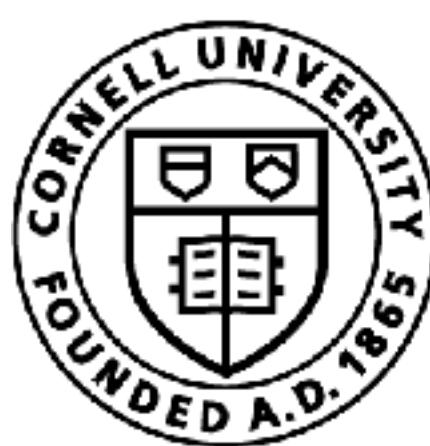


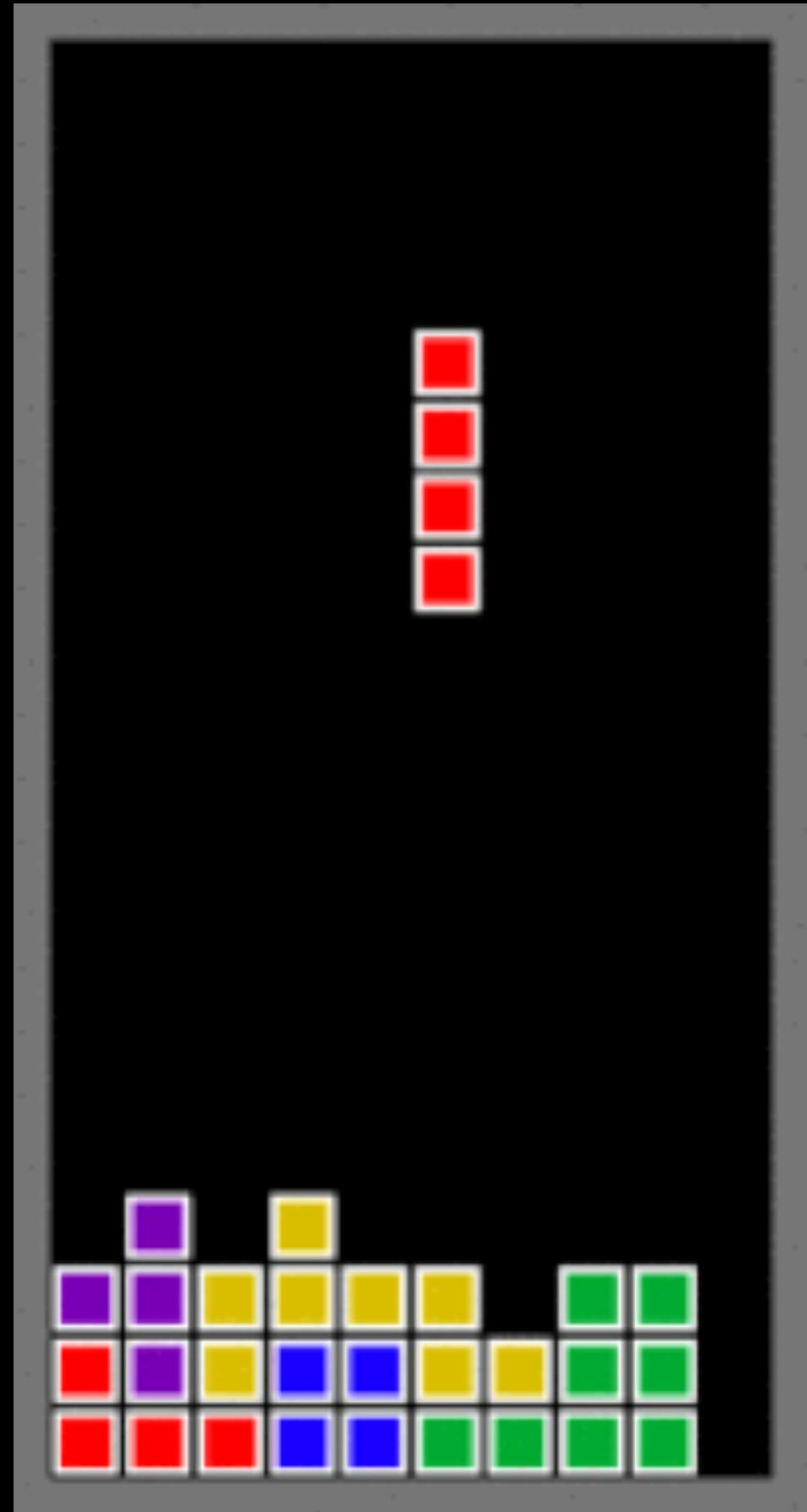
# Robots as Markov Decision Problems

Sanjiban Choudhury

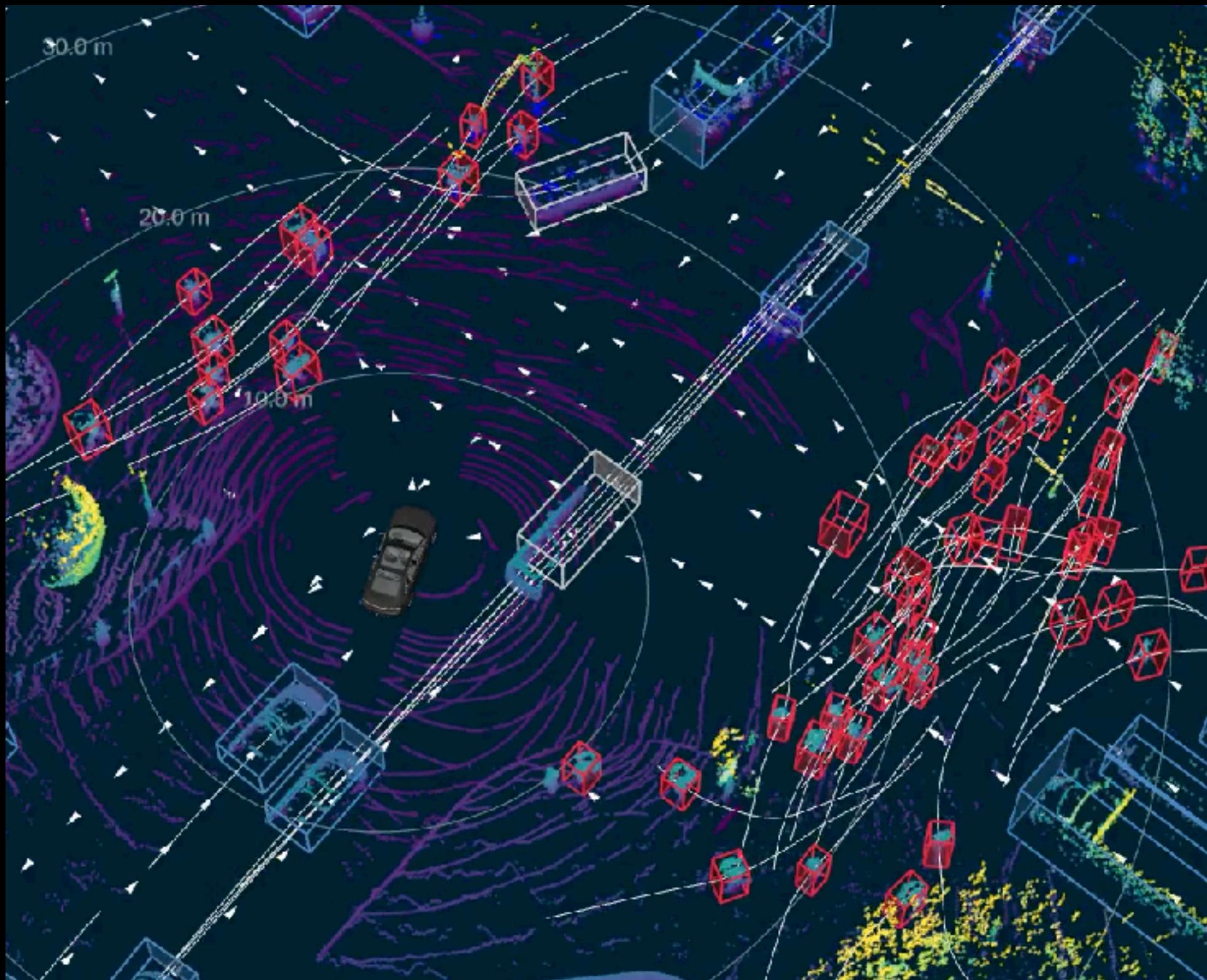


Cornell Bowers CIS  
**Computer Science**

# Decisions, decisions!



Tetris

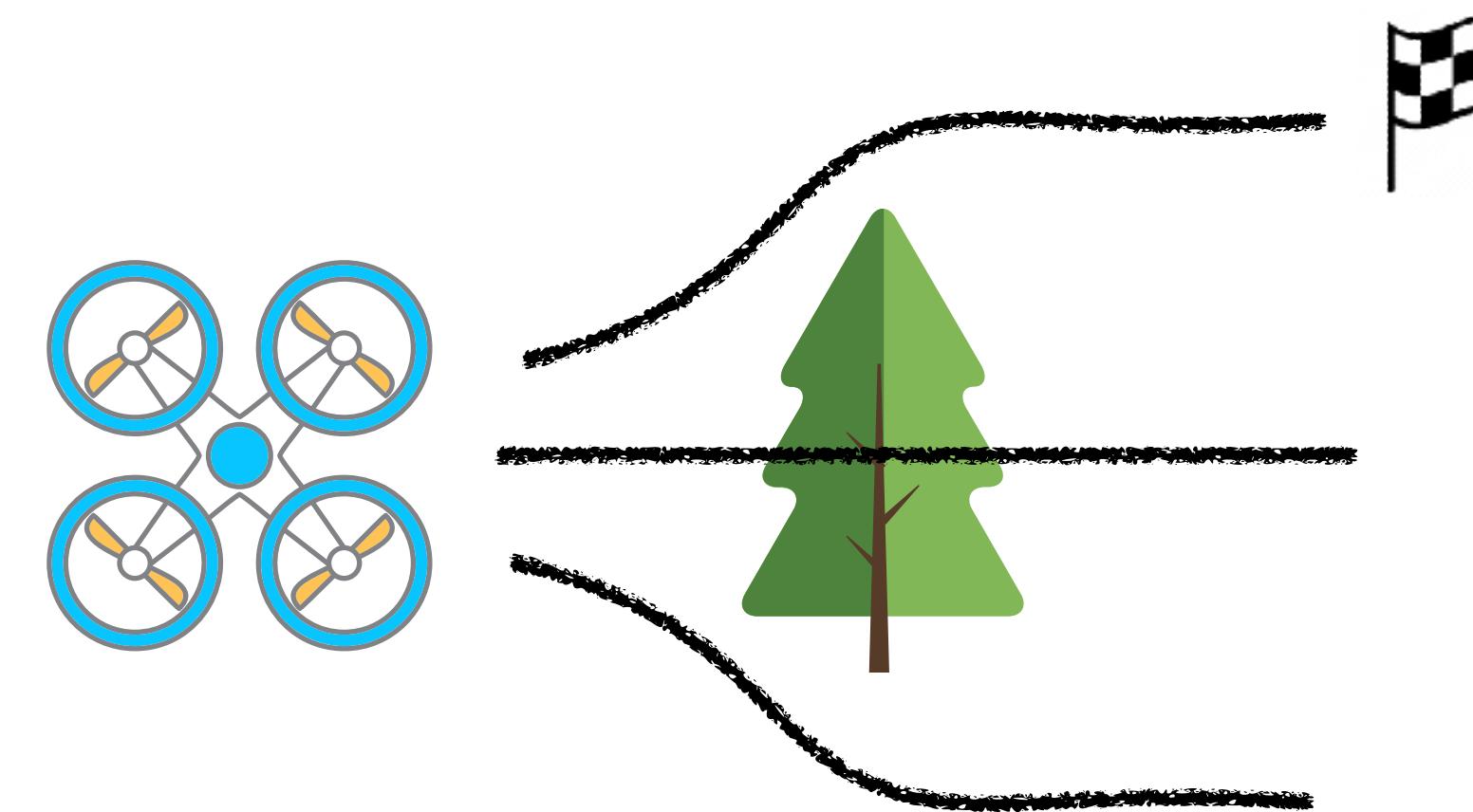


Self-driving



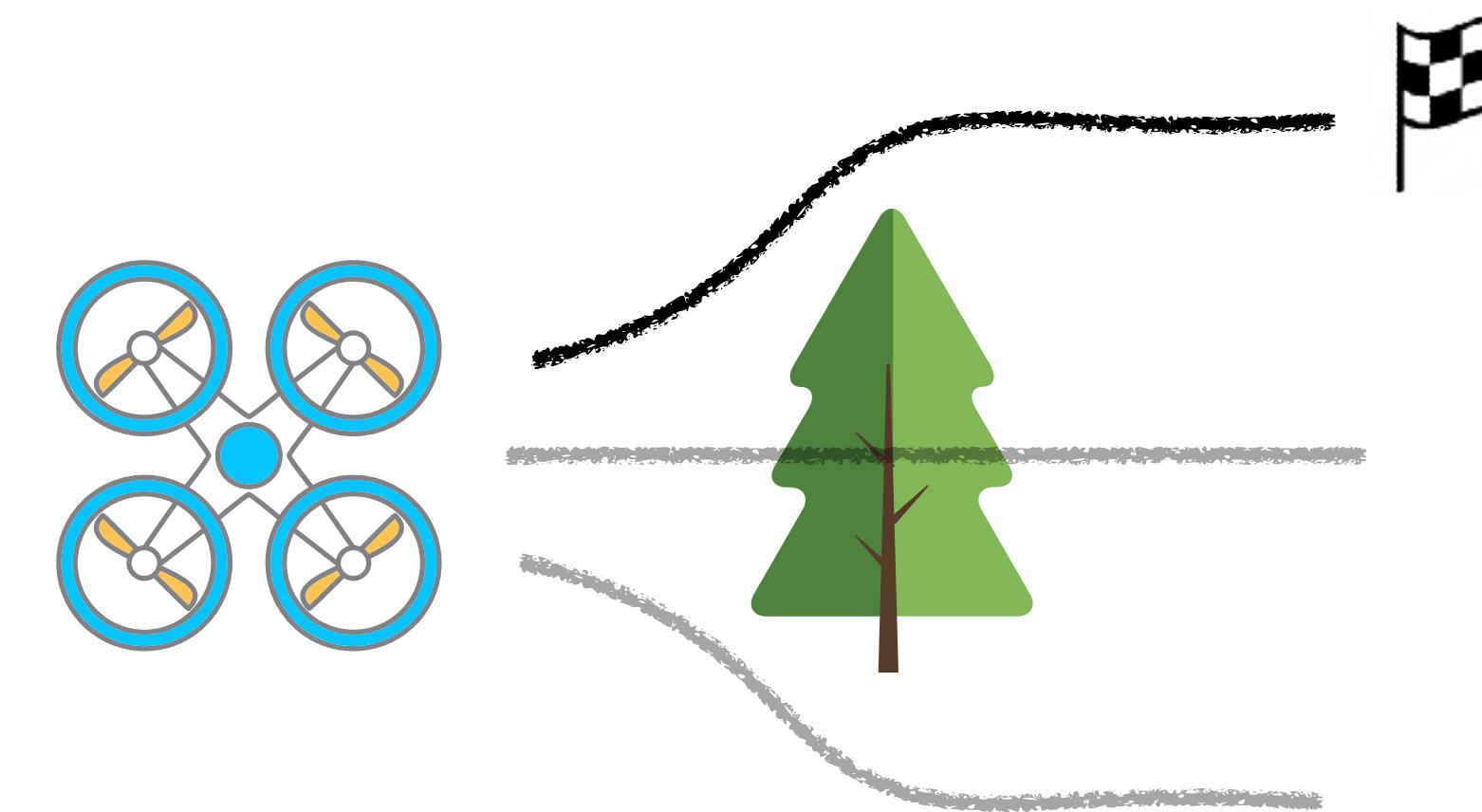
Robot Baristas

# What makes decision making hard?



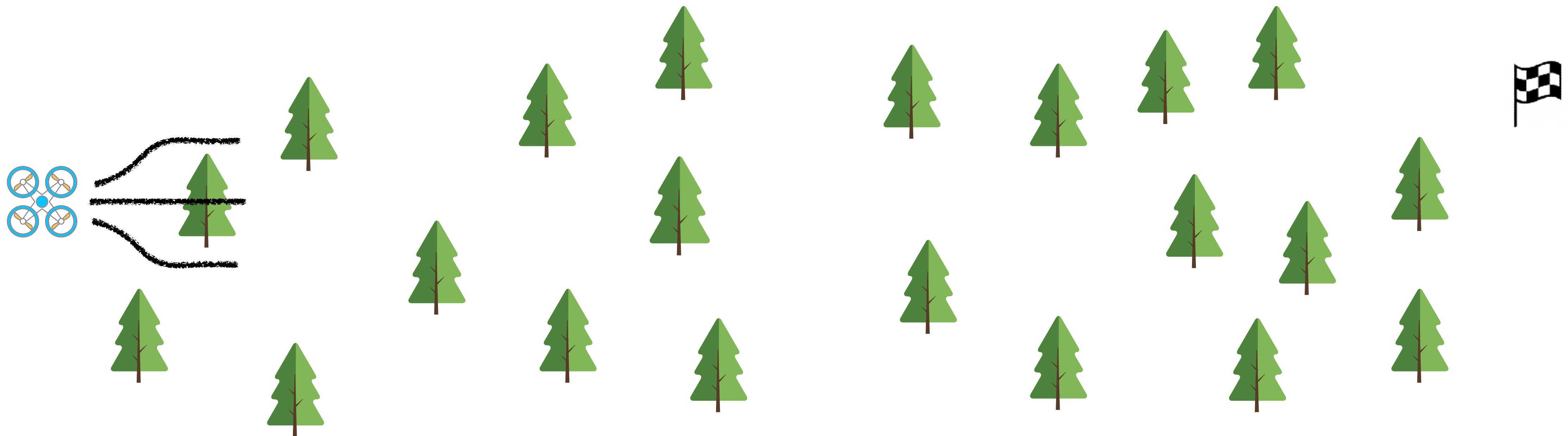
Single shot decision making

# What makes decision making hard?



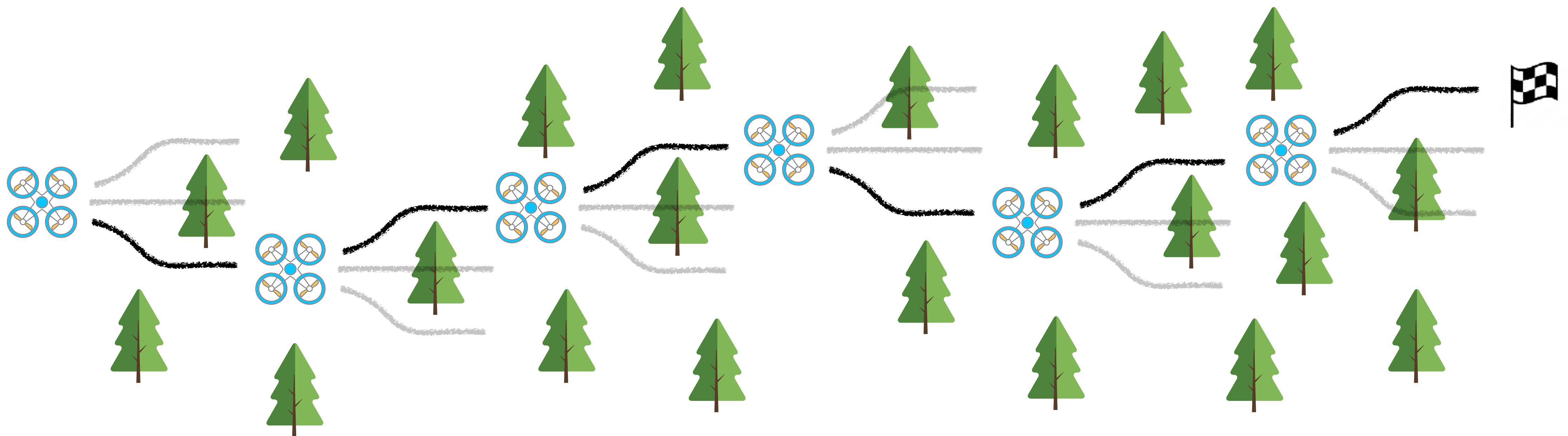
Single shot decision making

# What makes decision making hard?



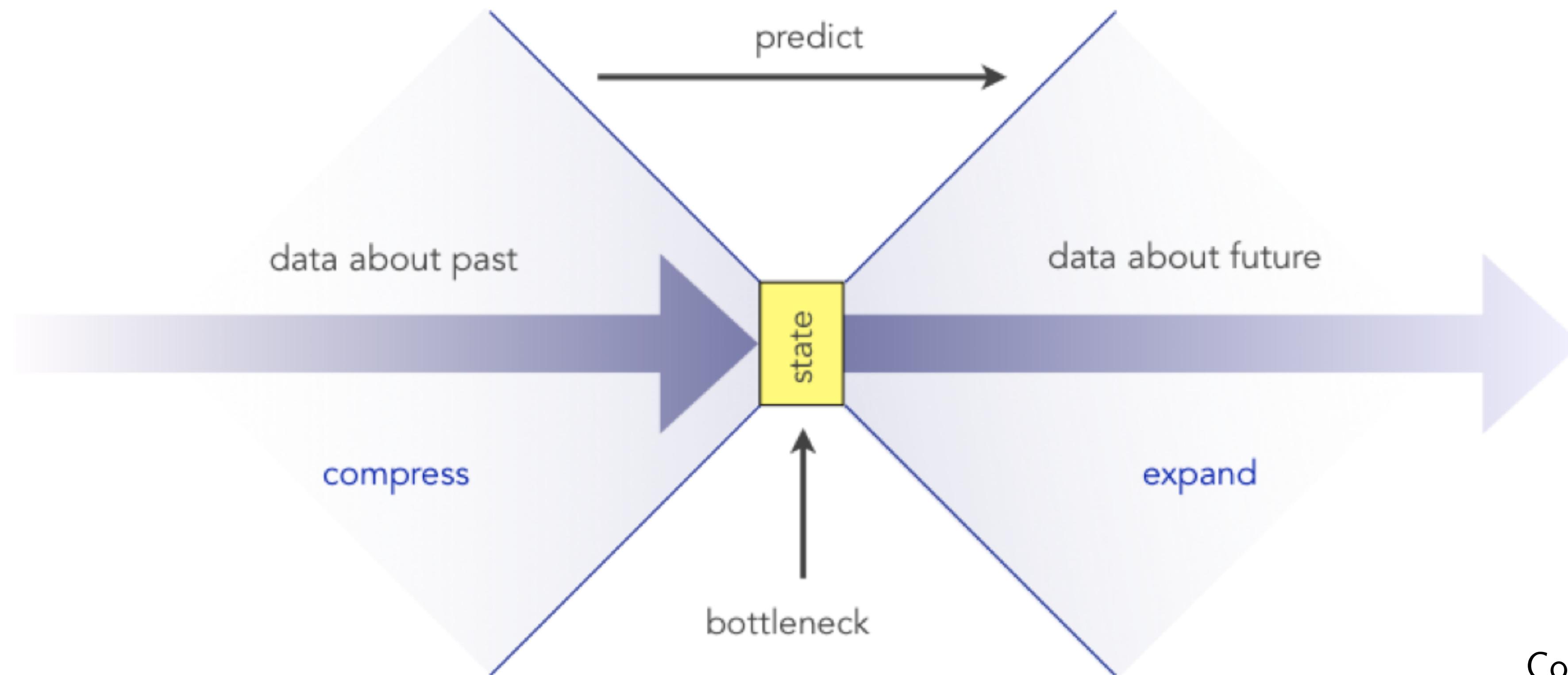
Sequential decision making

# What makes decision making hard?



How do we **tractably** reason over a sequence of decisions?

# Markov to the rescue!



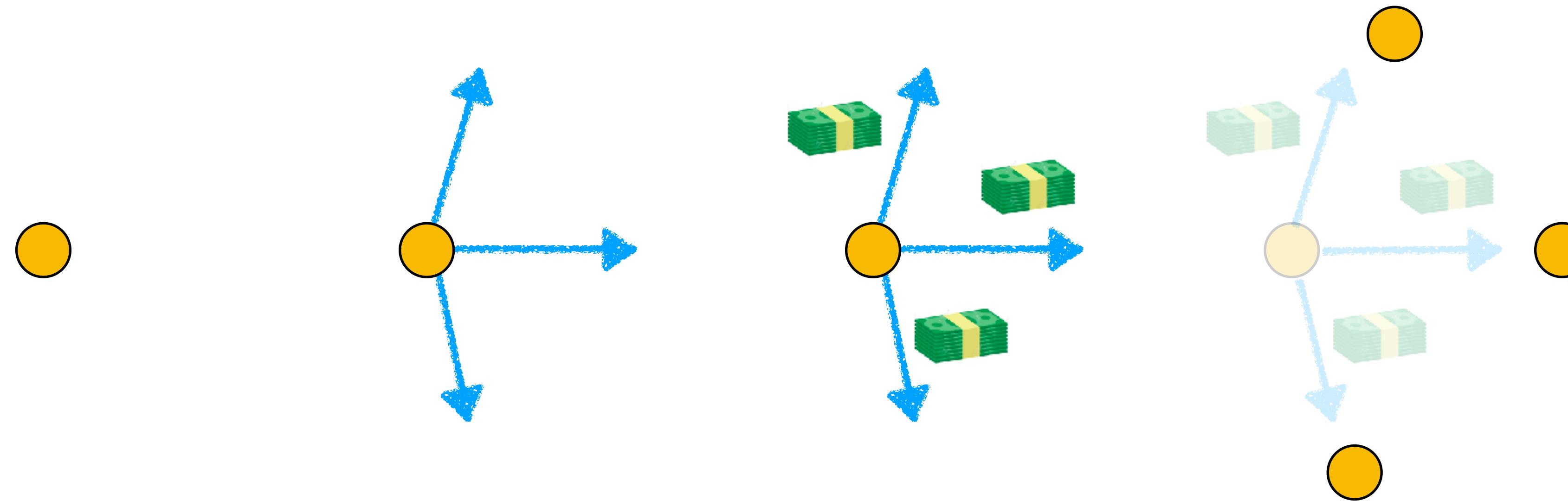
Courtesy: Byron Boots

State: statistic of history sufficient to predict the future

# Markov Decision Process

*A mathematical framework for modeling sequential decision making*

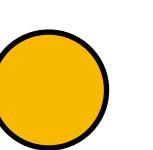
$\langle S, A, C, \mathcal{T} \rangle$



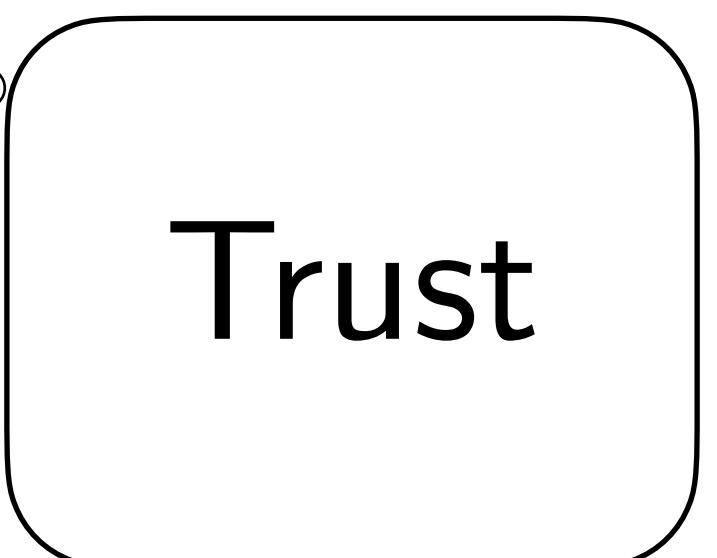
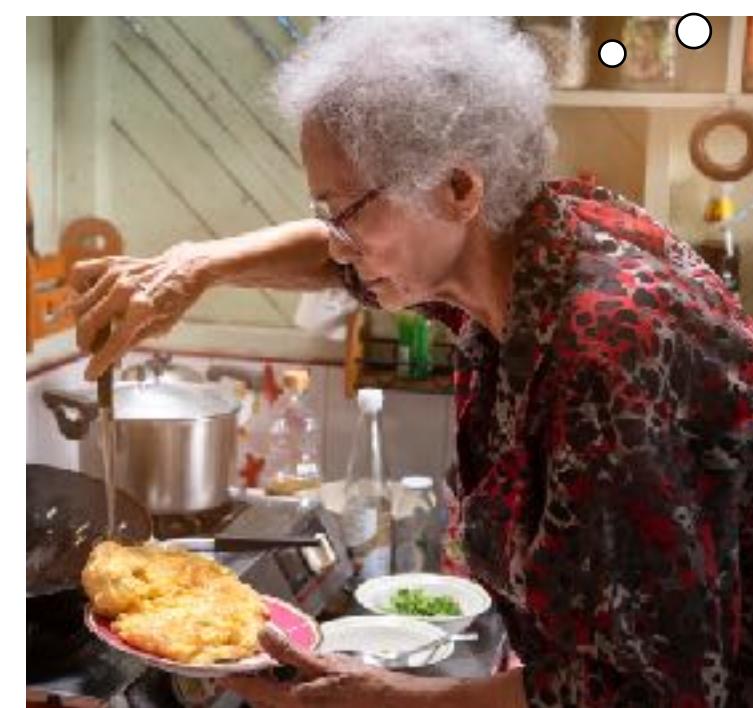
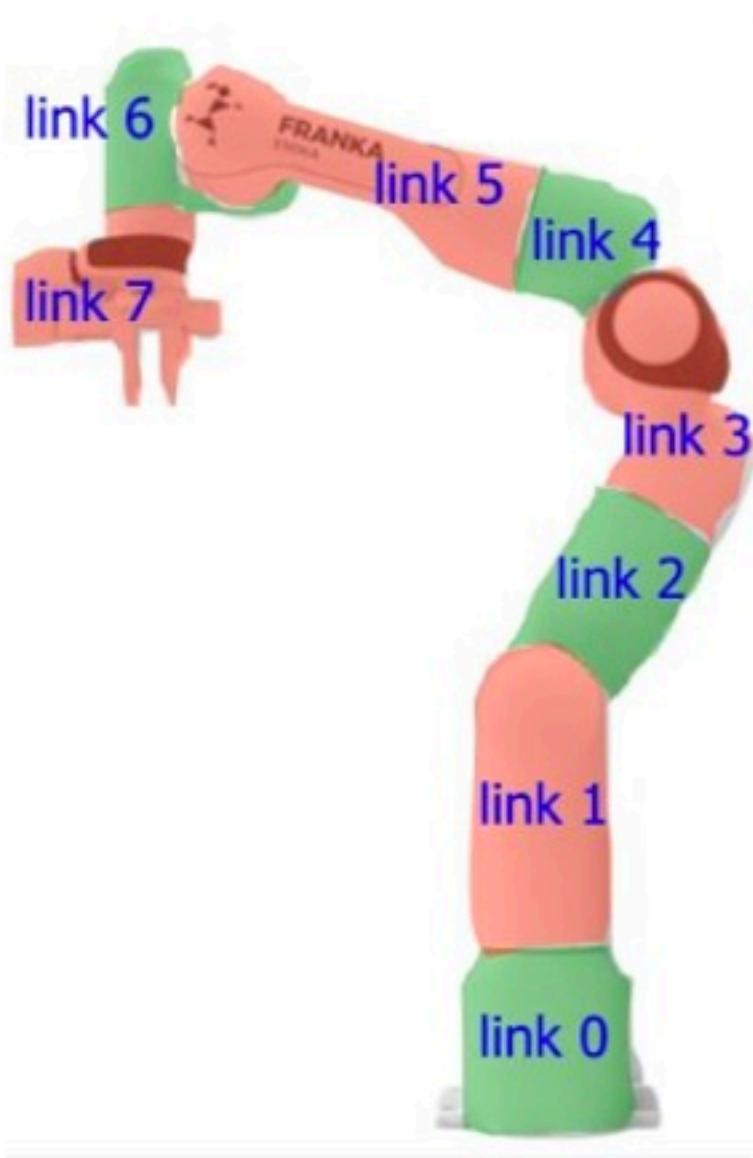
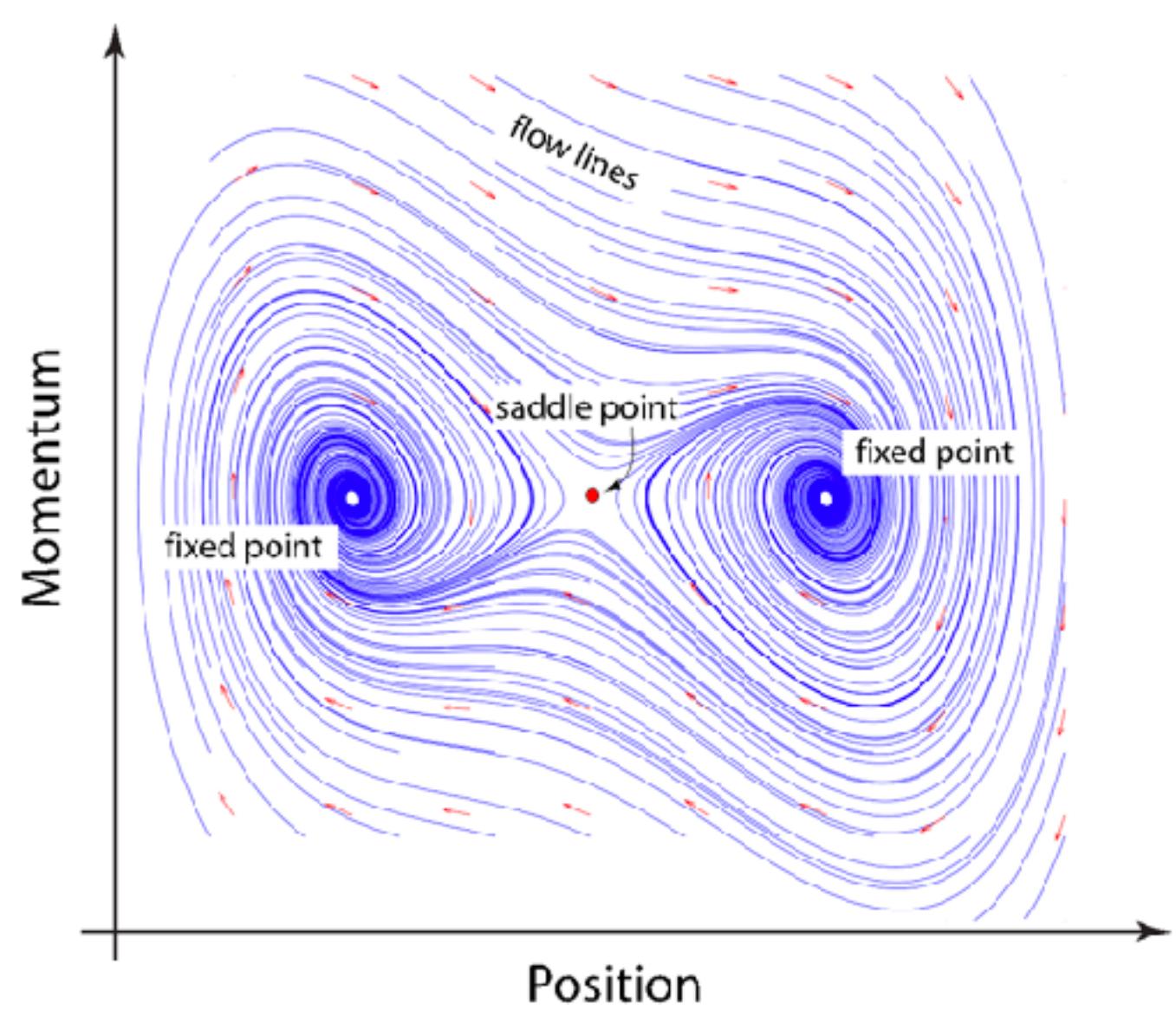
# State

$< S, A, C, \mathcal{T} >$

*Sufficient statistic of the system  
to predict future disregarding  
the past*



$s \in S$



# Activity!

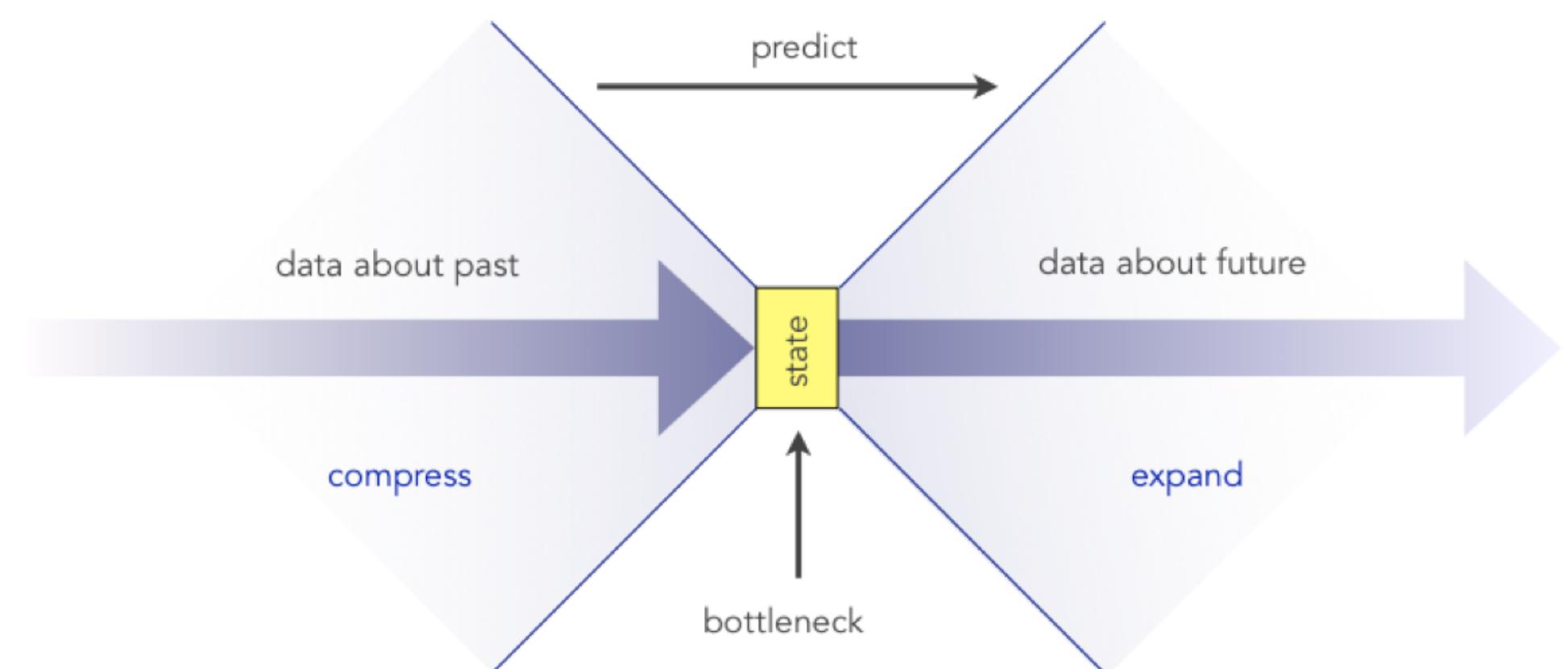


# Brainstorming

Think (30 sec): Example of MDPs with **shallow** state?  
(Current observation good enough)  
Example of MDPs with **deep** state?

Pair: Find a partner

Share (45 sec): Partners exchange ideas

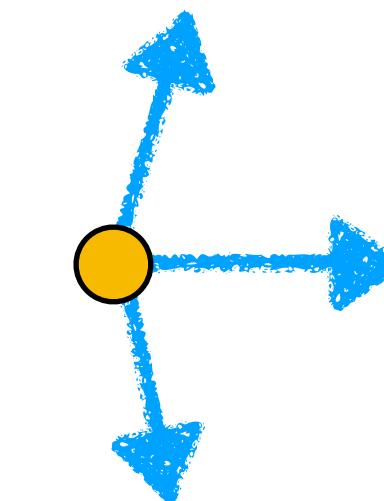


State: statistic of history sufficient to predict the future

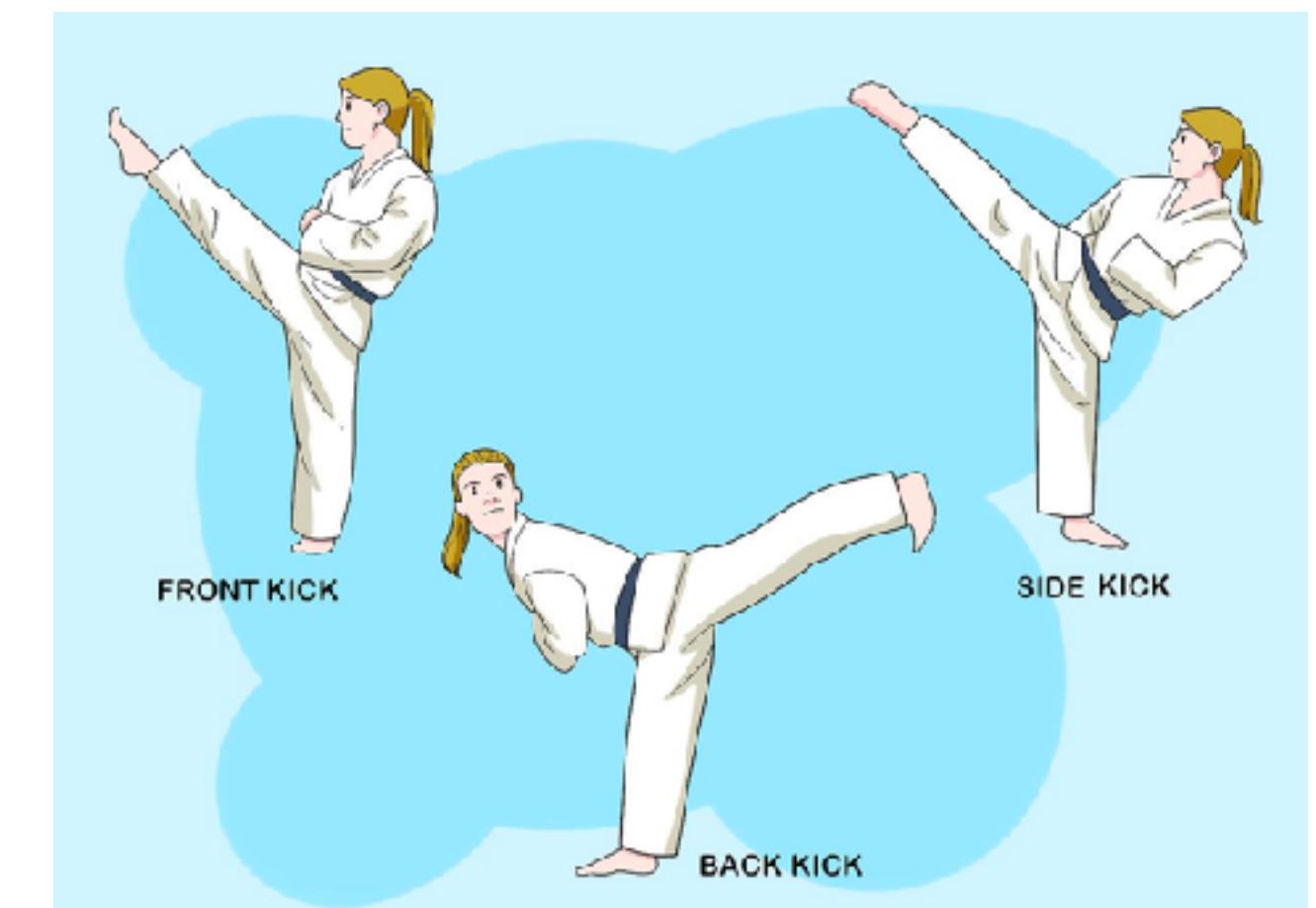
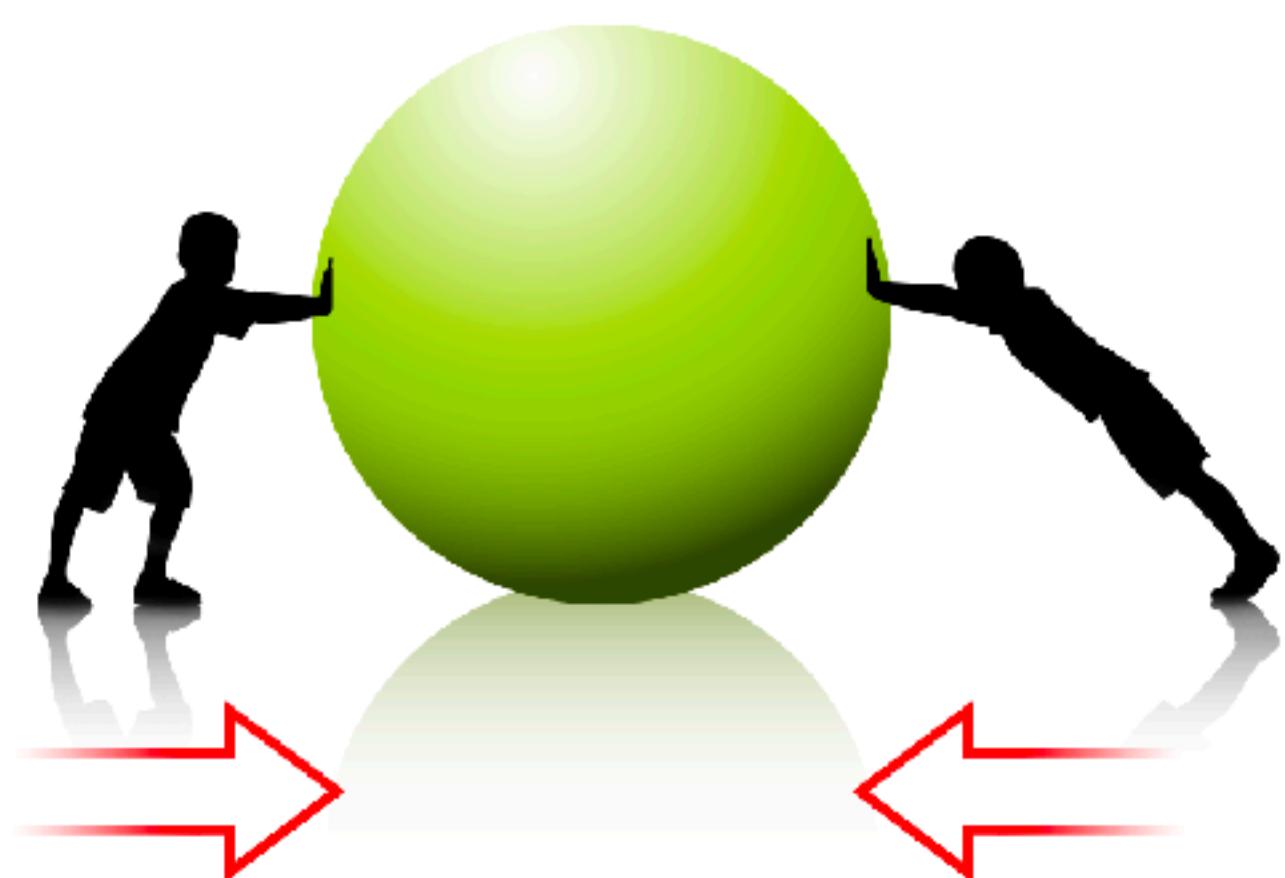
# Action

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

*Doing something:  
Control action / decisions*



$a \in A$

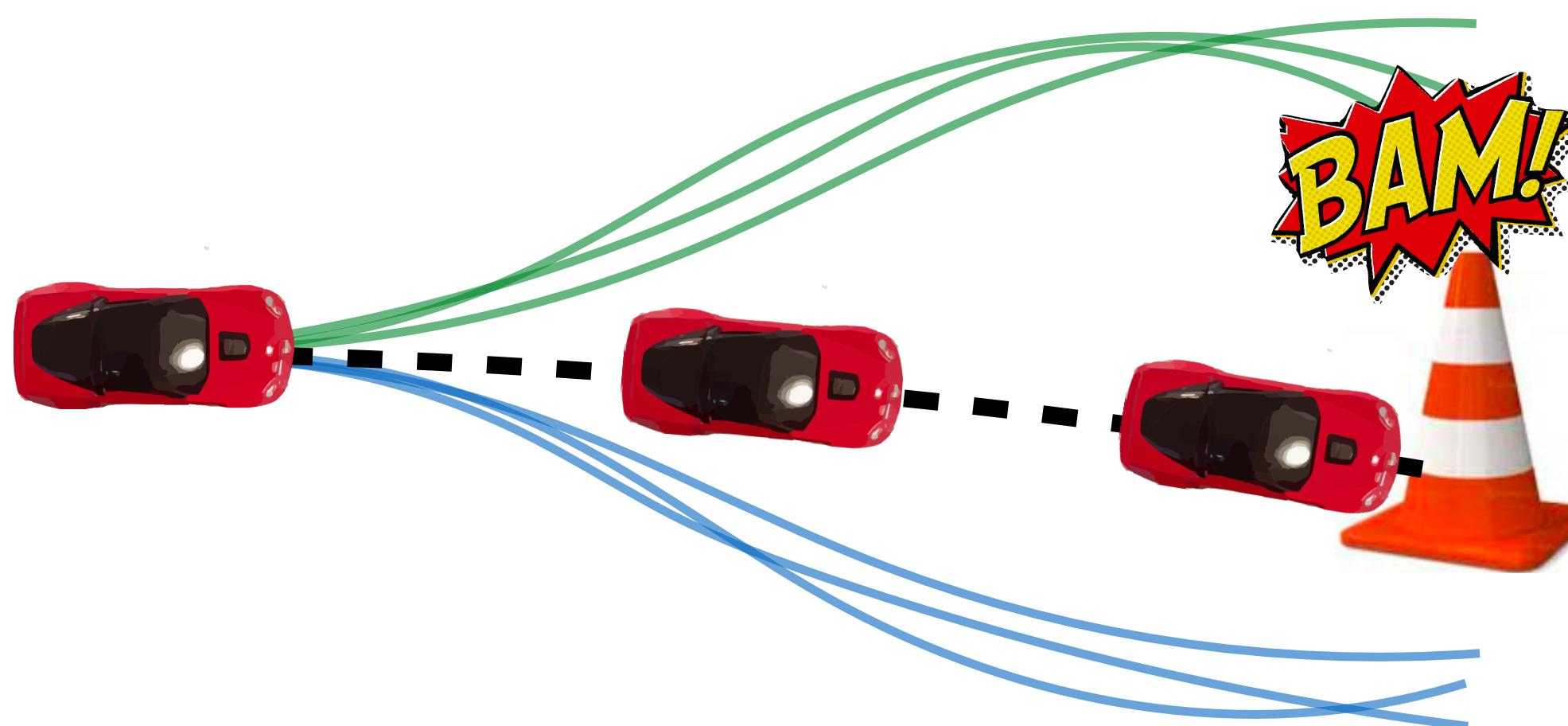
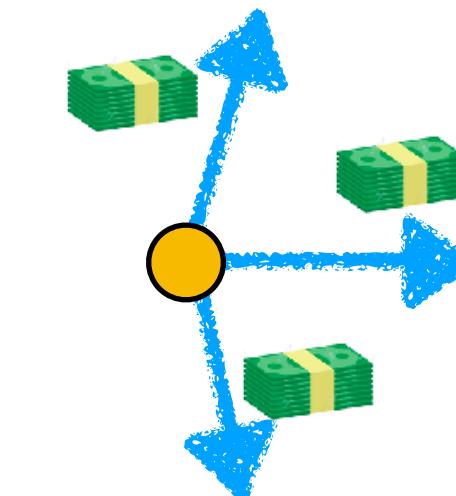


# Cost

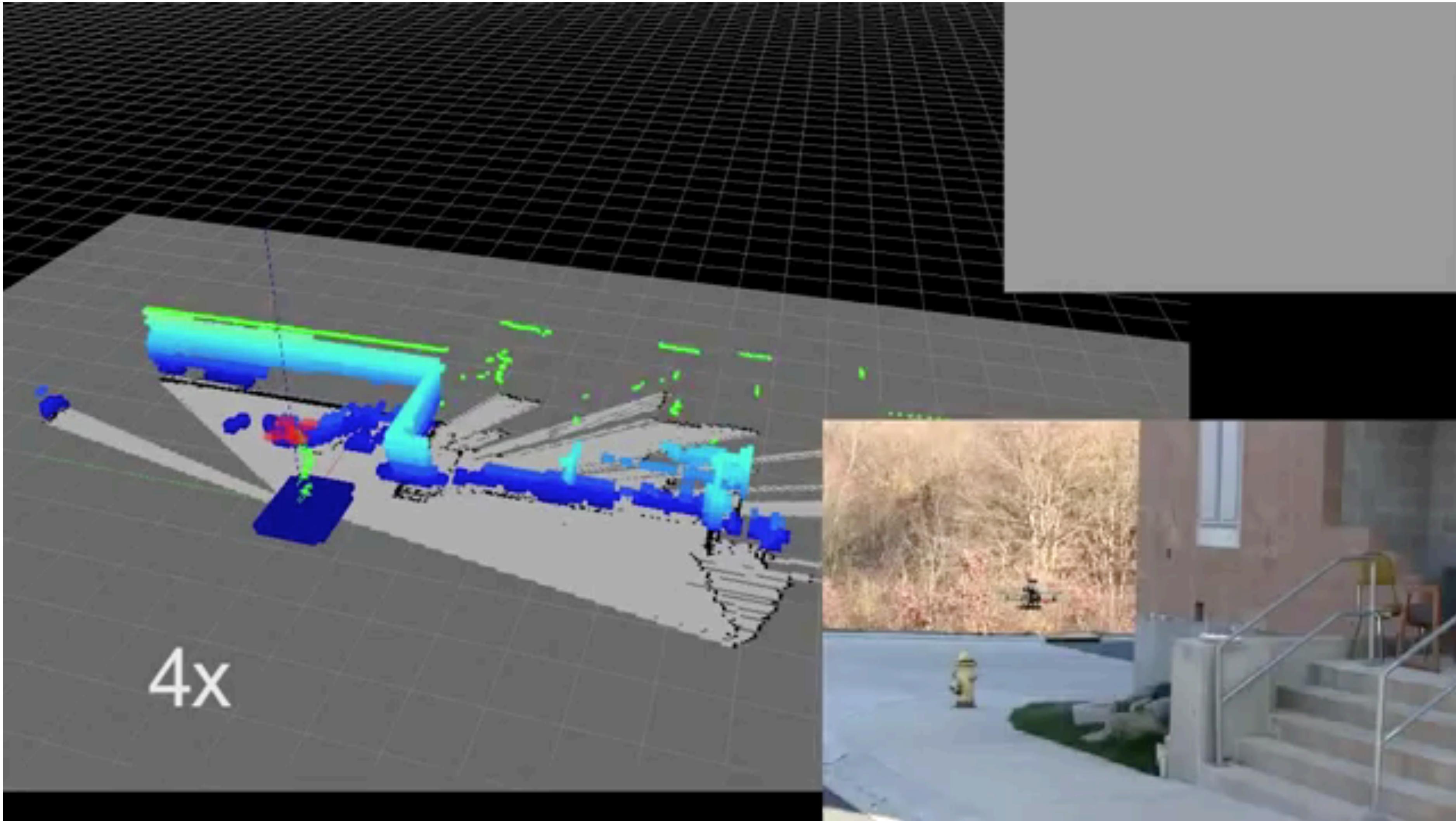
$\langle S, A, C, \mathcal{T} \rangle$

*The instantaneous cost of taking an action in a state*

$$c(s, a)$$



# Examples of *non-Markovian* cost?



# Transition

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

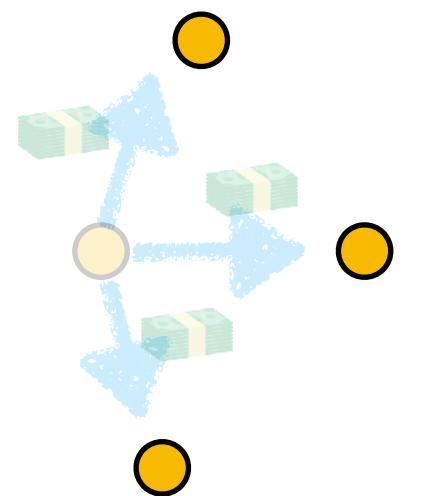
*The next state given state and action*

$$s' = \mathcal{T}(s, a)$$

Deterministic

$$s' \sim \mathcal{T}(s, a)$$

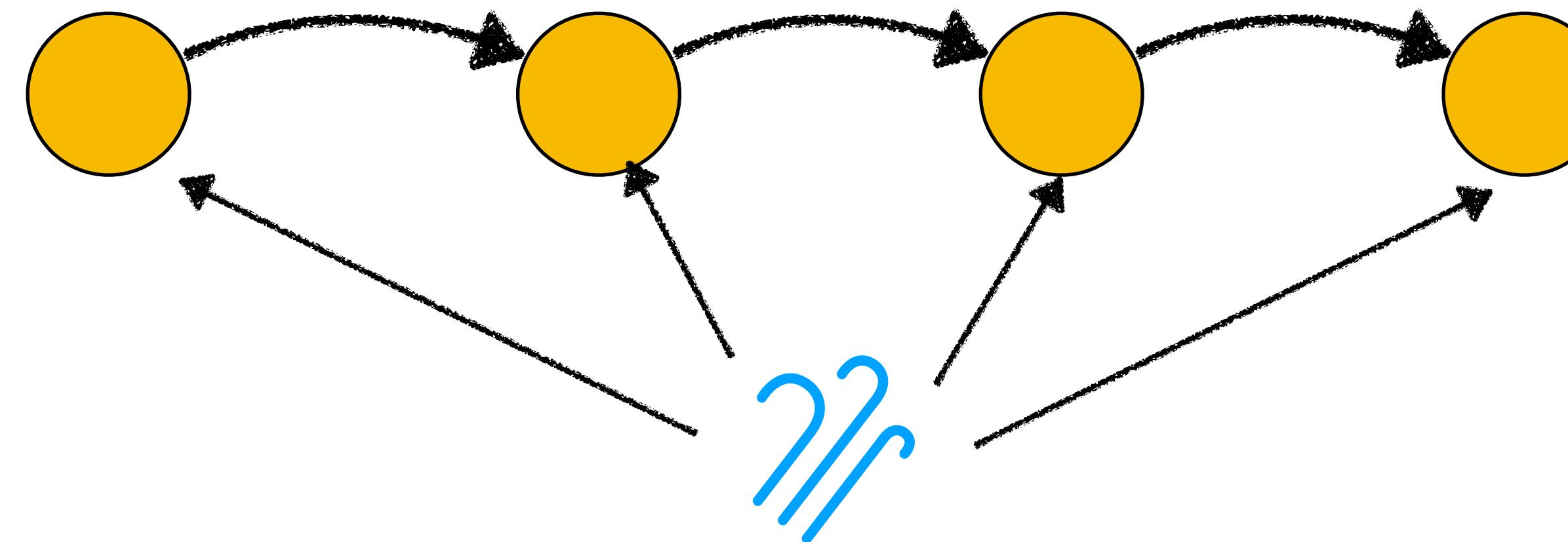
Stochastic



# Examples of *non-Markovian* dynamics?



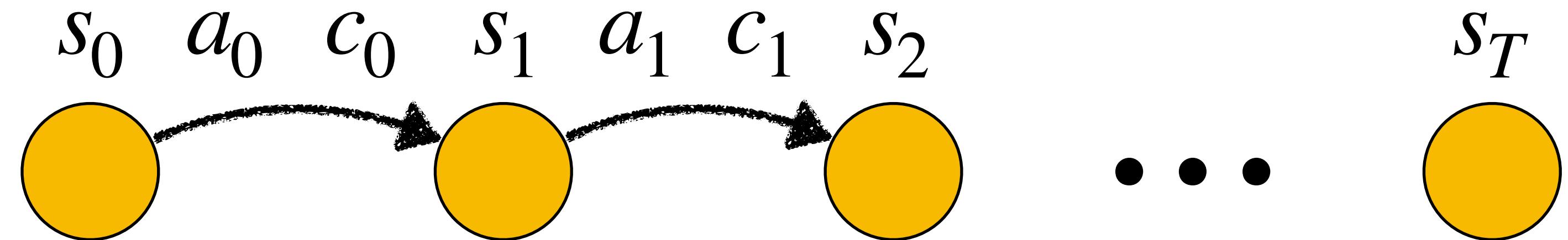
Wind correlates disturbance across time



# Markov Decision Process → Problem

Includes things to define an optimization problem

Horizon  $T \in \mathbb{N}$



Discount  $0 \leq \gamma \leq 1$

Return:  $c_0 + \gamma c_1 + \dots + \gamma^{T-1} c_{T-1}$   
(Costs are more valuable if they happen soon)

# Markov Decision Process → Problem

## Policy

$$\pi \in \Pi$$

$$\pi : s_t \rightarrow a_t \quad (\text{Deterministic})$$

$$\pi : s_t \rightarrow P(a_t) \quad (\text{Stochastic})$$

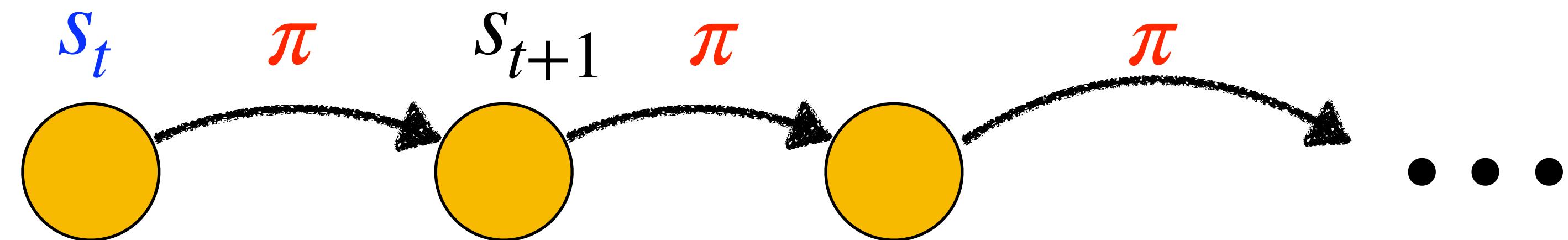
*A function that maps state (and time) to action*

## Objective Function

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

*Find policy that minimizes sum of discounted future costs*

# Value of a state

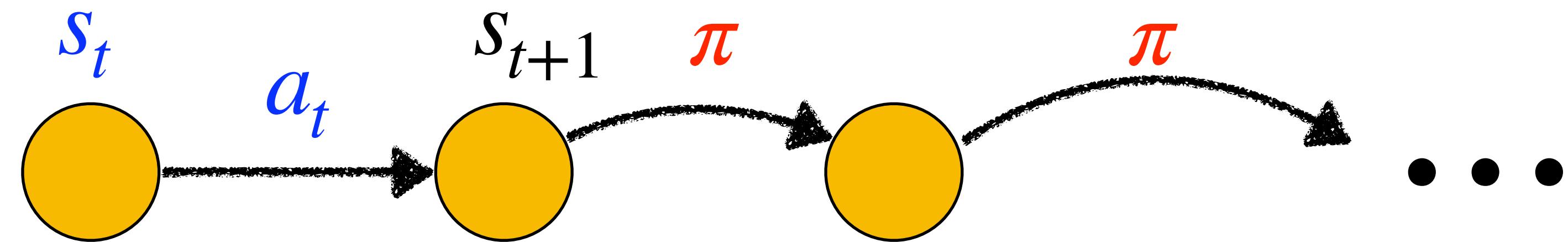


$$V^{\pi}(s_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} +$$

*Expected discounted sum of cost from  
starting at a state  
and following a policy from then on*

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

# Value of a state-action



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

*Expected discounted sum of cost from starting at a state, executing action and following a policy from then on*

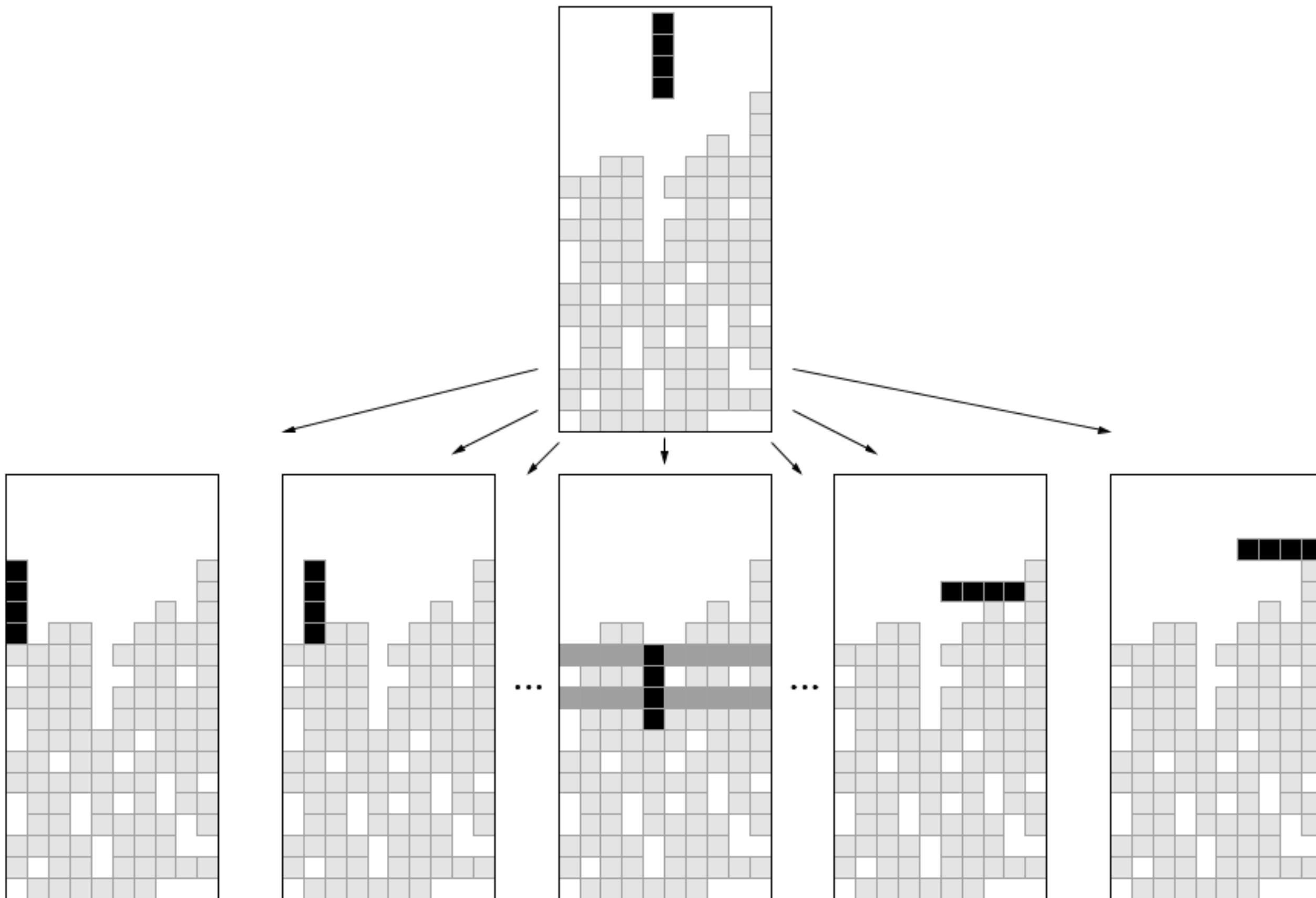
$$Q^{\pi}(s_t, a_t) = c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)} V^{\pi}(s_{t+1})$$



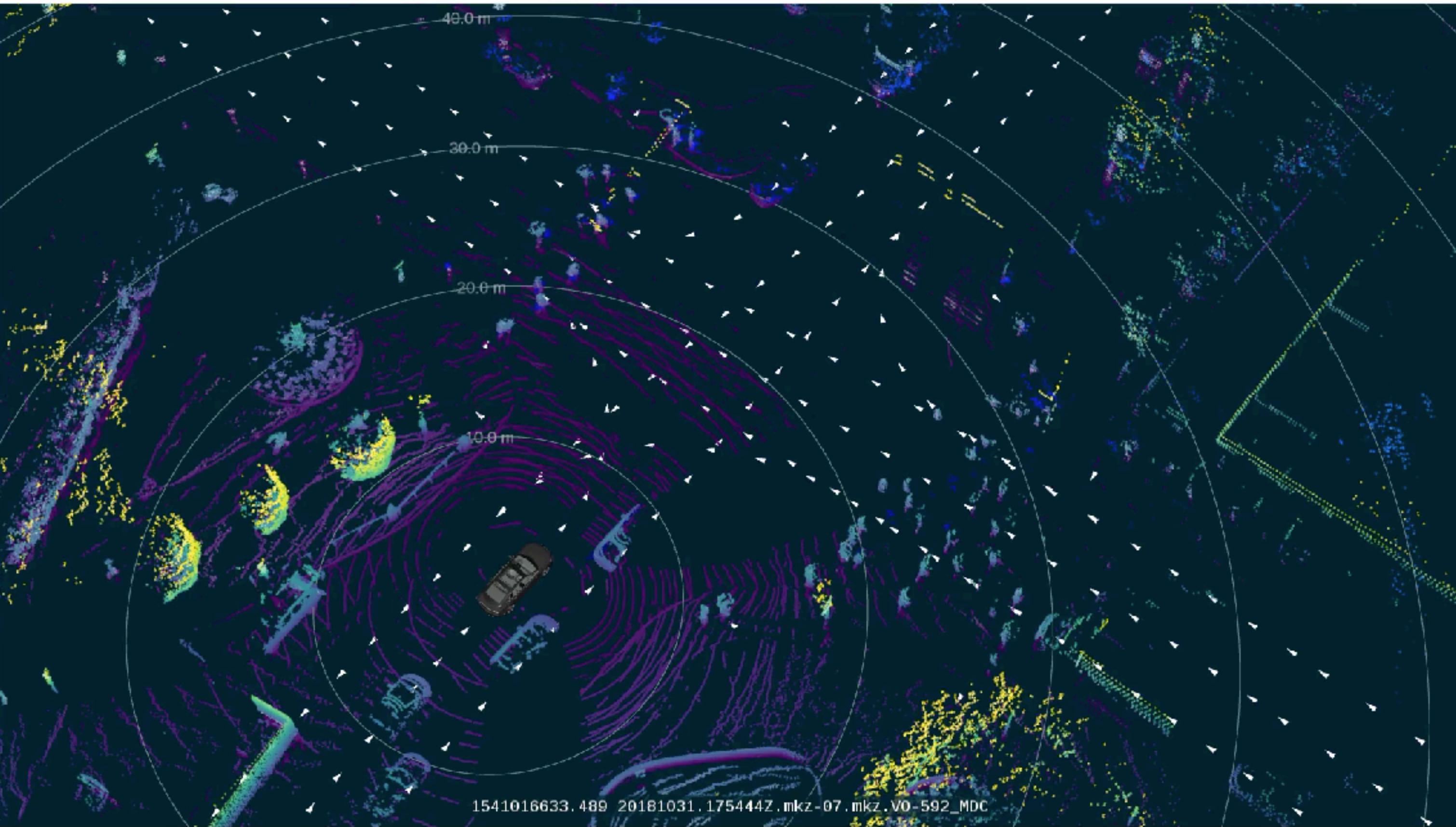
Values matter

# Example 1: Tetris!

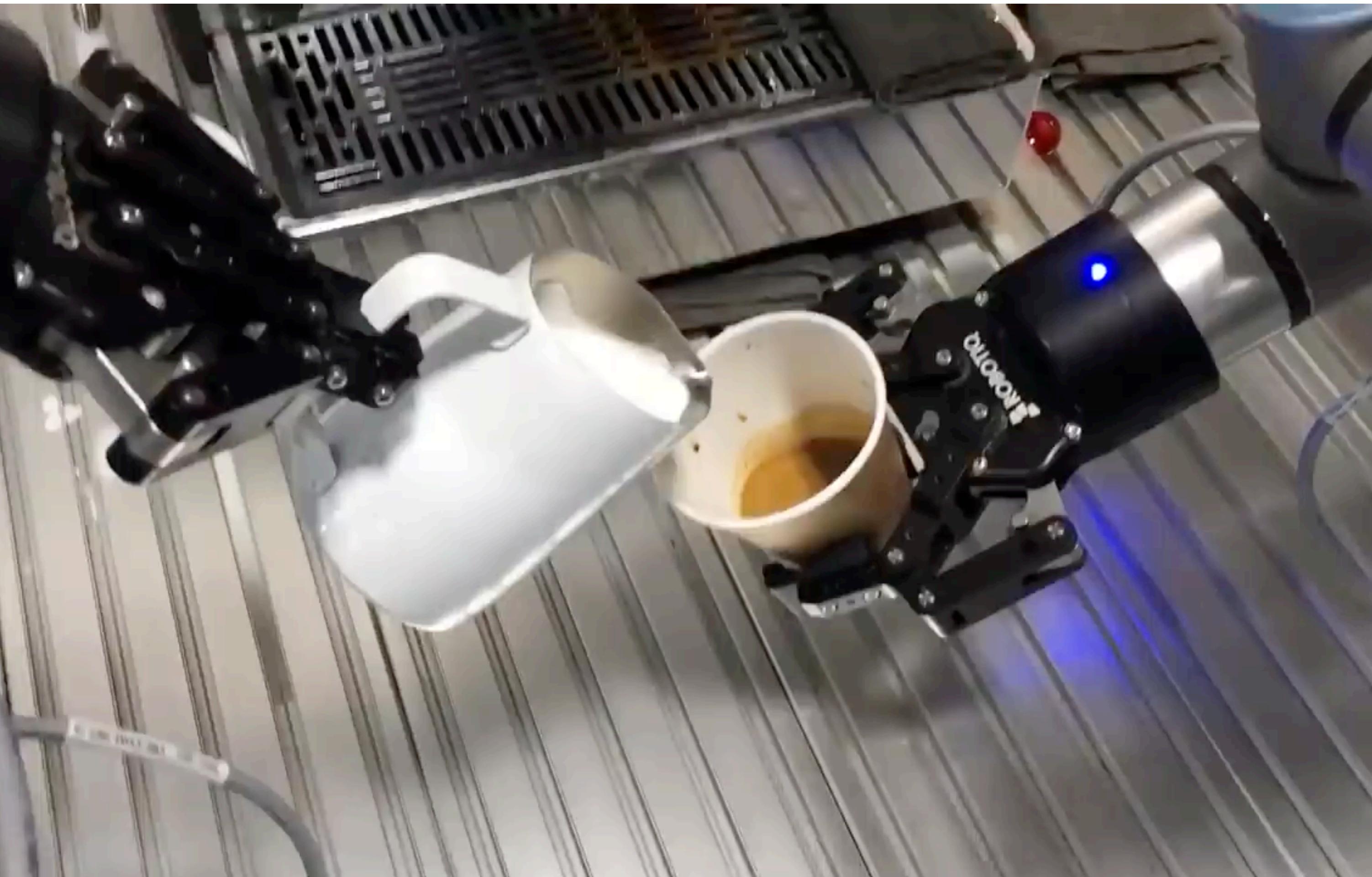
$\langle S, A, C, \mathcal{T} \rangle$



# Example 2: Self-driving


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


# Example 3: Coffee making robot


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

?

# Solving MDPs

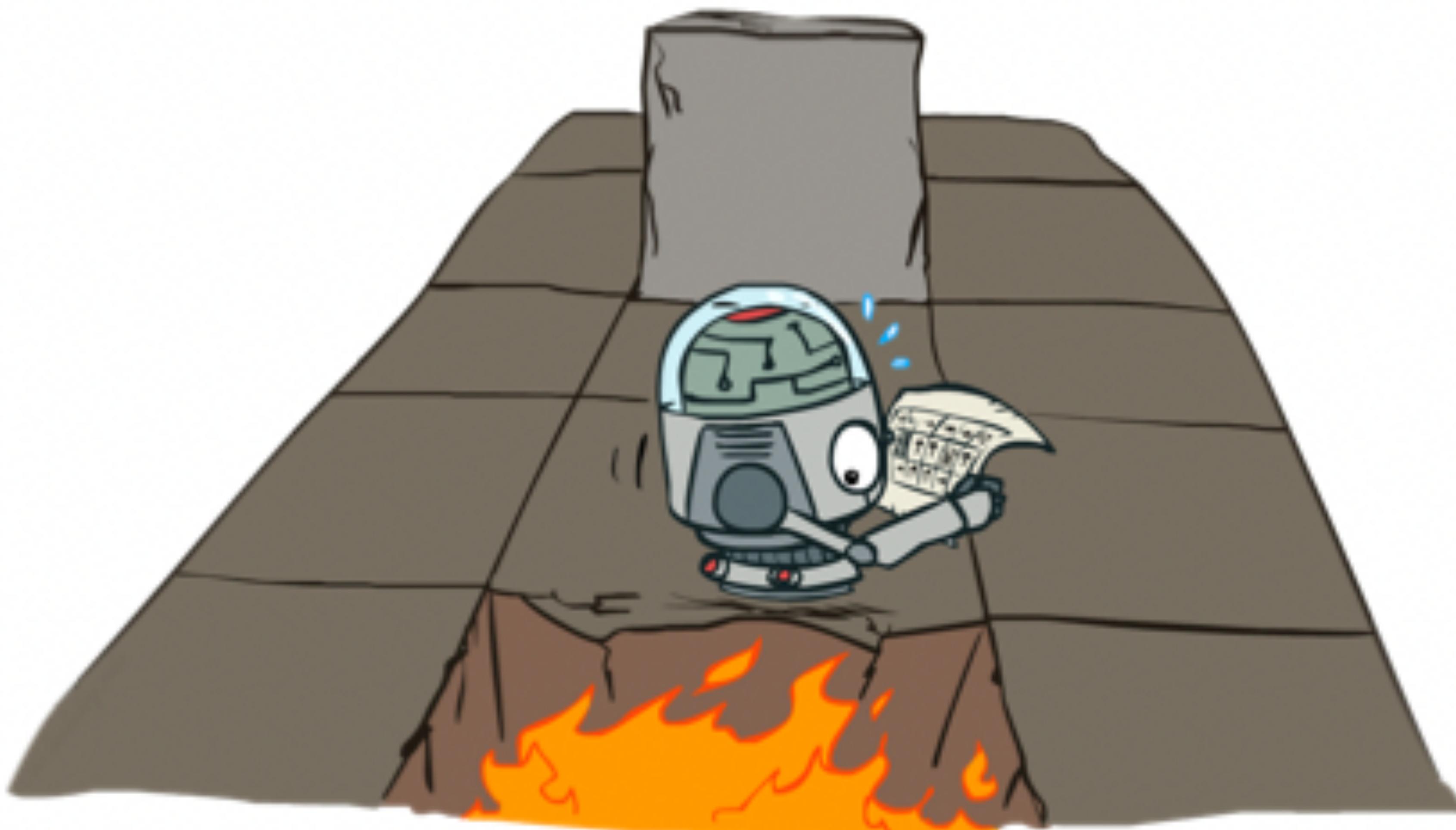
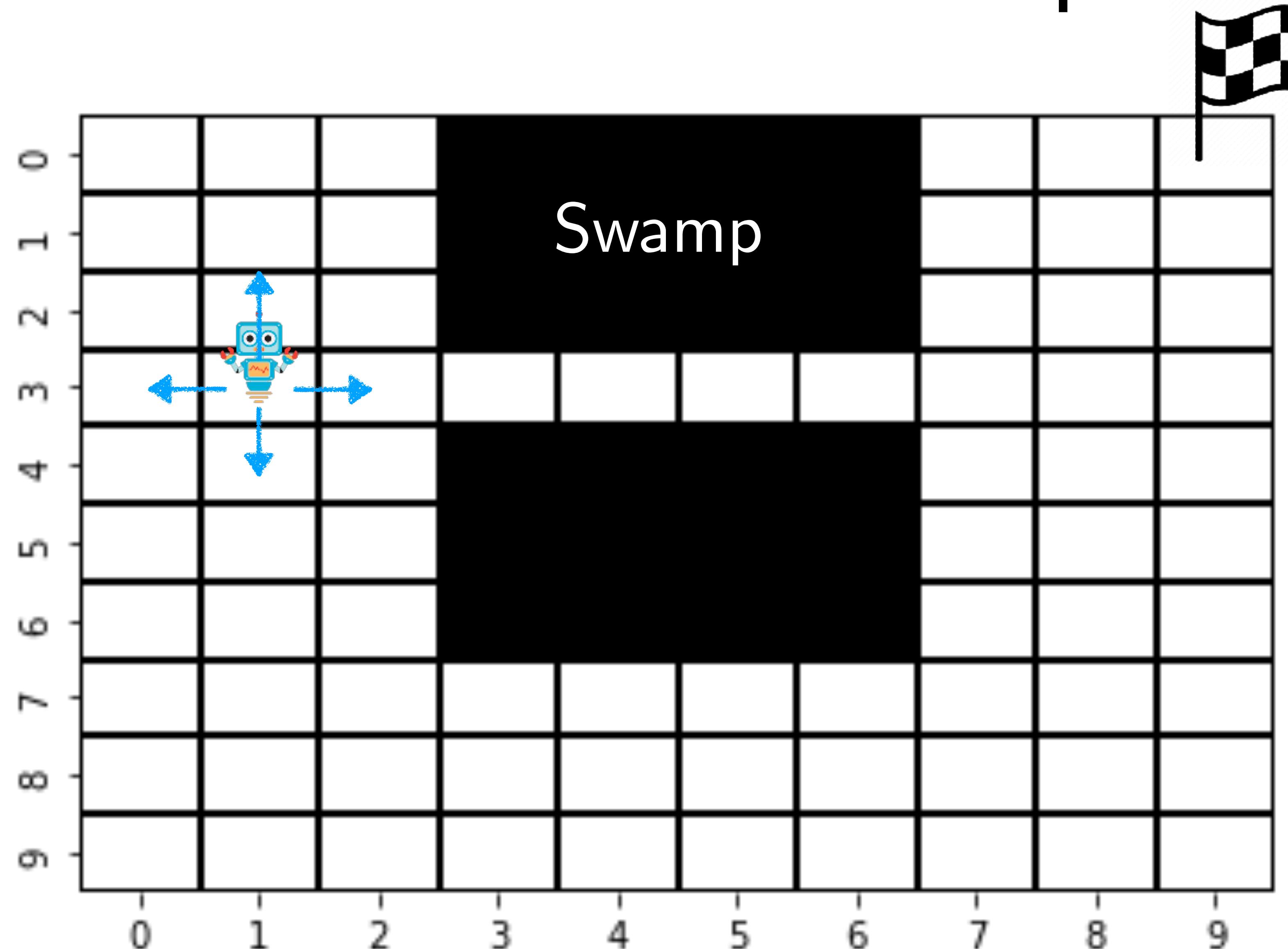


Image courtesy Dan Klein

# Setup



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

- Two absorbing states:  
Goal and Swamp
- Cost of each state is 1  
till you reach the goal
- Let's set  $T = 30$

# 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

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

---

**Algorithm 4:** Dynamic Programming Value Iteration for computing the optimal value function.

---

```

Algorithm OptimalValue( $x, T$ )
  for  $t = T - 1, \dots, 0$  do
    for  $x \in \mathbb{X}$  do
      if  $t = T - 1$  then
         $| V(x, t) = \min_a c(x, a)$ 
      end
      else
         $| V(x, t) = \min_a c(x, a) + \sum_{x' \in \mathbb{X}} p(x'|x, a)V(x, t + 1)$ 
      end
    end
  end

```

---

*What is  
the complexity?*

$$S \times A \times T$$

Deterministic

$$S^2 \times A \times T$$

Stochastic

$$k \times S \times A \times T$$

Efficient

# Why is the optimal policy a function of time?



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

# To infinity!



# Infinite horizon cases

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



Fixed point as  $t \rightarrow \infty$

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

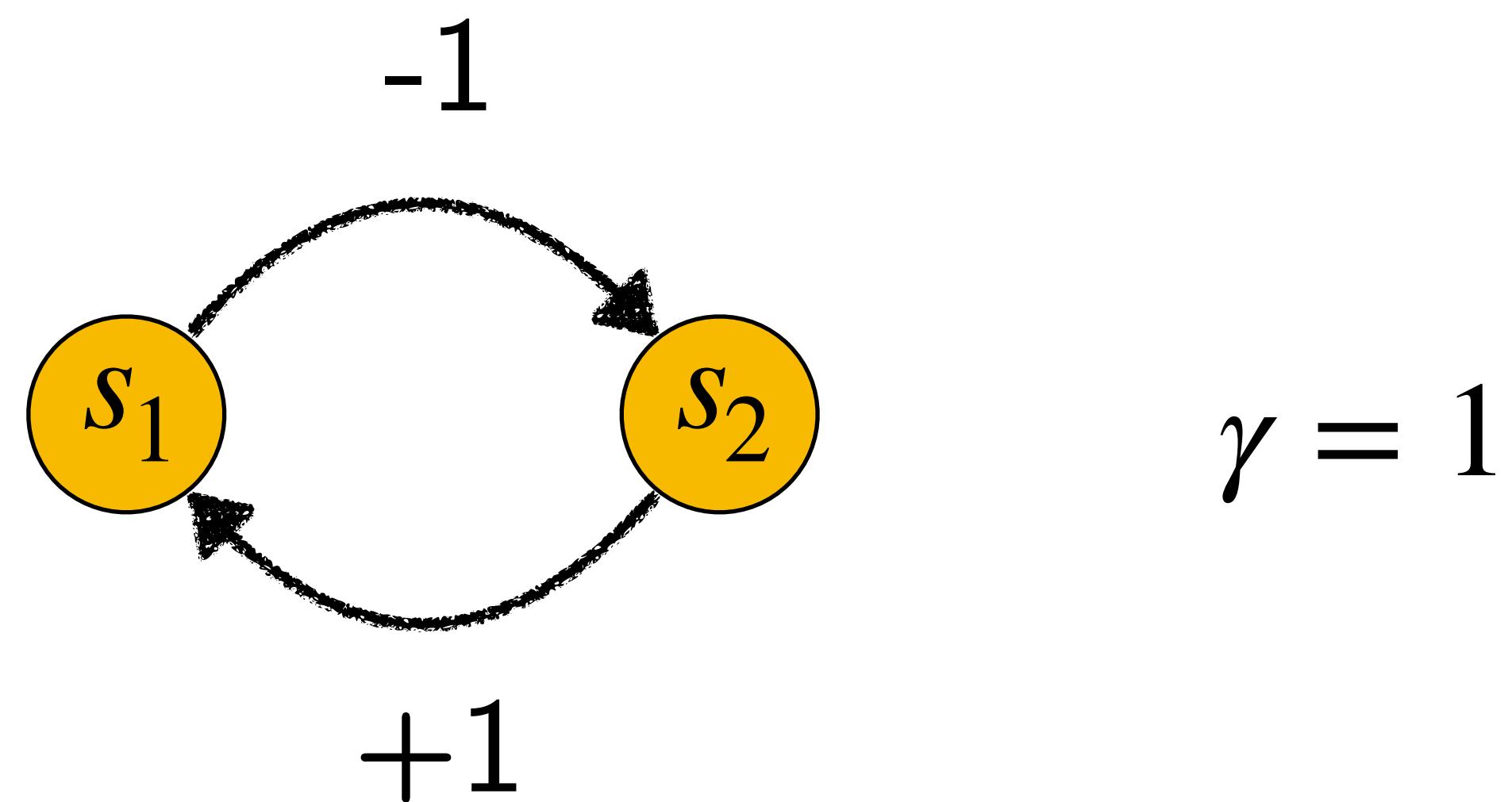
# Bellman Equation

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

Does this converge?

How fast does it converge?

# Does value iteration converge?



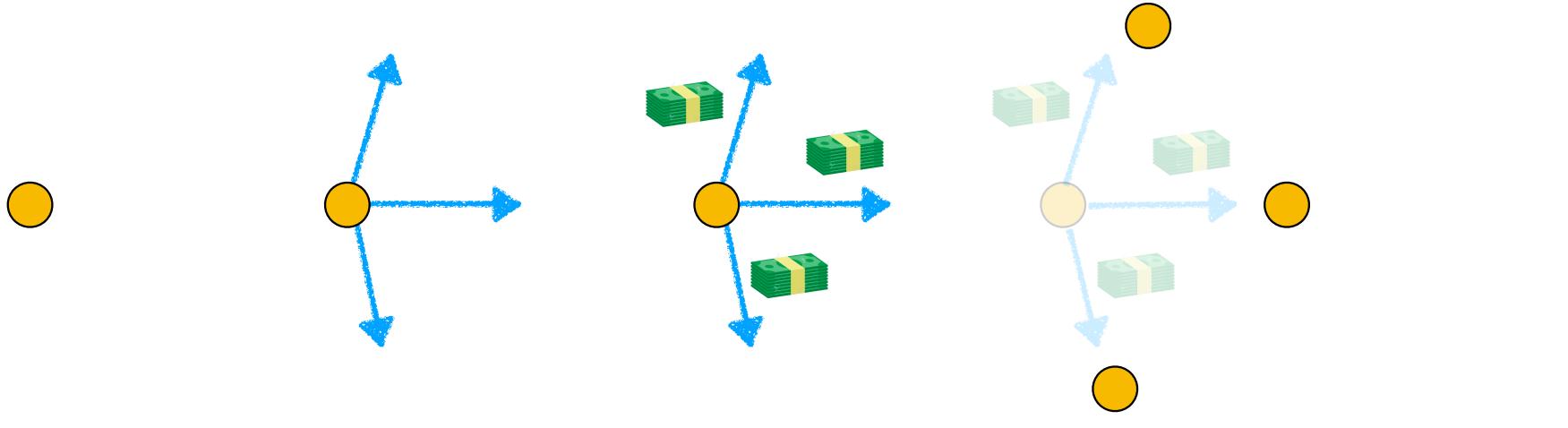
What is  $V^*(s_1)$ ? What is  $V^*(s_2)$ ?

# tl;dr

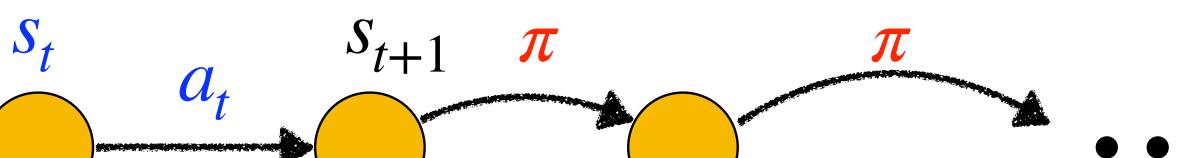
## Markov Decision Process

*A mathematical framework for modeling sequential decision making*

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



### Value of a state-action

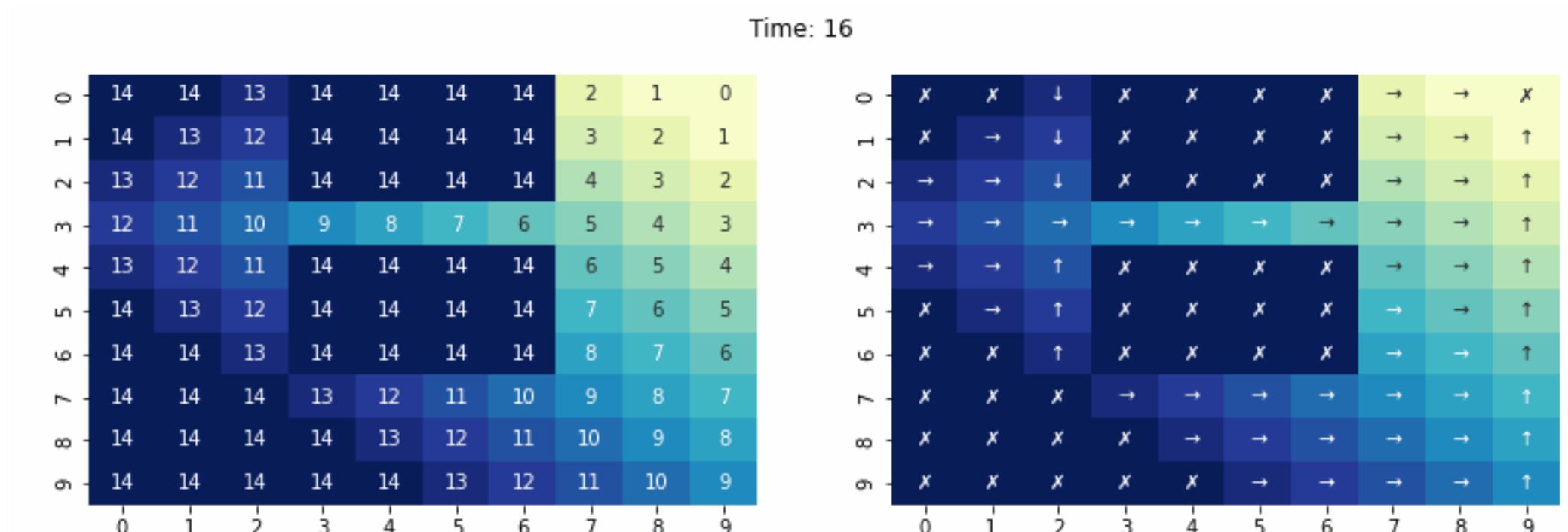


$$Q^\pi(s_t, a_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} + \dots$$

*Expected discounted sum of cost from starting at a state, executing action and following a policy from then on*

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

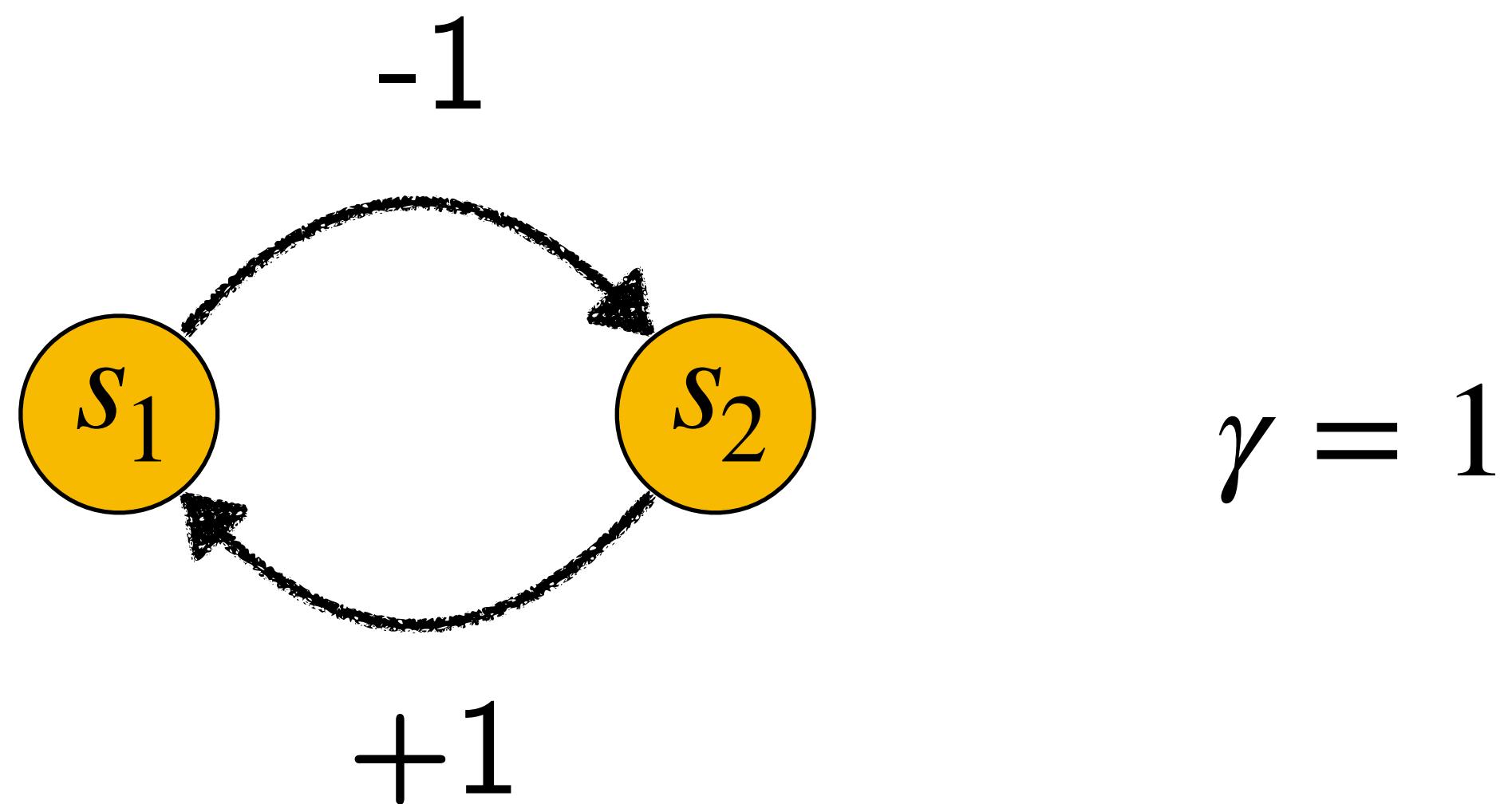
### Dynamic Programming all the way!



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

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

# Does value iteration converge?



What is  $V^*(s_1)$ ? What is  $V^*(s_2)$ ?

# What is the effect of discount factor?

Gamma: 0.0

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	x
1	x	x	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

# Activity!



# Think-Pair-Share

Think (30 sec): What are some attributes of a **hard** MDP?

Pair: Find a partner

Share (45 sec): Partners exchange  
ideas

# Policy Iteration

# How frequently does the best action change?

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

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 $\pi$

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

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

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