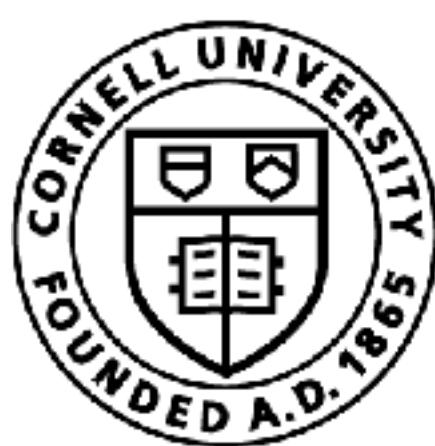


Solving Markov Decision Processes

Sanjiban Choudhury

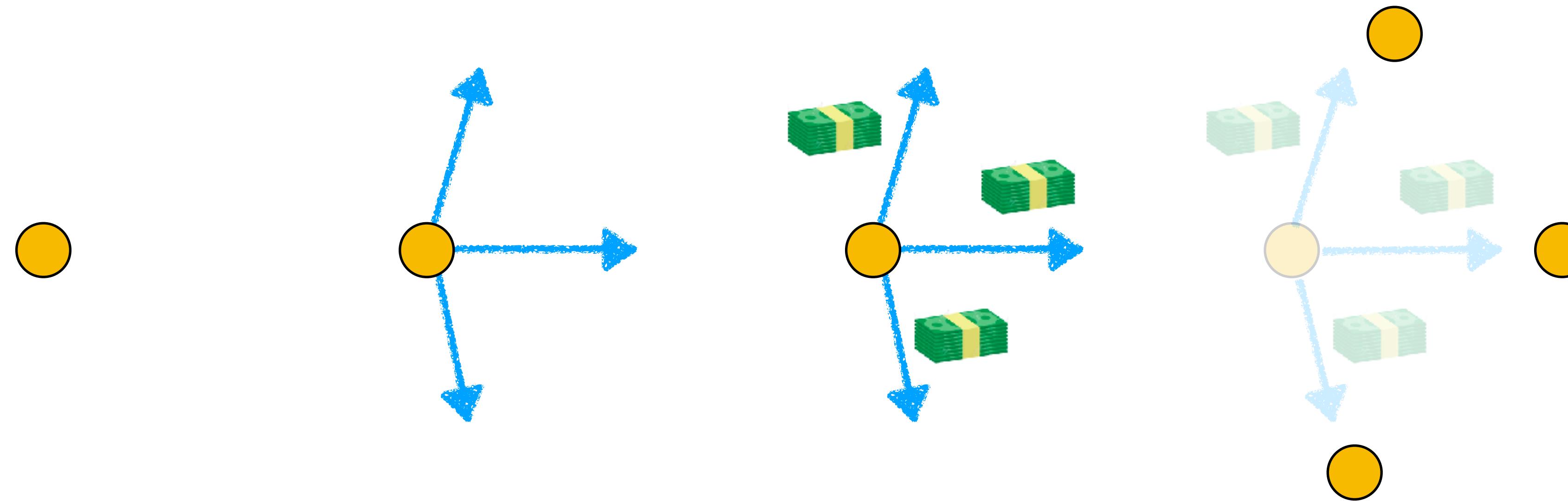


Cornell Bowers CIS
Computer Science

Markov Decision Process

A mathematical framework for modeling sequential decision making

$\langle S, A, C, \mathcal{T} \rangle$

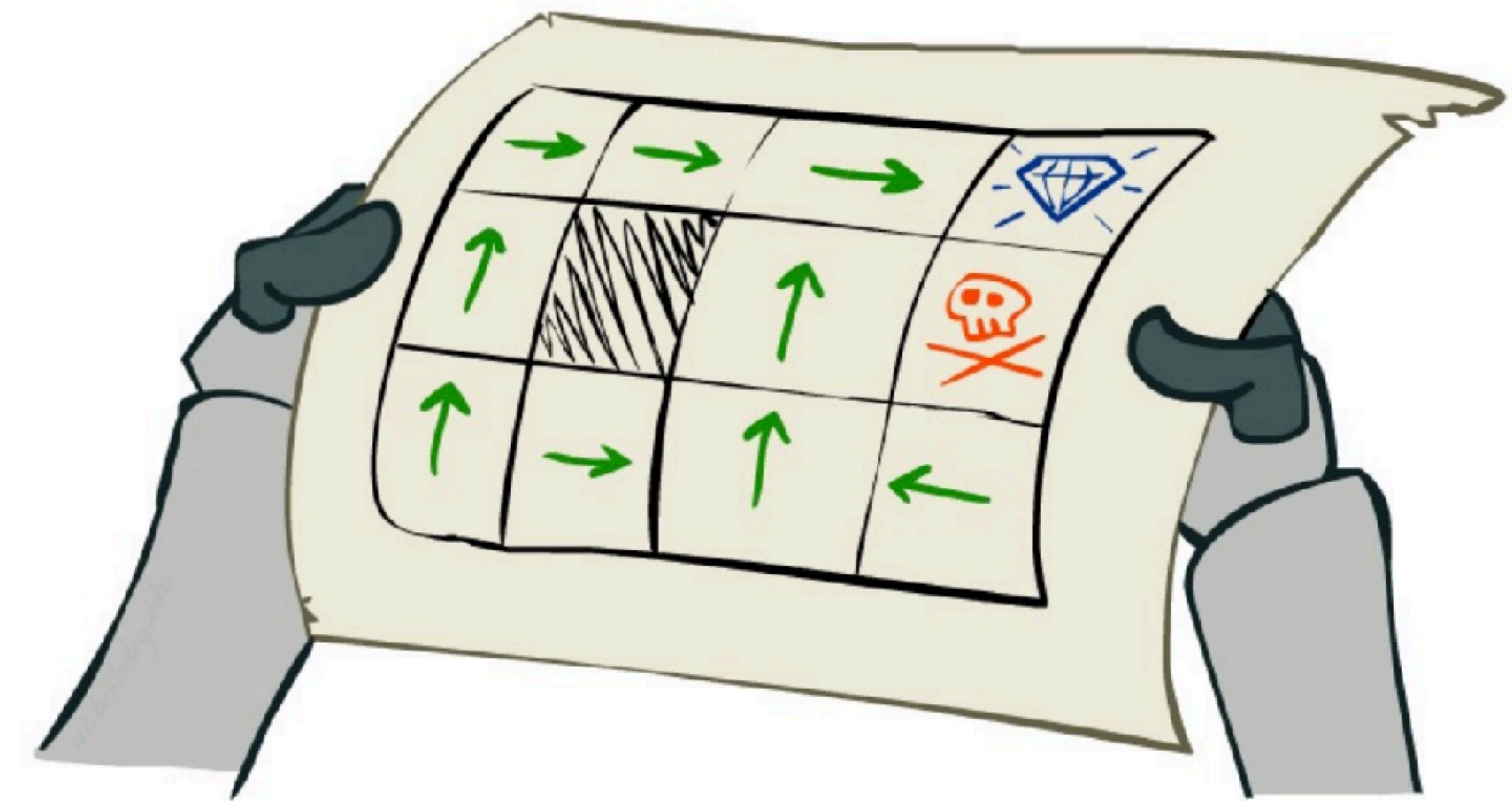


What does it mean to solve
a MDP?

Solving an MDP means finding a **Policy**

$$\pi : s_t \rightarrow a_t$$

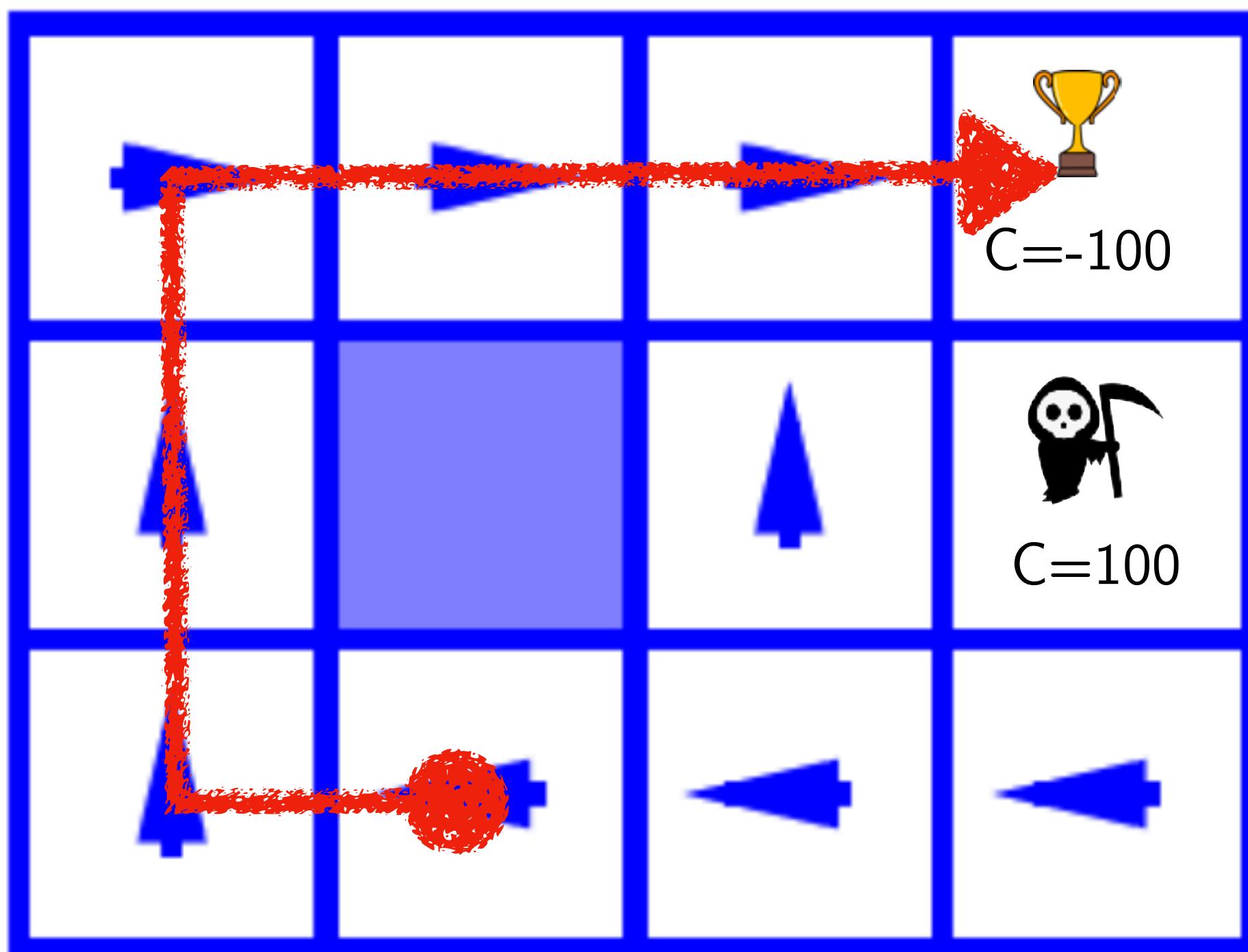
A function that maps state (and time) to action



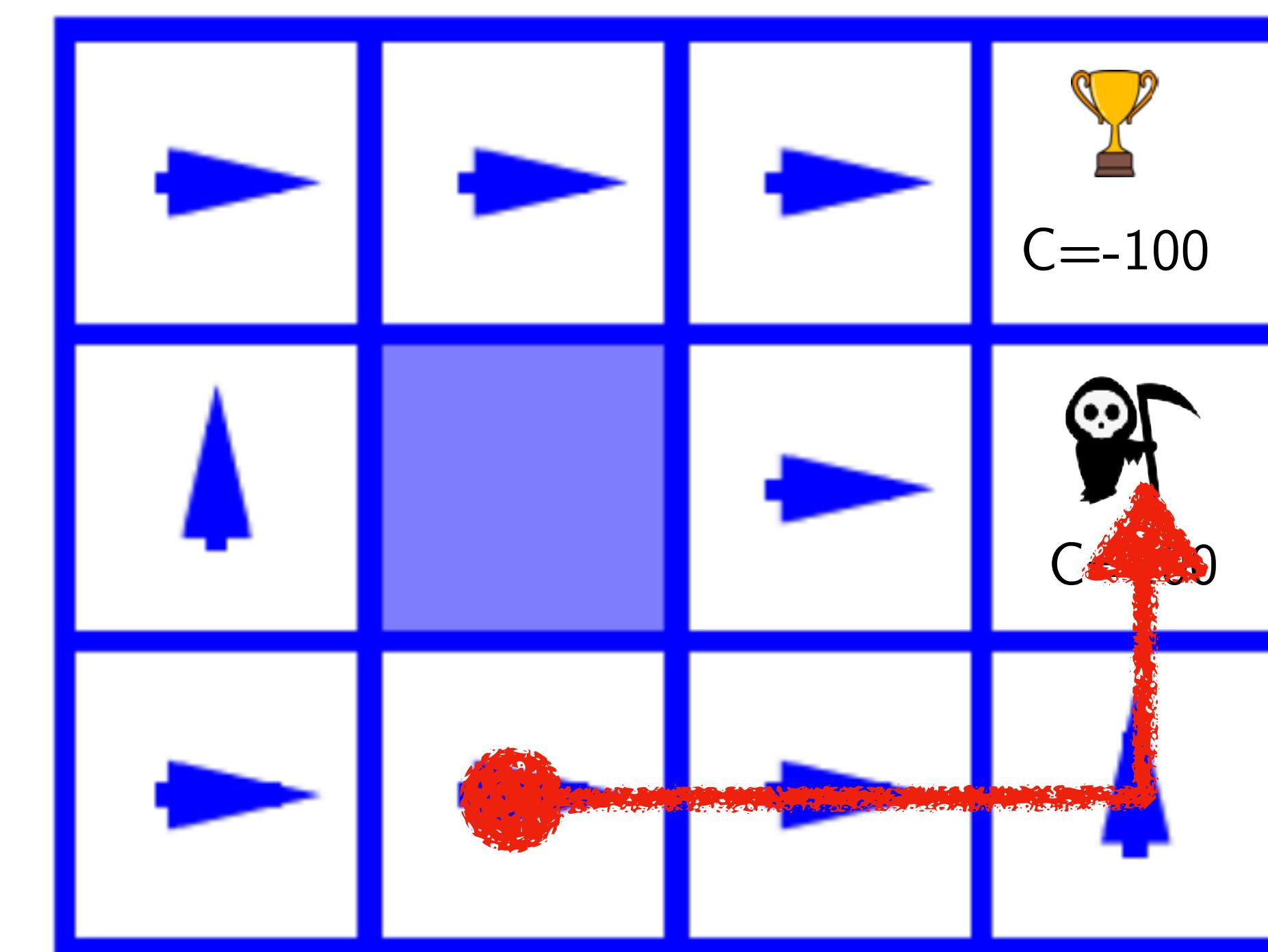
Policy: What action should I choose at any state?

What makes a policy *optimal*?

Which policy is better?



Policy π_1



Policy π_2

What makes a policy *optimal*?

$$\min_{\pi} \mathbb{E}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} \left[\sum_{t=0}^{T-1} c(s_t, a_t) \right]$$

(Search over Policies)

(Sample a start state, then follow π till end of episode)

(Sum over all costs)

One last piece ...

Which of the two outcomes do you prefer?

\$50 today

\$1 million
a 1000 days later

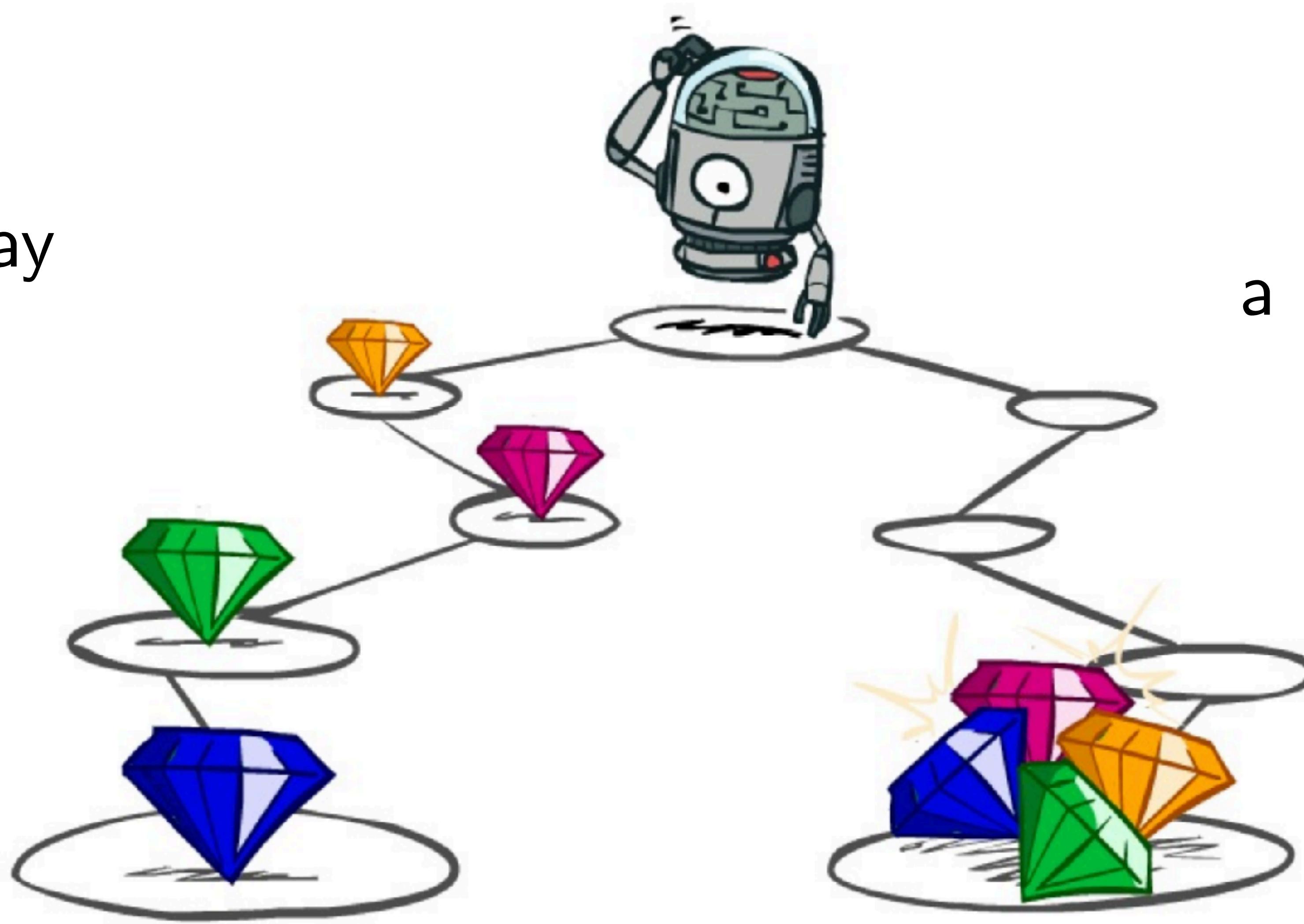


Image courtesy Dan Klein

Discount: Future rewards / costs matter less



1

Worth Now



γ

Worth Next Step



γ^2

Worth In Two Steps

At what discount value does it make sense to take
\$50 today than \$1million in 1000 days?

Image courtesy Dan Klein

What makes a policy *optimal*?

$$\min_{\pi} \mathbb{E}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} \left[\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t) \right]$$

(Search over Policies)

(Sample a start state, then follow π till end of episode)

(Discounted sum of costs)

How do we solve a MDP?

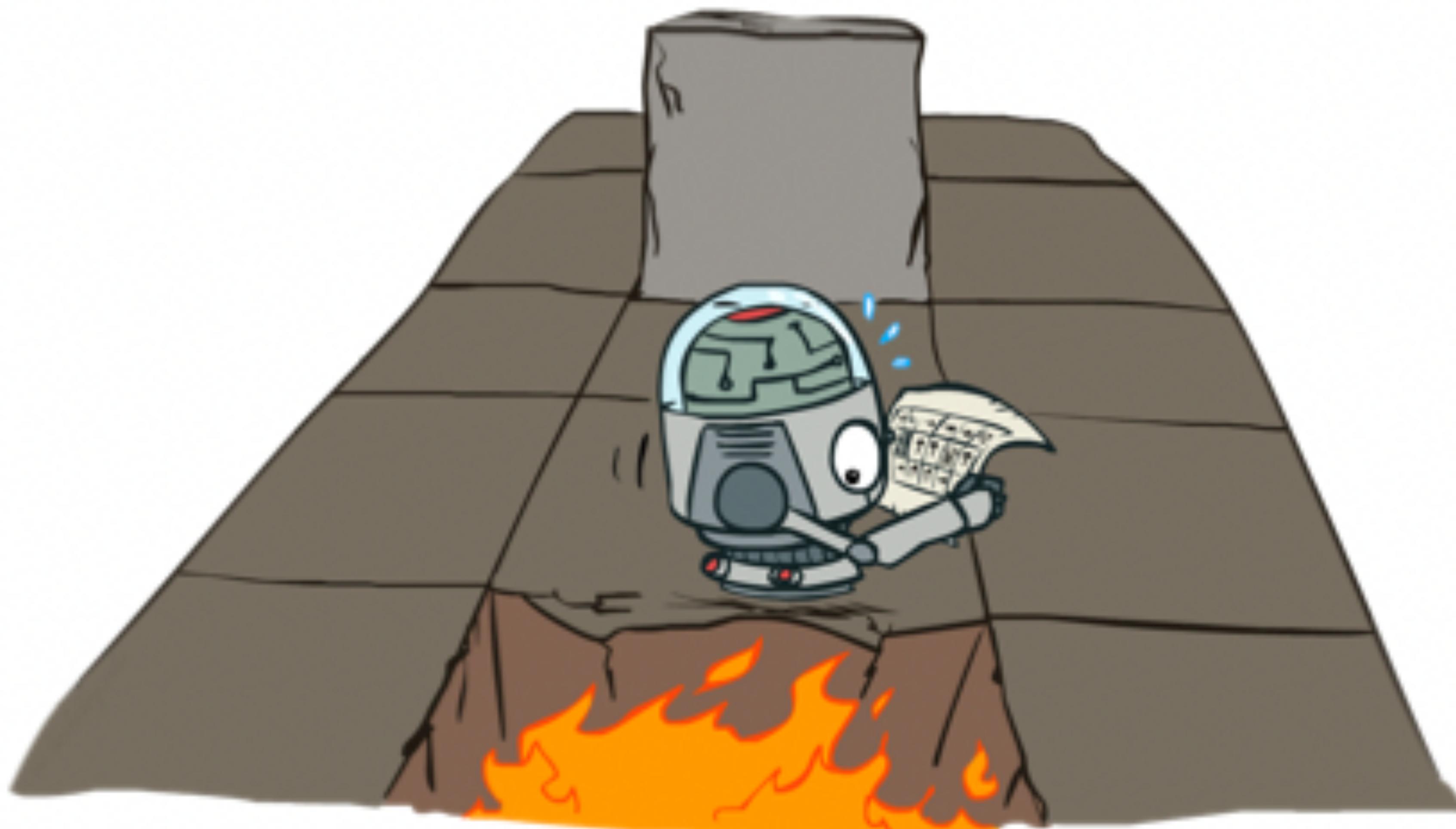


Image courtesy Dan Klein

Let's start with how NOT
to solve MDPs

What would brute force do?

$$\min_{\pi} \mathbb{E}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} \left[\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t) \right]$$

How much work would brute force have to do?

What would brute force do?

$$\min_{\pi} \mathbb{E}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} \left[\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t) \right]$$

1. Iterate over all possible policies
2. For every policy, evaluate the cost
3. Pick the best one

There are
 $(A^S)^T$
Policies!!!!

MDPs have a very special
structure

Introducing the “Value” Function

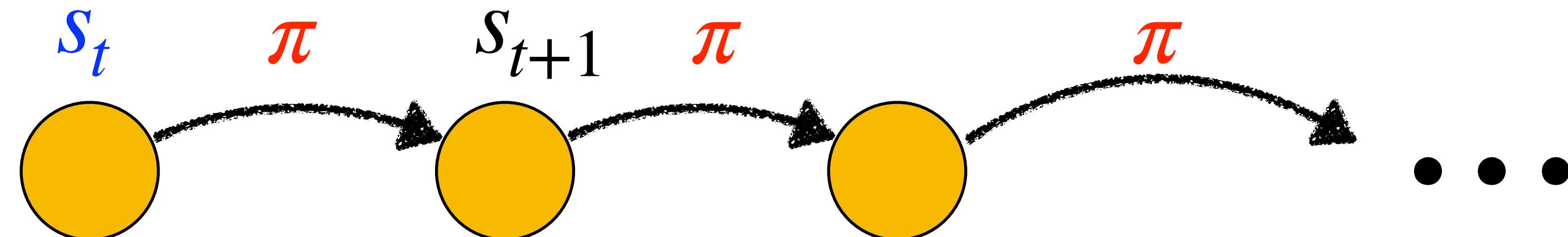
$$V^{\pi}(s_t)$$

Read this as: Value of a **policy** at a given **state and time**

Introducing the “Value” Function

$$V^{\pi}(s_t)$$

Read this as: Value of a **policy** at a given **state and time**



$$V^{\pi}(s_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} + \dots$$

The Bellman Equation

$$V^{\pi}(s_t) = c(s_t, \pi(s_t)) + \gamma \mathbb{E}_{s_{t+1}} V^{\pi}(s_{t+1})$$

*Value of
current state*

Cost

*Value of
future state*

Why is this true?

Optimal policy

$$\pi^* = \arg \min_{\pi} \mathbb{E}_{s_0} V^\pi(s_0)$$

Bellman Equation for the Optimal Policy

$$V^{\pi^*}(s_t) = \min_{a_t} [c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^*}(s_{t+1})]$$

*Optimal
Value*

Cost

*Optimal
Value of
Next State*

Why is this true?

We use V^* to denote optimal value

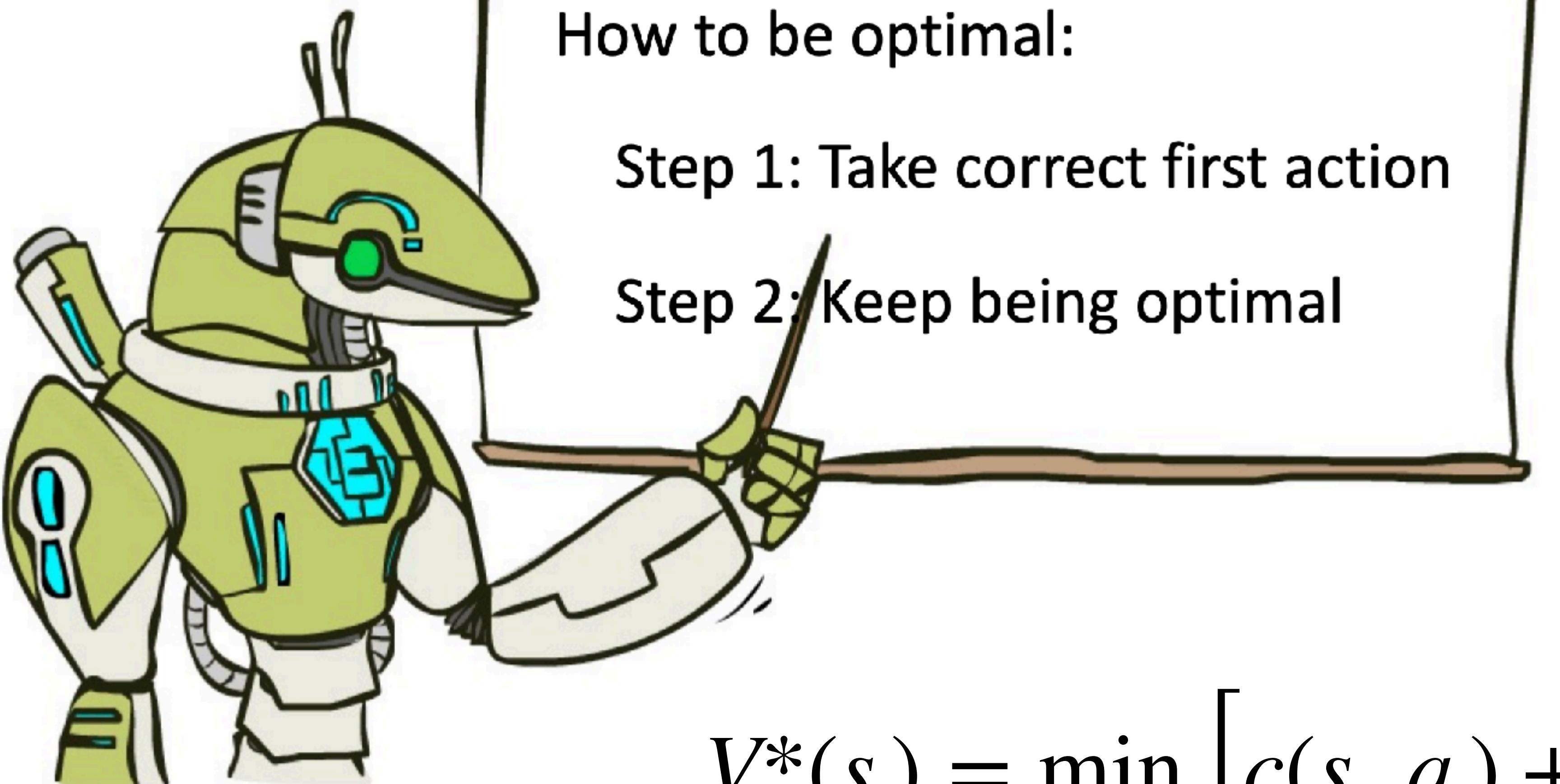
$$V^*(s_t) = \min_{a_t} \left[c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^*(s_{t+1}) \right]$$

*Optimal
Value*

Cost

*Optimal
Value of
Next State*

The Bellman Equation



Activity!



Value Iteration

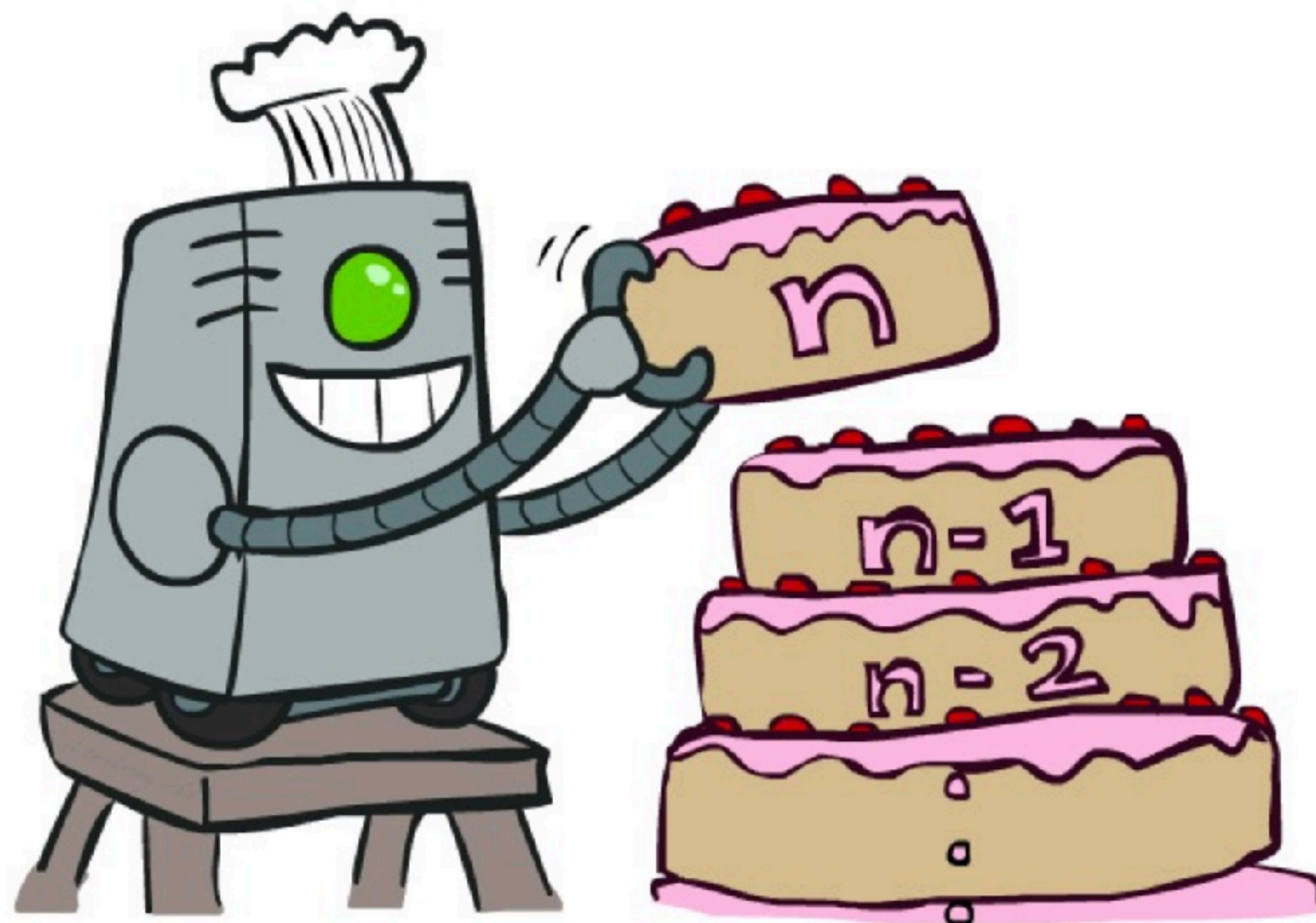
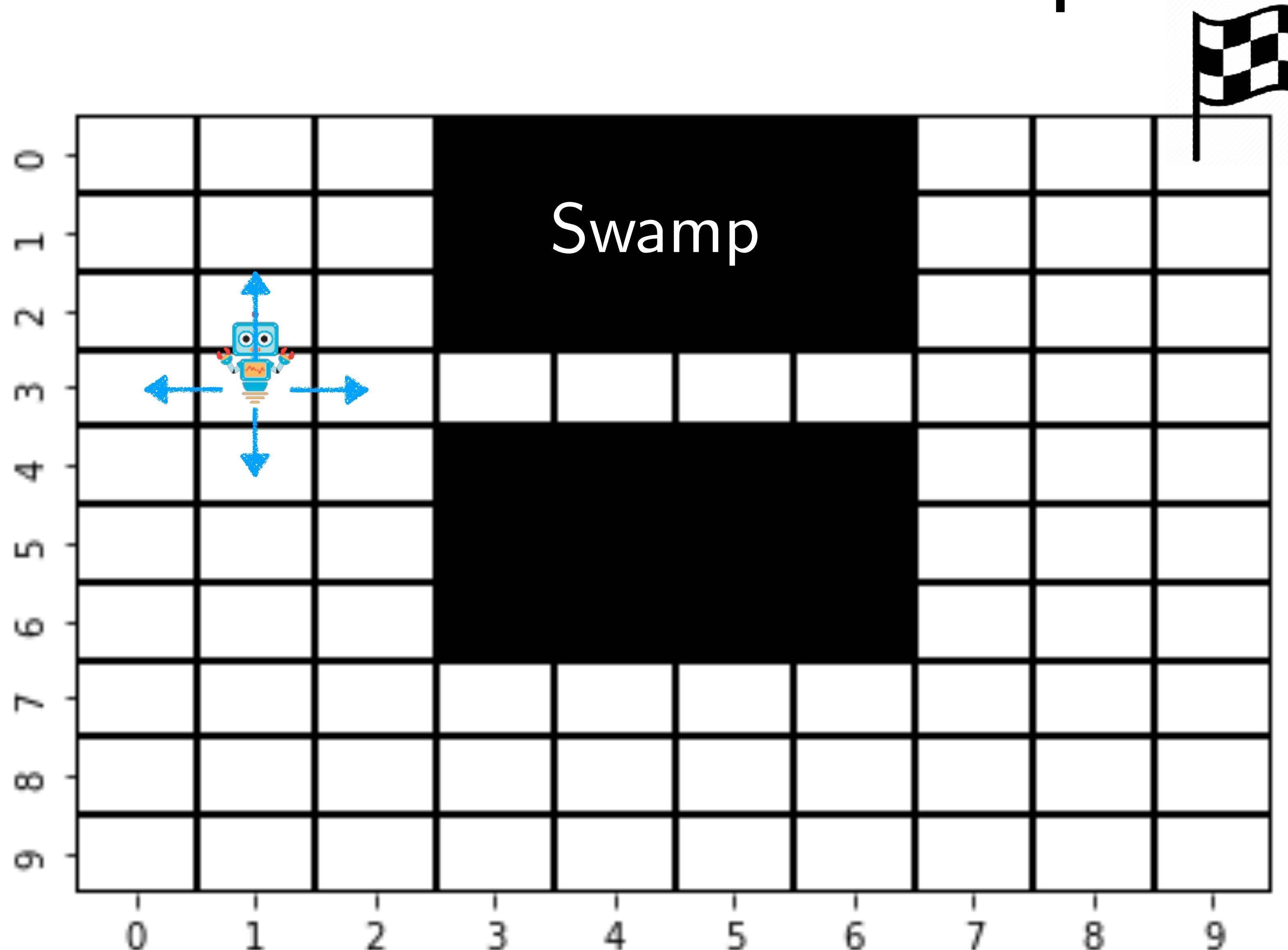


Image courtesy Dan Klein

Setup



$\langle S, A, C, \mathcal{T} \rangle$

- Two absorbing states:
Goal and Swamp
(can never leave)
- $c(s) = 0$ at the goal,
 $c(s) = 1$ everywhere else
- Transitions deterministic
- Time horizon $T = 30$
- Discount $\gamma = 1$

What is the optimal value at T-1?

Time: 29

0	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1

0	x	x	x	x	x	x	x	x	x	0
1	x	x	x	x	x	x	x	x	x	1
2	x	x	x	x	x	x	x	x	x	2
3	x	x	x	x	x	x	x	x	x	3
4	x	x	x	x	x	x	x	x	x	4
5	x	x	x	x	x	x	x	x	x	5
6	x	x	x	x	x	x	x	x	x	6
7	x	x	x	x	x	x	x	x	x	7
8	x	x	x	x	x	x	x	x	x	8
9	x	x	x	x	x	x	x	x	x	9

$$V^*(s_{T-1}) = \min_a c(s_{T-1}, a)$$

$$\pi^*(s_{T-1}) = \arg \min_a c(s_{T-1}, a)$$

What is the optimal value at $T-2$?

Time: 28

0	2	2	2	2	2	2	2	2	1	0
1	2	2	2	2	2	2	2	2	2	1
2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2	2	2
5	2	2	2	2	2	2	2	2	2	2
6	2	2	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	2	2	2	2
8	2	2	2	2	2	2	2	2	2	2
9	2	2	2	2	2	2	2	2	2	2
	0	1	2	3	4	5	6	7	8	9

0	x	x	x	x	x	x	x	x	→	x
1	x	x	x	x	x	x	x	x	↑	
2	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	x
5	x	x	x	x	x	x	x	x	x	x
6	x	x	x	x	x	x	x	x	x	x
7	x	x	x	x	x	x	x	x	x	x
8	x	x	x	x	x	x	x	x	x	x
9	x	x	x	x	x	x	x	x	x	x
	0	1	2	3	4	5	6	7	8	9

$$V^*(s_{T-2}) = \min_a [c(s_{T-2}, a) + V^*(s_{T-1})]$$

$$\pi^*(s_{T-2}) = \arg \min_a [c(s_{T-2}, a) + V^*(s_{T-1})]$$

Dynamic Programming all the way!

Time: 16

0	14	14	13	14	14	14	14	2	1	0
1	14	13	12	14	14	14	14	3	2	1
2	13	12	11	14	14	14	14	4	3	2
3	12	11	10	9	8	7	6	5	4	3
4	13	12	11	14	14	14	14	6	5	4
5	14	13	12	14	14	14	14	7	6	5
6	14	14	13	14	14	14	14	8	7	6
7	14	14	14	13	12	11	10	9	8	7
8	14	14	14	14	13	12	11	10	9	8
9	14	14	14	14	14	13	12	11	10	9

0	x	x	↓	x	x	x	x	→	→	x
1	x	→	↓	x	x	x	x	→	→	↑
2	→	→	↓	x	x	x	x	→	→	↑
3	→	→	→	→	→	→	→	→	→	↑
4	→	→	↑	x	x	x	x	→	→	↑
5	x	→	↑	x	x	x	x	→	→	↑
6	x	x	↑	x	x	x	x	→	→	↑
7	x	x	x	→	→	→	→	→	→	↑
8	x	x	x	x	→	→	→	→	→	↑
9	x	x	x	x	→	→	→	→	→	↑

$$V^*(s_t) = \min_a [c(s_t, a) + V^*(s_{t+1})]$$

$$\pi^*(s_t) = \arg \min_a [c(s_t, a) + V^*(s_{t+1})]$$

Value Iteration

Initialize value function at last time-step

$$V^*(s, T - 1) = \min_a c(s, a)$$

for $t = T - 2, \dots, 0$

Compute value function at time-step t

$$V^*(s, t) = \min_a \left[c(s, a) + \gamma \sum_{s'} \mathcal{T}(s' | s, a) V^*(s', t + 1) \right]$$

Quiz!



Computational complexity of value iteration

Initialize value function at last time-step

$$V^*(s, T - 1) = \min_a c(s, a)$$

for $t = T - 2, \dots, 0$

Compute value function at time-step t

$$V^*(s, t) = \min_a \left[c(s, a) + \gamma \sum_{s'} \mathcal{T}(s' | s, a) V^*(s', t + 1) \right]$$

When poll is active respond at PollEv.com/sc2582



Why is the optimal policy a function of time?

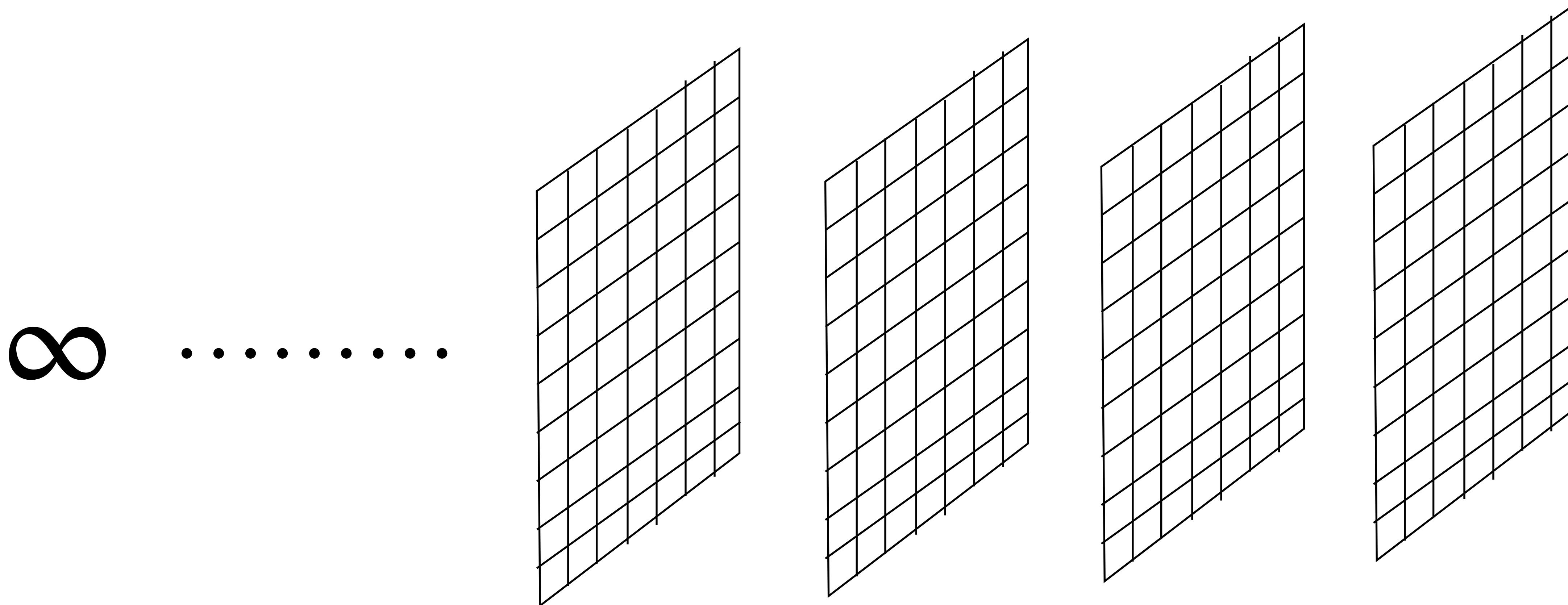


Pulling the goalie
when you
are losing and have
seconds left ..

What happens when horizon is infinity?

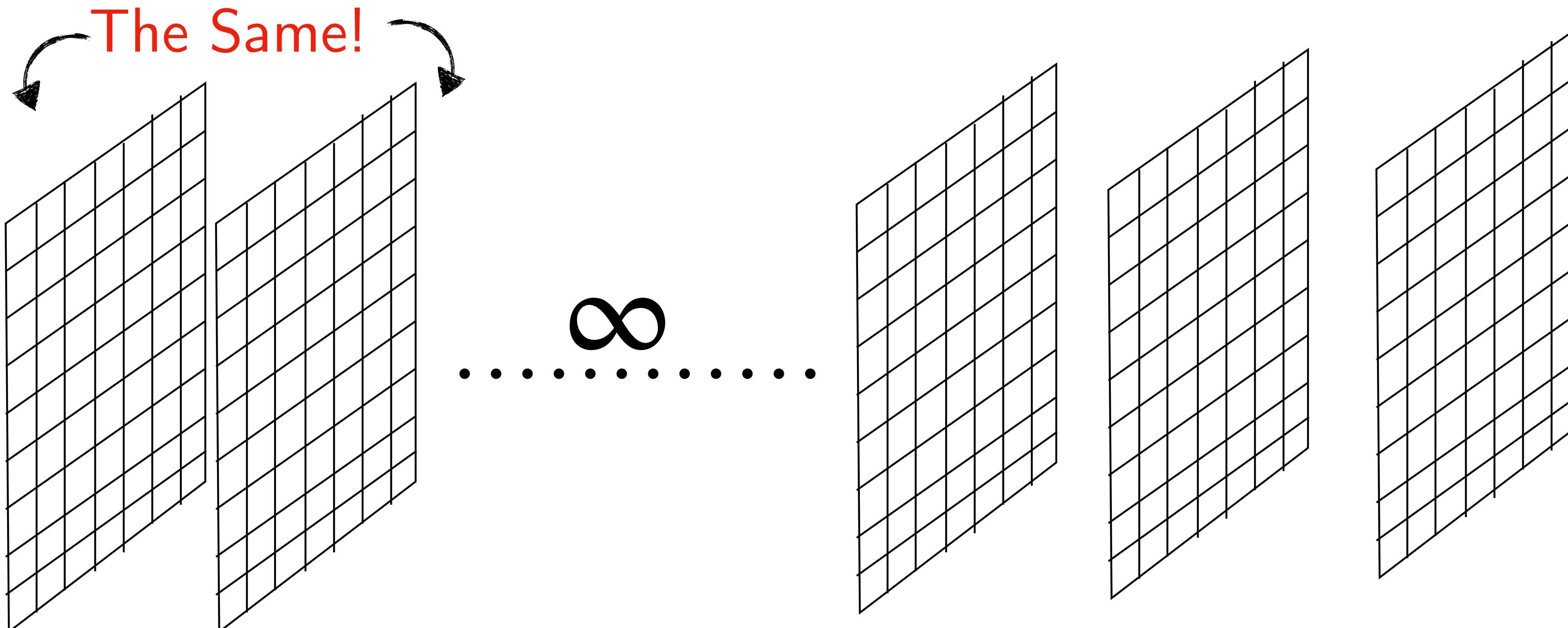


What happens when horizon is infinity?



$$V^{\pi^*}(s_t) = \min_{a_t} [c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^*}(s_{t+1})]$$

Value Function Converges! (For $\gamma < 1$)



$$V^*(s) = \min_a [c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^*(s')]$$

Infinite Horizon Value Iteration

Initialize with some value function $V^*(s)$

Repeat forever

Update values

$$V^*(s) = \min_a \left[c(s, a) + \gamma \sum_{s'} \mathcal{T}(s' | s, a) V^*(s') \right]$$

Policy Iteration

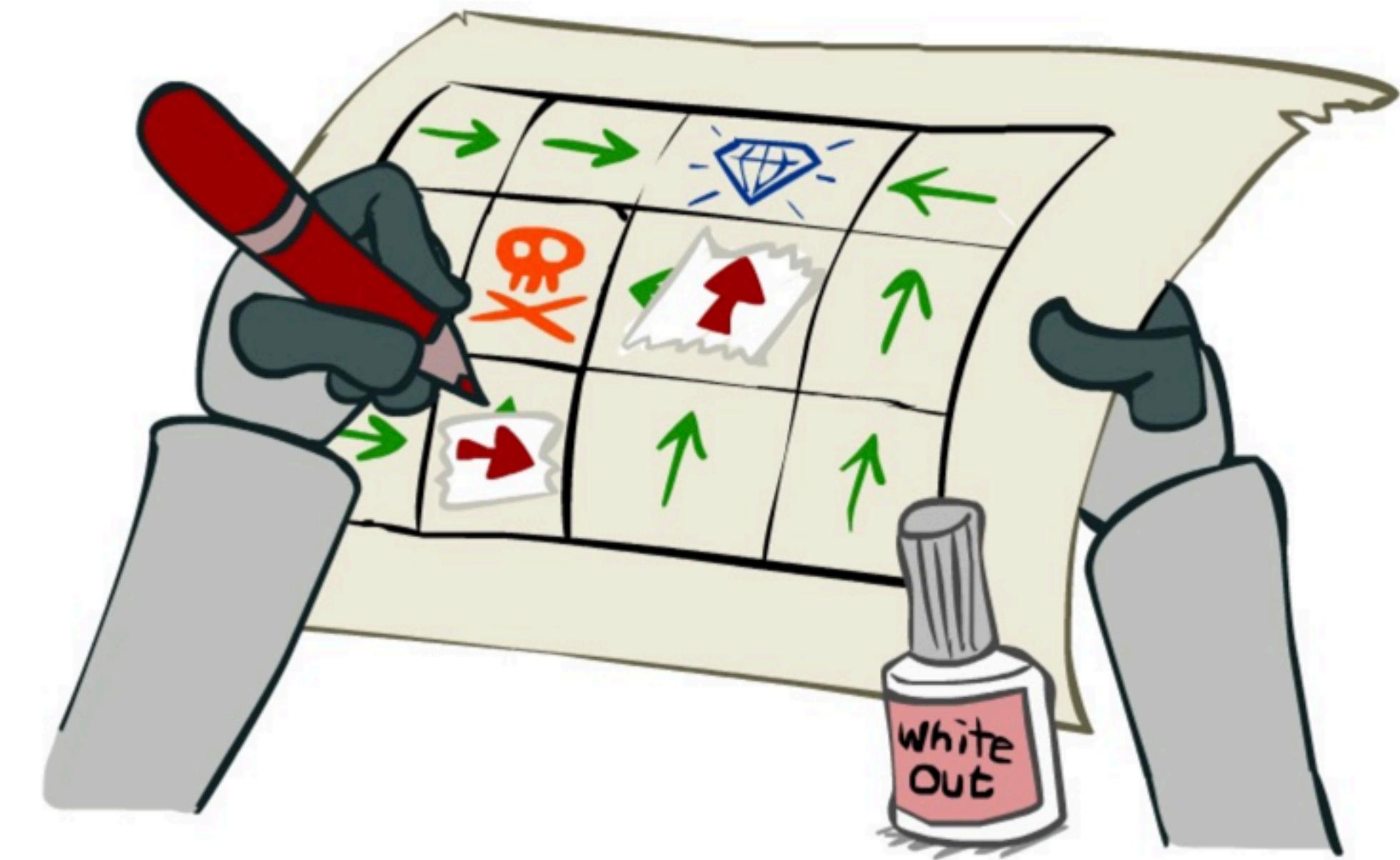
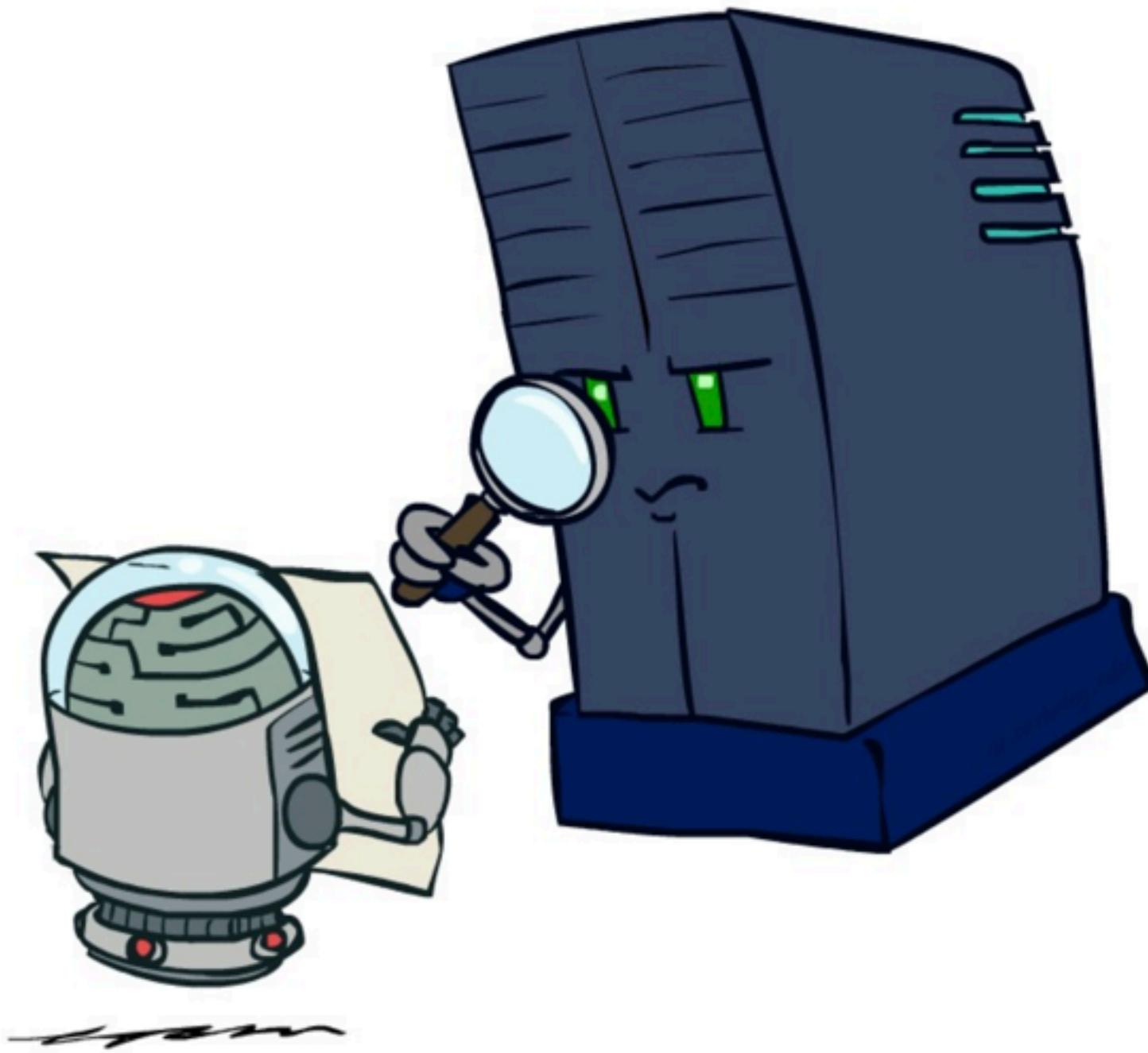


Image courtesy Dan Klein

Which converges faster: value or policy?

$\sigma = 0$	10	10	10	10	10	10	10	10	10	
$\sigma = 1$	10	10	10	10	10	10	10	10	10	
$\sigma = 2$	10	10	10	10	10	10	10	10	10	
$\sigma = 3$	10	10	10	10	10	10	10	10	10	
$\sigma = 4$	10	10	10	10	10	10	10	10	10	
$\sigma = 5$	10	10	10	10	10	10	10	10	10	
$\sigma = 6$	10	10	10	10	10	10	10	10	10	
$\sigma = 7$	10	10	10	10	10	10	10	10	10	
$\sigma = 8$	10	10	10	10	10	10	10	10	10	
$\sigma = 9$	10	10	10	10	10	10	10	10	10	
	0	1	2	3	4	5	6	7	8	9

Values

$\sigma = 0$	x	x	x	x	x	x	x	x	x	
$\sigma = 1$	x	x	x	x	x	x	x	x	x	
$\sigma = 2$	x	x	x	x	x	x	x	x	x	
$\sigma = 3$	x	x	x	x	x	x	x	x	x	
$\sigma = 4$	x	x	x	x	x	x	x	x	x	
$\sigma = 5$	x	x	x	x	x	x	x	x	x	
$\sigma = 6$	x	x	x	x	x	x	x	x	x	
$\sigma = 7$	x	x	x	x	x	x	x	x	x	
$\sigma = 8$	x	x	x	x	x	x	x	x	x	
$\sigma = 9$	x	x	x	x	x	x	x	x	x	
	0	1	2	3	4	5	6	7	8	9

Policy



Policy converges **faster**
than the value

Can we iterate over **policies**?

Policy Iteration

Init with some policy π

Repeat forever

Evaluate policy

$$V^\pi(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^\pi(s')$$

Improve policy

$$\pi^+(s) = \arg \min_a c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^\pi(s')$$

Init with some policy π

Iter: 0

0	-	→	→	→	→	→	→	→	→	→	↑
1	-	→	→	→	→	→	→	→	→	→	↑
2	-	→	→	→	→	→	→	→	→	→	↑
3	-	→	→	→	→	→	→	→	→	→	↑
4	-	→	→	→	→	→	→	→	→	→	↑
5	-	→	→	→	→	→	→	→	→	→	↑
6	-	→	→	→	→	→	→	→	→	→	↑
7	-	→	→	→	→	→	→	→	→	→	↑
8	-	→	→	→	→	→	→	→	→	→	↑
9	-	→	→	→	→	→	→	→	→	→	↑
	↓	0	1	2	3	4	5	6	7	8	9

Iteration 1

Iter: 1

0	74	75	76	77	77	77	77	2	1	0
1	74	75	76	77	77	77	77	3	2	1
2	74	75	76	77	77	77	77	3.9	3	2
3	55	56	56	57	50	40	26	4.9	3.9	3
4	74	75	76	77	77	77	77	5.9	4.9	3.9
5	74	75	76	77	77	77	77	6.8	5.9	4.9
6	74	75	76	77	77	77	77	7.7	6.8	5.9
7	15	14	13	12	11	10	9.6	8.6	7.7	6.8
8	16	15	14	13	12	11	10	9.6	8.6	7.7
9	17	16	15	14	13	12	11	10	9.6	8.6
	0	1	2	3	4	5	6	7	8	9

0	x	←	←	x	x	x	x	→	→	x
1	x	←	←	x	x	x	x	→	→	↑
2	↓	↓	↓	x	x	x	x	→	→	↑
m	x	←	←	→	→	→	→	→	→	↑
4	↑	↑	↑	x	x	x	x	→	→	↑
5	x	←	←	x	x	x	x	→	→	↑
6	↓	↓	↓	x	x	x	x	→	→	↑
7	→	→	→	→	→	→	→	→	→	↑
8	→	→	→	→	→	→	→	→	→	↑
9	→	→	→	→	→	→	→	→	→	↑
	0	1	2	3	4	5	6	7	8	9

$$V^\pi(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} V^\pi(s')$$

$$\pi^+(s) = \arg \min_a c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} V^\pi(s')$$

Policy Iteration

Iter: 0									
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9
1	0	1	2	3	4	5	6	7	8
2	0	1	2	3	4	5	6	7	8
3	0	1	2	3	4	5	6	7	8
4	0	1	2	3	4	5	6	7	8
5	0	1	2	3	4	5	6	7	8
6	0	1	2	3	4	5	6	7	8
7	0	1	2	3	4	5	6	7	8
8	0	1	2	3	4	5	6	7	8
9	0	1	2	3	4	5	6	7	8

$$V^\pi(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} V^\pi(s')$$

$$\pi^+(s) = \arg \min_a c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} V^\pi(s')$$

How do we evaluate policy?

$$V^\pi(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^\pi(s')$$

Idea 1: Start with an initial guess, and update (like value iteration)

$$V^{i+1}(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^i(s')$$

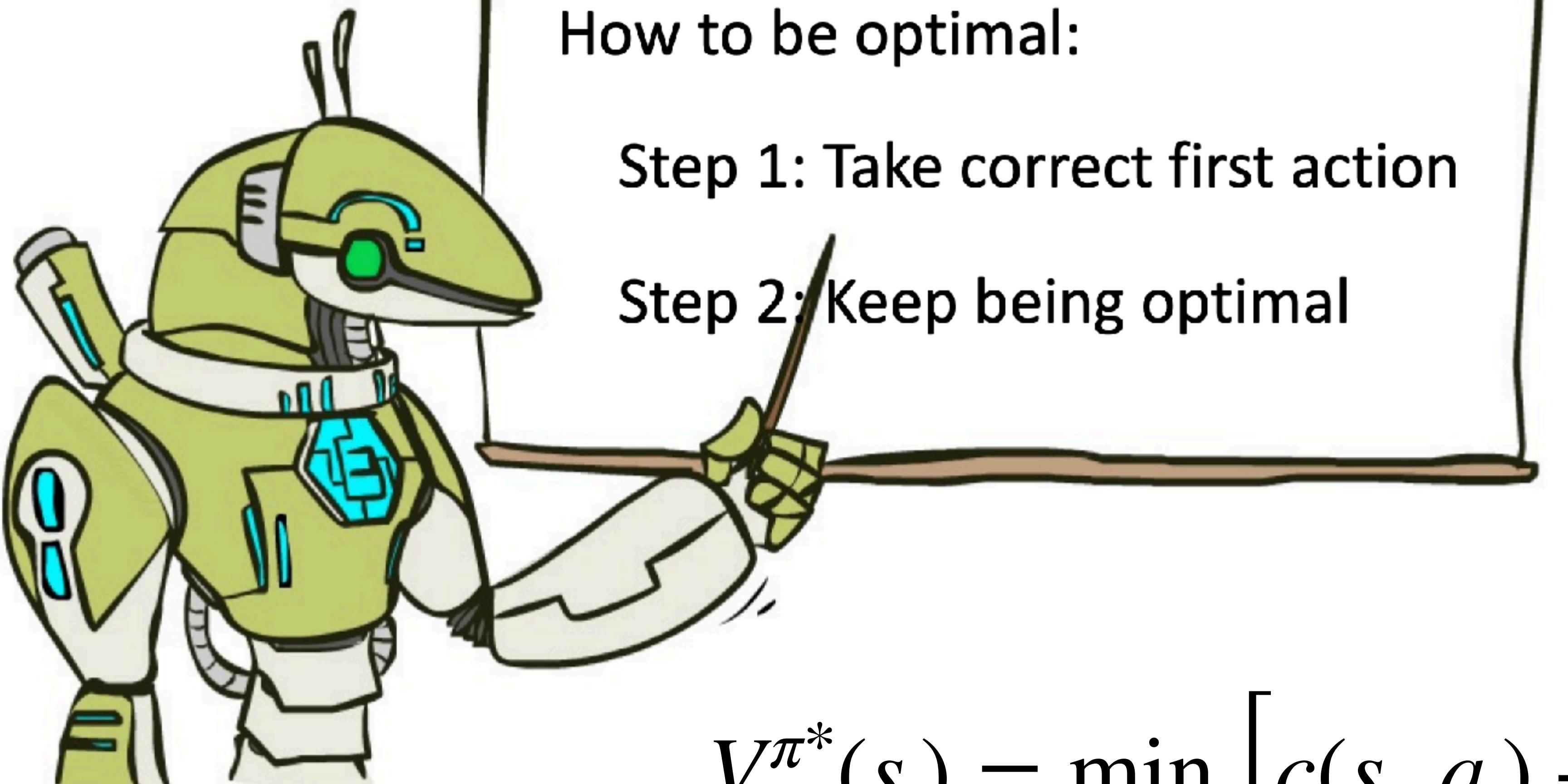
Idea 2: It's a linear set of equations (no max)! Solve for Eigen values

$$\overrightarrow{V^\pi} = \overrightarrow{c^\pi} + \gamma \mathcal{T}^\pi \overrightarrow{V^\pi} \longrightarrow \overrightarrow{V^\pi} = (1 - \mathcal{T}^\pi)^{-1} \overrightarrow{c^\pi}$$

Value Iteration vs Policy Iteration

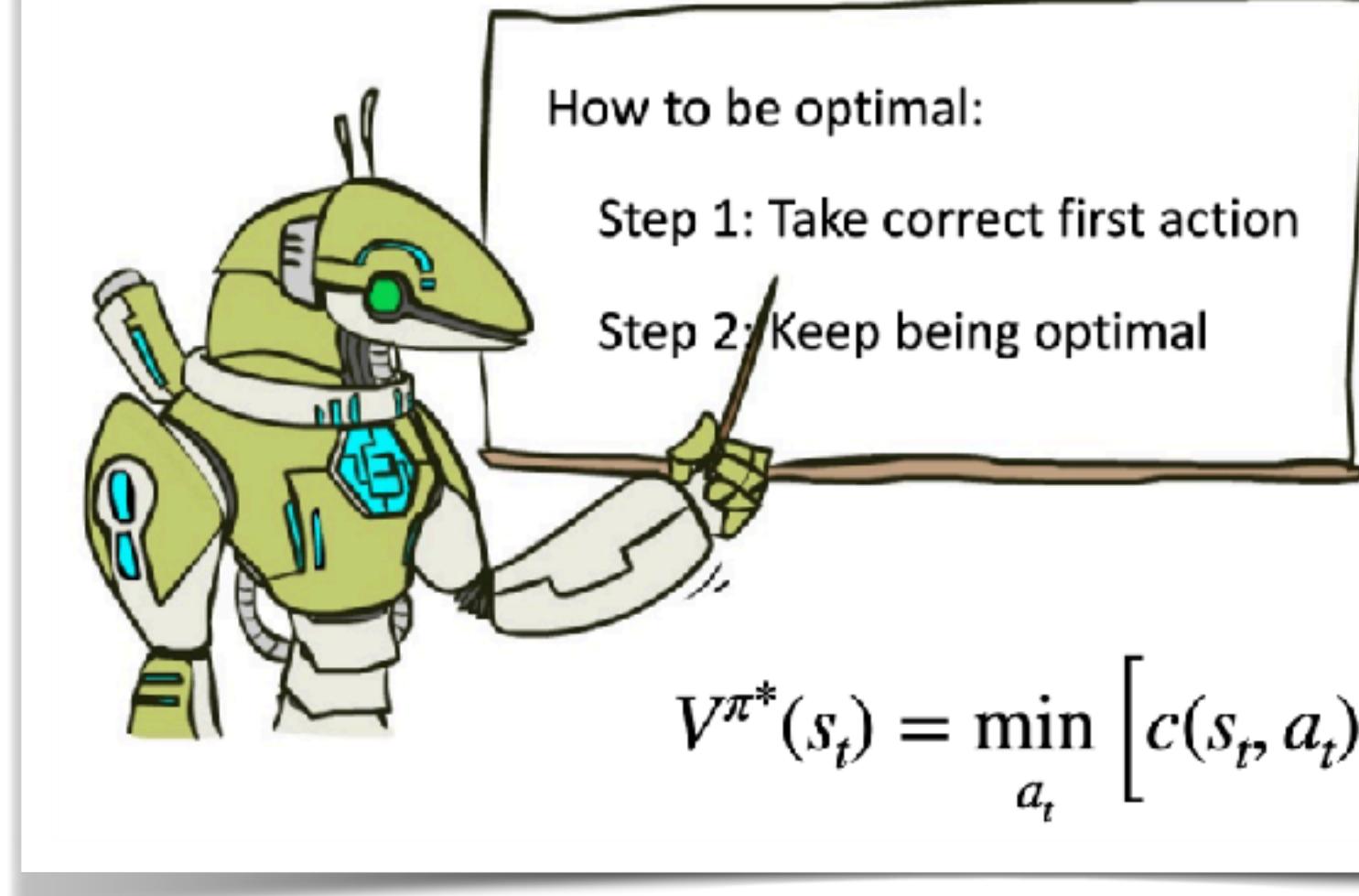
- Both value iteration and policy iteration compute the same thing (all optimal values)
- In value iteration:
 - Every iteration updates both the values and (implicitly) the policy
 - We don't track the policy, but taking the max over actions implicitly recomputes it
- In policy iteration:
 - We do several passes that update utilities with fixed policy (each pass is fast because we consider only one action, not all of them)
 - After the policy is evaluated, a new policy is chosen (slow like a value iteration pass)
 - The new policy will be better (or we're done)
- Both are dynamic programs for solving MDPs

The Bellman Equation



tl;dr

The Bellman Equation



How to be optimal:

- Step 1: Take correct first action
- Step 2: Keep being optimal

$$V^{\pi^*}(s_t) = \min_{a_t} \left[c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^{\pi^*}(s_{t+1}) \right]$$

Value Iteration

Initialize value function at last time-step

$$V^*(s, T-1) = \min_a c(s, a)$$

for $t = T-2, \dots, 0$

Compute value function at time-step t

$$V^*(s, t) = \min_a \left[c(s, a) + \gamma \sum_{s'} \mathcal{T}(s' | s, a) V^*(s', t+1) \right]$$

Policy Iteration

Init with some policy π

Repeat forever

Evaluate policy

$$V^\pi(s) = c(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^\pi(s')$$

Improve policy

$$\pi^+(s) = \arg \min_a c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^\pi(s')$$