Robots as Markov Decision Problems

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Announcement!

Assignment 0

https://github.com/sanjibanc/cs4756-robot-learning-fa24/tree/main/assignments/A0

Link in website!

Checks familiarity with PyTorch!

Due Thursday 9/5!

Course Policies

All policies are posted on the Website!

Course Website: 3 TOTAL late days. Any assignment turned in late will incur a reduction in score by 33% for each late day

Academic Integrity: Any work presented as your own must be your own, with no exceptions tolerated. Submitting work created by ChatGPT, or copied from a bot or a website, as your own work violates academic integrity.

Generative AI

The work you do consists of writing code and natural language descriptions.

To some extent, the new crop of "generative AI" (GAI) tools can do both of these things for you.

However, **we require that the vast majority of the intellectual work must be originated by you,** not by GAI. You may use GAI to look up helper functions, or to proofread your text, but clearly document how you used it.

Generative AI

In this class, for every assignment and final project, you can choose between two options:

Option 1: Avoid all GAI tools. Disable GitHub Copilot in your editor, do not ask chatbots any questions related to the assignment, etc. If you choose this option, you have nothing more to do.

Option 2: Use GAI tools with caution and include a one-paragraph description of everything you used them for along with your writeup. This paragraph must:

1. Link to exactly which tools you used and describe how you used each of them, for which parts of the work.
2. Give at least one concrete example (e.g., generated code or Q&A output) that you think is particularly illustrative of the "help" you got from the tool.
3. Describe any times when the tool was unhelpful, especially if it was wrong in a particularly hilarious way.
4. Conclude with your current opinion about the strengths and weaknesses of the tools you used for real-world compiler implementation.

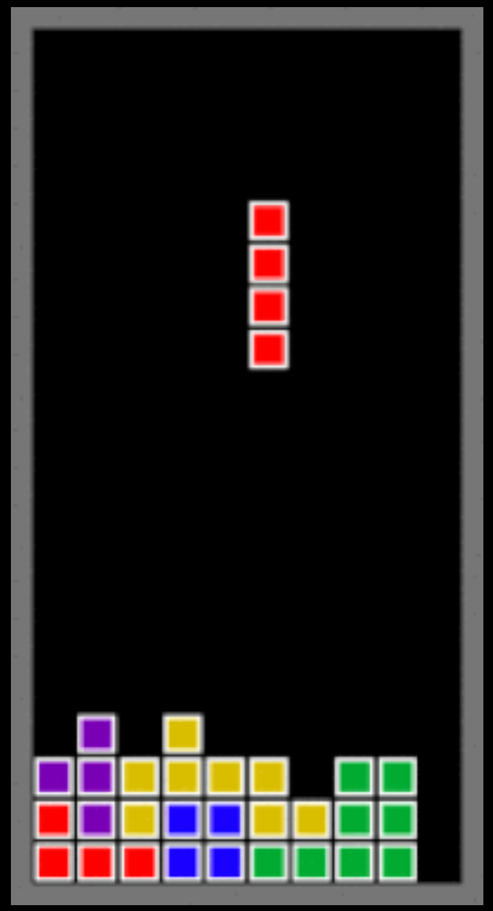
Remember that you can pick whether to use GAI tools for every assignment, so using them on one set of tasks doesn't mean you have to keep using them forever.

Let’s get started

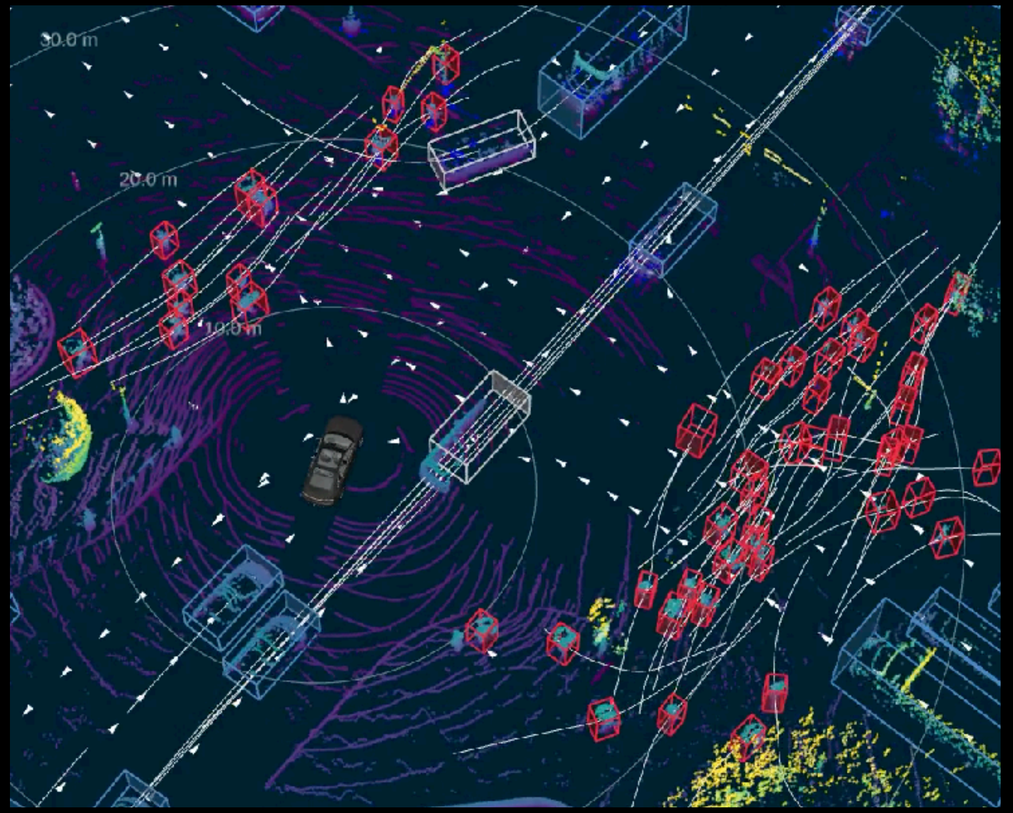
Today's class

- What makes sequential decision making hard?
- What is a Markov Decision Process (MDP)?
- Identifying MDPs in the wild
- What does it mean to solve a MDP?

Sequential Decision Making



Tetris

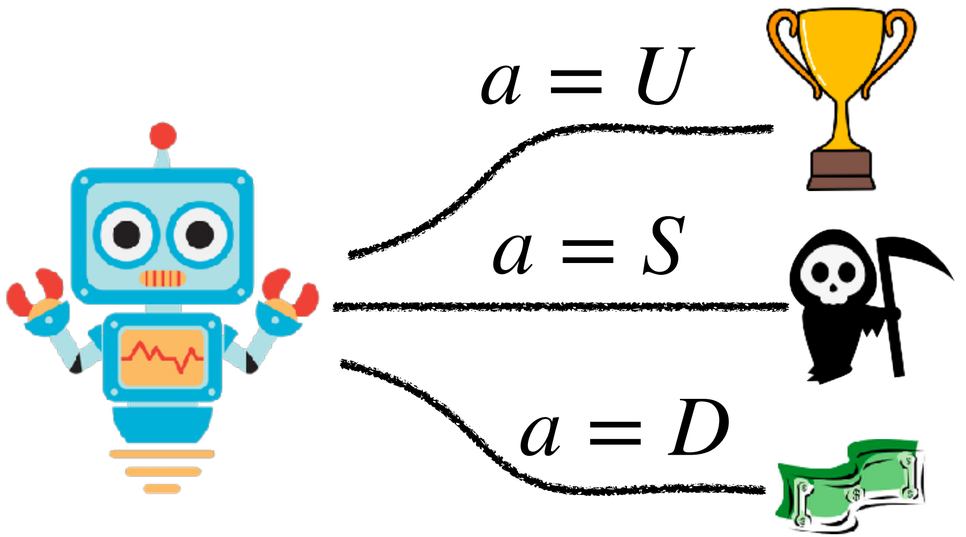


Self-driving

Robot Baristas

What makes *sequential* decision making hard?

An Easy Example: *Non*-sequential decision making

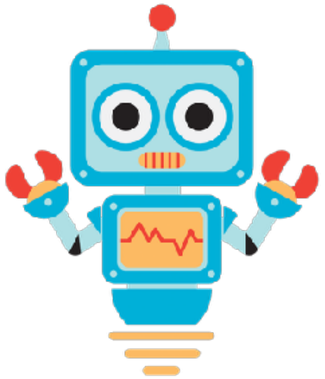


	Reward
$a = U$	100
$a = S$	-1000
$a = D$	1

Goal: Pick the action that maximizes reward $\arg\max_a R(a)$

What is the complexity of this optimization problem?

A Hard Example: *Sequential* decision making



A sequence of actions
for T time steps

$a_1, a_2, a_3, a_4, \ldots a_T$

Black Box

Reward

-100

Goal: Pick the sequence of actions that maximizes reward

$$\arg \max_{a_1, a_2, \ldots, a_T} R(a_1, a_2, \ldots, a_T)$$

What is the complexity of this optimization problem?

Today's class

- [x] What makes sequential decision making hard?
- [ ] What is a Markov Decision Process (MDP)?
- [ ] Identifying MDPs in the wild
- [ ] What does it mean to solve a MDP?

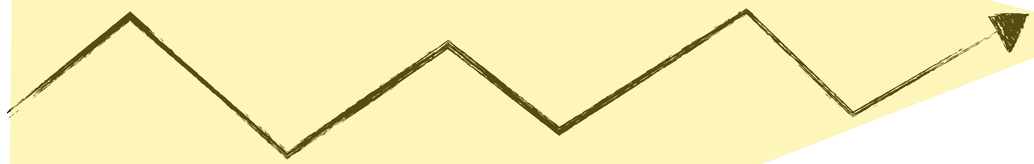
What assumption makes the optimization problem tractable?

The Markov Assumption

Summarize all past information into a compact state ...

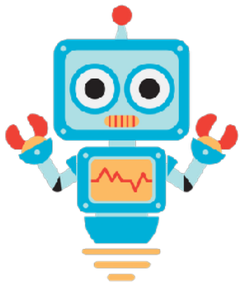
State

... that is sufficient to predict the future



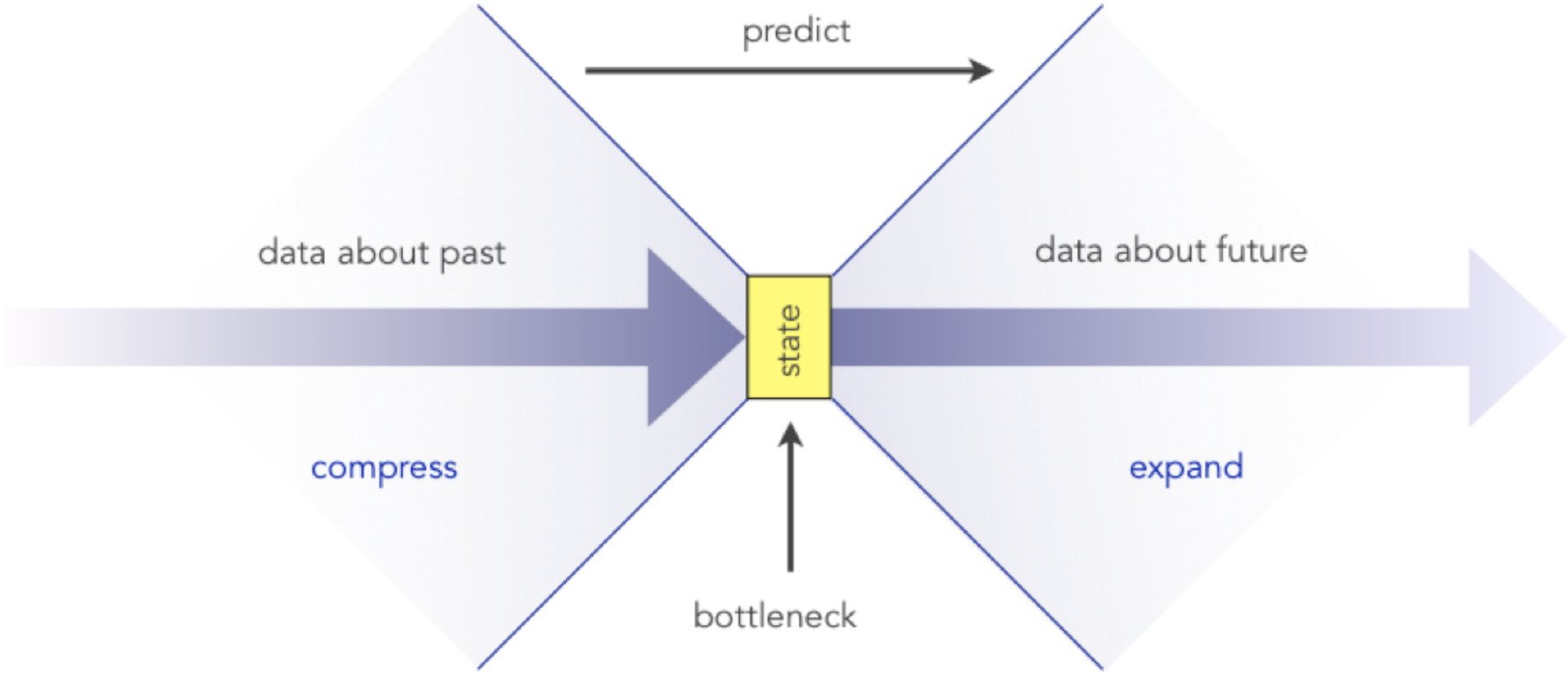
$a_1, a_2, \ldots a_{t-1}$

Previous actions



t

The Markov Assumption

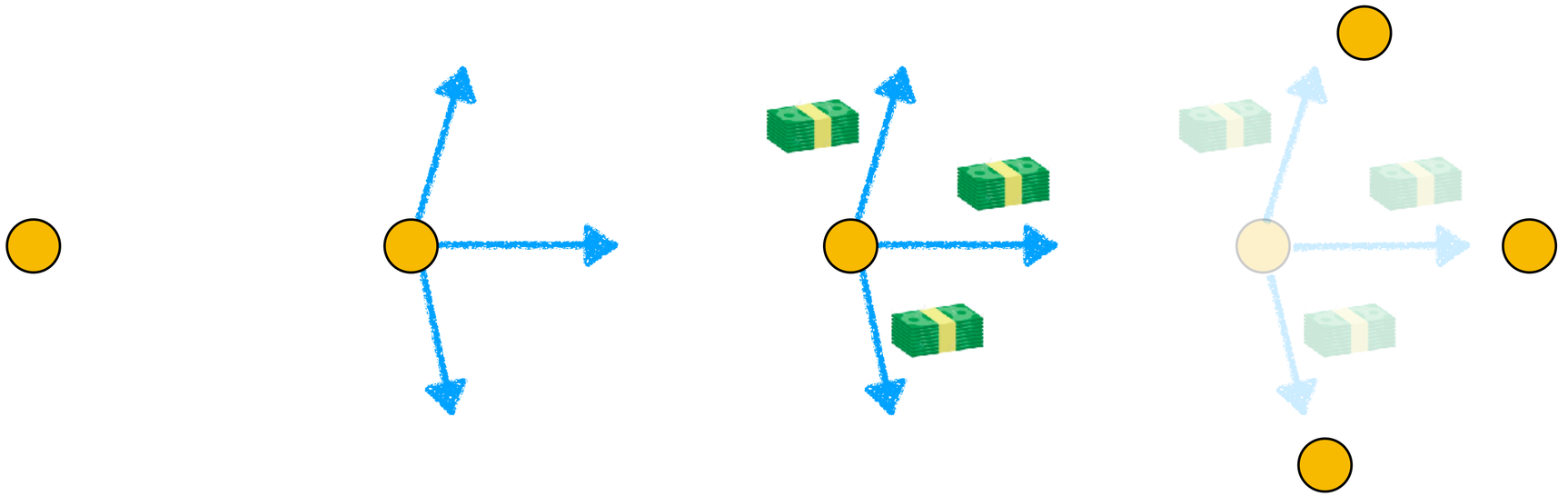


State: statistic of history sufficient to predict the future

Markov Decision Process

A mathematical framework for modeling sequential decision making

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

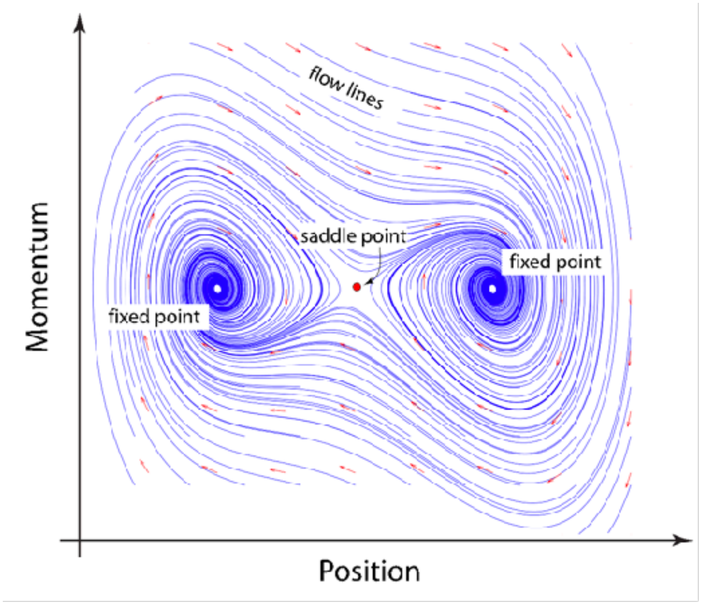


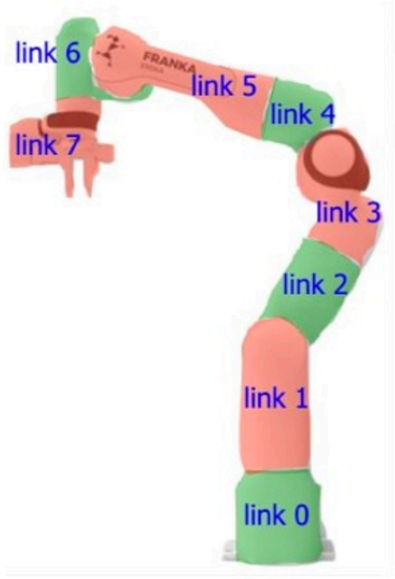
State

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

Sufficient statistic of the system to predict future disregarding the past

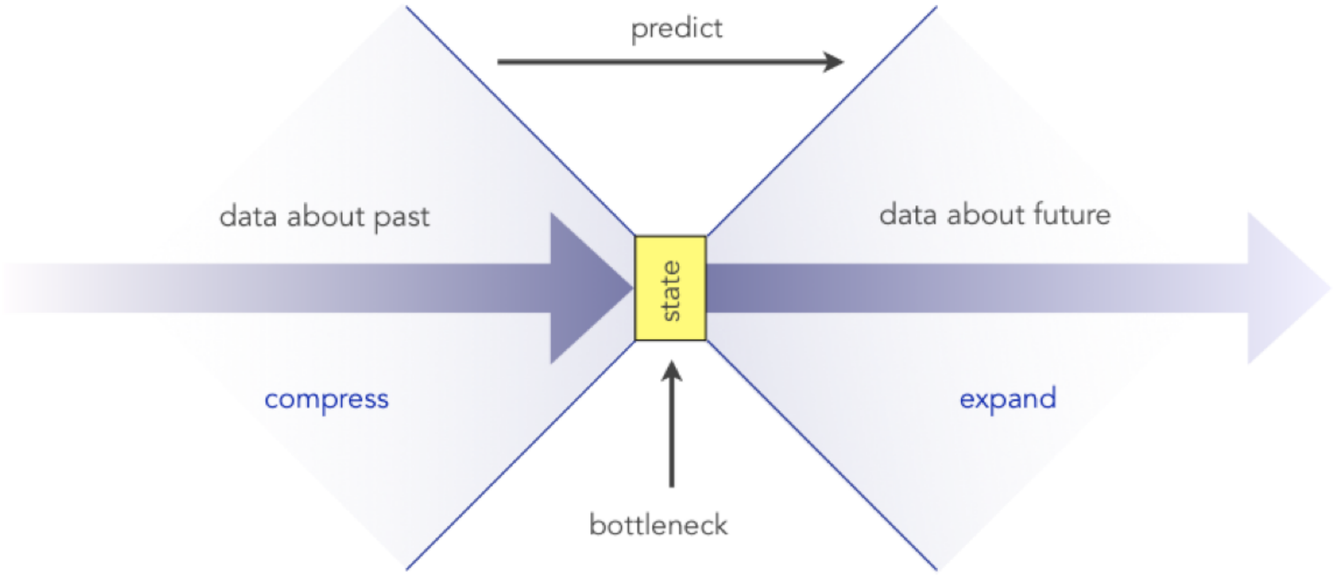
- $s \in S$





Trust

States can be shallow or deep



Shallow state looks at only the past few time steps

Deeps state requires looking far back into the past

Activity!

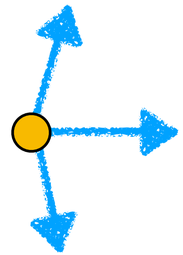
Rank according to depth of state

When poll is active respond at **PollEv.com/sc2582**

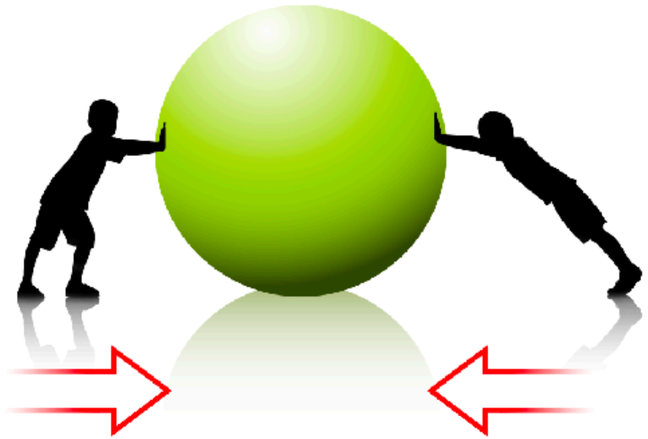
Action

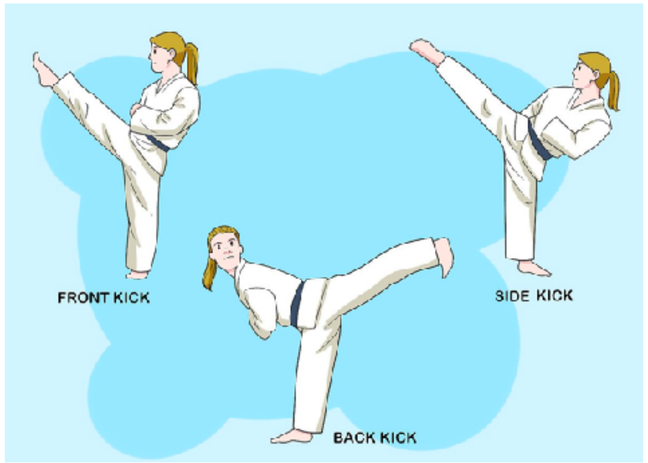
$< S, A, C, \mathcal{T} >$

Doing something:
Control action / decisions



$a \in A$



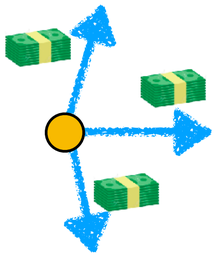


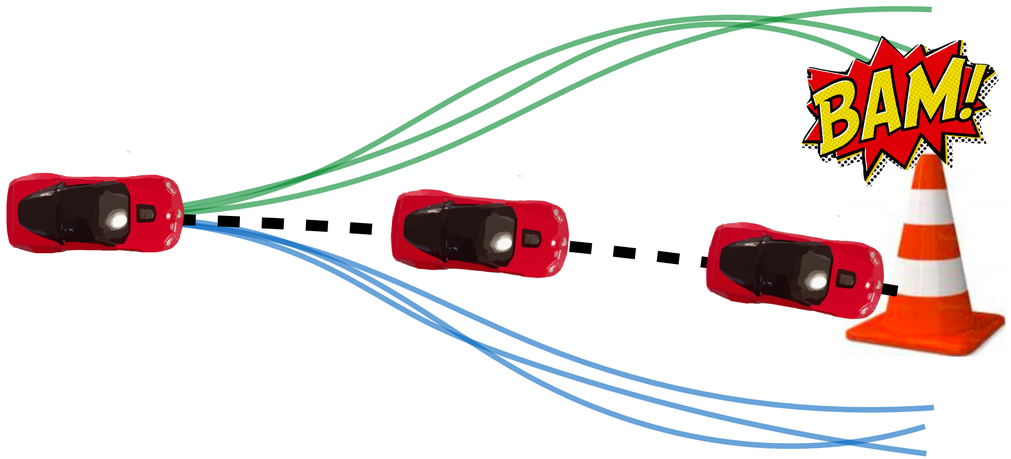
Cost

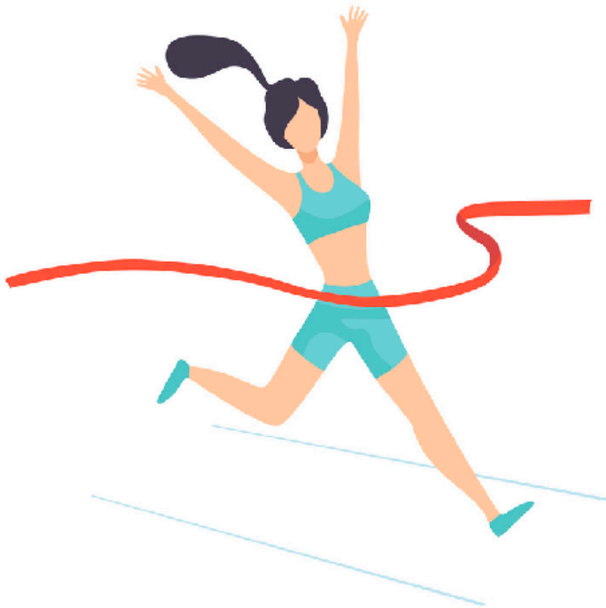
$< S, A, C, T >$

The instantaneous cost of taking an action in a state

$$c(s, a)$$





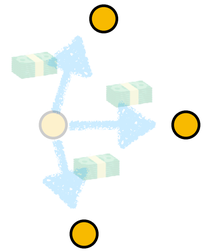


Cost = -Reward

We will use these two interchangeably based on what makes sense

Transition

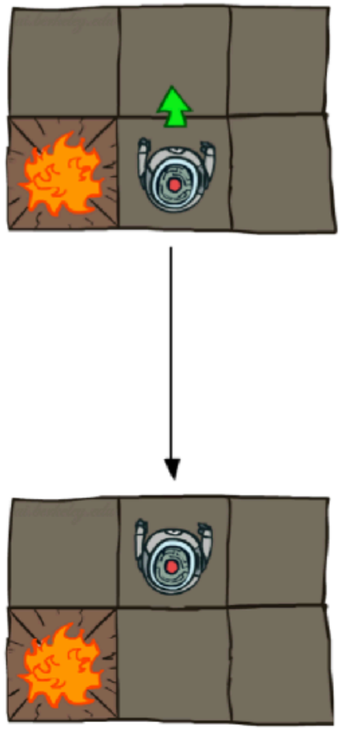
$< S, A, C, \mathcal{T} >$



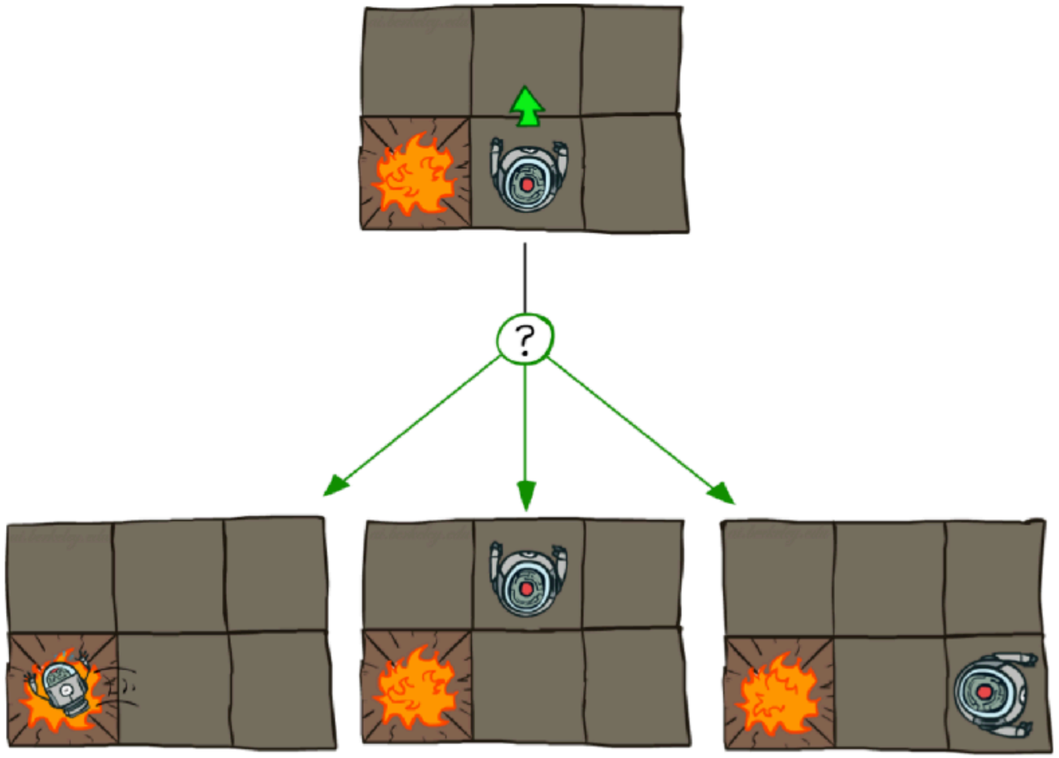
The next state given state and action

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

Deterministic



Stochastic



Credit: Dan Klein

State, action, cost, next state ..

Cost $c(s_0, a_0)$ $c(s_1, a_1)$

"Episode":
A sequence of
state, action, costs

s_0 sampled from an initial distribution over states $P(s_0)$

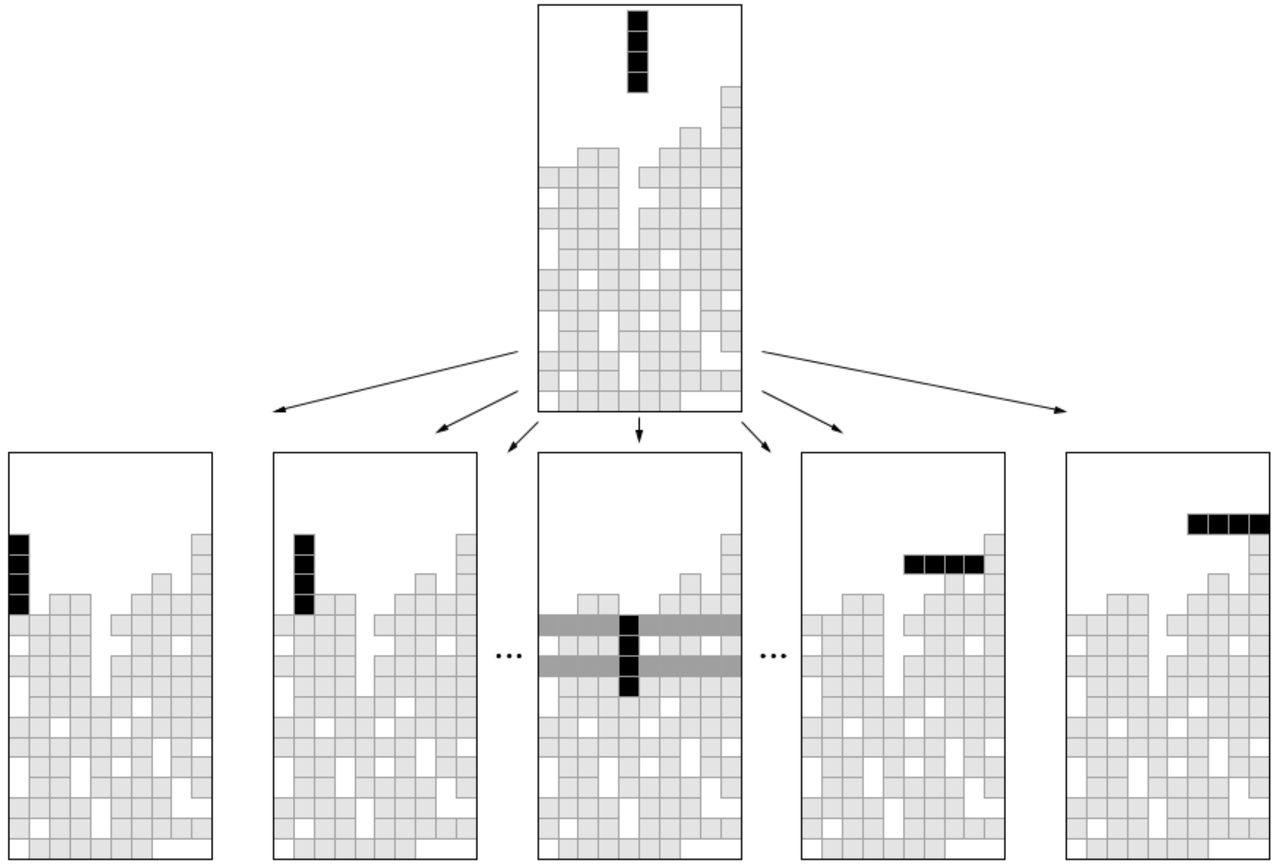
s_1 sampled from transition function $\mathcal{T}(s_0, a_0)$

Goal: Minimize total sum of costs $\sum_{t=0}^{T-1} c(s_t, a_t)$

Today's class

- [x] What makes sequential decision making hard?
- [x] What is a Markov Decision Process (MDP)?
- [ ] Identifying MDPs in the wild
- [ ] What does it mean to solve a MDP?

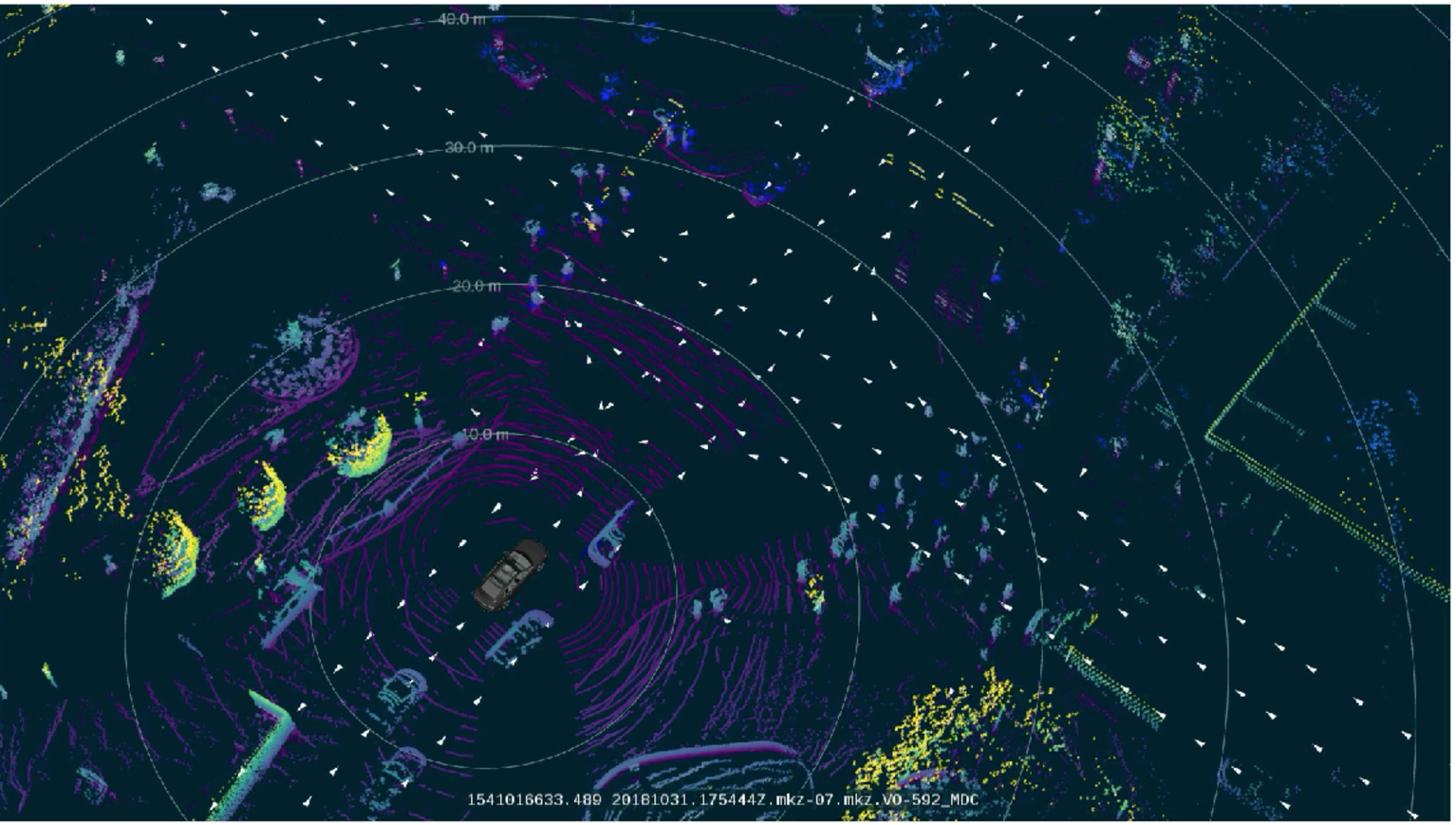
Example 1: Tetris!



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

?

Example 2: Self-driving



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

?

Example 3: Coffee making robot

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

?

Today's class

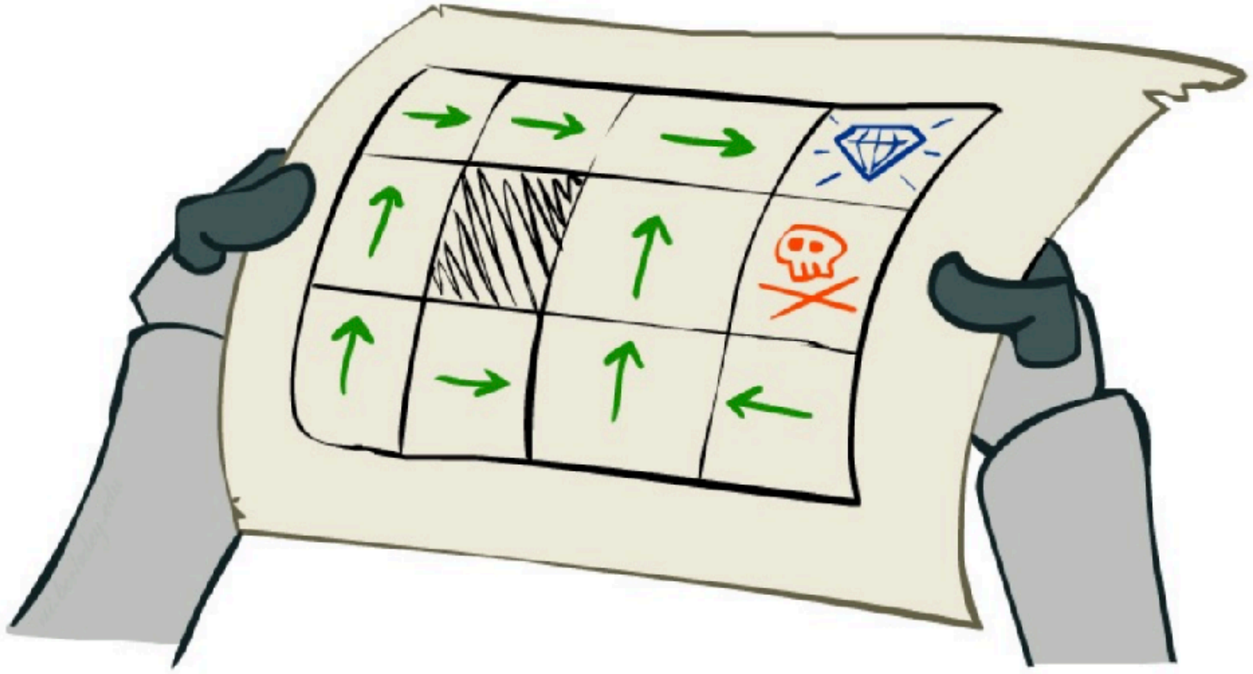
- [x] What makes sequential decision making hard?
- [x] What is a Markov Decision Process (MDP)?
- [x] Identifying MDPs in the wild
- [ ] What does it mean to solve a MDP?

What does it mean to solve a MDP?

Solving an MDP means finding a Policy

$$\pi : s_t \rightarrow a_t$$

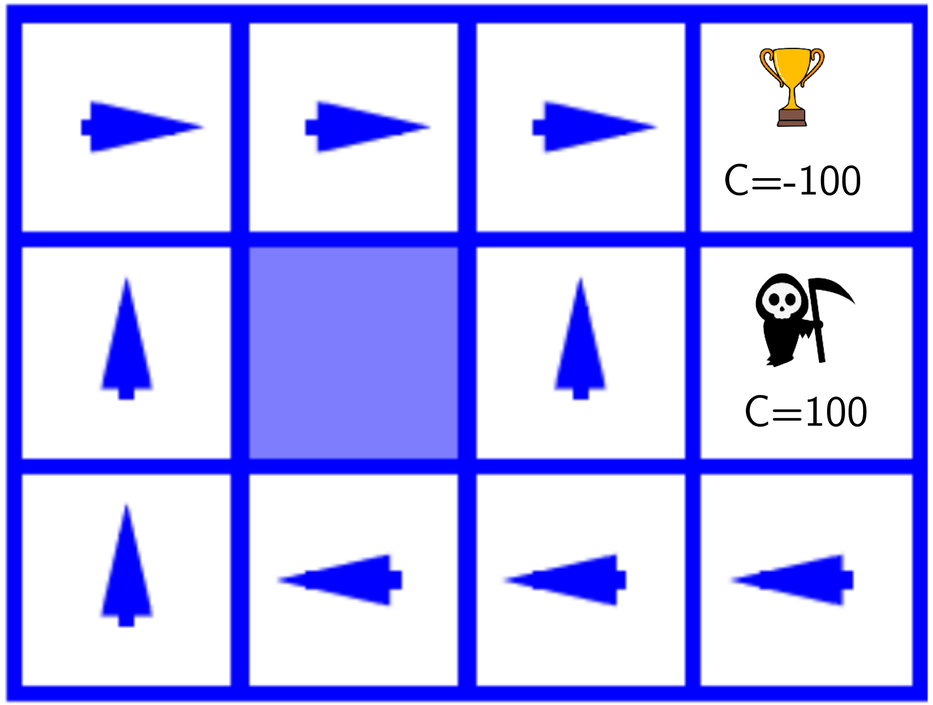
A function that maps state (and time) to action



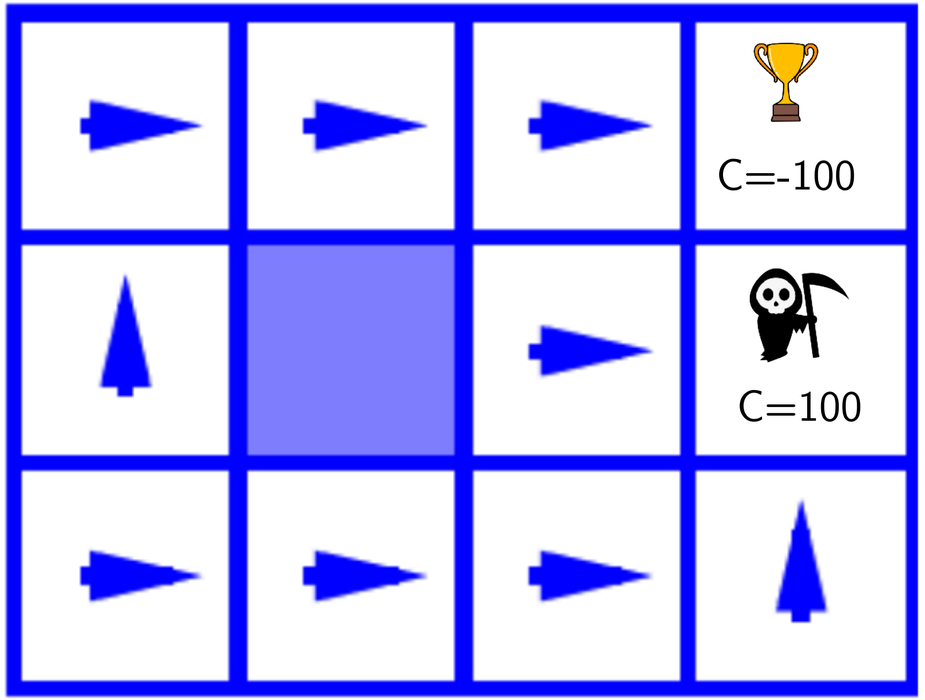
Policy: What action should I choose at any state?

What makes a policy *optimal?*

Which policy is better?



Policy π_1



Policy π_2

What makes a policy *optimal?*

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

(Search over Policies)

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

(Sum over all costs)

Today's class

- ☑ What makes sequential decision making hard?
- ☑ What is a Markov Decision Process (MDP)?
- ☑ Identifying MDPs in the wild
- ☑ What does it mean to solve a MDP?