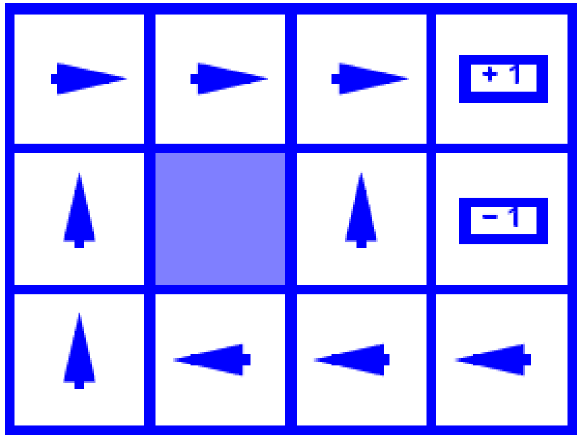
D)


Piazza Poll 2

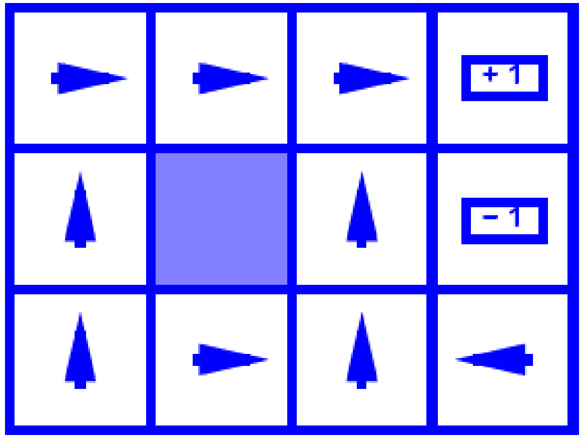
Optimal Policies



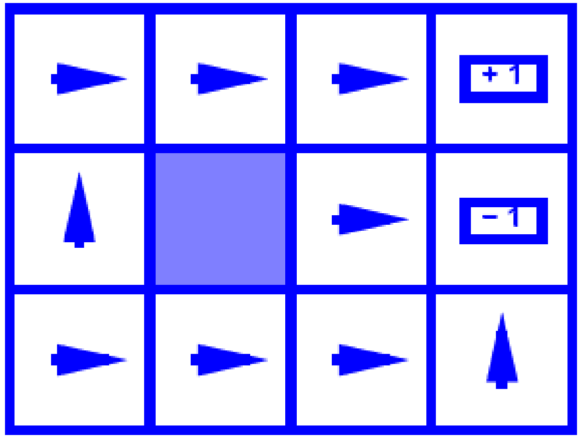
R(s) = -0.01



R(s) = -0.03

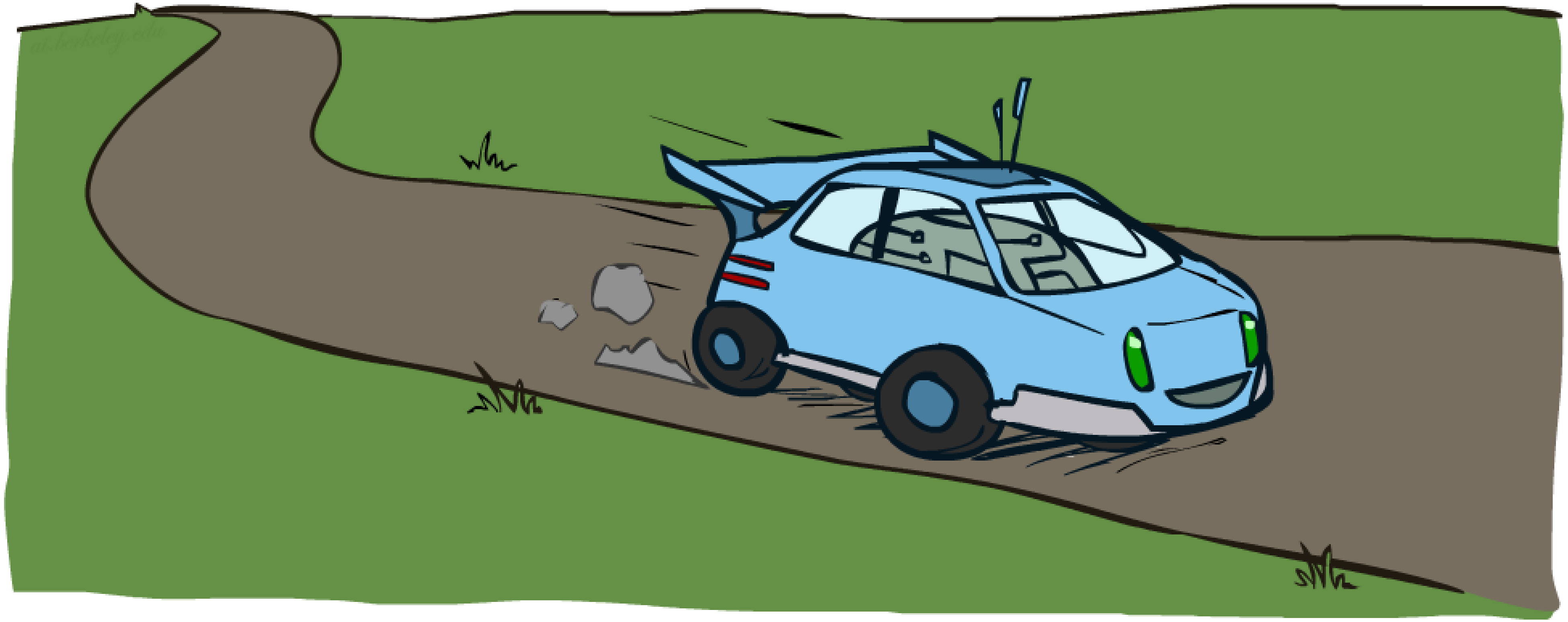


R(s) = -0.4



R(s) = -2.0

Example: Racing



Example: Racing

$R(s,a,s') = 1$ if $a =$ slow, $\forall s, s'$

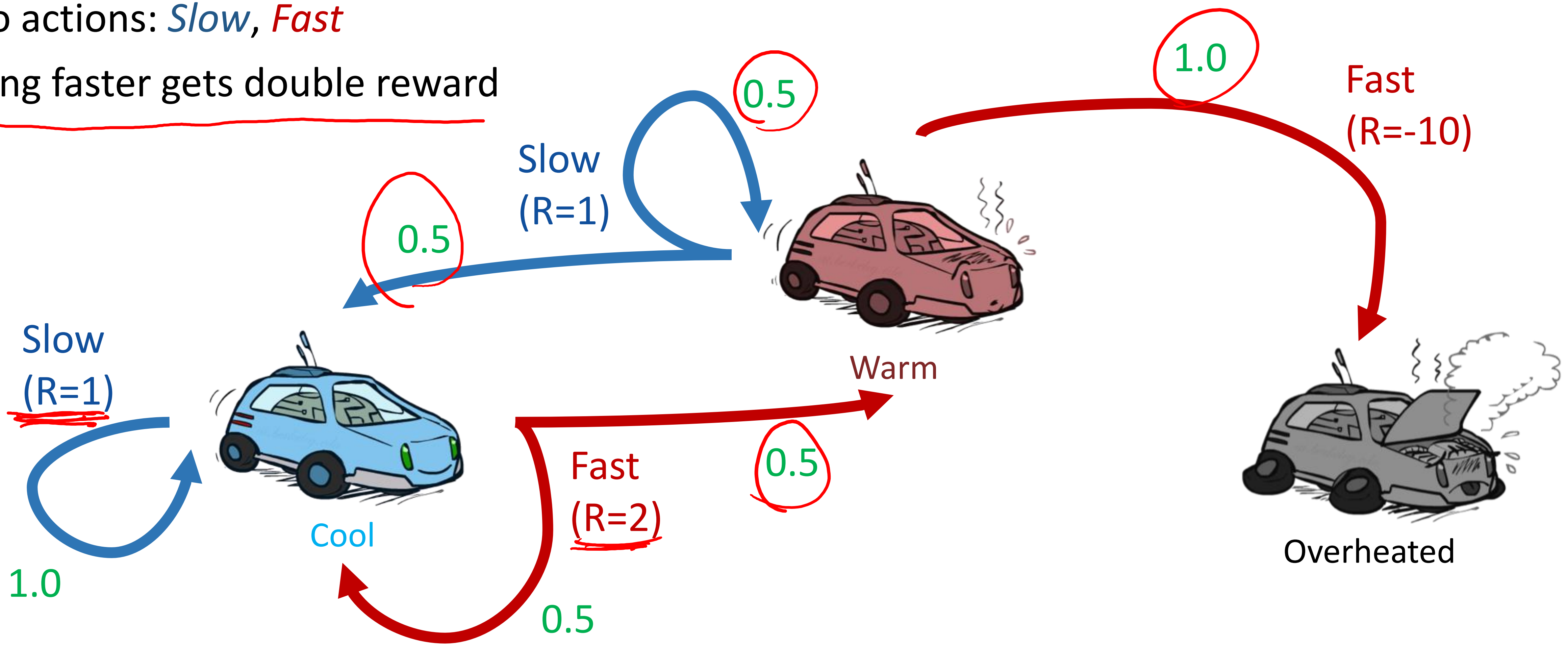
- - - - 2 - - fast - - -

A robot car wants to travel far, quickly

Three states: Cool, Warm, Overheated

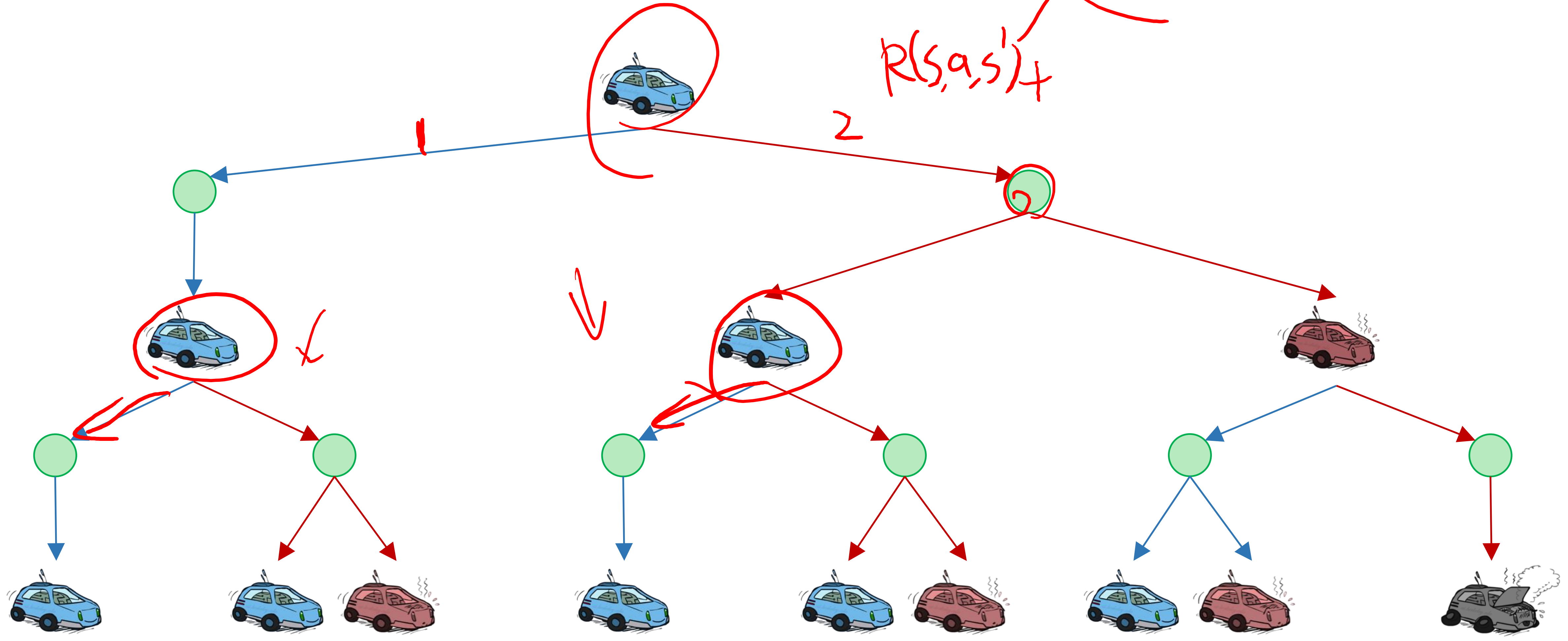
Two actions: *Slow*, *Fast*

Going faster gets double reward



Racing Search Tree

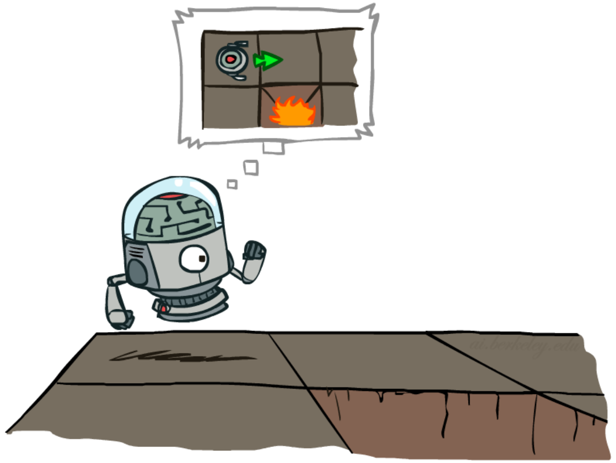
$$V(s) = \sum_{s'} R(s,a,s') + P(s'|a,s) \cdot V(s')$$

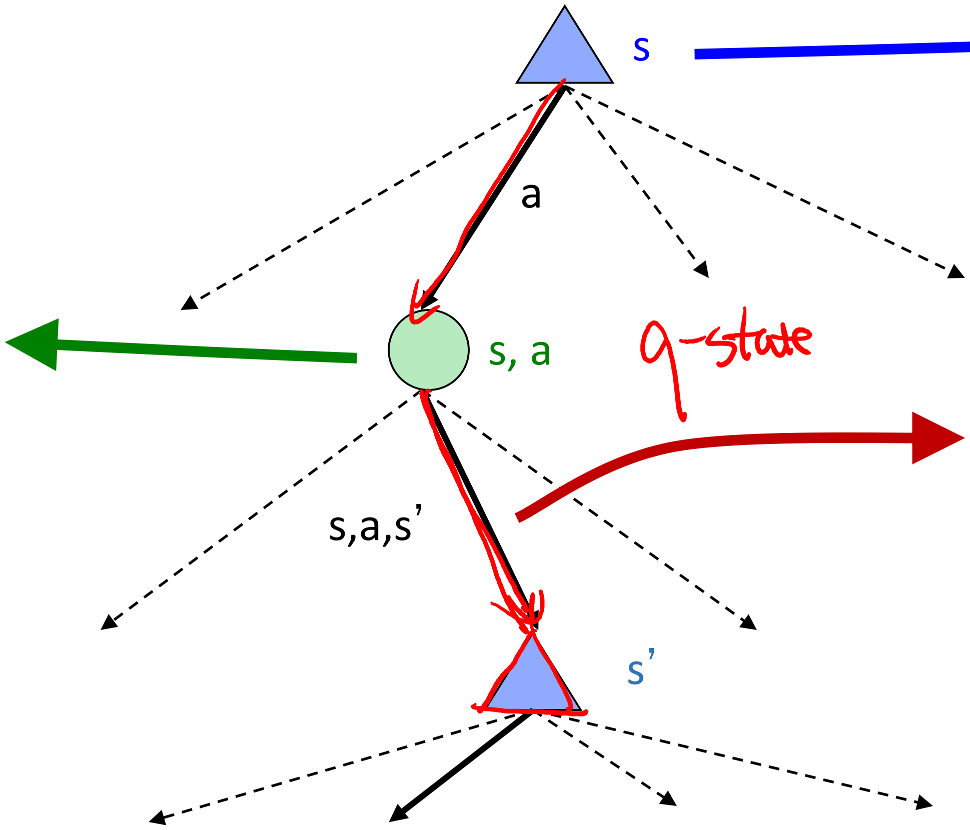


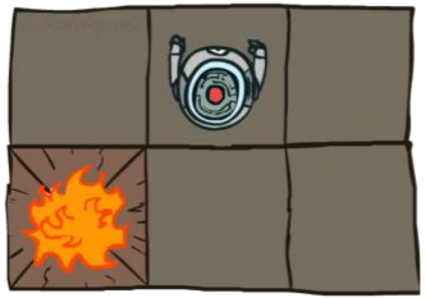
Can we use expectimax for MDP directly?

MDP Search Trees

Each MDP state projects an expectimax-like search tree







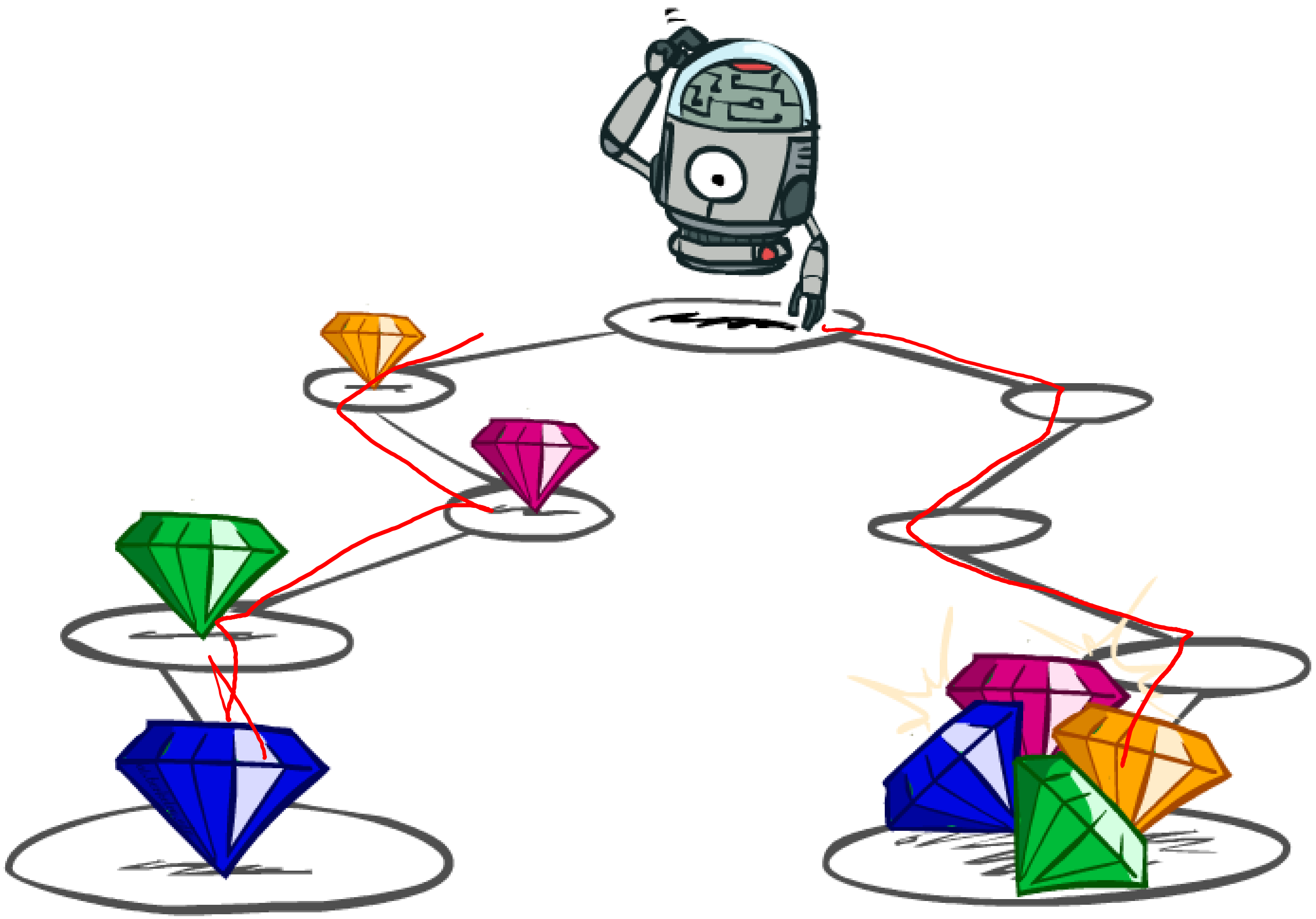
s is a *state*

(s,a,s') called a *transition*

$T(s,a,s') = P(s'|s,a)$

$R(s,a,s')$

Utilities of Sequences

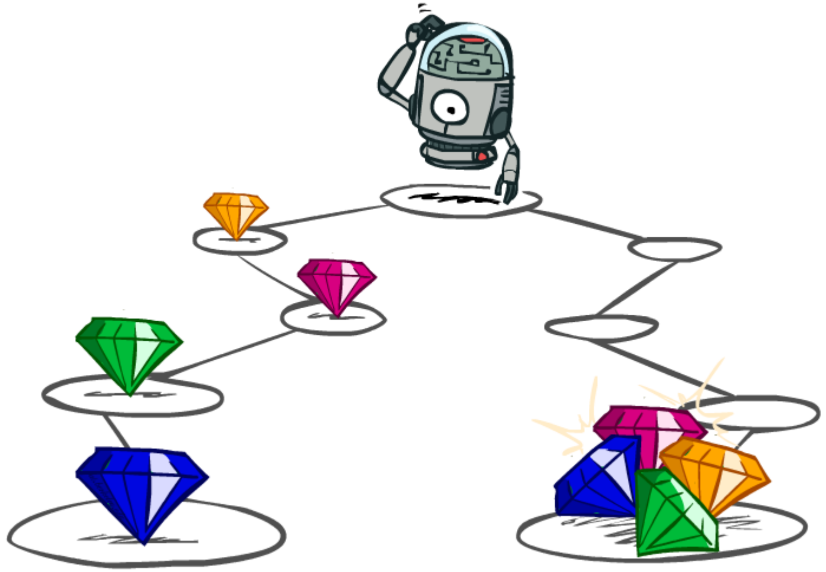


Utilities of Sequences

What preferences should an agent have over reward sequences?

More or less? [1, 2, 2] or [2, 3, 4]

Now or later? [0, 0, 1] or [1, 0, 0]



Discounting

It’s reasonable to maximize the sum of rewards

It’s also reasonable to prefer rewards now to rewards later

One solution: utility of rewards decay exponentially

1

Worth Now

$\gamma < 1$

Worth Next Step

$\gamma^2 < \gamma < 1$

Worth In Two Steps

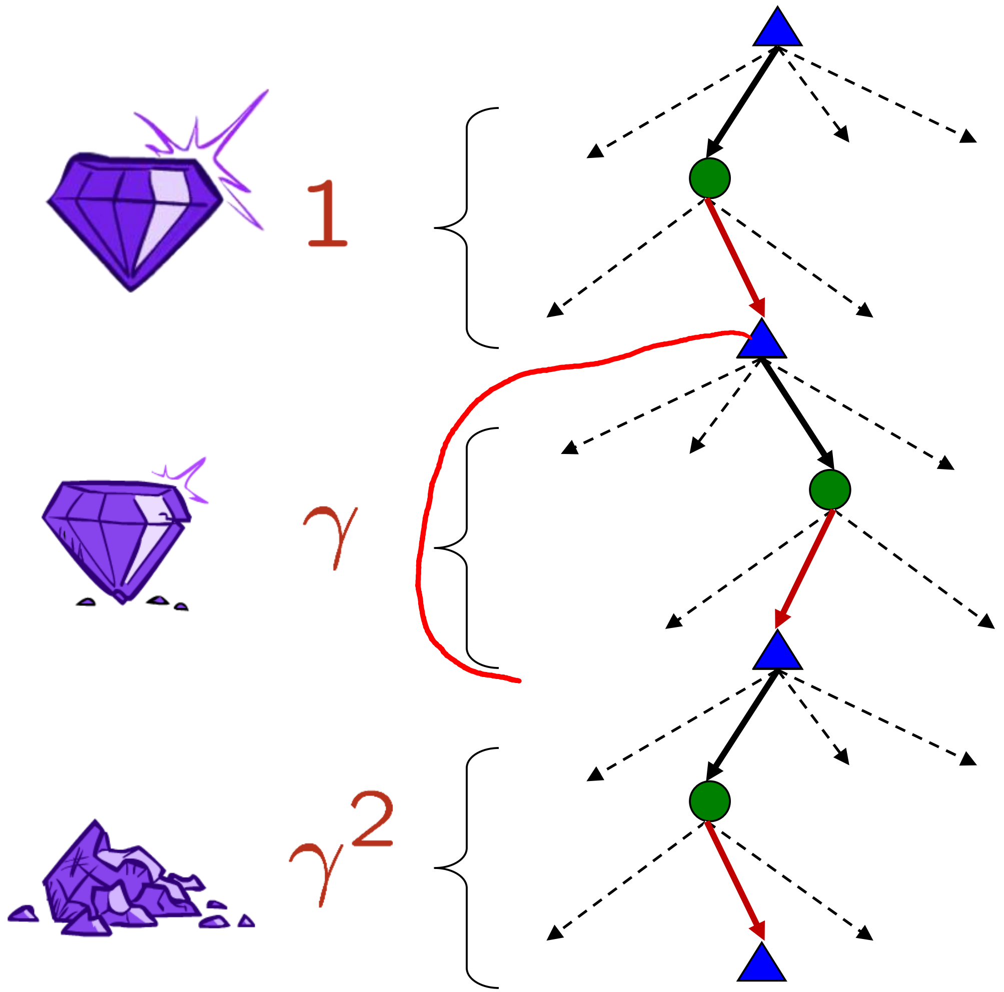
Discounting

How to discount?

- Each time we descend a level, we multiply in the discount once

Why discount?

- Sooner rewards probably do have higher utility than later rewards
- Also helps our algorithms converge



Piazza Poll 3

$1, r, r^2, r^3, ---$

r^0

What is the value of U([2,4,8]) with $\gamma = 0.5$?

U(·) is the total utility of a reward sequence

$2 + r \cdot 4 + r^2 \cdot 8 = 6$

$0.5 \cdot 4$ $\quad$ $0.5^2 \cdot 8$

2 $\quad$ 2

A. 3

75% B. 6

17% C. 7

D. 14

$\sum_k r^k \cdot R_k$

$8 + 2 + 0.5 = 10.5$

Bonus: What is the value of U([8,4,2]) with $\gamma = 0.5$?

Piazza Poll 3

What is the value of U([2,4,8]) with $\gamma = 0.5$?

U(·) is the total utility of a reward sequence

A. 3
B. 6
C. 7
D. 14

$$\gamma^0 \times 2 + \gamma^1 \times 4 + \gamma^2 \times 8 = 2 + 0.5 \times 4 + 0.5 \times 0.5 \times 8 = 2 + 2 + 2 = 6$$

Bonus: What is the value of U([8,4,2]) with $\gamma = 0.5$?

$$\gamma^0 \times 8 + \gamma^1 \times 4 + \gamma^2 \times 2 = 8 + 0.5 \times 4 + 0.5 \times 0.5 \times 2 = 8 + 2 + 0.5 = 10.5$$

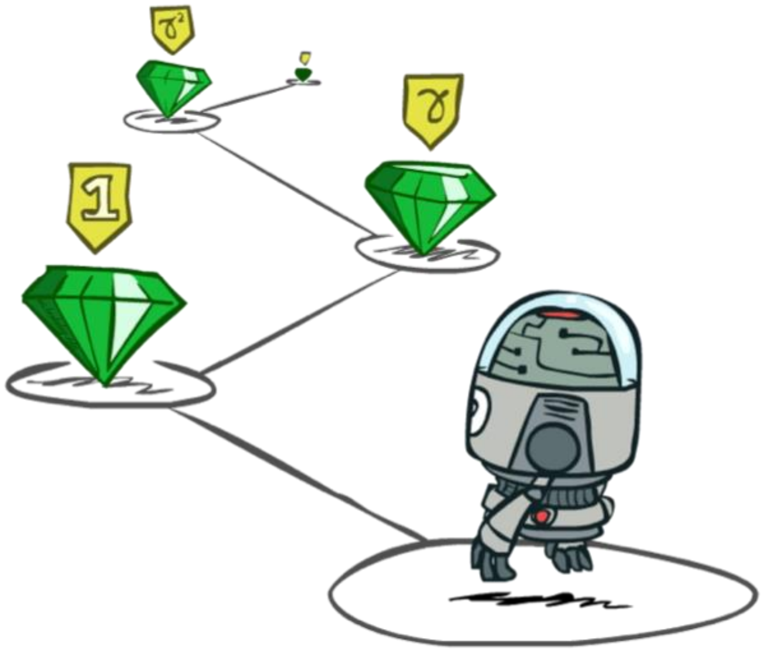
Stationary Preferences

Theorem: if we assume stationary preferences:

$$[a_1, a_2, \ldots] \succ [b_1, b_2, \ldots]$$

$$\Updownarrow$$

$$[r, a_1, a_2, \ldots] \succ [r, b_1, b_2, \ldots]$$



Then: there are only two ways to define utilities

$\gamma = 1$

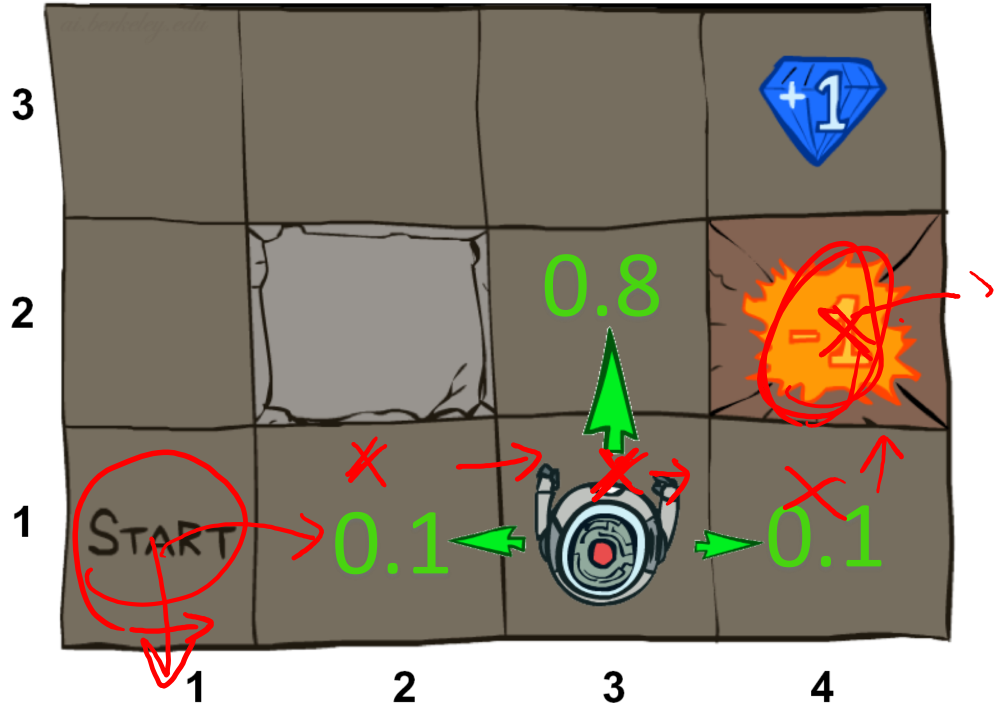
- Additive utility: $U([r_0, r_1, r_2, \ldots]) = r_0 + r_1 + r_2 + \cdots$
- Discounted utility: $U([r_0, r_1, r_2, \ldots]) = r_0 + \gamma r_1 + \gamma^2 r_2 \cdots$

Question

What is the minimum and maximum possible length of reward sequence in this grid world problem?

$r(s, a, s') = -0.1$

$r(s_{42}, \text{exit}, \text{terminal state}) = -1$



5 4, 3, 2

-0.1, -0.1, -0.1, -1

-0.1 -0.1 -0.1 -0.1

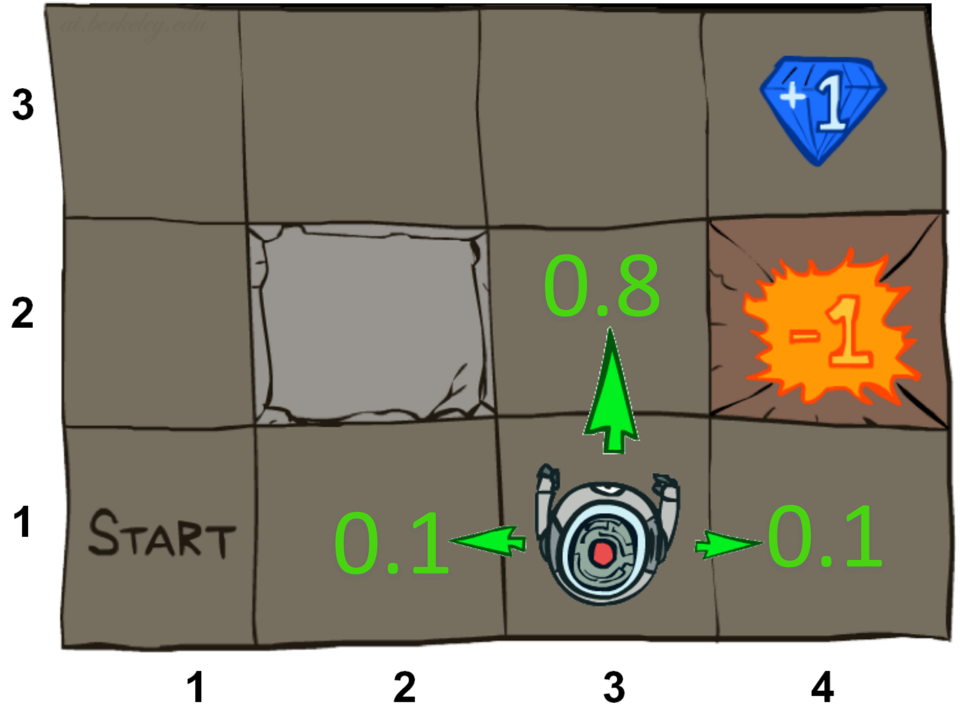
-1

+∞

Question

What is the minimum and maximum possible length of reward sequence in this grid world problem?

5, $+\infty$

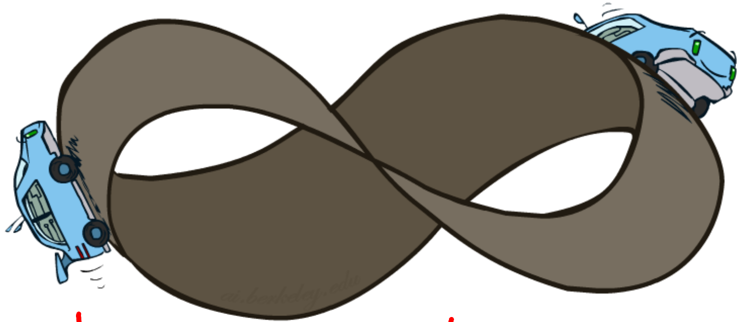


Infinite Utilities?!

$r_t \leq R_{max}$

What if the sequence is infinite? Do we get infinite utility?

- With discounting γ where $0 < \gamma < 1$ Assume $|r_t| \leq R_{max}$



$$U([r_0, \ldots r_\infty]) = \sum_{t=0}^{\infty} \gamma^t r_t \leq R_{\max}/(1-\gamma)$$

$$\leq \sum_{t=0} \gamma^t R_{max} = R_{max} \cdot \sum_{t=0}^{\infty} \gamma^t = R_{max} \frac{1}{1-\gamma}$$

- Smaller γ means smaller "horizon" – shorter term focus

Optimal Policy with Discounting

$1, \gamma, \gamma^2, \gamma^3, \ldots$

$R(s,a,s')=0$

Given:

a	b	c	d	e
10				1

- Actions: East, West, and Exit (only available in exit states a, e)
- Transitions: deterministic

For $\gamma = 1$, what is the optimal policy?

10	←	←	←	1

For $\gamma = 0.1$, what is the optimal policy?

10	←	←	←	1

$0.1^2 \cdot 10 > 0.1^2 \cdot 1$

For which γ are West and East equally good when in state d?

$U([0,0,0,10])$

$0.1^3 \cdot 10 < 0.1 \times 1$

Optimal Policy with Discounting

Given:

10				1
a	b	c	d	e

- Actions: East, West, and Exit (only available in exit states a, e)
- Transitions: deterministic

For $\gamma = 1$, what is the optimal policy?

10	←	←	←	1

For $\gamma = 0.1$, what is the optimal policy?

10	←	←	→	1

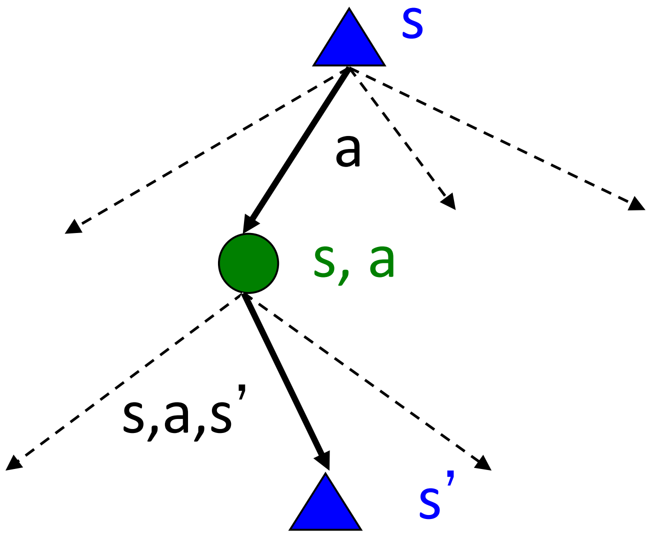
For which γ are West and East equally good when in state d?

$$\gamma^3 \times 10 = \gamma^1 \times 1$$

MDP Quantities

Markov decision processes:

- States S
- Actions A
- Transitions P(s'|s,a) (or T(s,a,s'))
- Rewards R(s,a,s') (and discount γ)
- Start state s_0



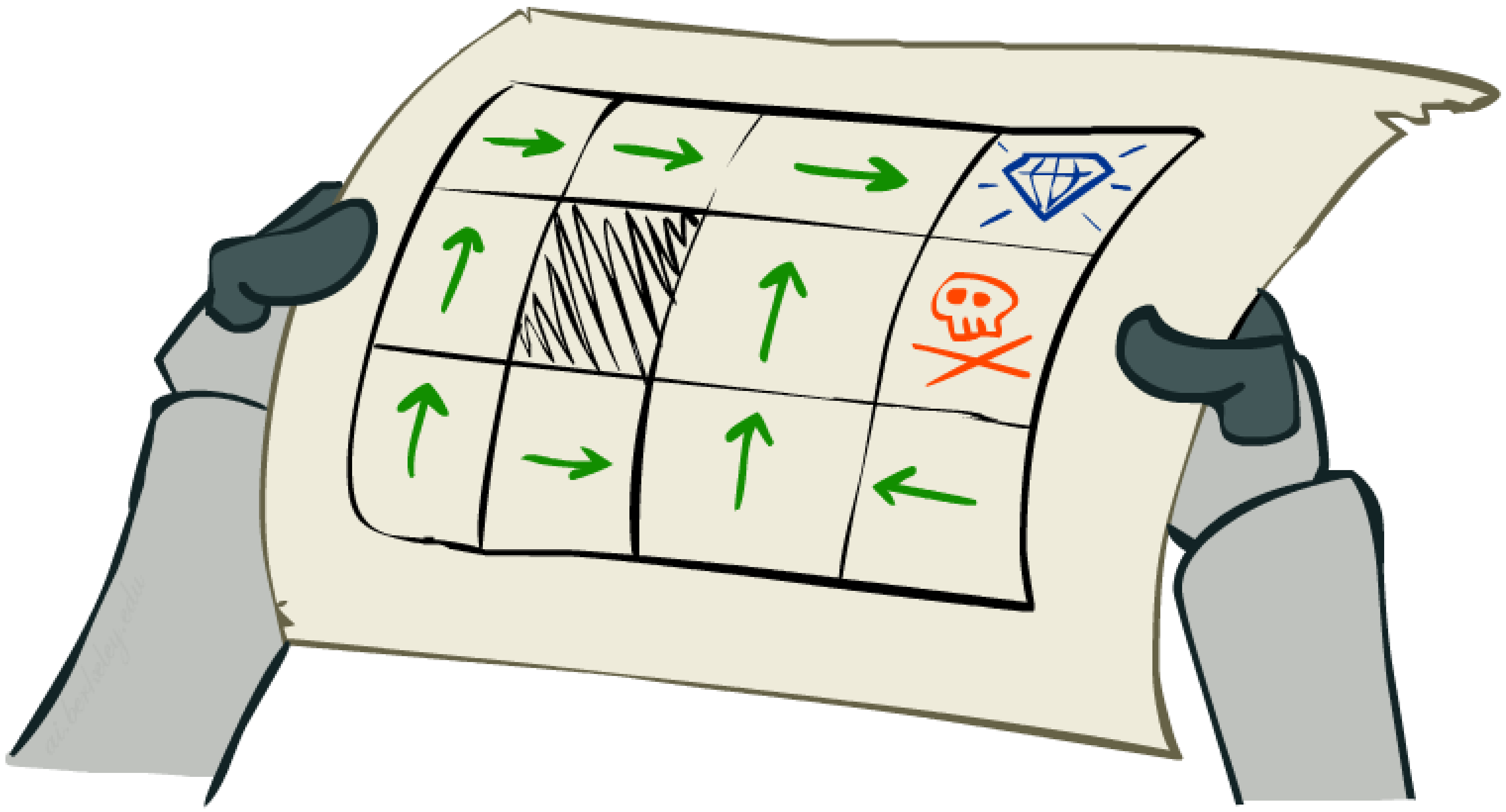
MDP quantities so far:

- Policy = map of states to actions
- Utility = sum of (discounted) rewards

$\pi(s) = a, \; a \in A$

$r^0, r^1, \ldots r^\infty$

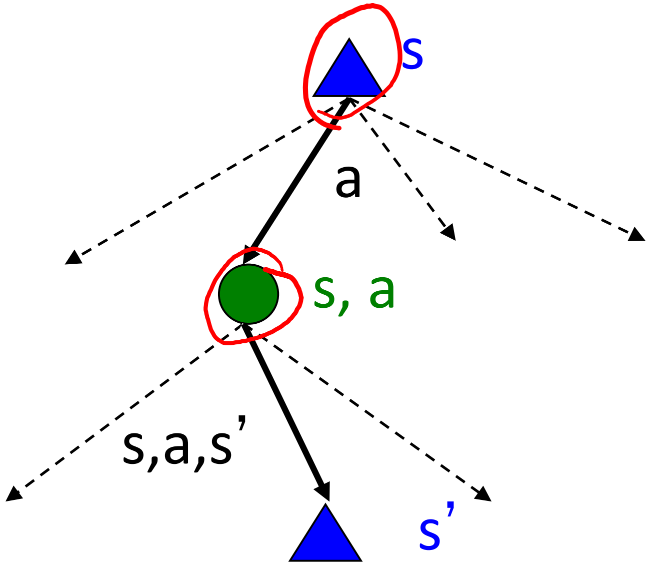
Solving MDPs



MDP Quantities

Markov decision processes:

- States S
- Actions A
- Transitions P(s'|s,a) (or T(s,a,s'))
- Rewards R(s,a,s') (and discount γ)
- Start state s_0



MDP quantities:

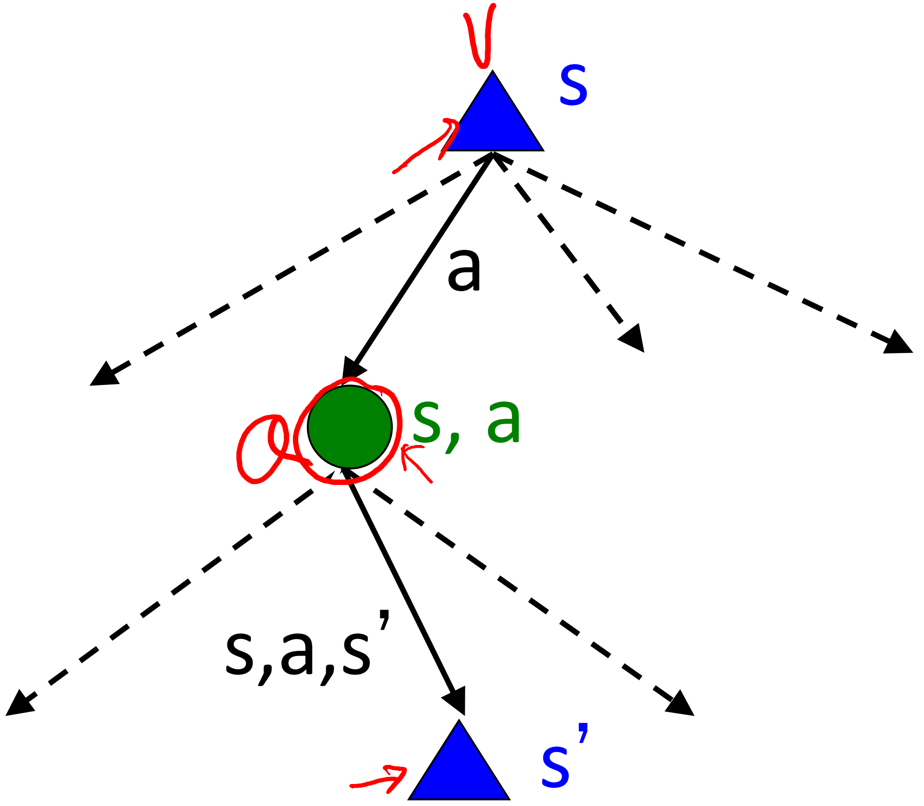
- Policy = map of states to actions
- Utility = sum of (discounted) rewards
- (State) Value = expected utility starting from a state (max node)
- Q-Value = expected utility starting from a state-action pair, i.e., q-state (chance node)

MDP Optimal Quantities

- The optimal policy:
  $\pi^*(s)$ = optimal action from state s

- The (true) value (or utility) of a state s:
  $V^*(s)$ = expected utility starting in s and acting optimally

- The (true) value (or utility) of a q-state (s,a):
  $Q^*(s,a)$ = expected utility starting out having taken action a from state s and (thereafter) acting optimally

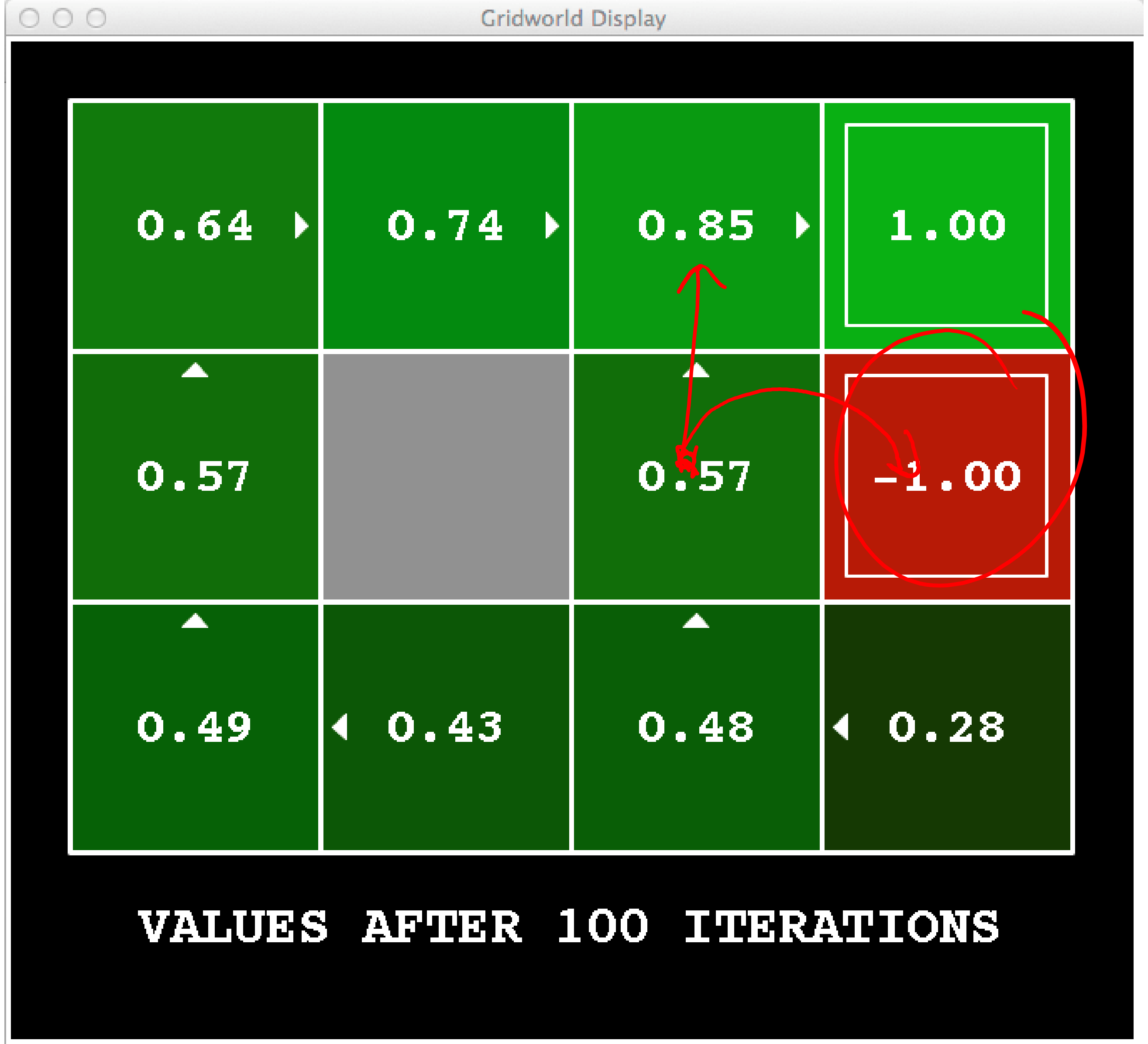


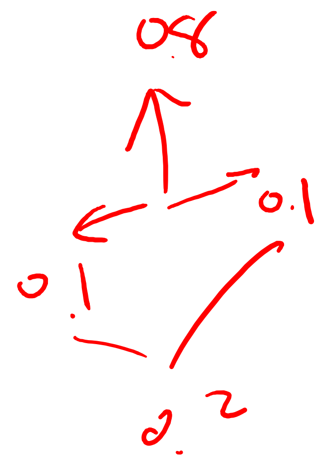
Solve MDP: Find π^*, V^* and/or Q^*

[Demo: gridworld values (L9D1)]

Snapshot of Demo – Gridworld V Values

What is the optimal policy?





Noise = 0.2
Discount = 0.9
Living reward = 0

Snapshot of Demo – Gridworld V Values

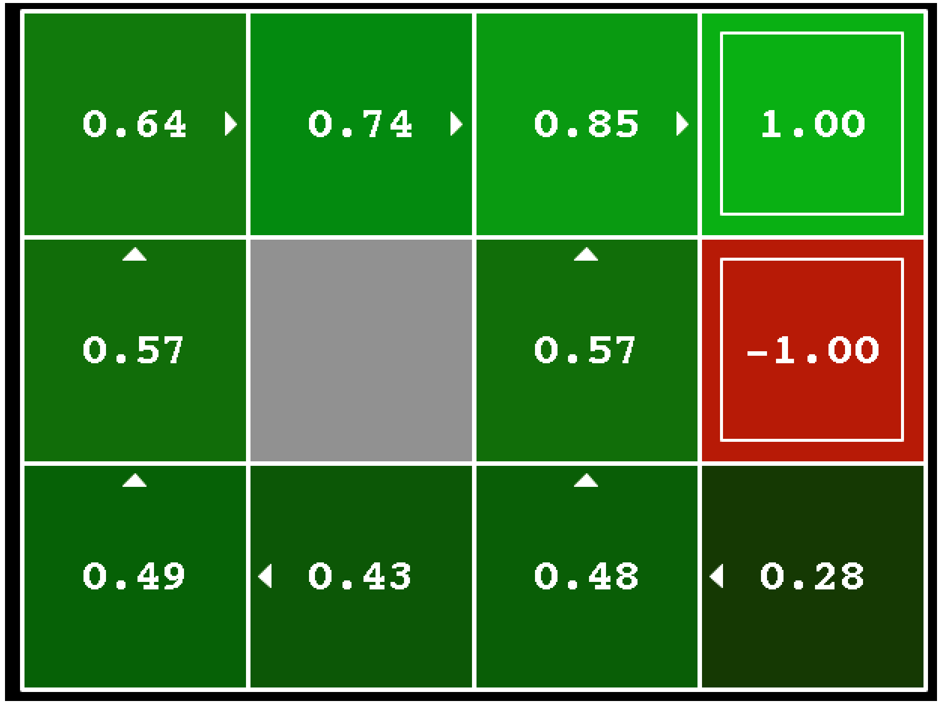
What is the
optimal policy?

May not equal $\underset{a}{\operatorname{argmax}} V^*(s')$ where s' is the most likely state after taking action a

V*(s) = expected utility starting in s and acting optimally

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a) * (\underbrace{\boxed{R(s,a,s')}}_{=0} + \gamma V^*(s'))$$

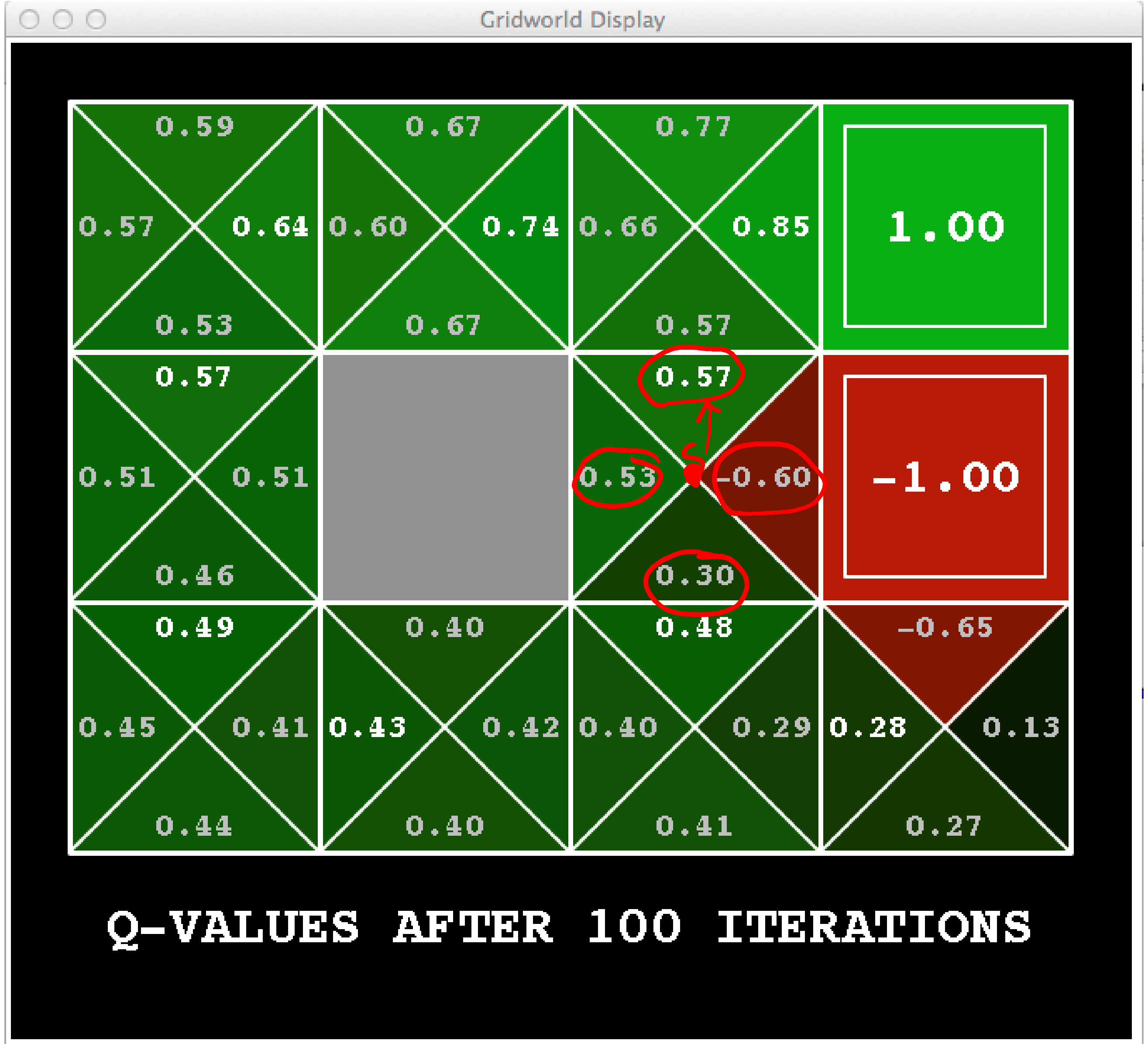
$$= \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a) V^*(s'))$$



Noise = 0.2
Discount = 0.9
Living reward = 0

Snapshot of Demo – Gridworld Q Values

What is the optimal policy?



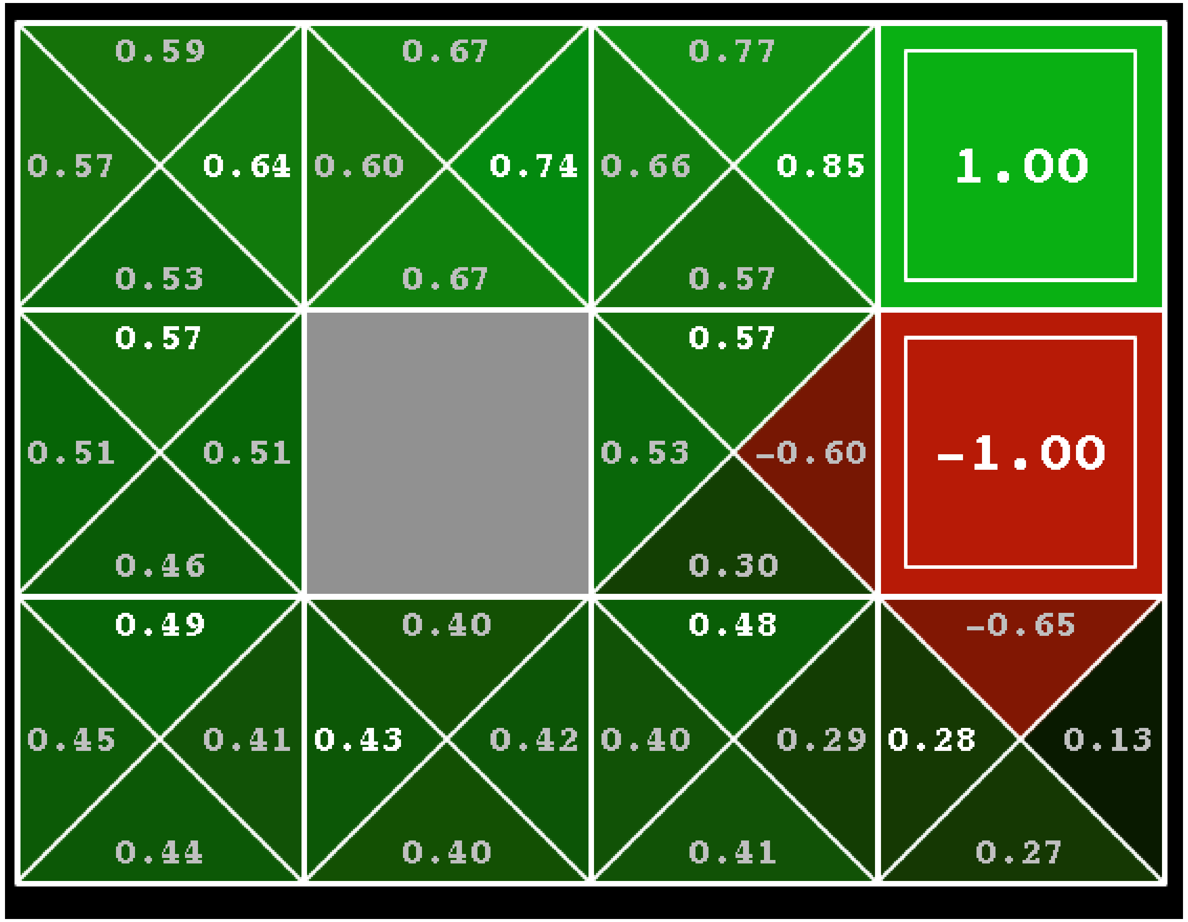
Noise = 0.2
Discount = 0.9
Living reward = 0

Snapshot of Demo – Gridworld Q Values

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q^*(s,a)$$

What is the optimal policy?

$Q^*(s,a)$ = expected utility starting out having taken action a from state s and (thereafter) acting optimally



Noise = 0.2
Discount = 0.9
Living reward = 0

Snapshot of Demo – Gridworld V Values

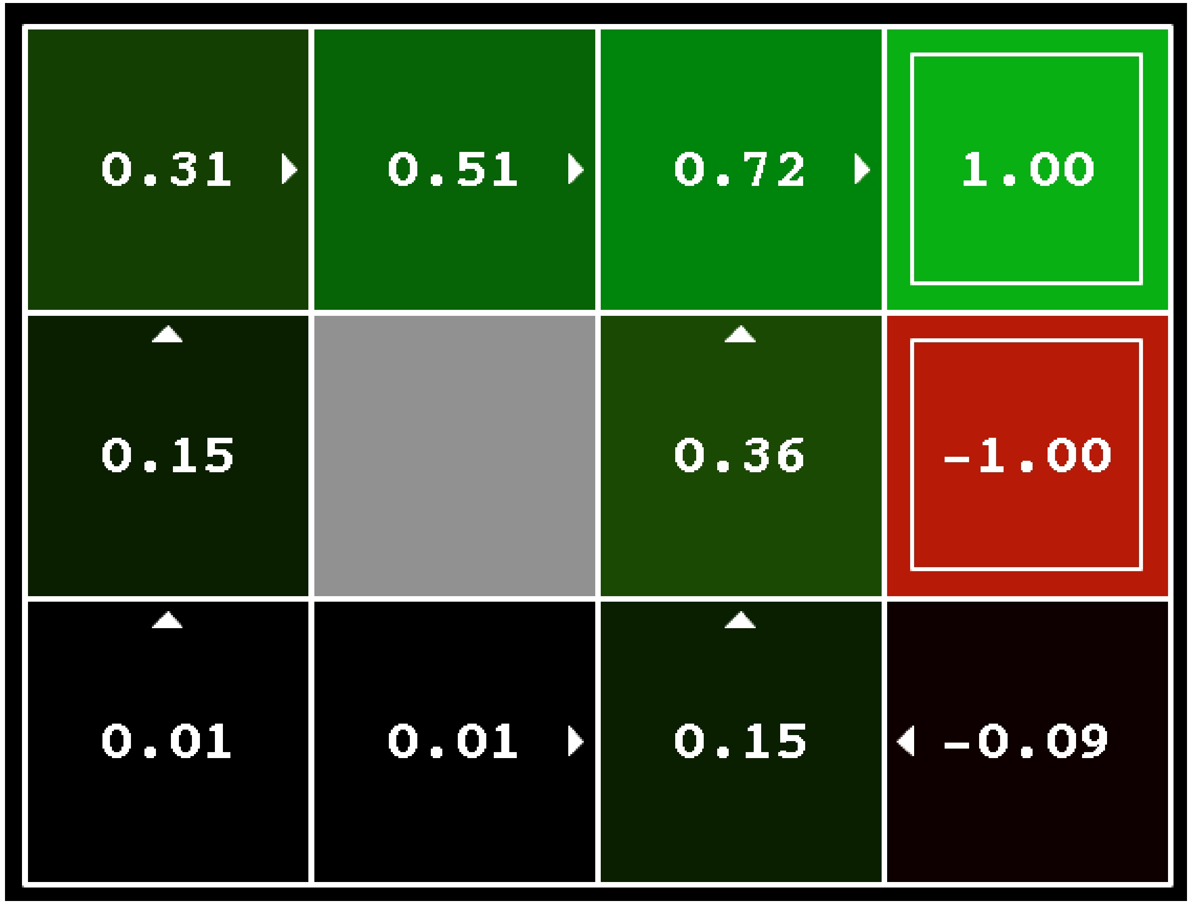
What is the optimal policy?

Noise = 0.2
Discount = 0.9
Living reward = -0.1

Snapshot of Demo – Gridworld V Values

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a) * (R(s,a,s') + \gamma V^*(s')) \neq \underset{a}{\operatorname{argmax}}\, V^*(s')$$

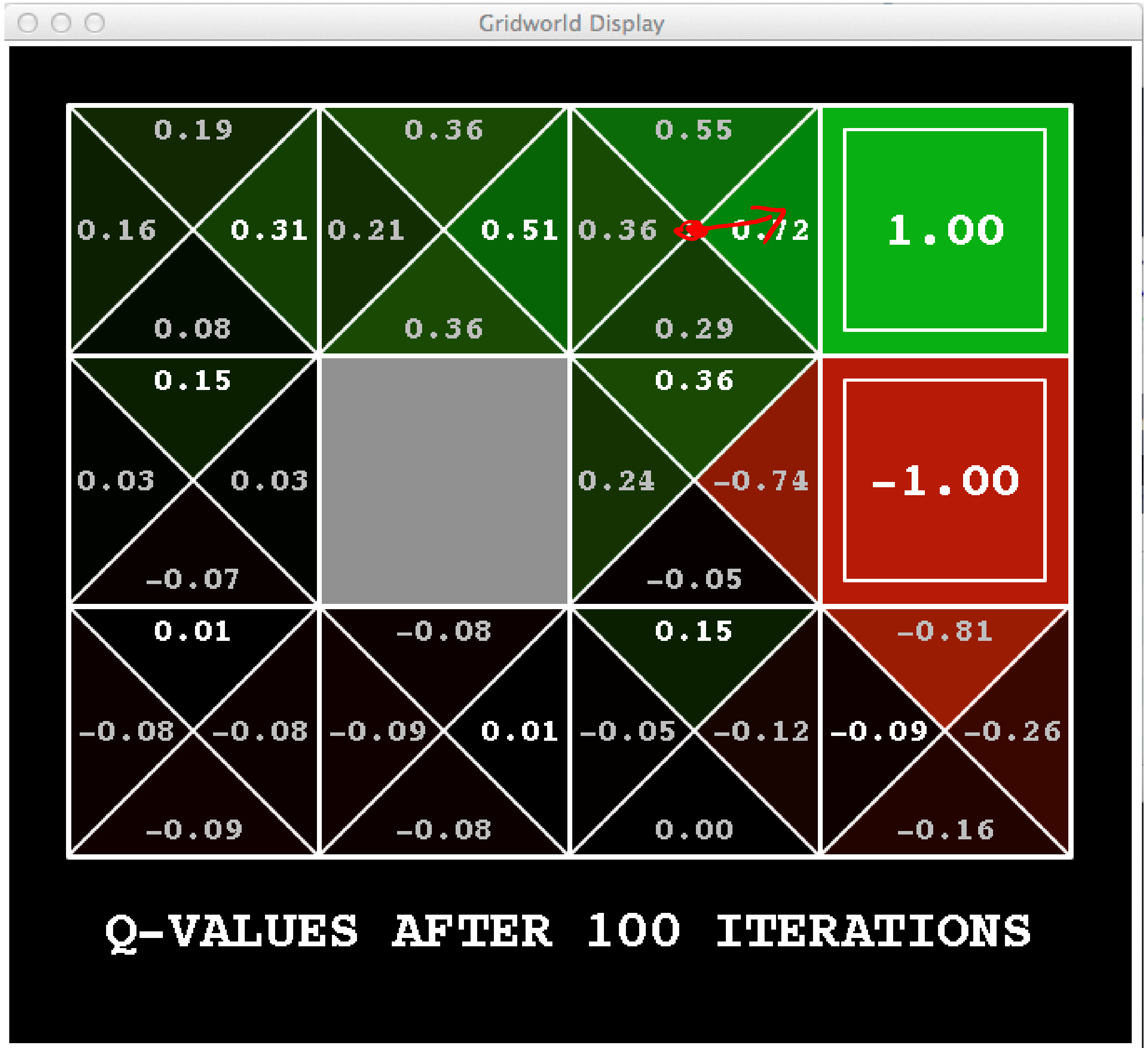
What is the optimal policy?



Noise = 0.2
Discount = 0.9
Living reward = -0.1

Snapshot of Demo – Gridworld Q Values

What is the optimal policy?

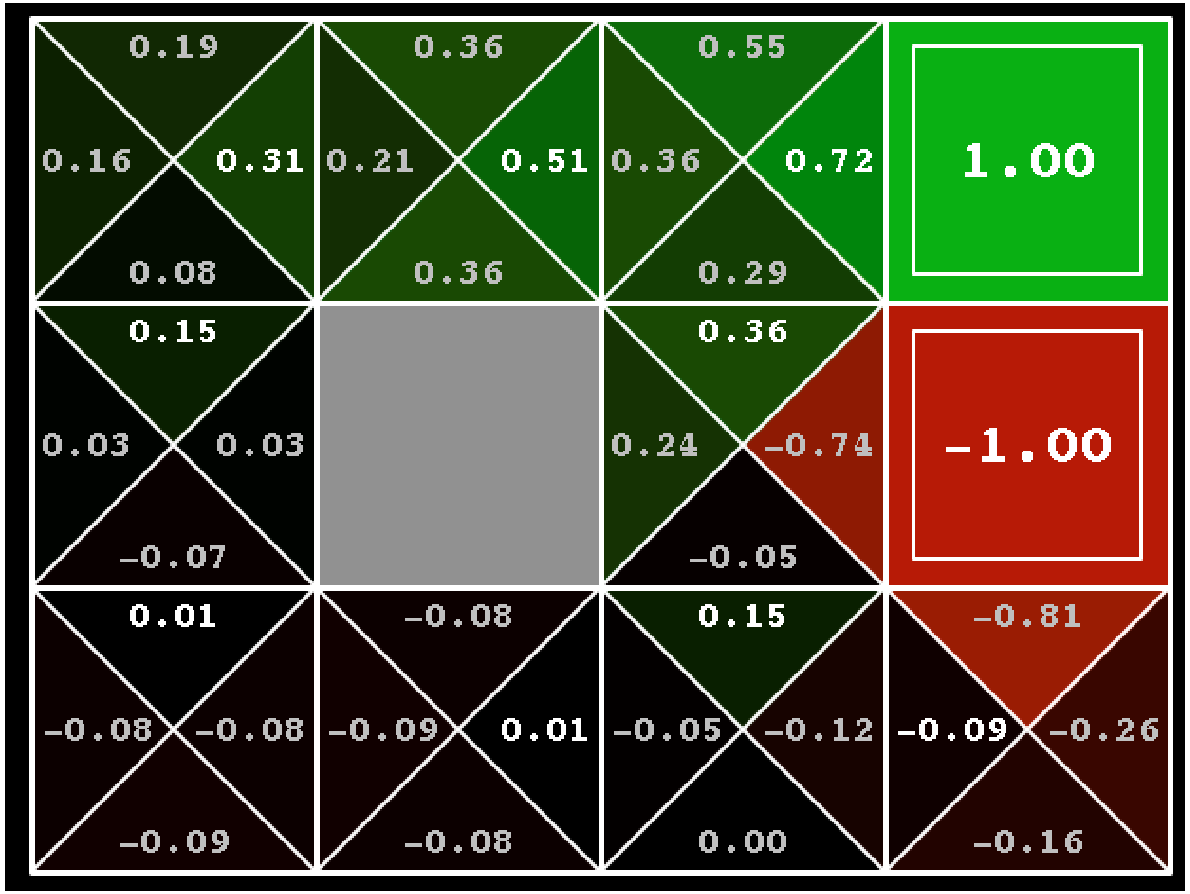


Noise = 0.2
Discount = 0.9
Living reward = -0.1

Snapshot of Demo – Gridworld Q Values

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q^*(s,a)$$

What is the optimal policy?



Noise = 0.2
Discount = 0.9
Living reward = -0.1

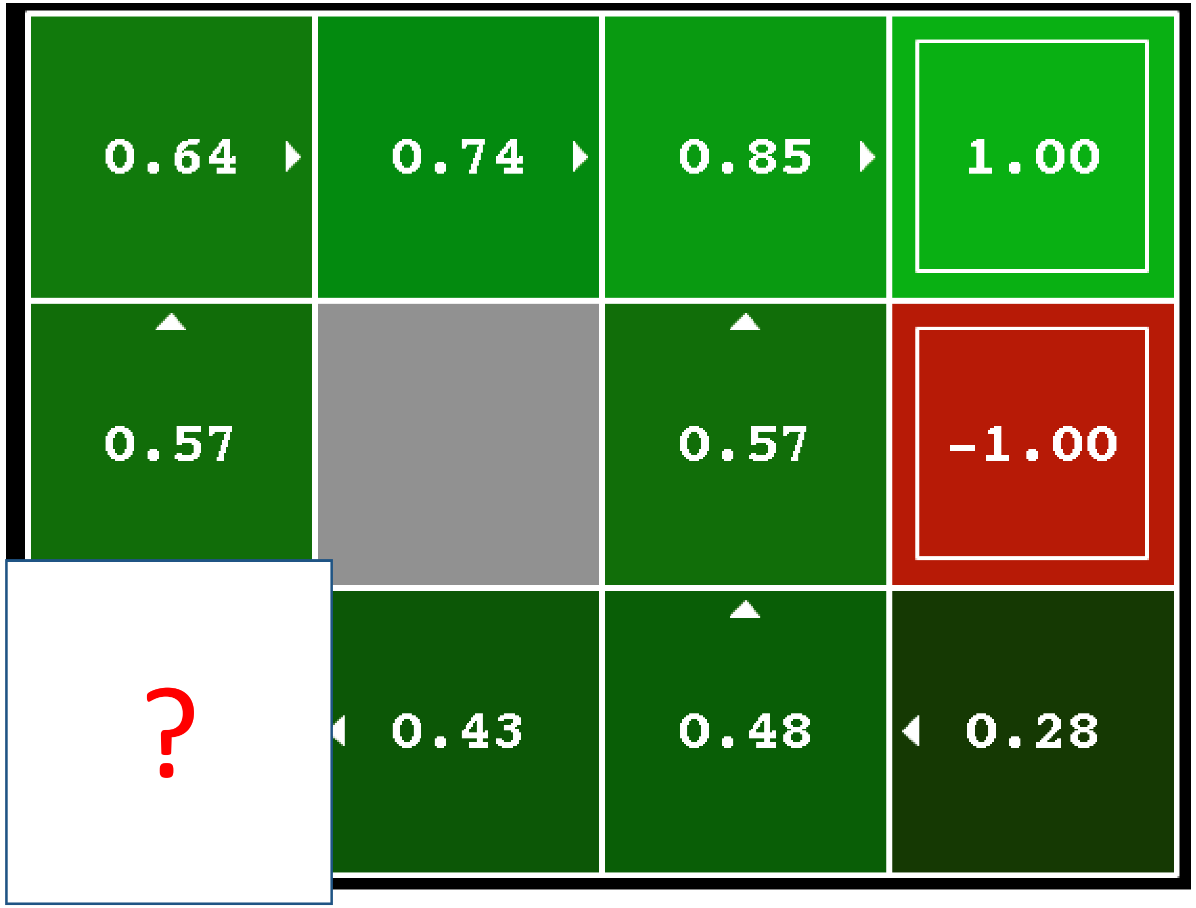
Backup Slides

Snapshot of Demo – Gridworld V Values

What is $V^*(s_{1,1})$?

What is $\pi^*(s_{1,1})$?

$V^*(s)$



Noise = 0.2
Discount = 0.9
Living reward = 0

Snapshot of Demo – Gridworld V Values

What is $V^*(s_{1,1})$?

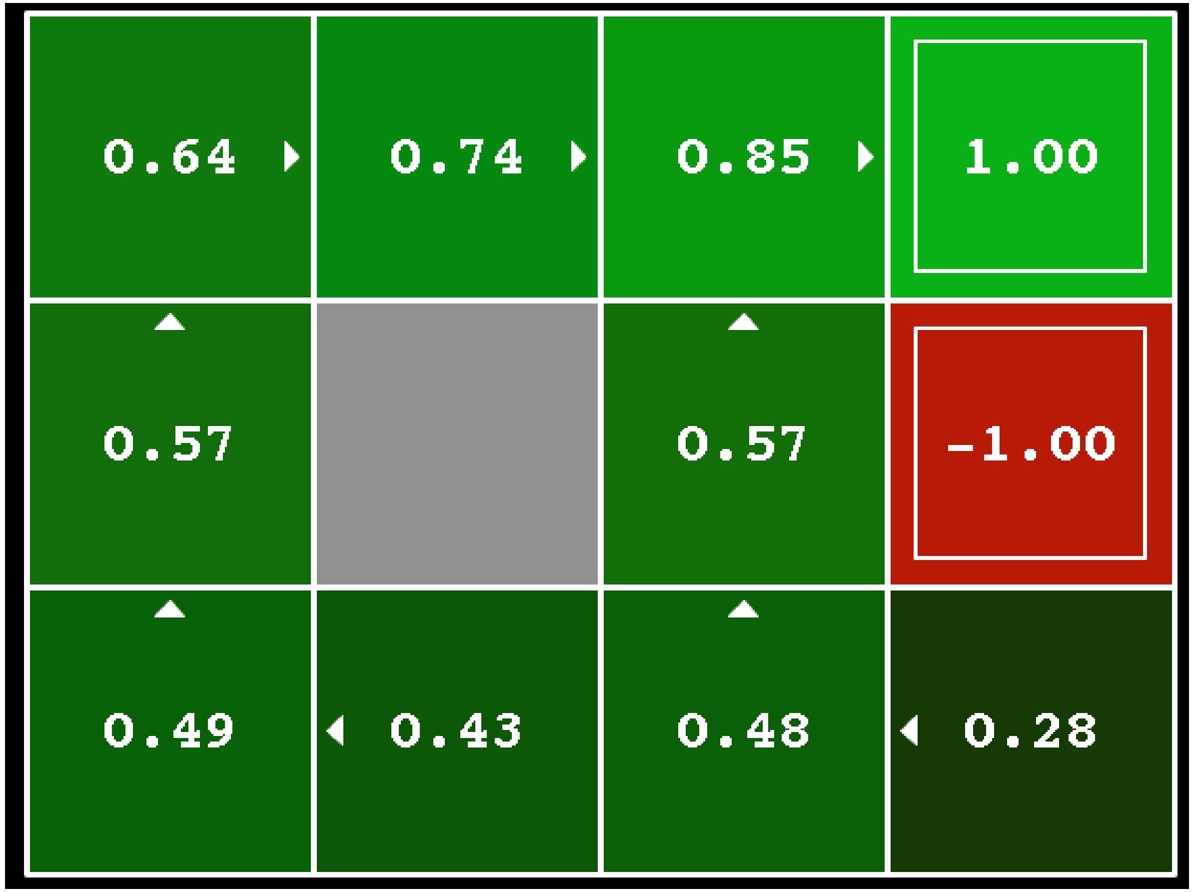
What is $\pi^*(s_{1,1})$?

Let $V^*(s_{1,1}) = x$.

If $\pi^*(s_{1,1})$ =North: x=0+0.9*(0.8*0.57+0.1*x+0.1*0.43) ⇒ x=0.493

If $\pi^*(s_{1,1})$ =South: x=0+0.9*(0.8*x+0.1*x+0.1*0.43) ⇒ x=0.203

But when x=0.203, taking action North leads to higher expected value. Contradiction.



Noise = 0.2
Discount = 0.9
Living reward = 0