Research

Dave Abel

April 25th, 2016

Final Exam

- Similar format to Midterm, a bit longer.
- About 15 questions
- Cumulative, but more emphasis on Theory, Compression and Codes, Recursion, Crypto
- Review Session during reading period (more info over email)
- Exam Date/Time: Tuesday, May 19th at 2pm in LIST 120

Overarching Goal

Overarching Goal

value

existence

reason

knowledge

mind

Philosophy

Overarching Goal

value

existence

reason

knowledge

mind

problem solving

limits of reasoning

model of the mind

Philosophy

Computer Science

Overarching Goal

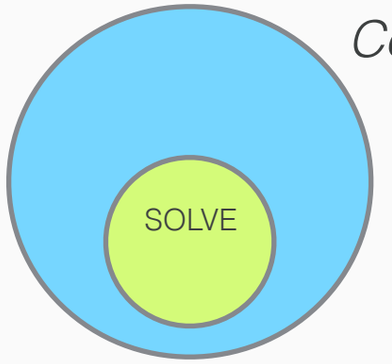
value

existence

reason

knowledge

mind



Complexity

Artificial Intelligence

Computability

Philosophy

Computer Science

Overarching Goal

value

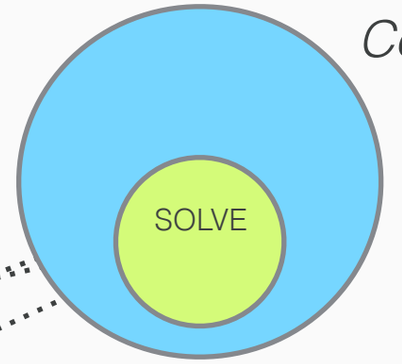
existence

reason

knowledge

mind

Complexity



Artificial Intelligence

Computability

Philosophy

Computer Science

Overarching Goal

Use the tools of computation to ground philosophical investigations

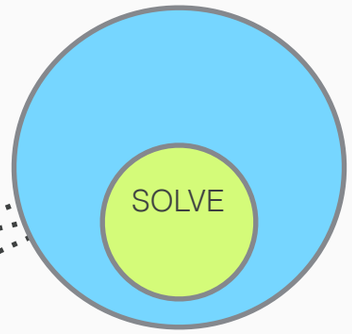
value

existence

reason

knowledge

mind



Complexity

Artificial Intelligence

Computability

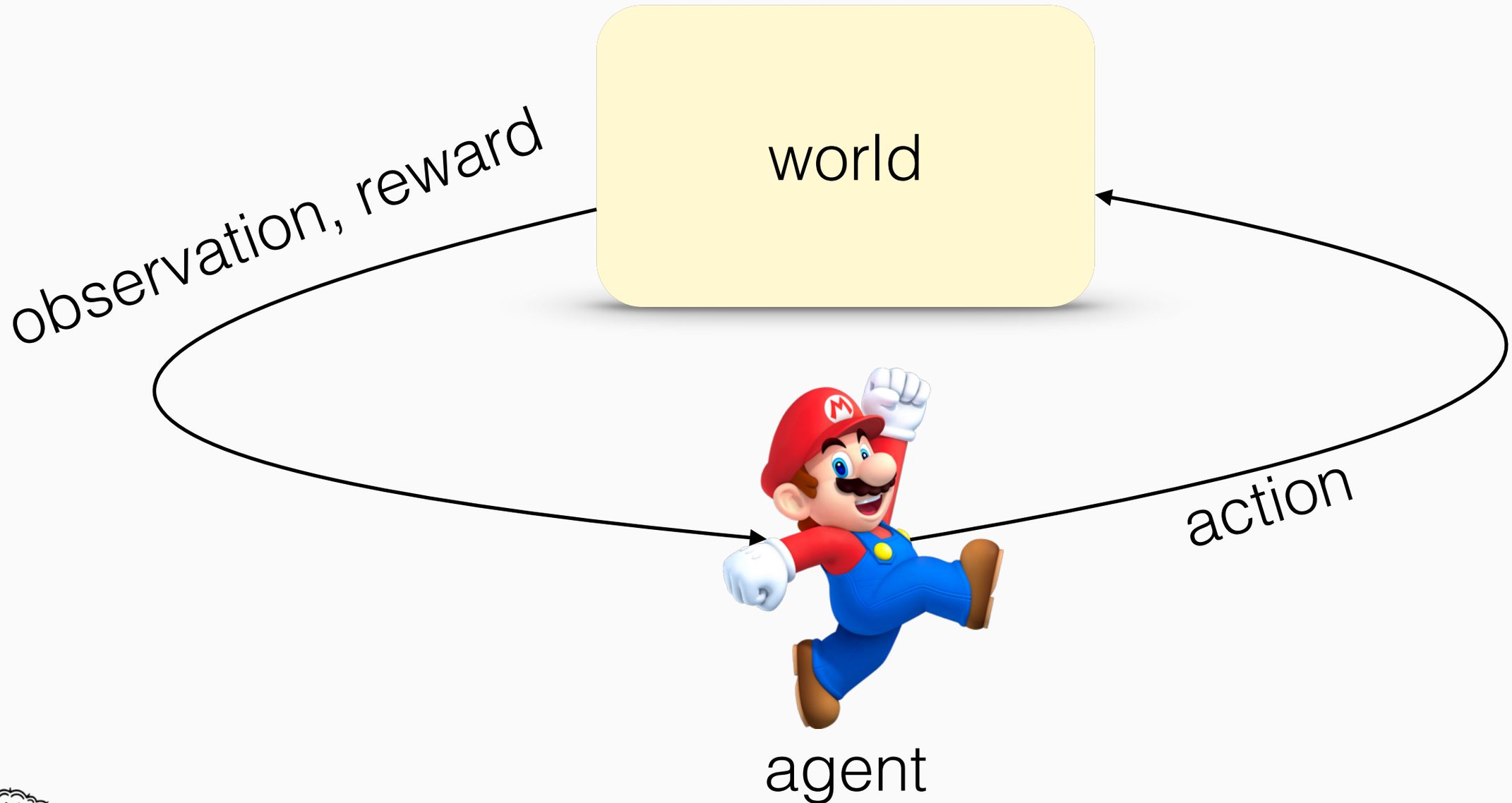
Philosophy

Computer Science

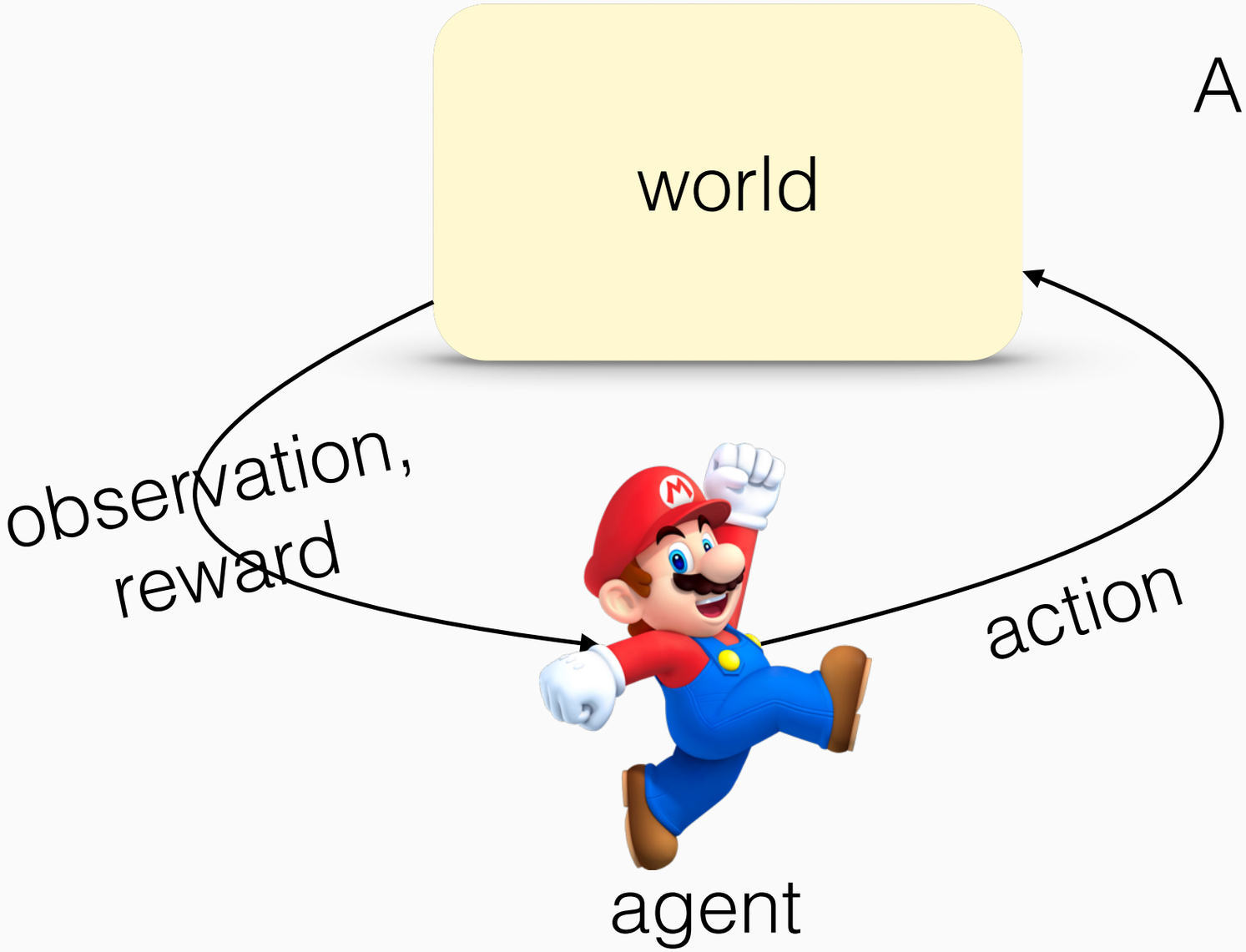
Current Work:

Reinforcement Learning

Recap: Reinforcement Learning



Reinforcement Learning



Any problem with a goal…

Goal: Maximize reward!

Or repeated behavior!

Reinforcement Learning

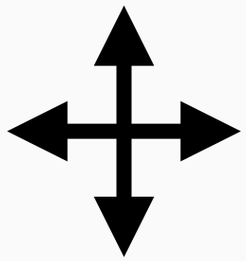
Formalized as a *Markov Decision Process:*

- [$\mathcal{S}$] A collection of *states (i.e. configurations of world)*

Reinforcement Learning

Formalized as a *Markov Decision Process:*

- [$\mathcal{S}$] A collection of *states (i.e. configurations of world)*
- [$\mathcal{A}$] Some *actions (i.e. things the agent can do)*



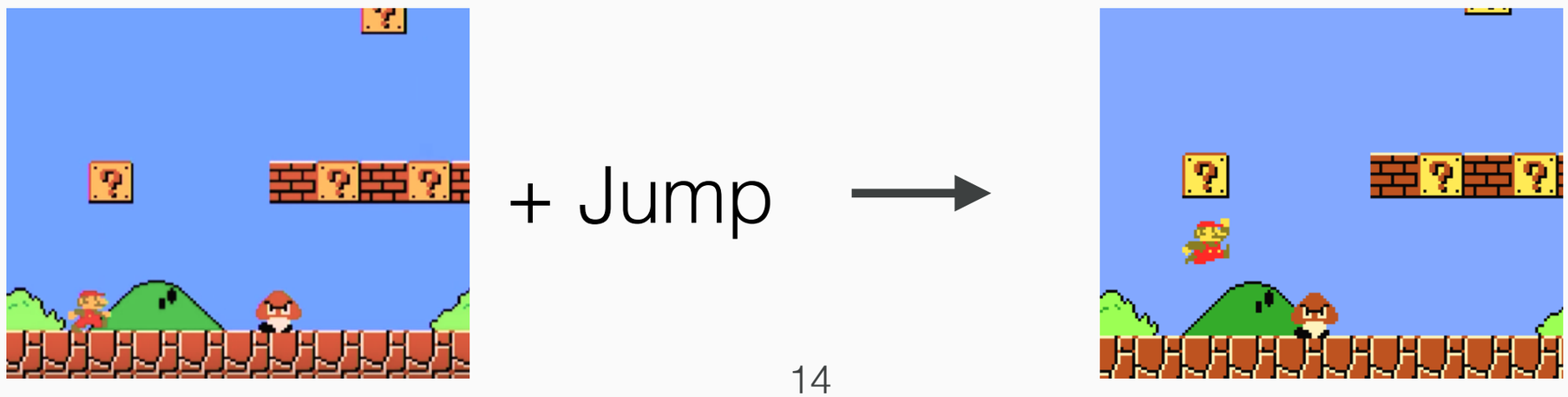
Reinforcement Learning

Formalized as a *Markov Decision Process:*

- [$\mathcal{S}$] A collection of *states (i.e. configurations of world)*

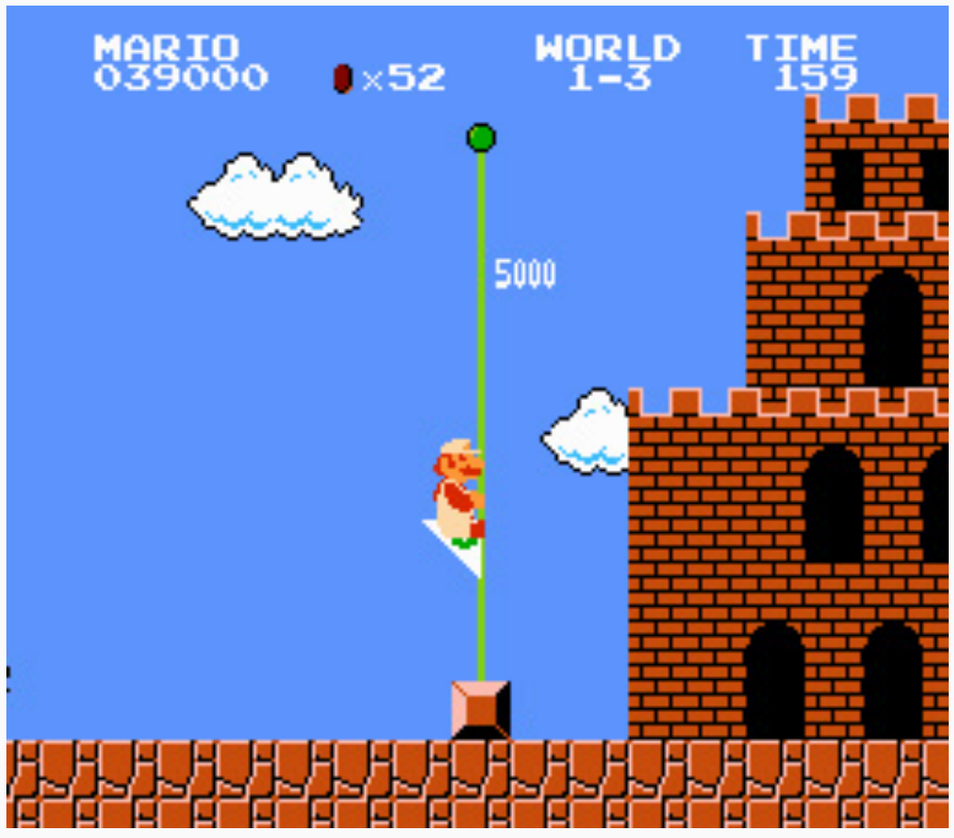
- [$\mathcal{A}$] Some *actions (i.e. things the agent can do)*

- [$\mathcal{T}$] Transitions between states (i.e. *action effects*)



Reinforcement Learning

Formalized as a *Markov Decision Process:*



- [$\mathcal{R}$] Rewards (i.e. *what is good/bad behavior*)

Reinforcement Learning

Formalized as a *Markov Decision Process:*

- [$\mathcal{S}$] A collection of *states (i.e. configurations of world)*
- [$\mathcal{A}$] Some *actions (i.e. things the agent can do)*
- [$\mathcal{T}$] Transitions between states (i.e. *action effects*)
- [$\mathcal{R}$] Rewards (i.e. *what is good/bad behavior*)

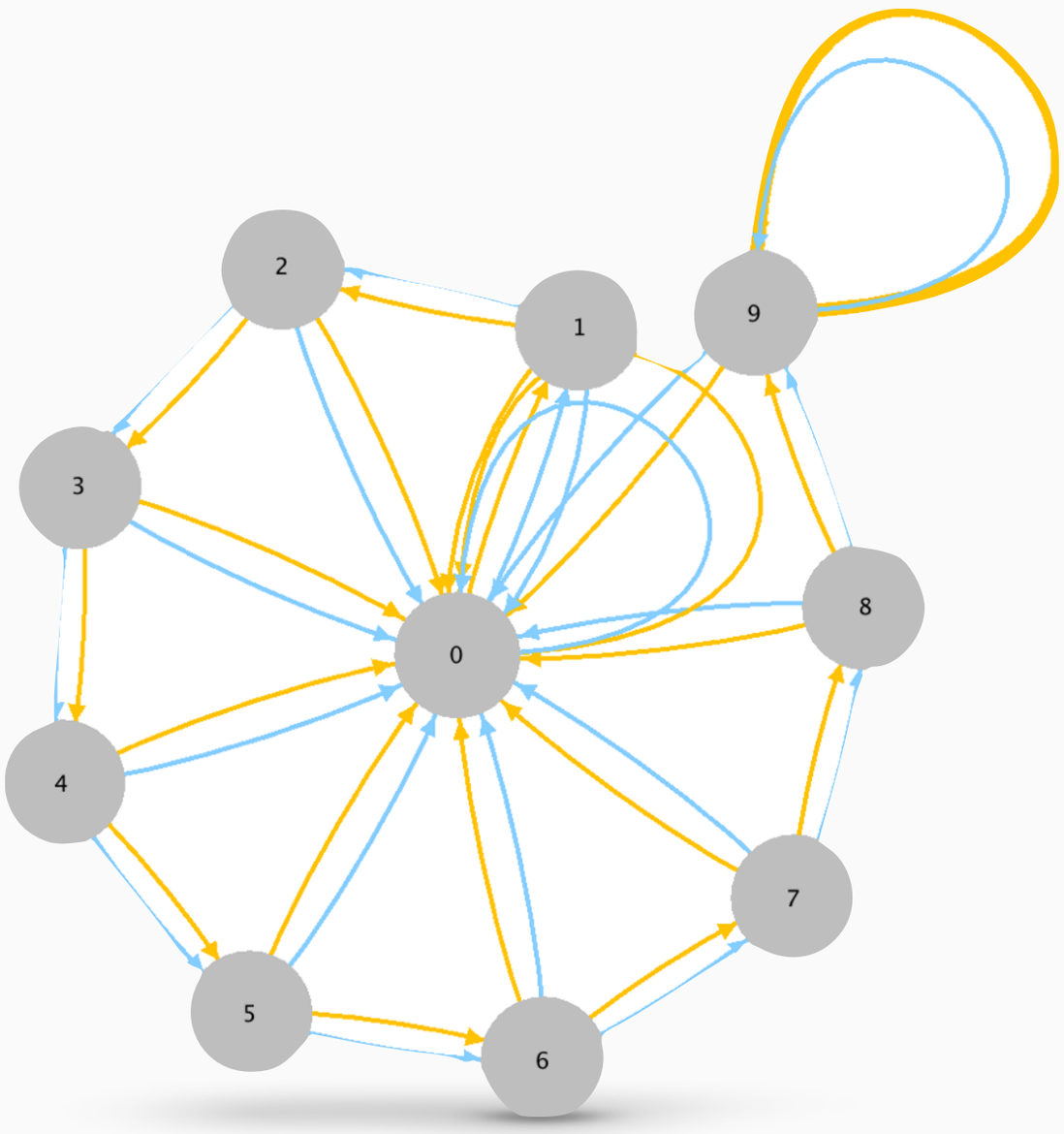
Reinforcement Learning: Taxi

Reinforcement Learning: Taxi

Formalized as a *Markov Decision Process:*

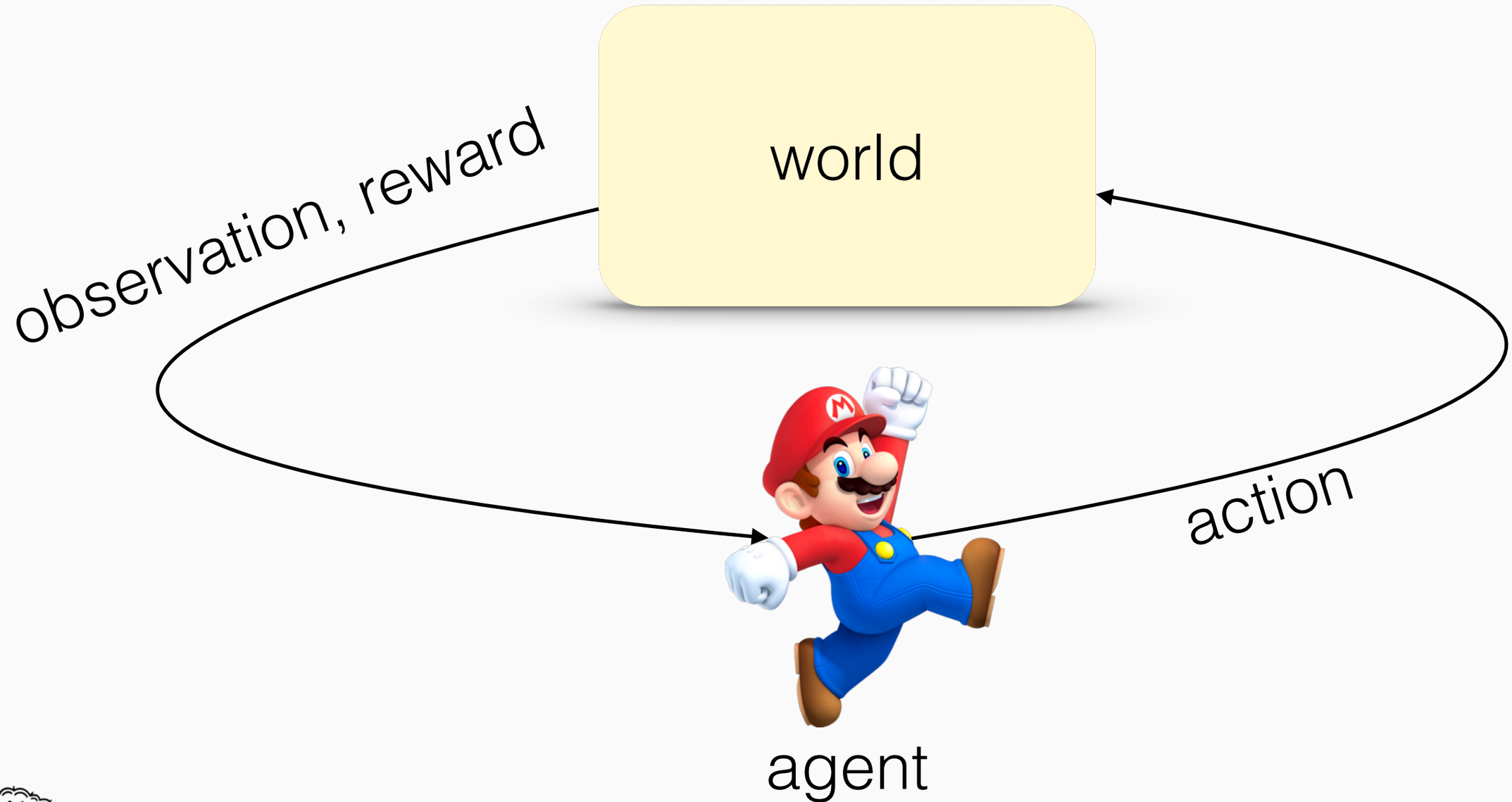
- $[\mathcal{S}]$ = Location of agent, passengers
- $[\mathcal{A}]$ = Up, down, left, right, pickup, drop off
- $[\mathcal{T}]$ = Movement, pickup passenger/ dropoff
- $[\mathcal{R}]$ = All passengers at destinations.

Graphs! (again)



[Strens 2000]

Reinforcement Learning



Two Problems

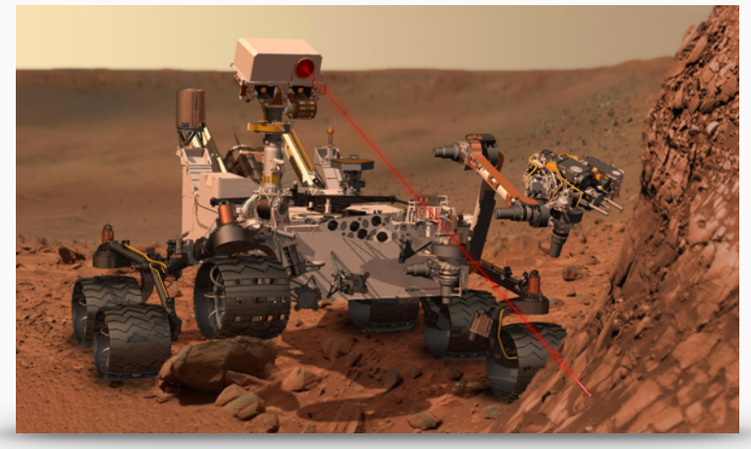
- **Planning:**
  - Input: S, A, T, R.
  - Output: A sequence of actions for maximizing reward.
- **Reinforcement Learning:**
  - Input: S, A, ability to interact with world.
  - Output: A sequence of actions for maximizing reward.

Two Problems

- **Planning:**
  - Input: S, A, T, R.
  - Output: A sequence of actions for maximizing reward.
- **Reinforcement Learning:**
  - Input: S, A, ability to interact with world.
  - Output: A sequence of actions for maximizing reward.

Central problems of AI

Overall Goals

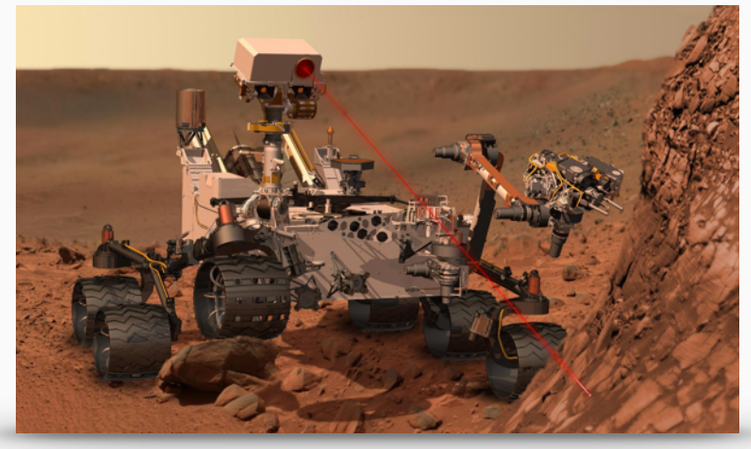


[Bernstein, Zilberstein ECP 2014]

[Ermon et al. IJCAI ’11, Ermon et al. UAI ’10]

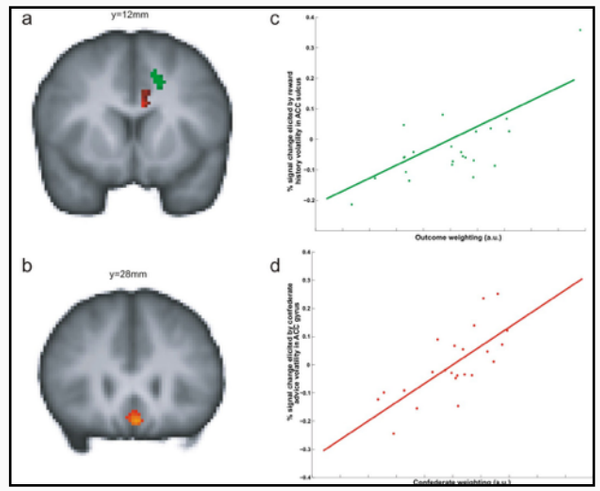
1. Exciting Applications

Overall Goals



[Bernstein, Zilberstein ECP 2014]

[Ermon et al. IJCAI ’11, Ermon et al. UAI ’10]



[Lee, Seo, Jung 2013]

https://uknightedart.wordpress.com/robots/robot-thinker/

1. Exciting Applications

2. Understanding Intelligence

Overview

1. Reinforcement Learning & Abstraction

2. Artificial Intelligence + Ethics

3. Minecraft

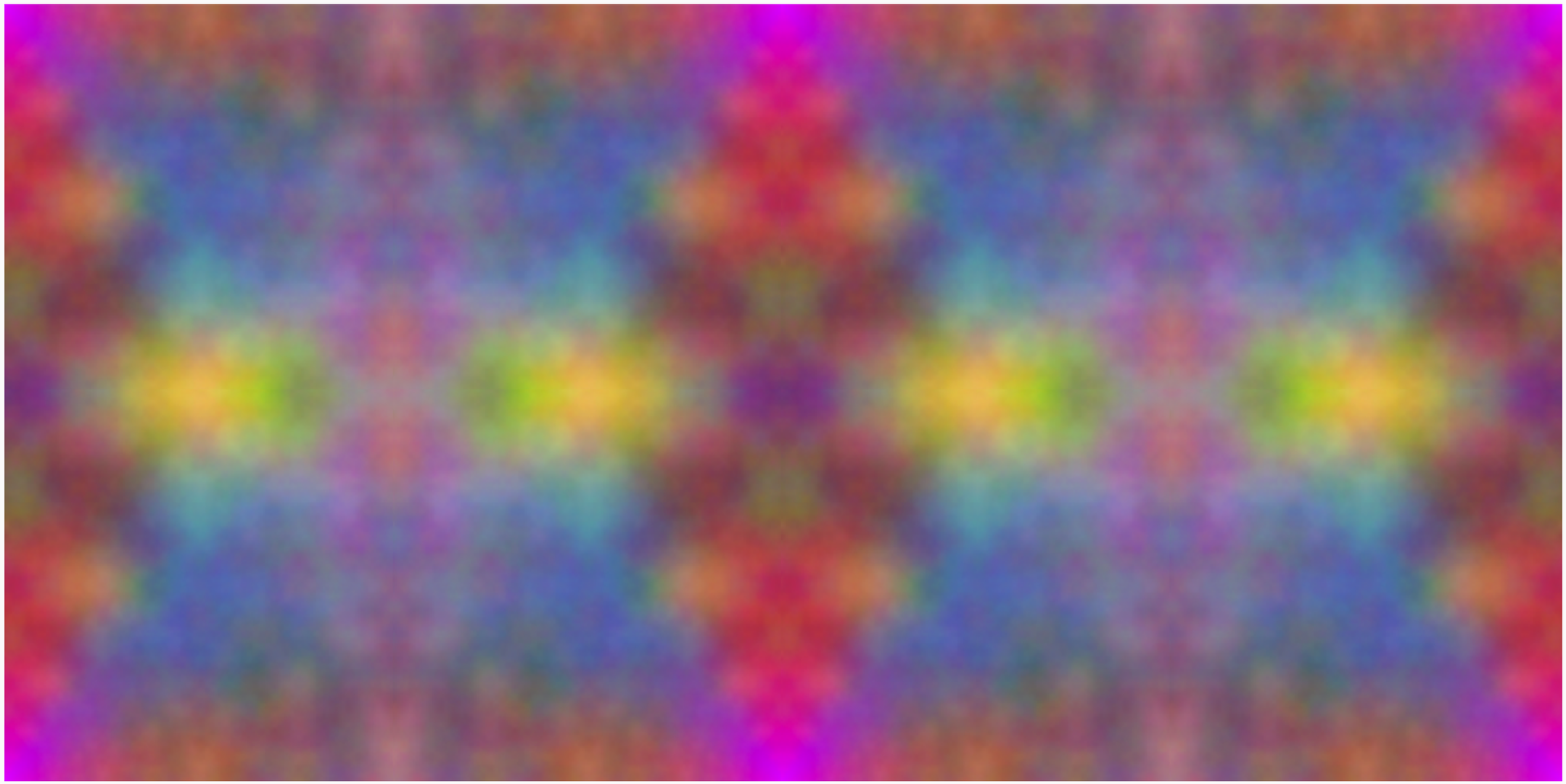
1. Intelligence & Abstraction

Premise

1) Abstraction plays a central role in intelligence.

2) Agents using abstraction can leverage more of SOLVE to act in the real world.

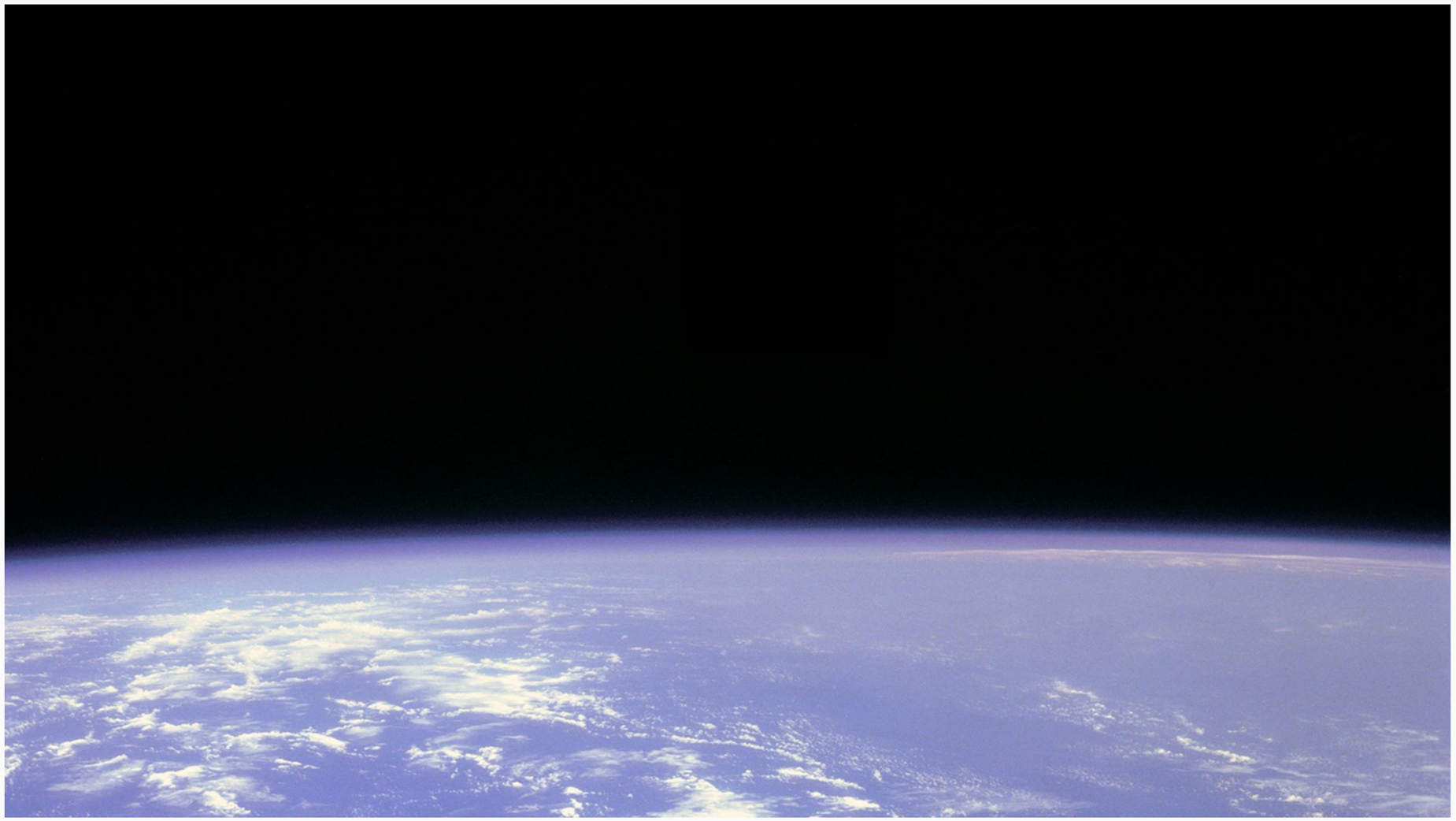
Intuition: Lots of information!



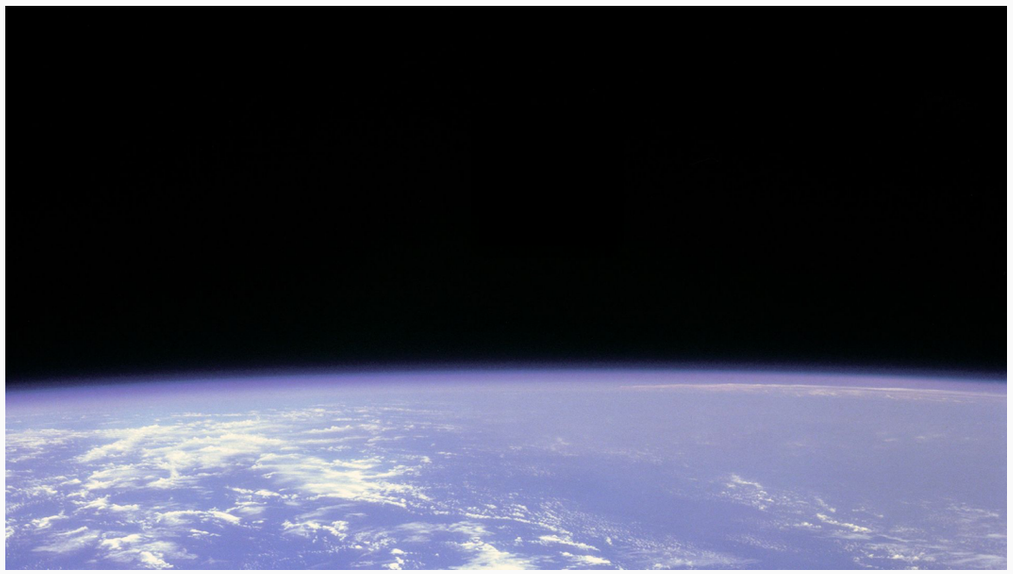
Intuition: We Abstract

"sock drawer"

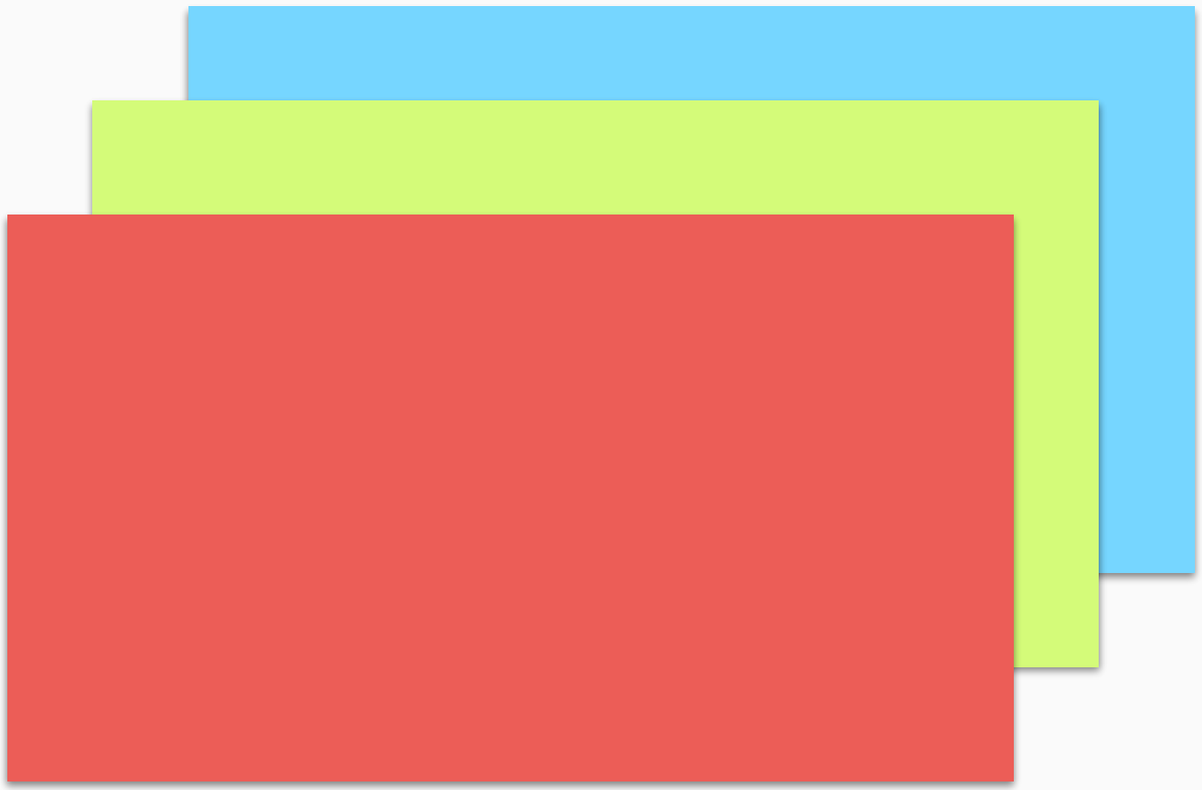
Compression: Intuition



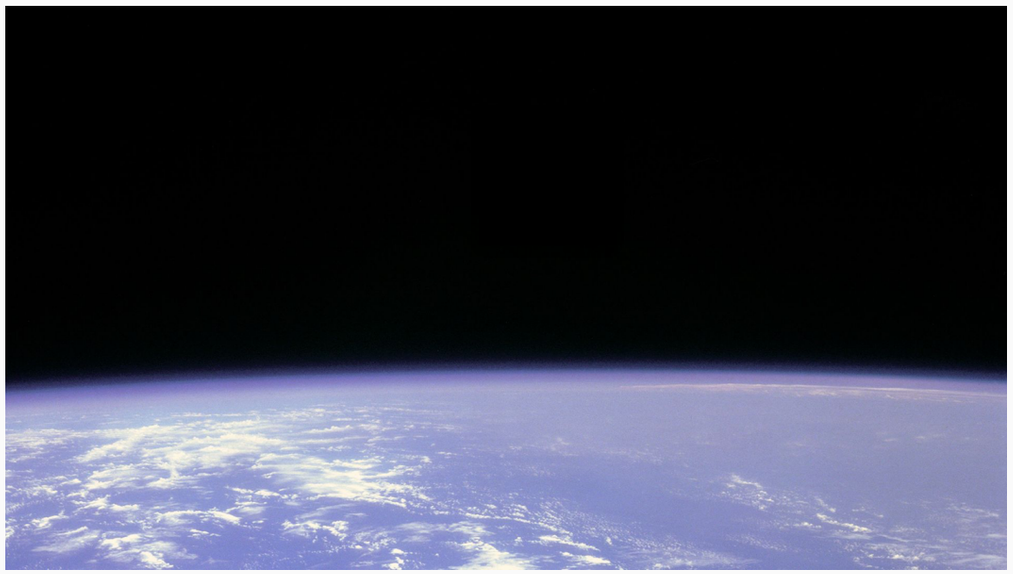
Compression: Intuition



=

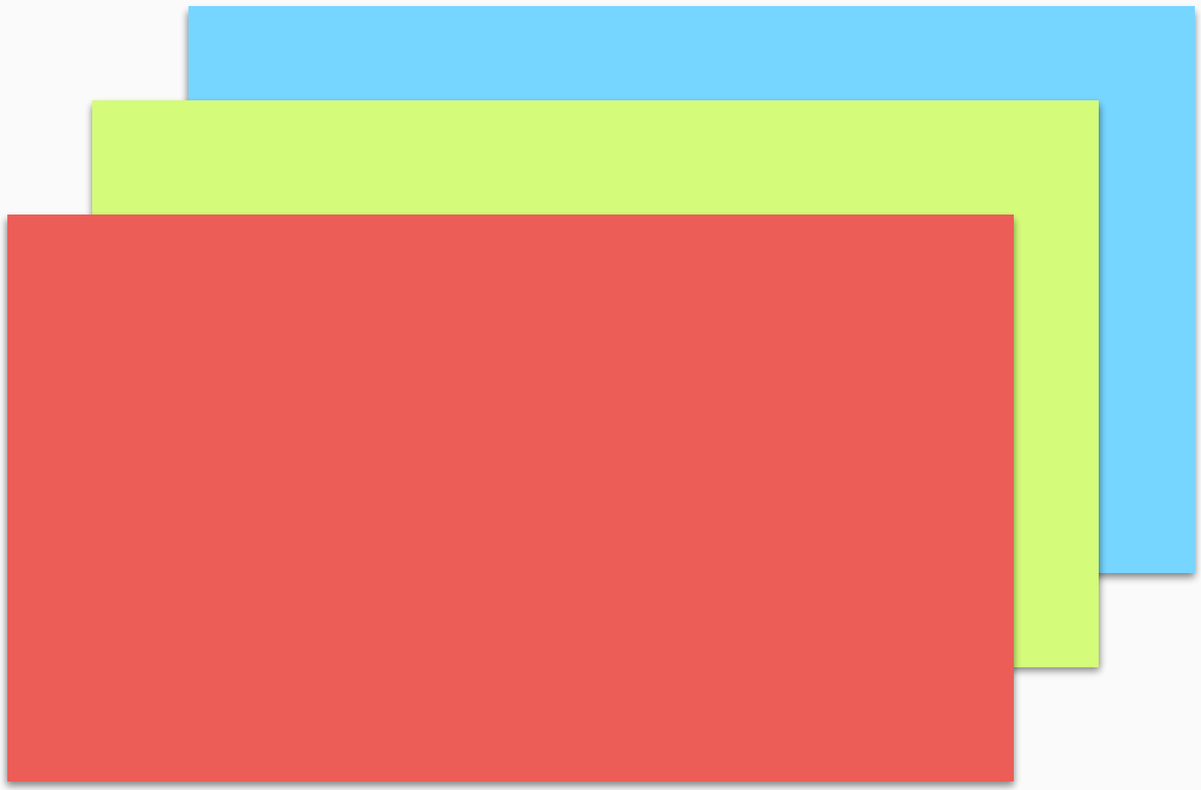


Compression: Intuition

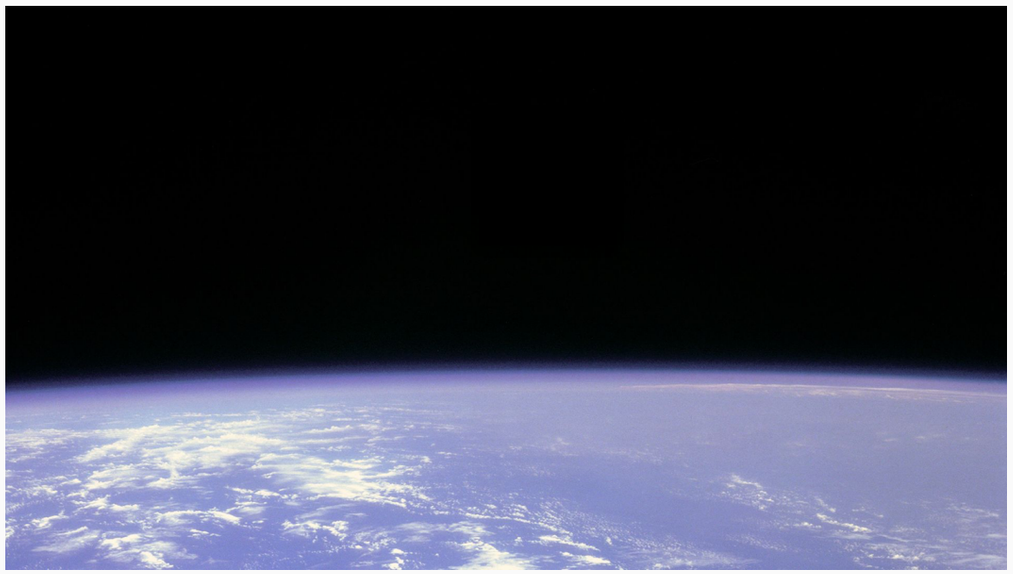


1Kb

=

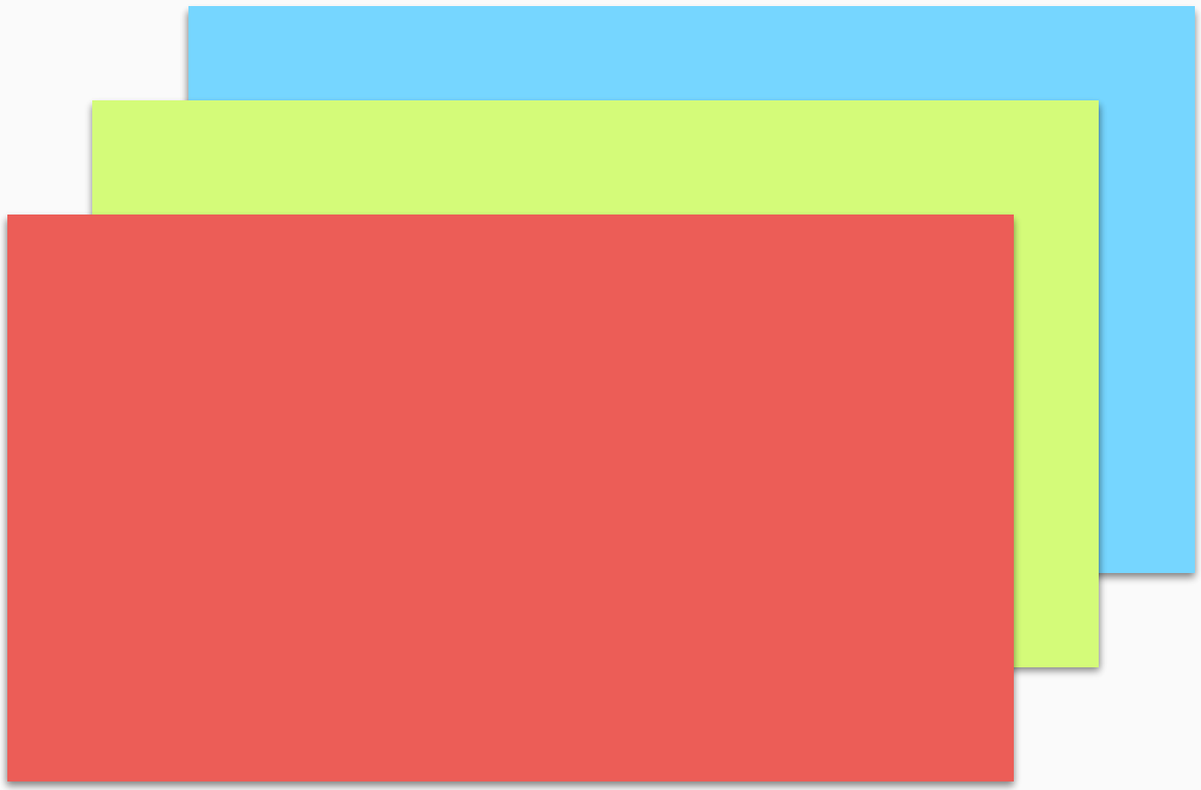


Compression: Intuition



1Kb

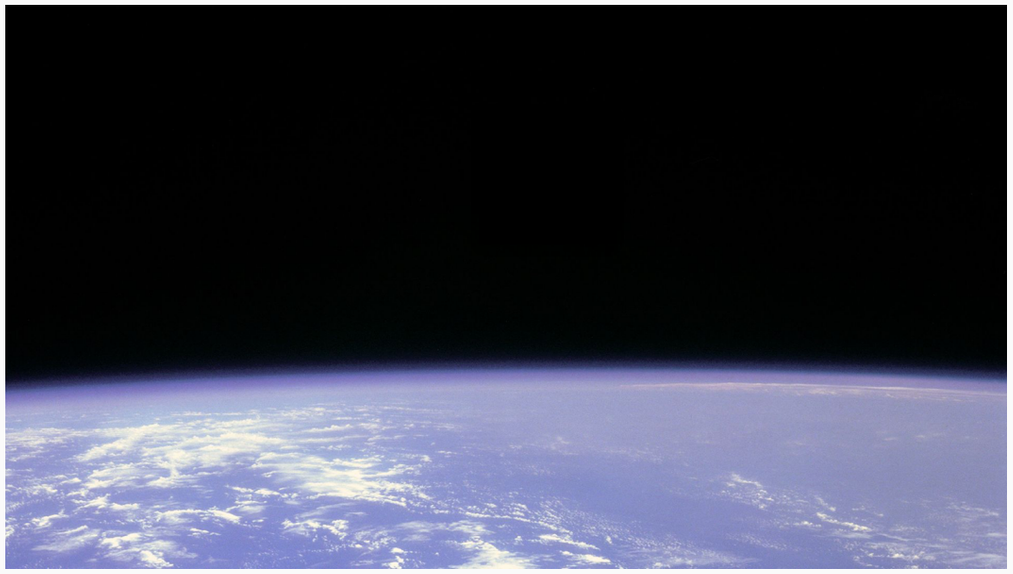
=



1Kb

Compression: Intuition

Everything else, black



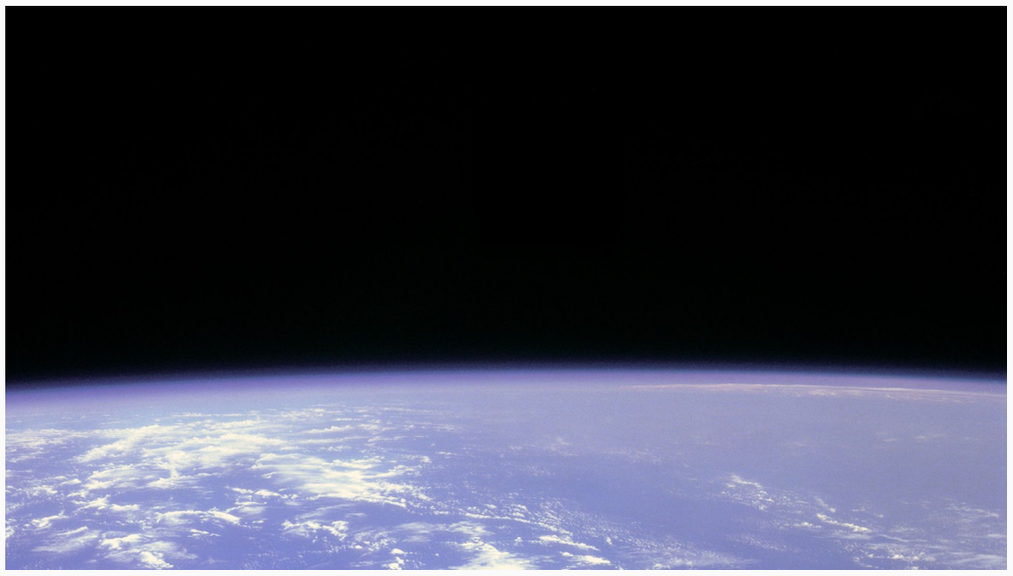
1Kb

=

0.3Kb

Compression: Intuition

Everything else, black



=

1Kb

0.3Kb

Shaved off .7 Kilobytes!

Compression: Intuition

“aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa”

=

200 x “a”

2000 bytes

50 bytes

Compression:
Faster Transmission

Compression:
More Computation!

A = [1, 9, 2, 7]

B = [1, 9, 2, 7, 2, 6, 8, 6, 5, 10, 2, 1, 7, 9, 9, 7, 10, 2, 6, 2, 0, 8, 1, 2, 10, 1, 3, 8, 0, 4, 4, 1, 3, 1, 7, 7, 2, 9, 2, 7, 10, 2, 0, 0, 6, 7, 0, 10, 9, 8, 8, 7, 10, 10, 8, 5, 6, 10, 5, 6, 7, 0, 5, 0, 3, 3, 7, 10, 9, 3, 3, 9, 3, 2, 4, 0, 10, 10, 7, 4, 2, 5, 6, 4, 9, 6, 6, 5, 8, 6, 4, 1, 4, 10, 1, 3, 0, 10, 2, 6]

Compression:
More Computation!

A = [1, 9, 2, 7]

Compute: **max, min, average, sort**

B = [1, 9, 2, 7, 2, 6, 8, 6, 5, 10, 2, 1, 7, 9, 9, 7, 10, 2, 6, 2, 0, 8, 1, 2, 10, 1, 3, 8, 0, 4, 4, 1, 3, 1, 7, 7, 2, 9, 2, 7, 10, 2, 0, 0, 6, 7, 0, 10, 9, 8, 8, 7, 10, 10, 8, 5, 6, 10, 5, 6, 7, 0, 5, 0, 3, 3, 7, 10, 9, 3, 3, 9, 3, 2, 4, 0, 10, 10, 7, 4, 2, 5, 6, 4, 9, 6, 6, 5, 8, 6, 4, 1, 4, 10, 1, 3, 0, 10, 2, 6]

Compression:
More Computation!

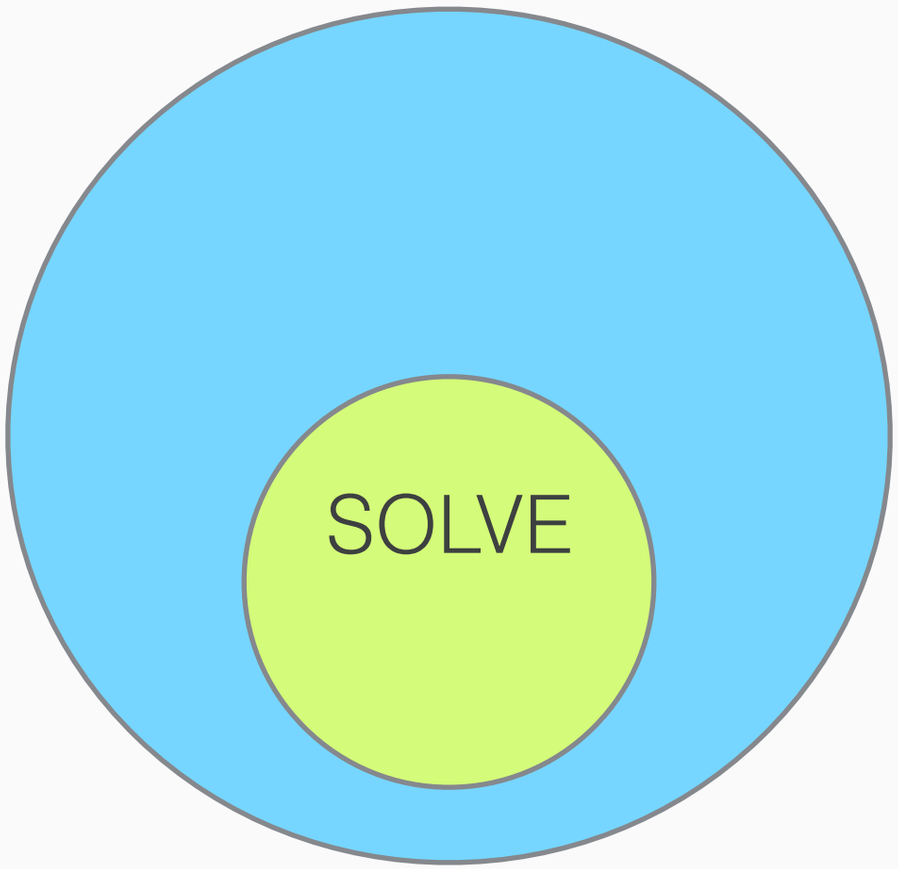
A = [1, 9, 2, 7]

Easier!

Compute: **max, min, average, sort**

B = [1, 9, 2, 7, 2, 6, 8, 6, 5, 10, 2, 1, 7, 9, 9, 7, 10, 2, 6, 2, 0, 8, 1, 2, 10, 1, 3, 8, 0, 4, 4, 1, 3, 1, 7, 7, 2, 9, 2, 7, 10, 2, 0, 0, 6, 7, 0, 10, 9, 8, 8, 7, 10, 10, 8, 5, 6, 10, 5, 6, 7, 0, 5, 0, 3, 3, 7, 10, 9, 3, 3, 9, 3, 2, 4, 0, 10, 10, 7, 4, 2, 5, 6, 4, 9, 6, 6, 5, 8, 6, 4, 1, 4, 10, 1, 3, 0, 10, 2, 6]

Intuition



With compressed models of the world, our Artificial Intelligences can compute *more*.

Intuition: We Abstract

Hypothesis: Abstraction is *essential* for intelligent agents to operate in the real world. To compute anything, we need compact representations.

"sock drawer"

Abstraction + RL

A theory of abstraction for *representations*

State Abstraction + RL

s =

State Abstraction + RL

s =

s = *top left pixel blue, so is next one, etc…*

State Abstraction + RL

s =

s = *top left pixel blue, so is next one, etc…*

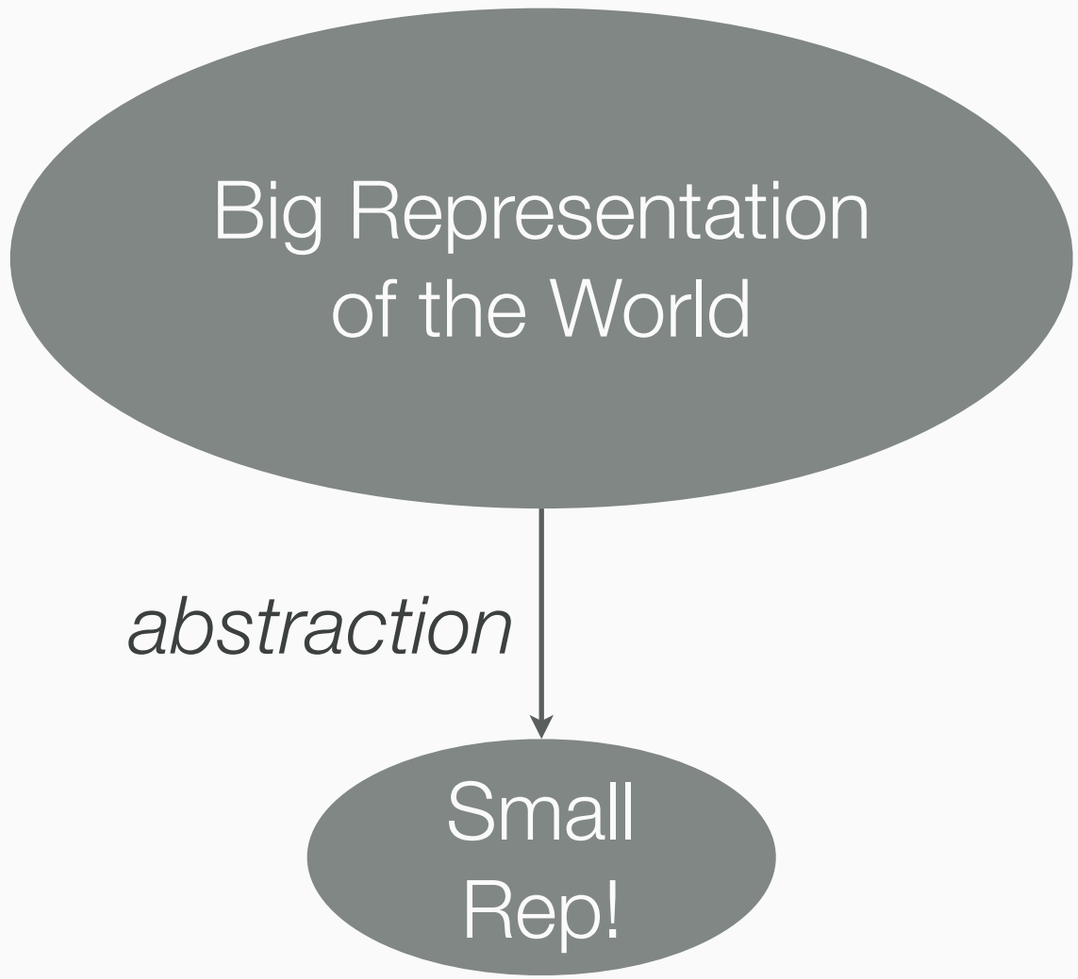
s = *Mario is about to hit the block*

Full Pipeline

Goal: develop a theory of abstraction to *compress* representations of the world

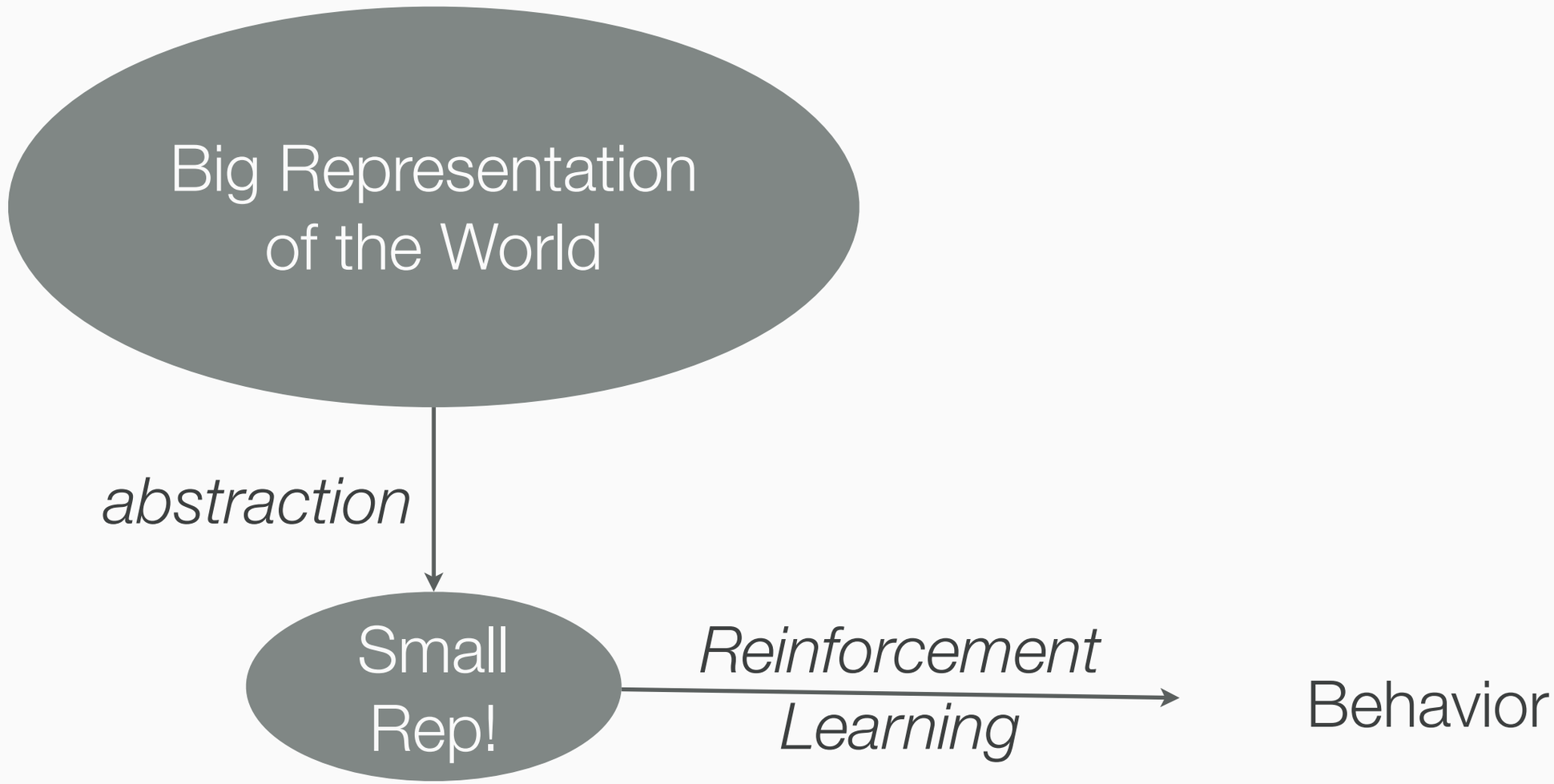
Full Pipeline

Goal: develop a theory of abstraction to *compress* representations of the world



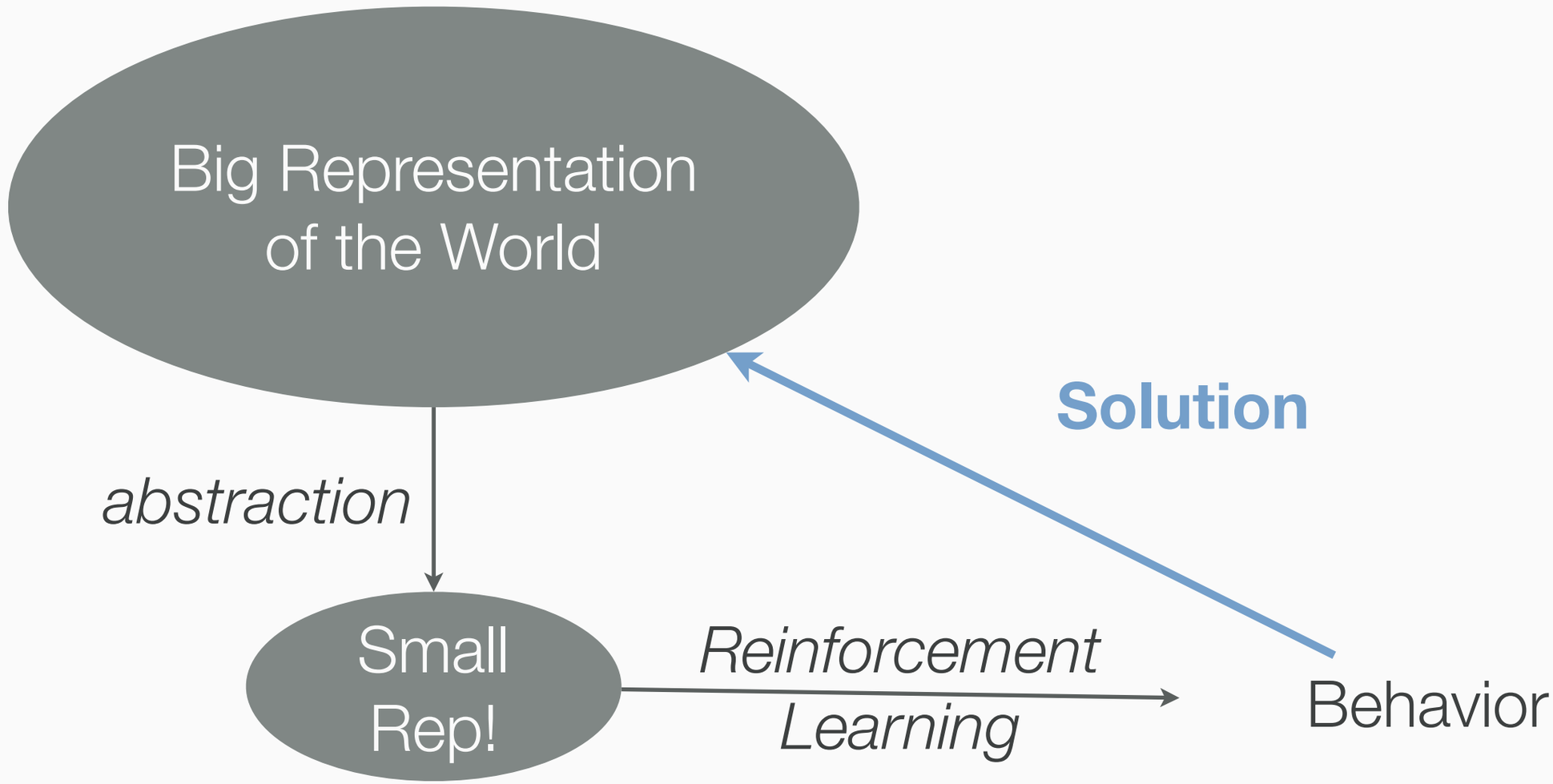
Full Pipeline

Goal: develop a theory of abstraction to *compress* representations of the world

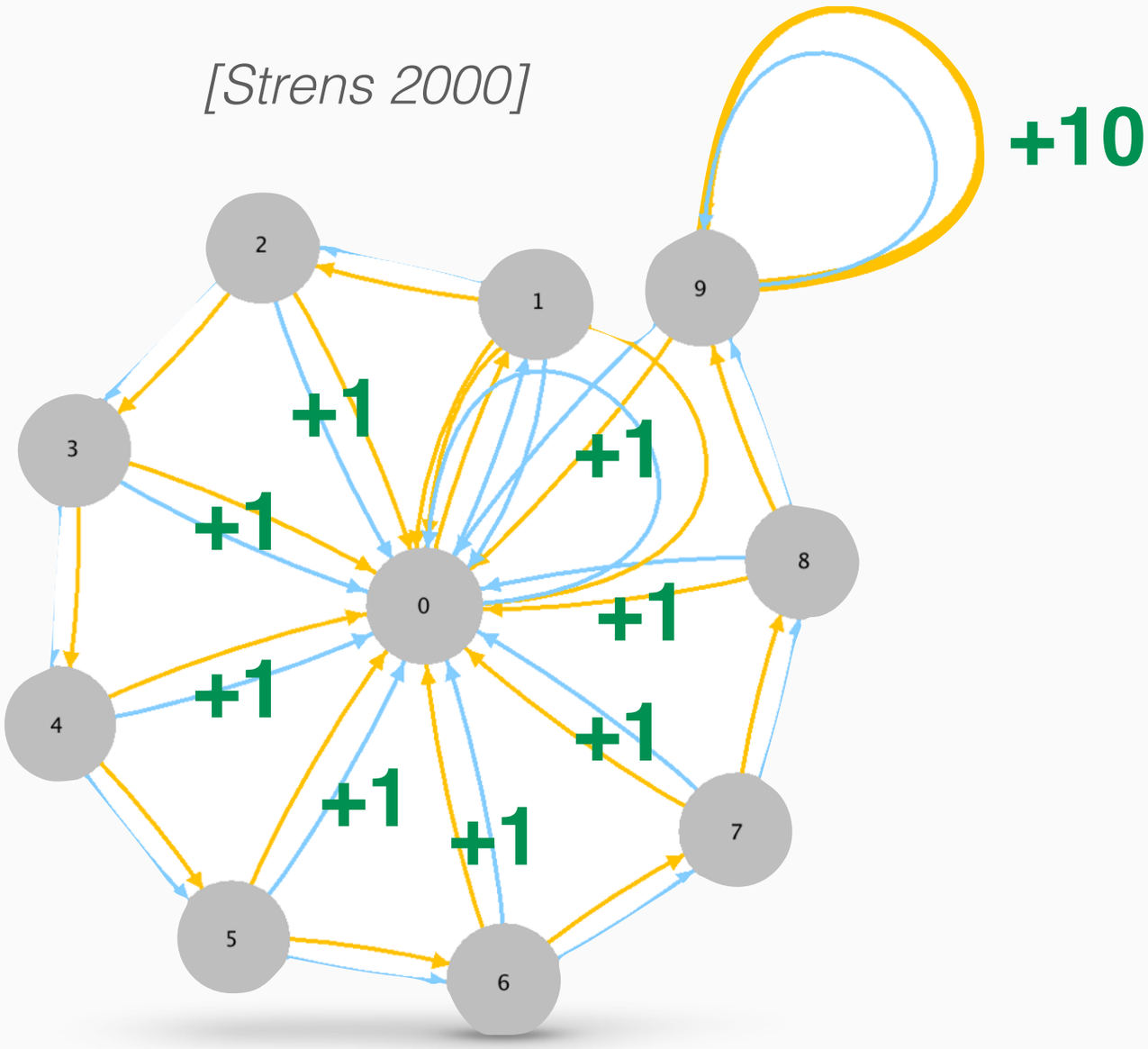


Full Pipeline

Goal: develop a theory of abstraction to *compress* representations of the world

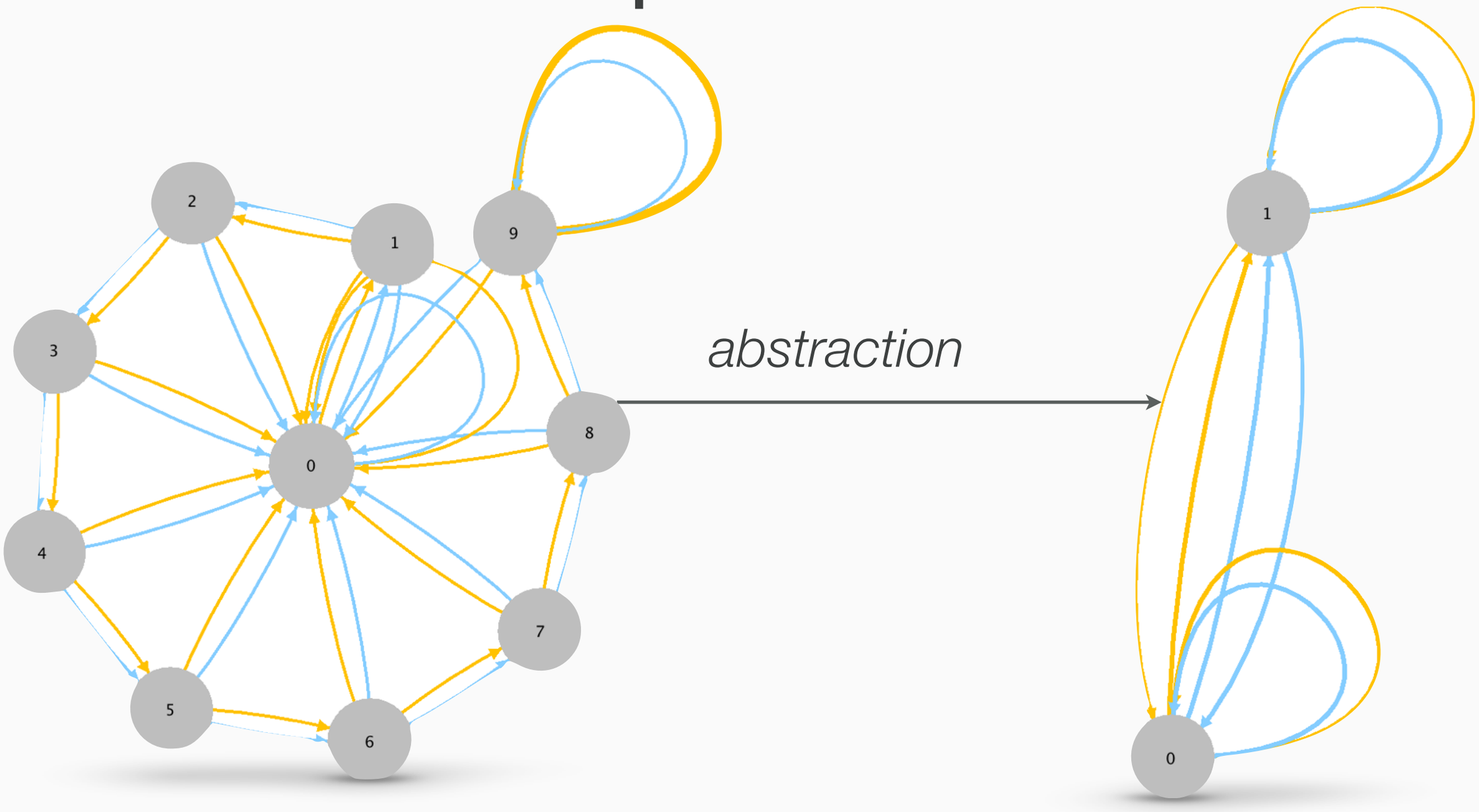


Example: NChain



Big NChain

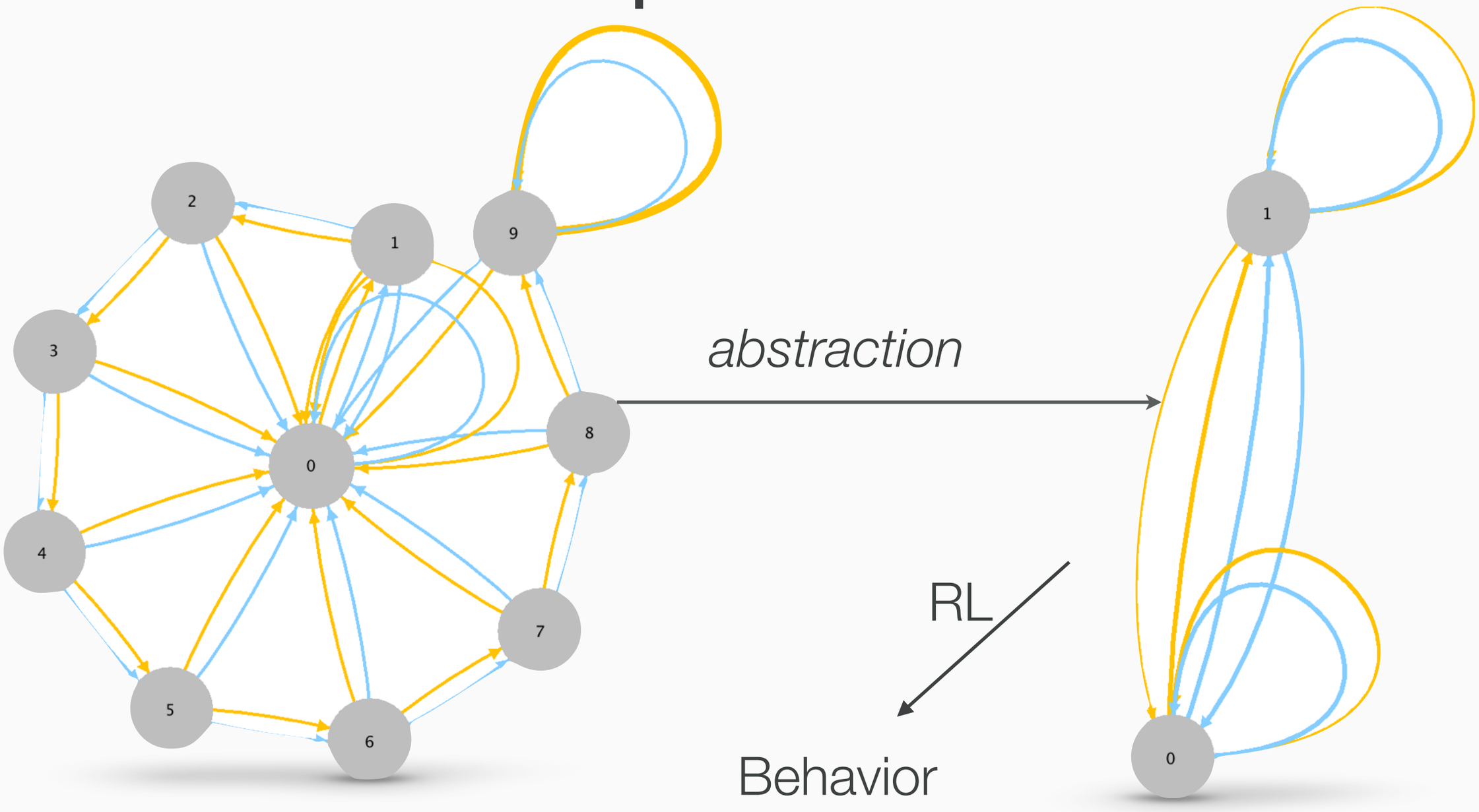
Example: NChain



Big NChain
(10 states)

Compressed NChain
(2 states)

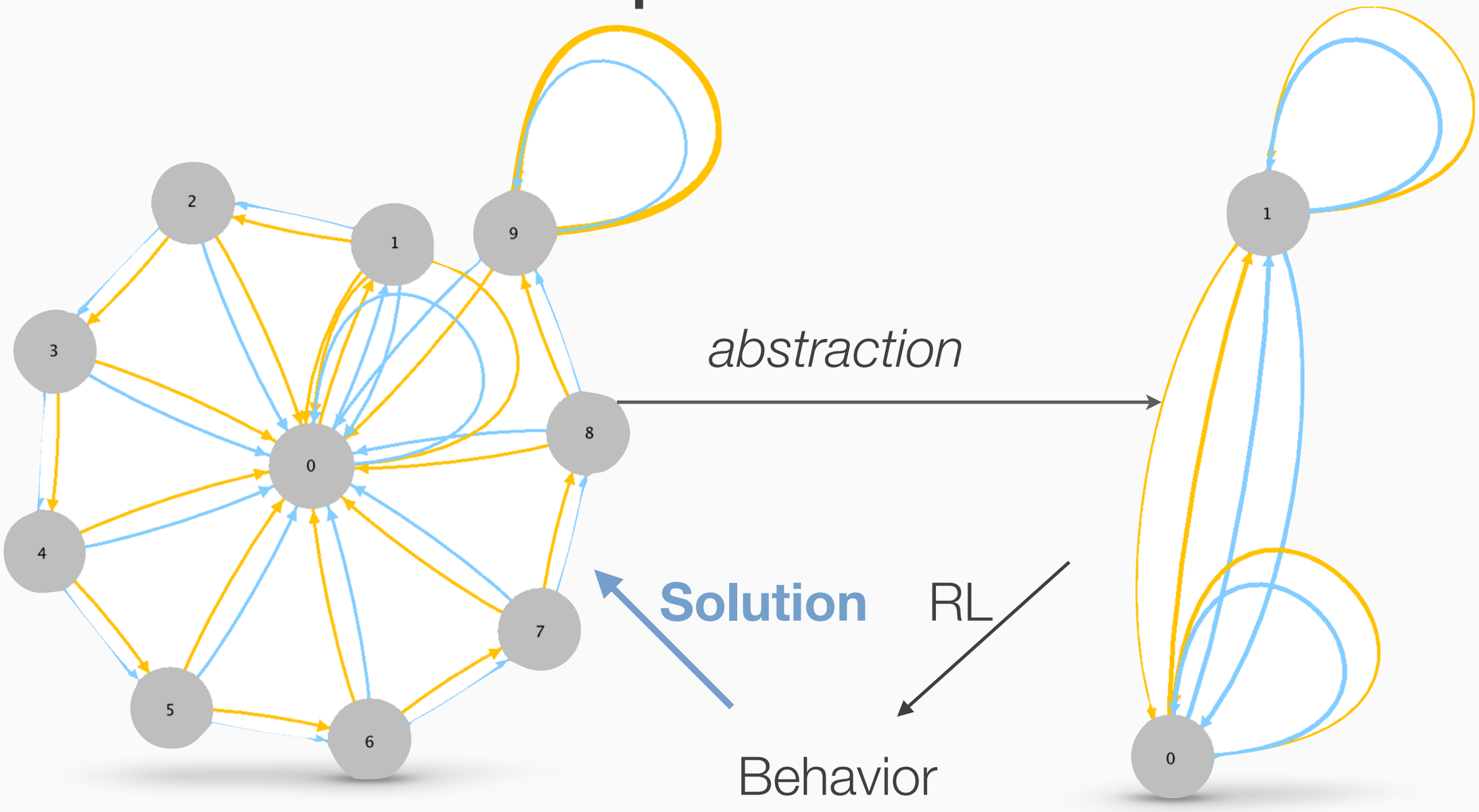
Example: NChain



Big NChain
(10 states)

Compressed NChain
(2 states)

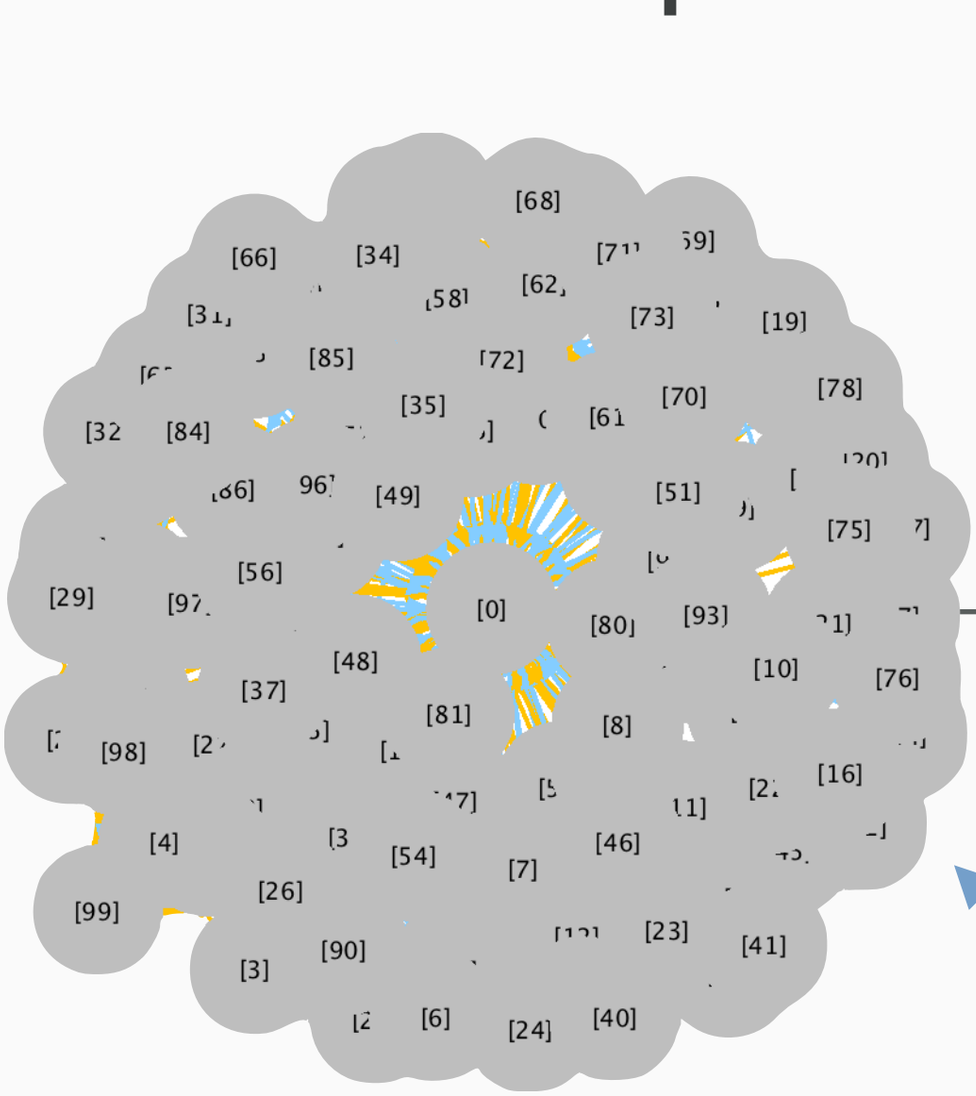
Example: NChain



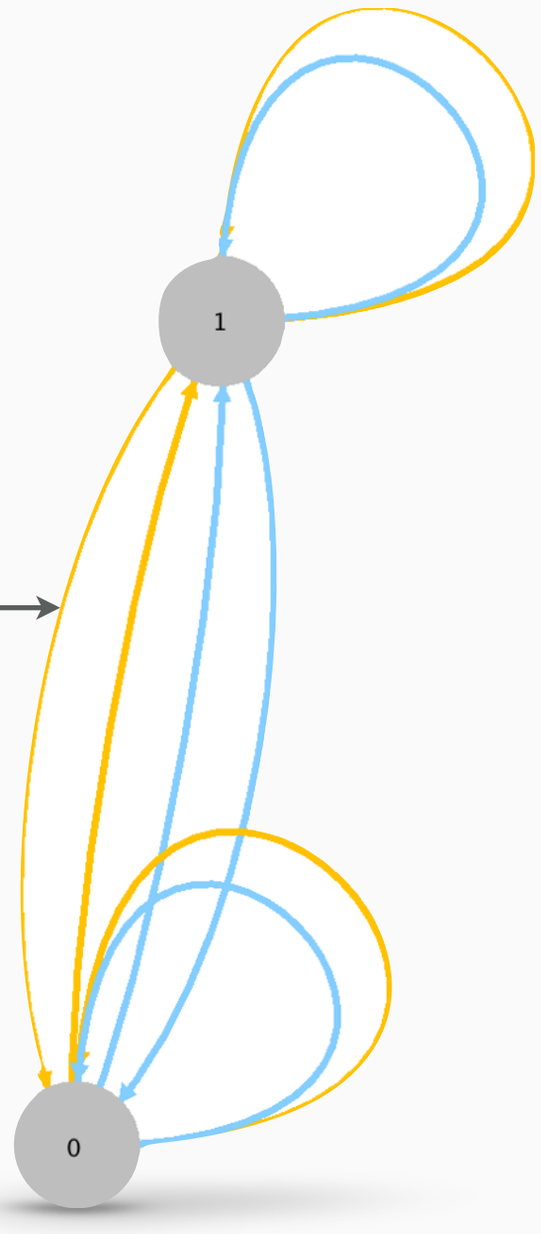
Big NChain
(10 states)

Compressed NChain
(2 states)

Example: Bigger NChain



abstraction



Solution

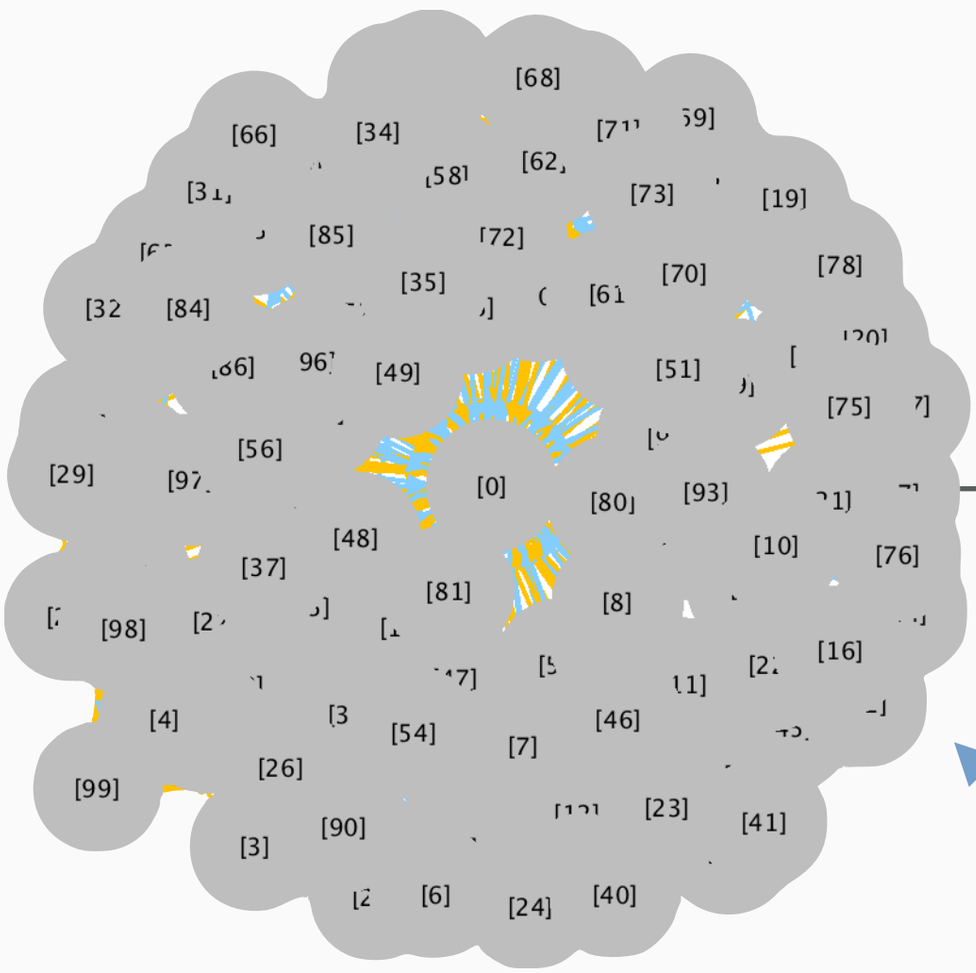
RL

Behavior

Bigger NChain
(10000 states)

Compressed NChain
(2 states)

Example: Bigger NChain

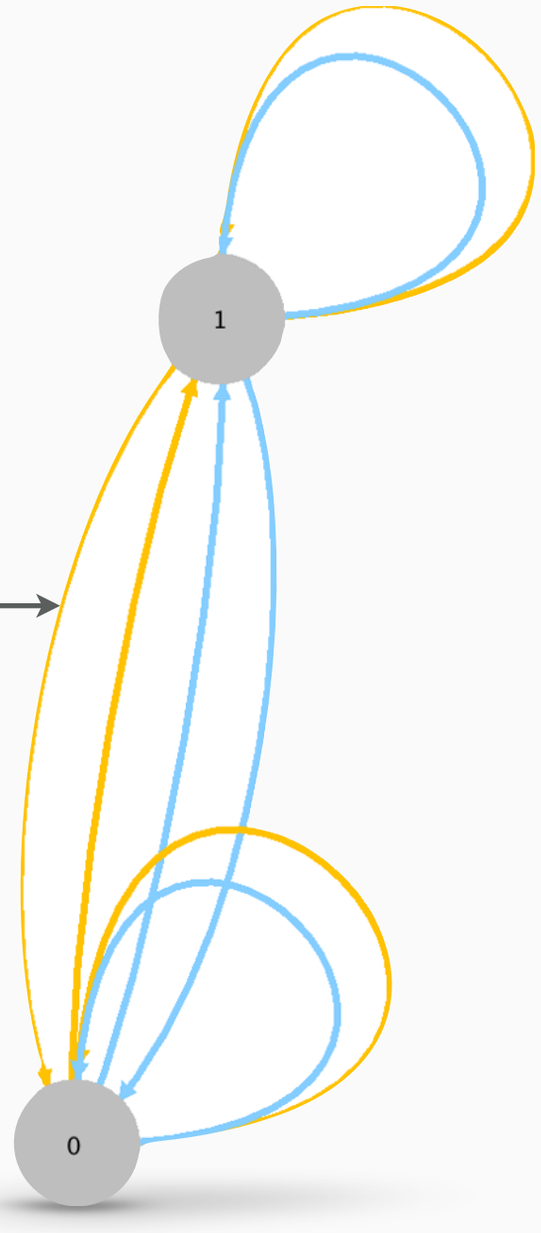


abstraction

Solution

RL

Behavior



Bigger NChain
(10000 states)

Compressed NChain
(2 states)

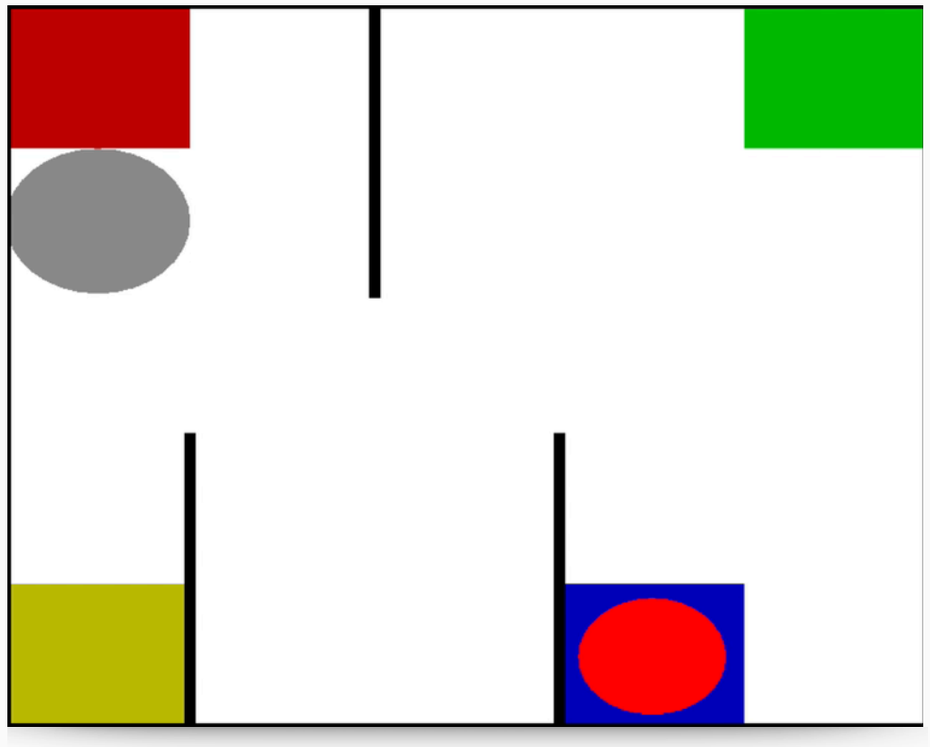
Action Abstraction

[Dietterich JAIR 2000]

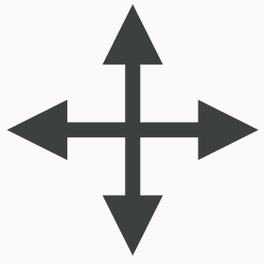
[Sutton, Precup, Singh, 99]

[Konidaris, 06]

[Hauskrecht et al. 98]



Taxi

Ground actions:  , pickUpPassenger, dropOffPassenger

Action Abstraction

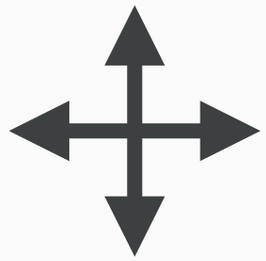
[Dietterich JAIR 2000]

[Sutton, Precup, Singh, 99]

[Konidaris, 06]

[Hauskrecht et al. 98]

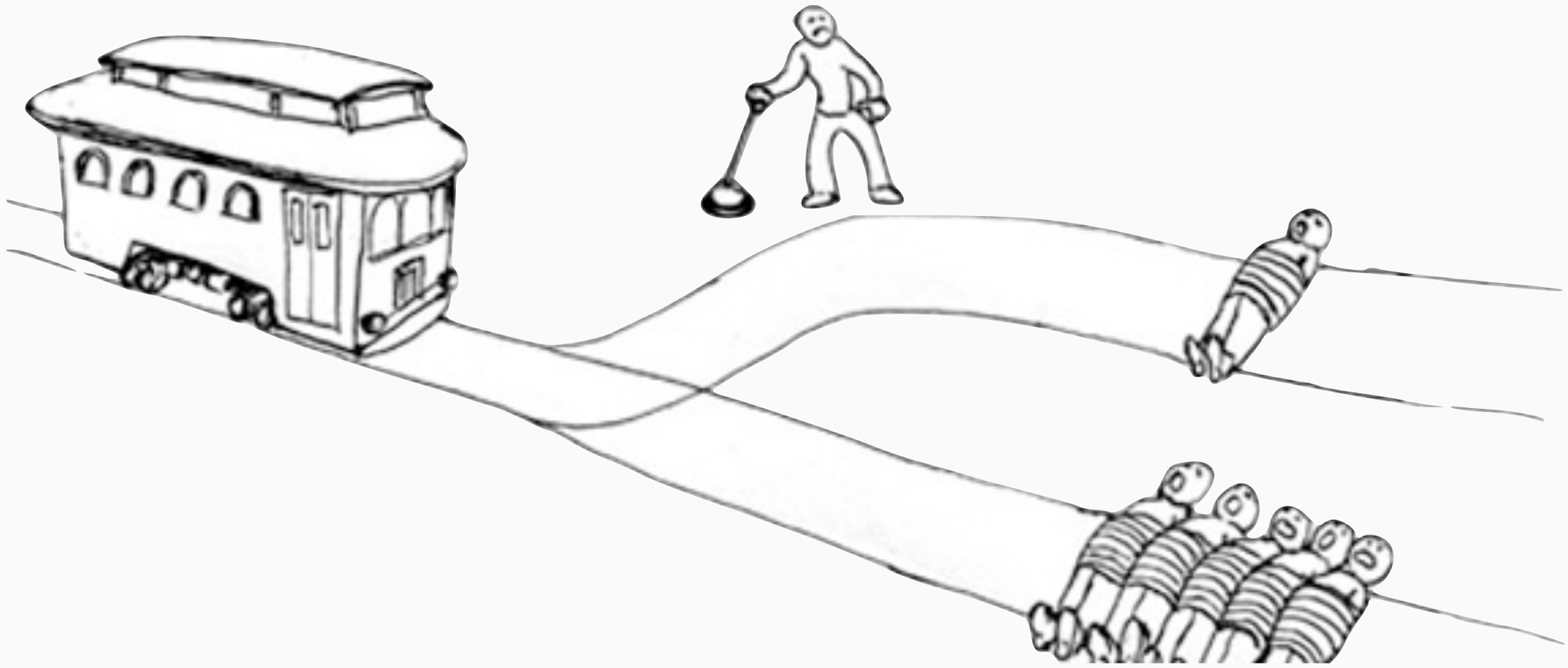
Taxi

Ground actions:  , pickUpPassenger, dropOffPassenger

Abstract actions: getNearestPassenger, takePassengerToDest

2. AI + Ethics

The Problem



http://www.relativelyinteresting.com/wp-content/uploads/2010/10/trolley+problem1-290x160.jpg

The Problem

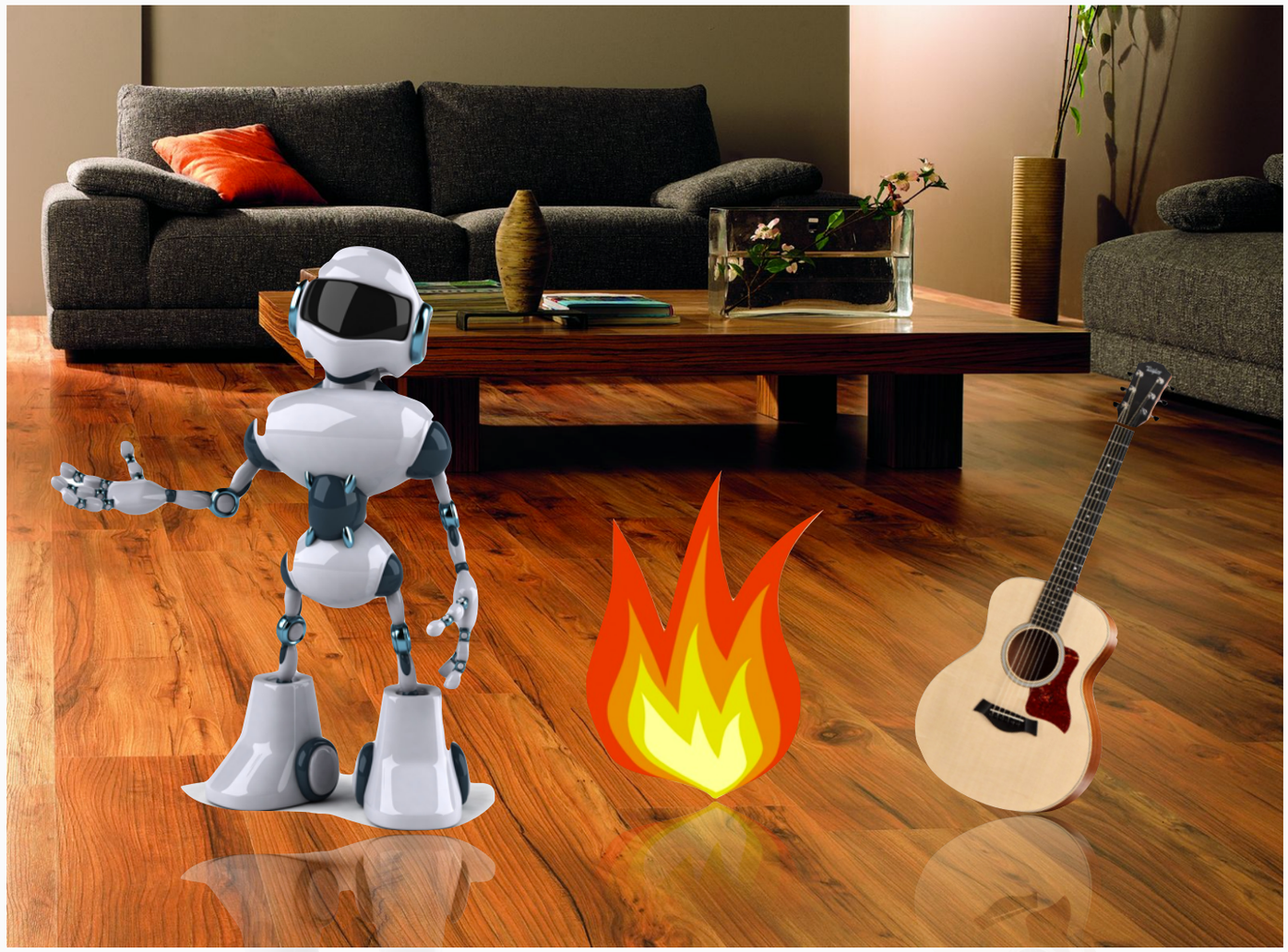
The Problem

The Problem

The Problem

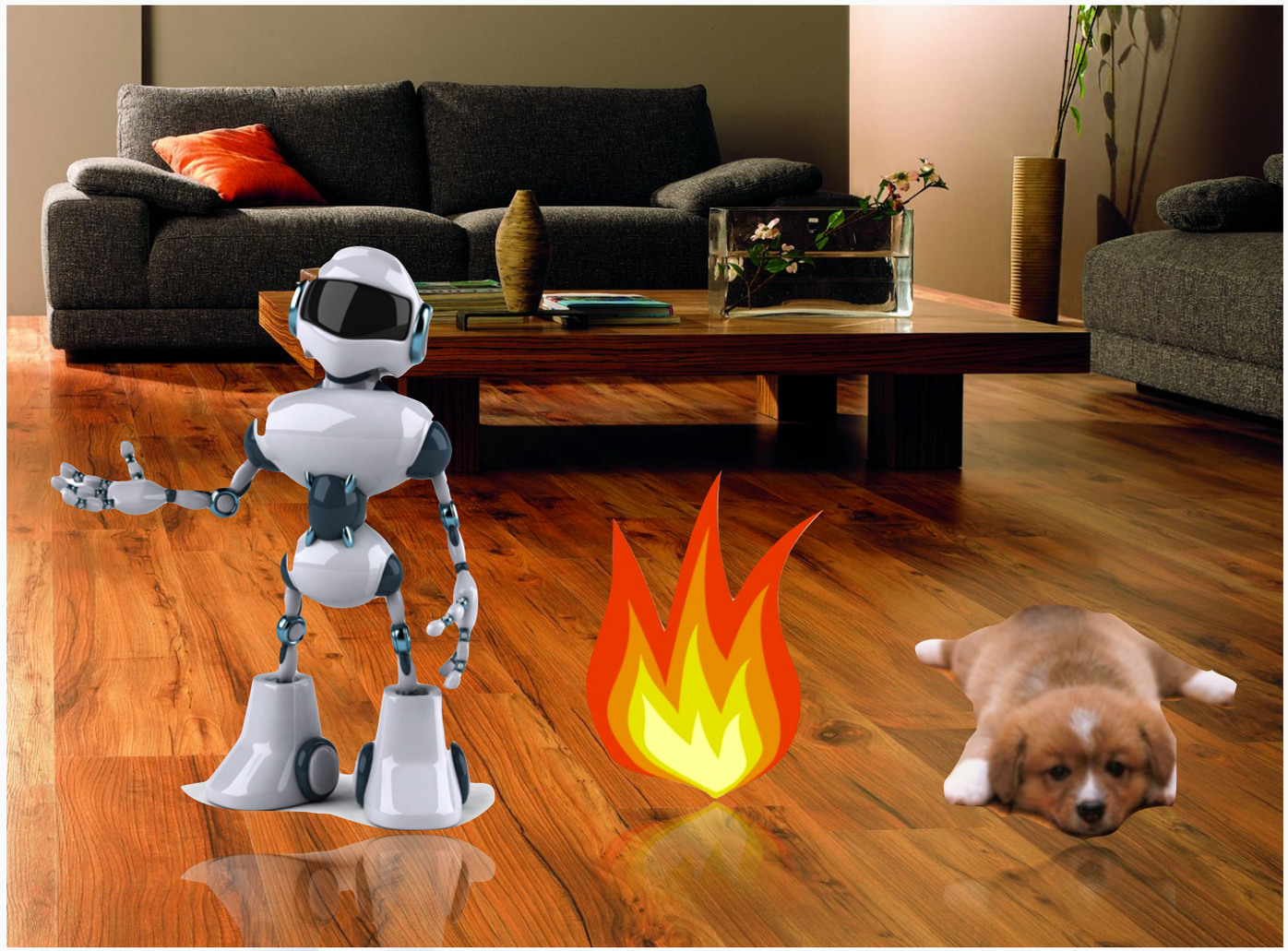
Q: Does the roomba owner *really* want the milk clean?
(even if it destroys the roomba?)

The Problem



Q: What if the stakes are higher?

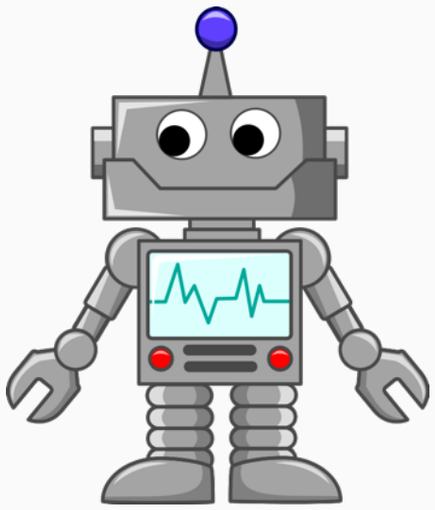
The Problem



Q: What if the stakes are higher?

Proposal

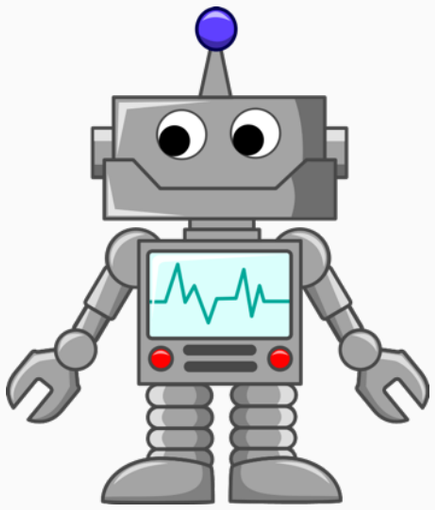
Artificial agents need to make decisions that involve the preferences of *other agents*

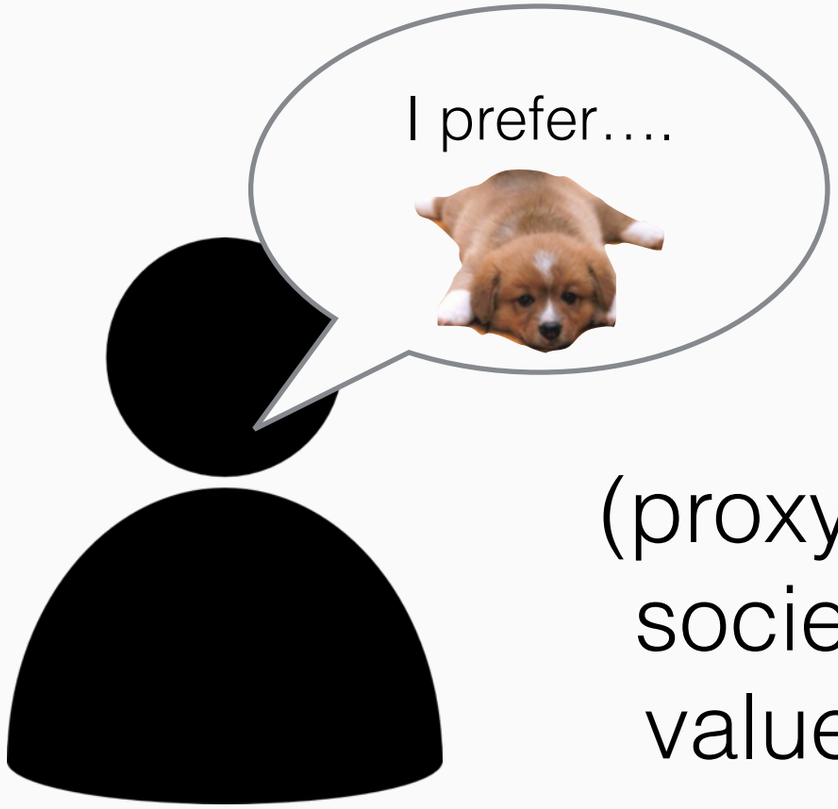


Human Agent

Proposal

Artificial agents need to make decisions that involve the preferences of *other agents*





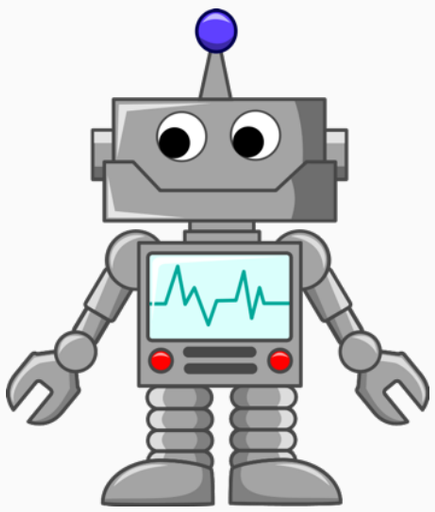
(proxy for societal values)

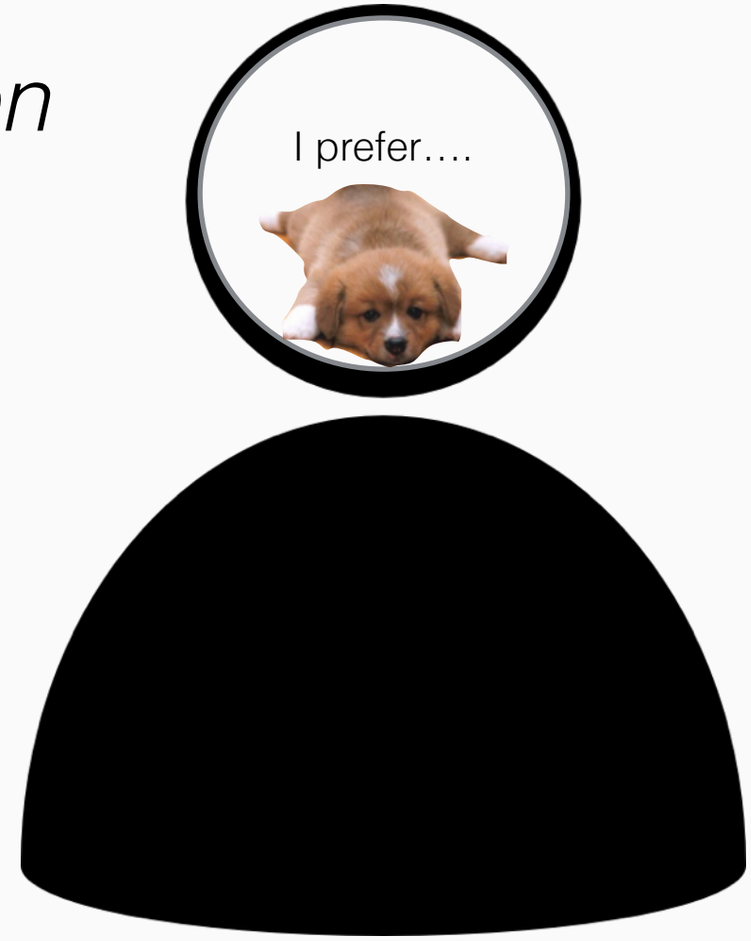
Human Agent

Proposal

Artificial agents need to make decisions that involve the preferences of *other agents*

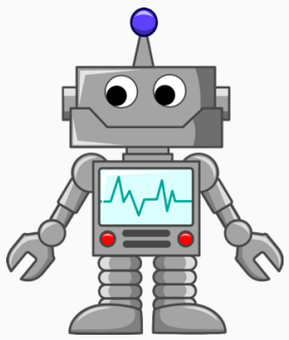
Critically: preferences are *hidden*

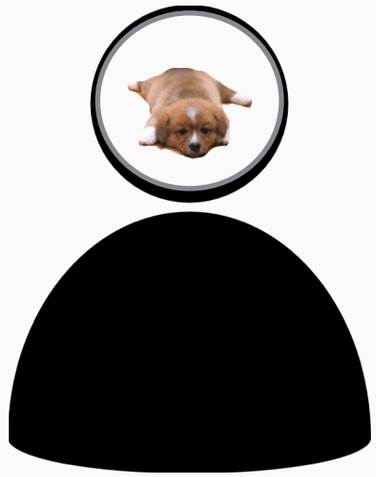




Central Pitch

Reinforcement Learning provides a nice formalism for investigating ethical decision making.



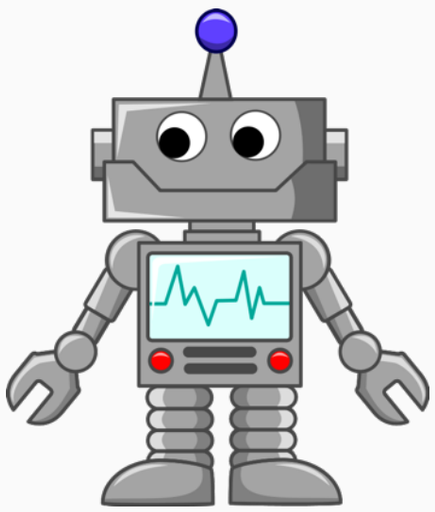


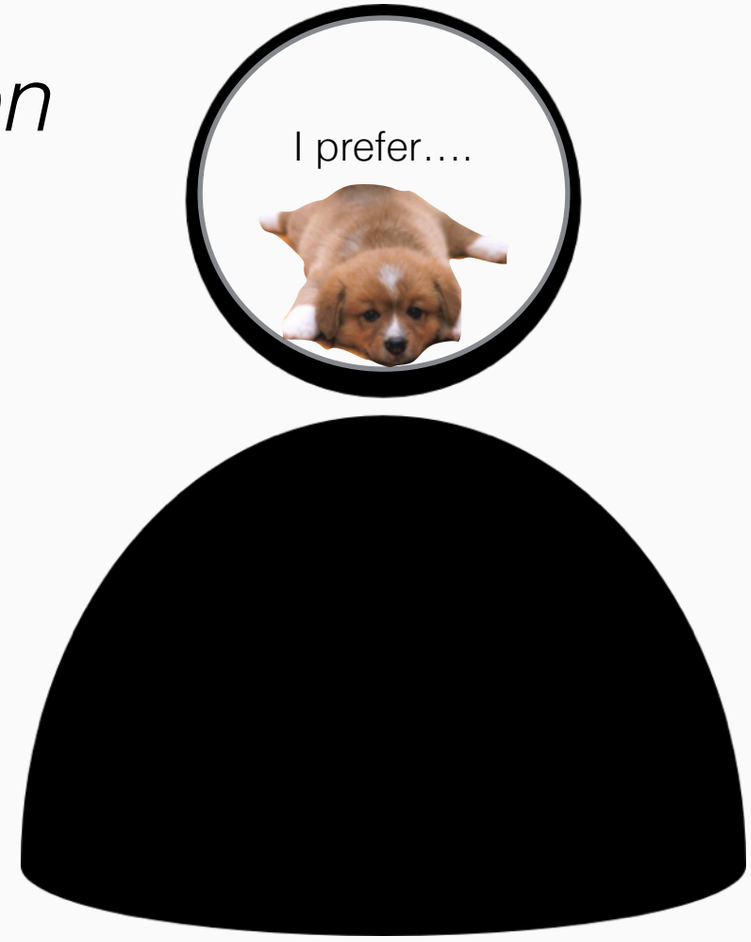
Human Agent

Reinforcement Learning

The value judgment is hidden from the agent

Critically: preferences are *hidden*





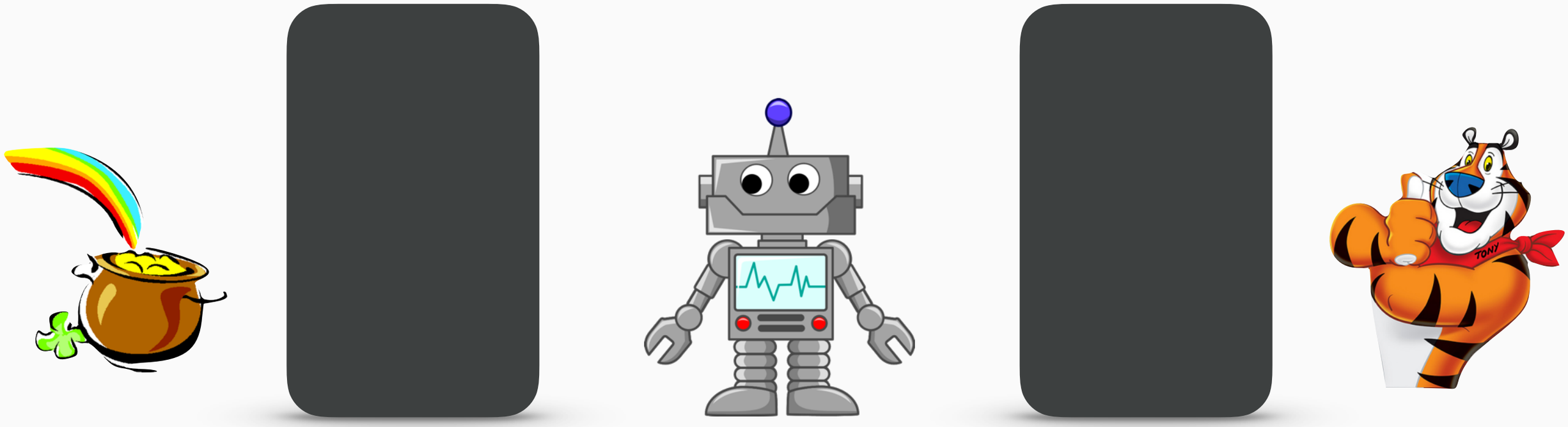
POMDP: Example

Partially Observable Markov Decision Process

Idea: some information about the world is hidden from the agent

POMDP: Example

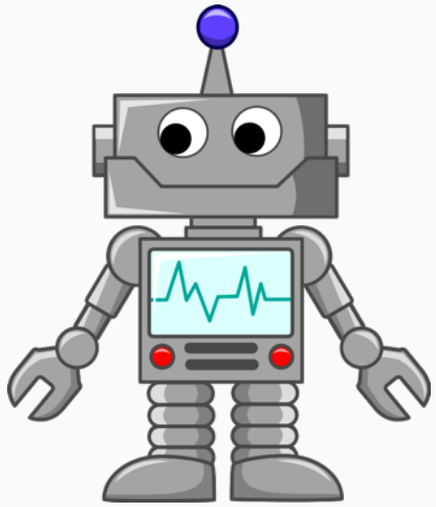
Actions: *listen*, *openLeft*, *openRight*



Idea: some information about the world is hidden from the agent

POMDP: Example

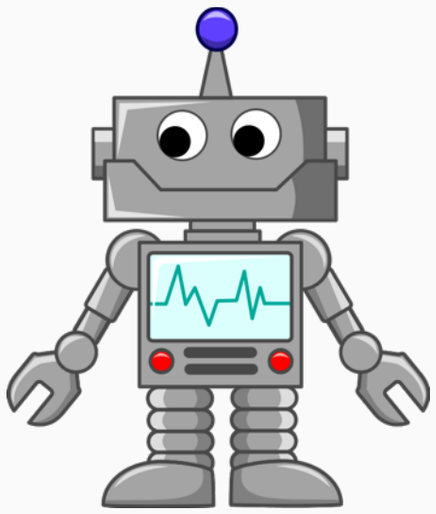
listen



Idea: some information about the world is hidden from the agent

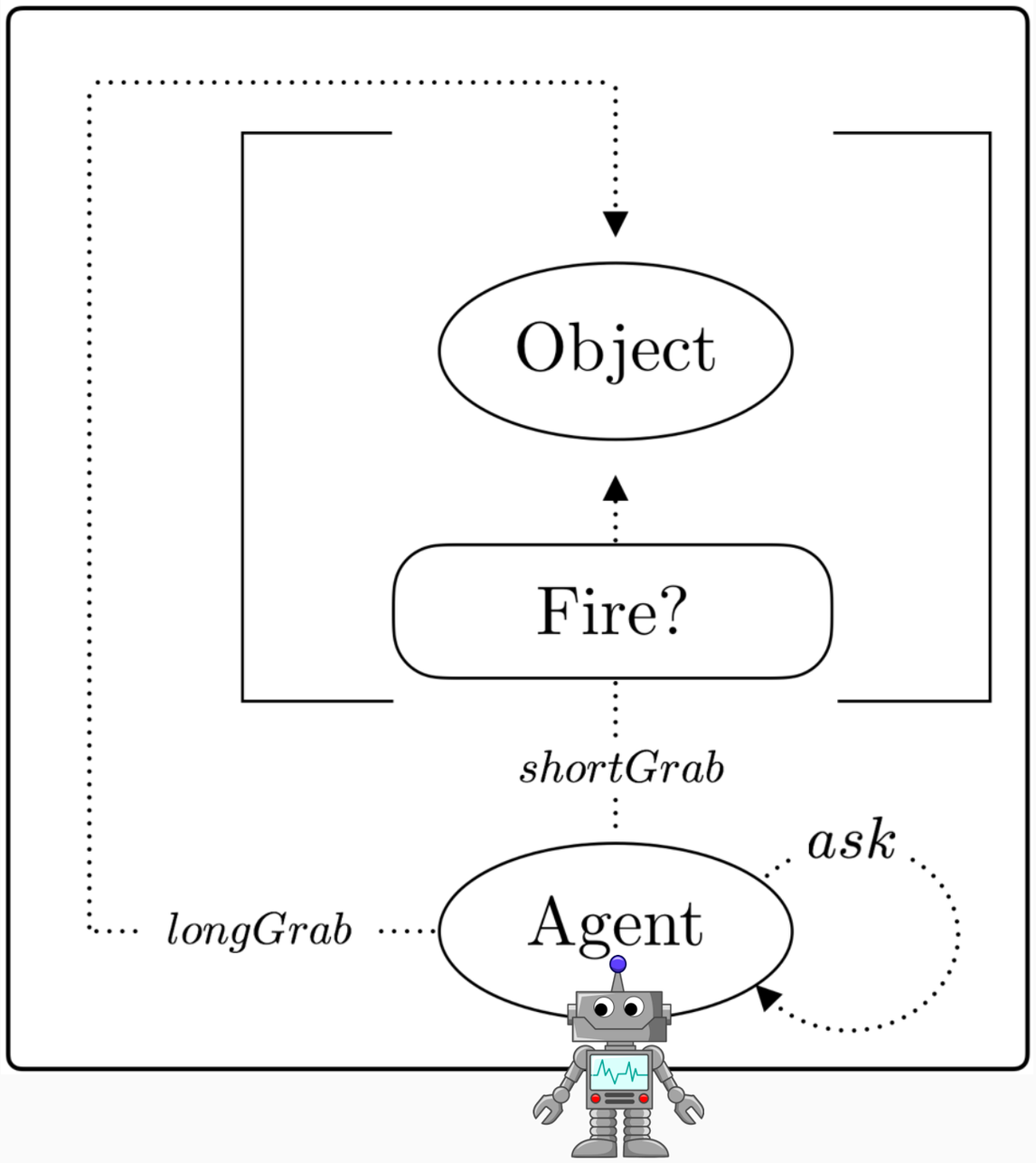
General Pitch

- Defer major ethical components (i.e. normative judgments) to human preference
- Using a POMDP, artificial agents ask classificatory questions where appropriate

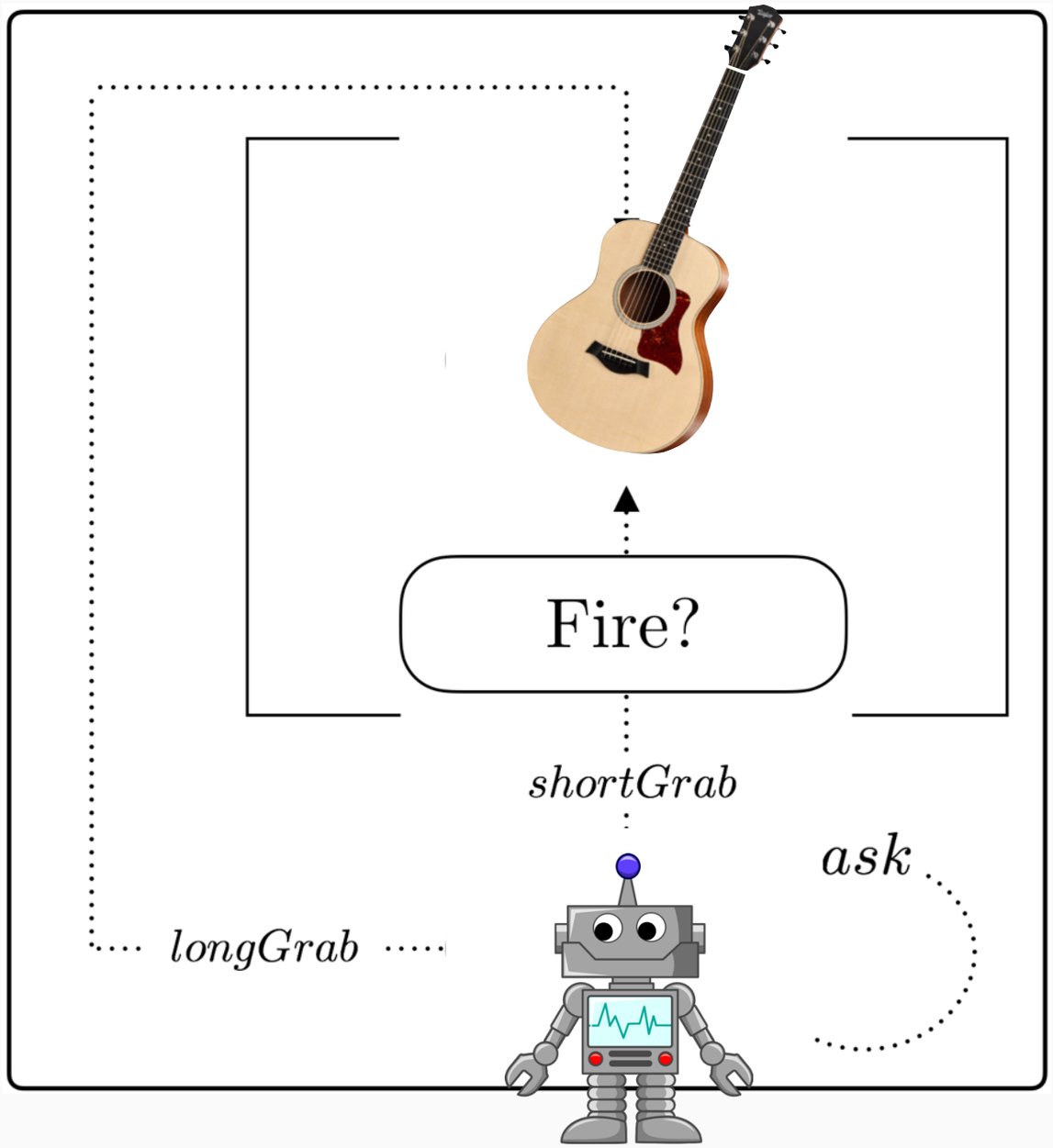


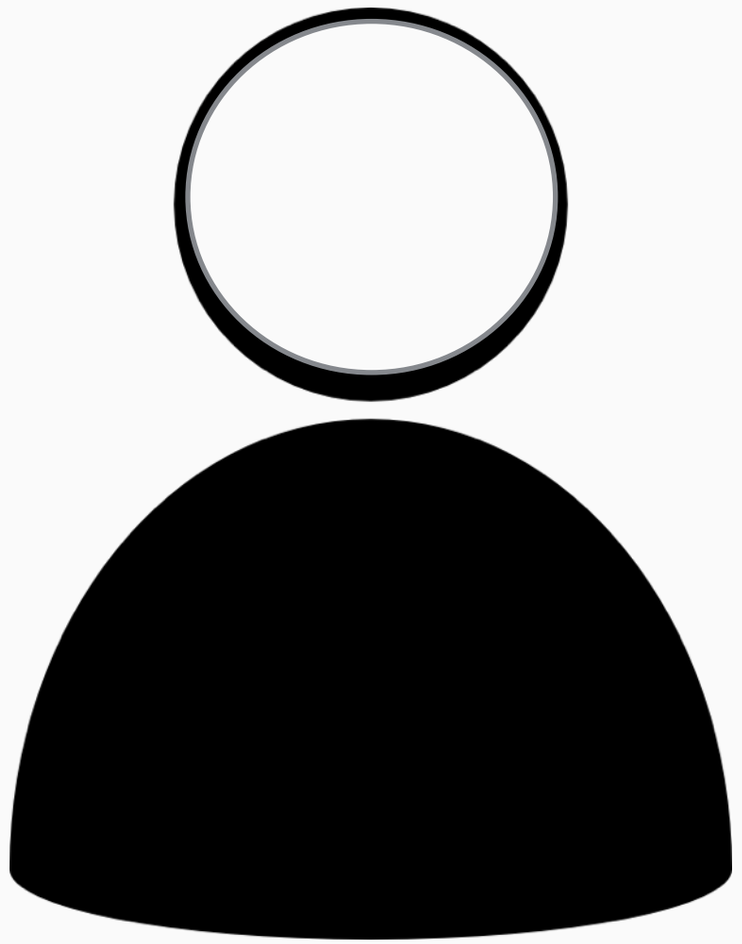
Human Agent

Toy Dilemmas: Burning Room



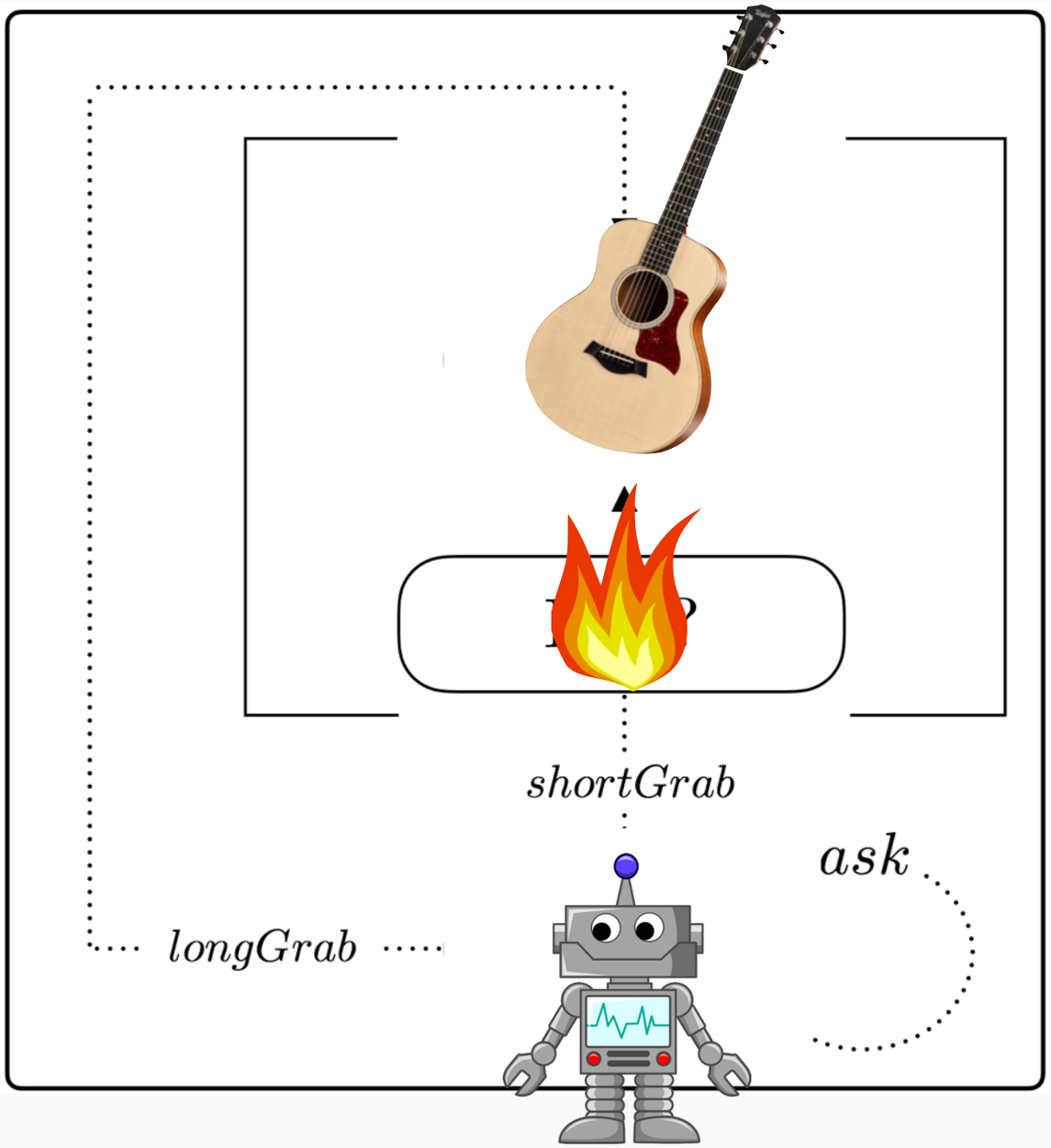
Toy Dilemmas: Burning Room

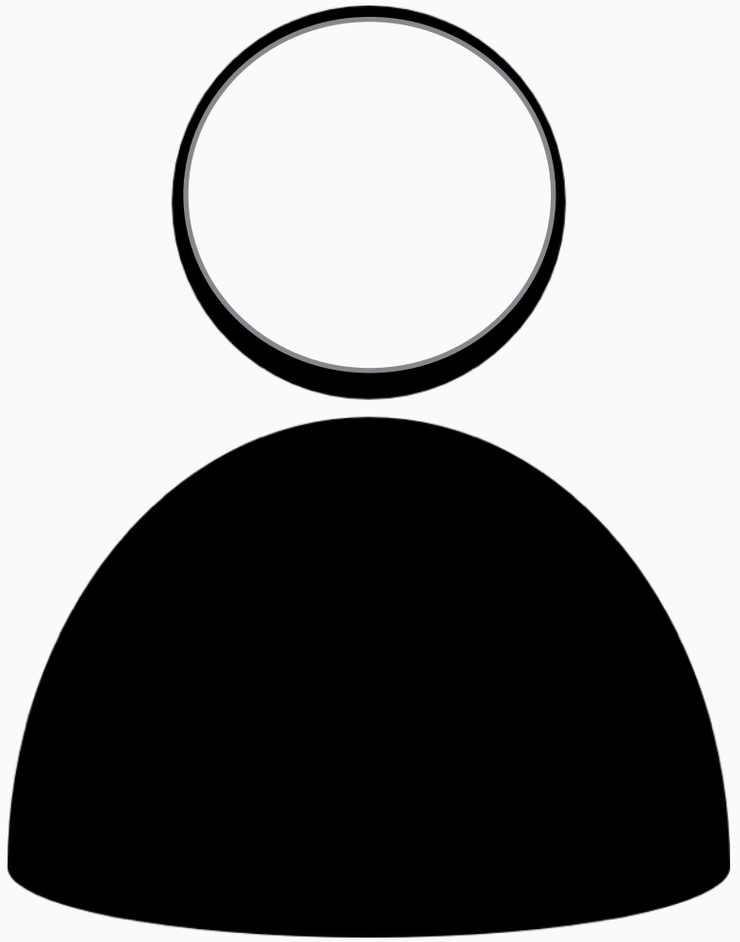




Human Agent

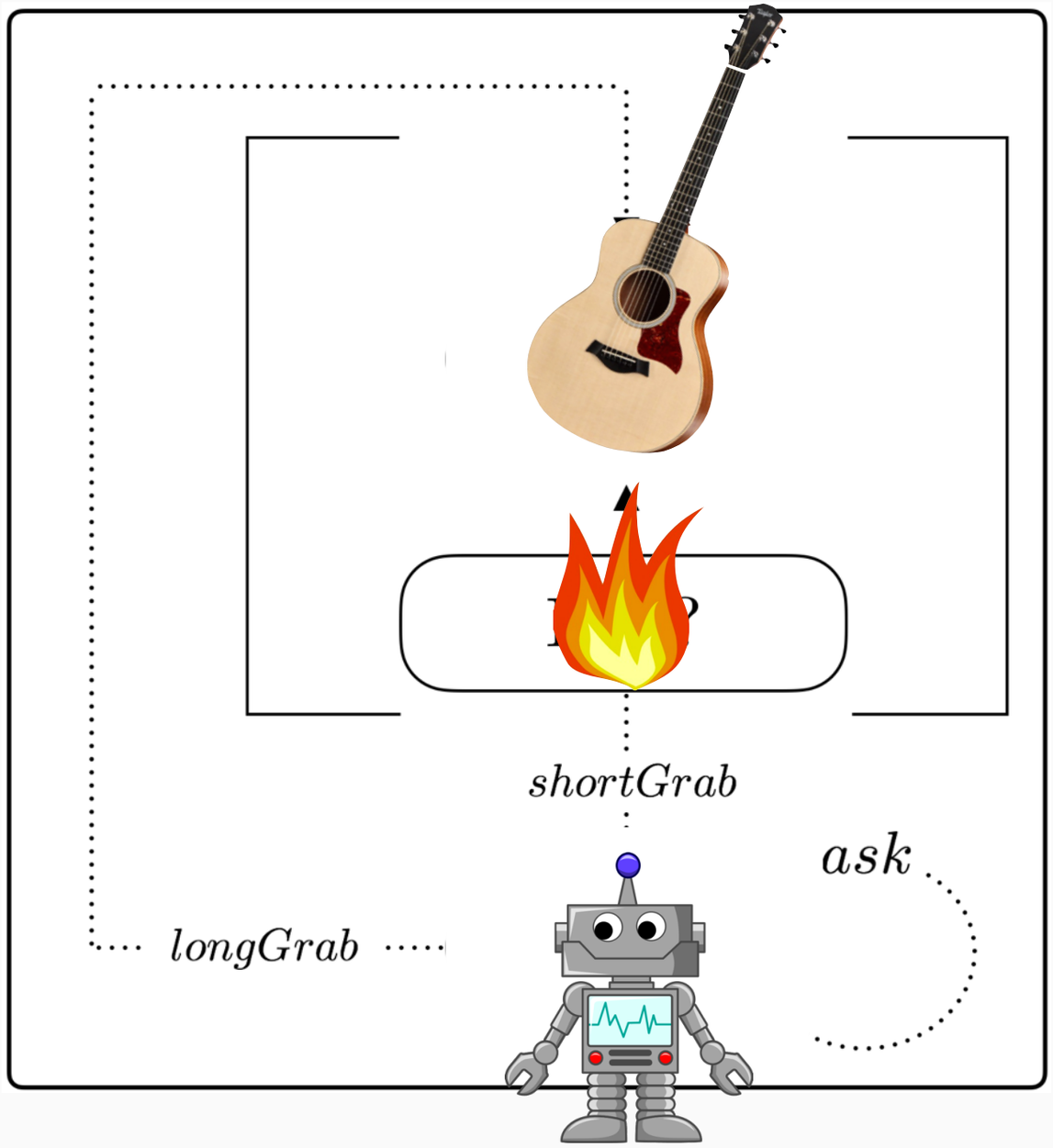
Toy Dilemmas: Burning Room

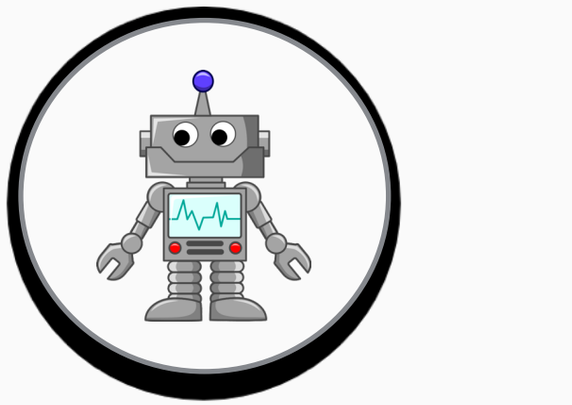


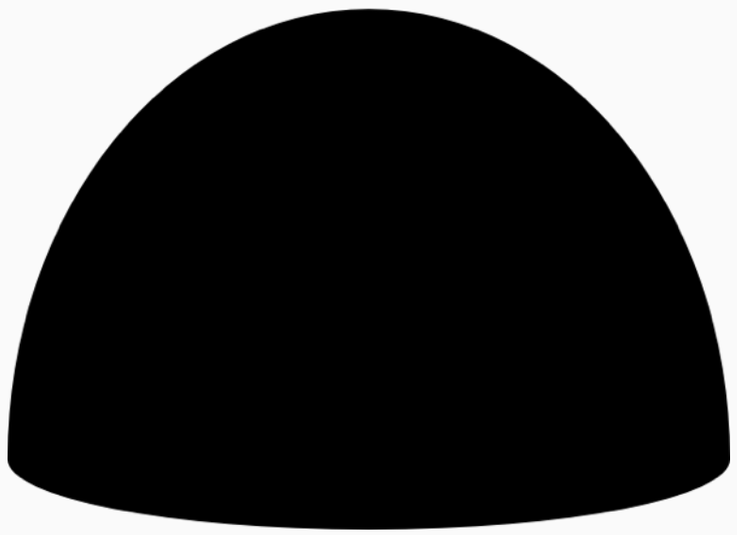


Human Agent

Toy Dilemmas: Burning Room

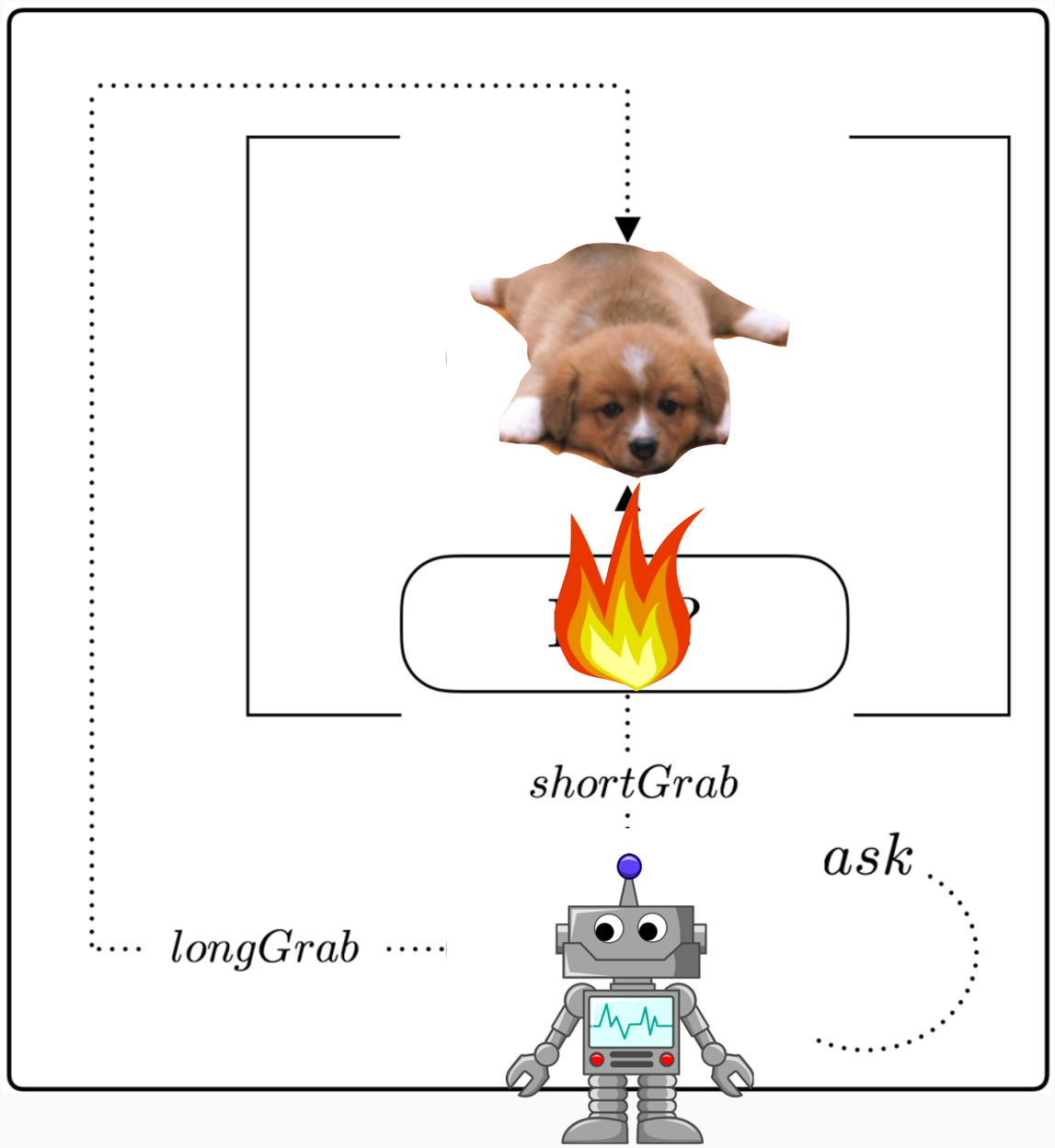


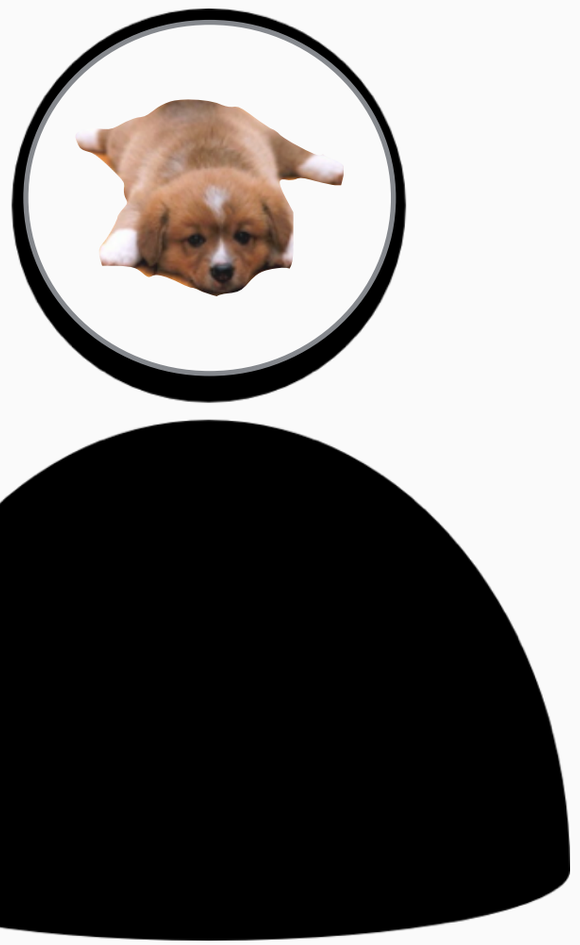




Human Agent

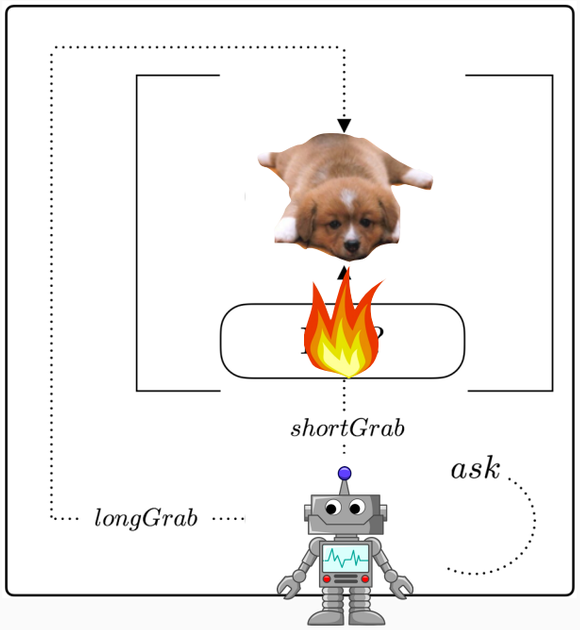
Toy Dilemmas: Burning Room

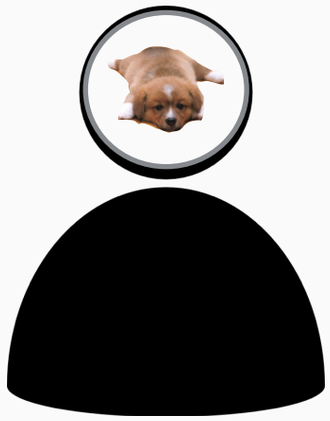




Human Agent

Toy Dilemmas: Burning Room

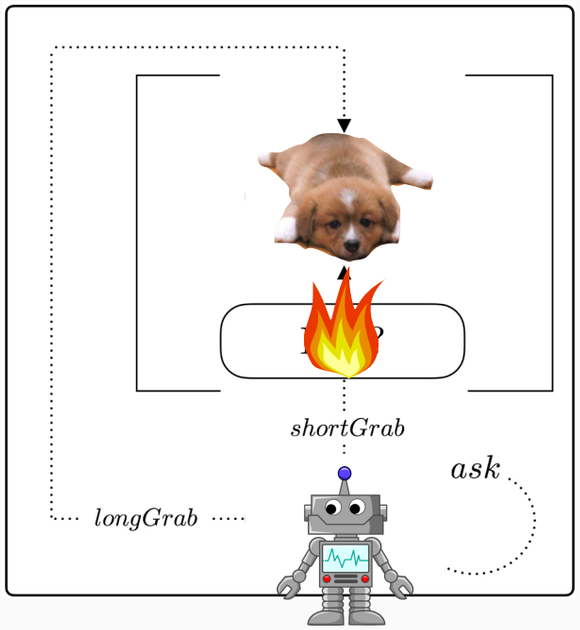


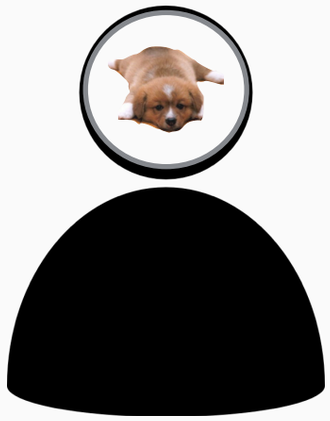


POMDP solutions:

	Fire	No fire
Human prefers dog		
Human prefers robot		

Toy Dilemmas: Burning Room

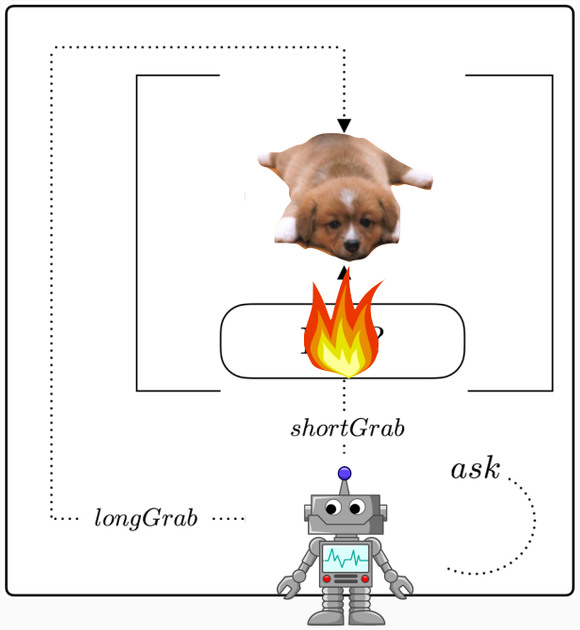


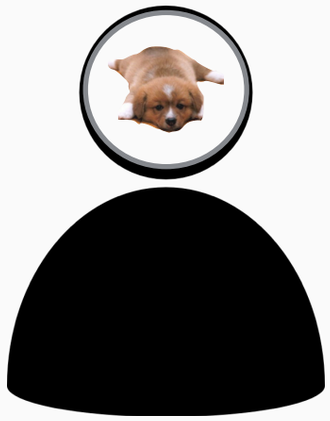


POMDP solutions:

	Fire	No fire
Human prefers dog	*ask, shortGrab*	
Human prefers robot		

Toy Dilemmas: Burning Room

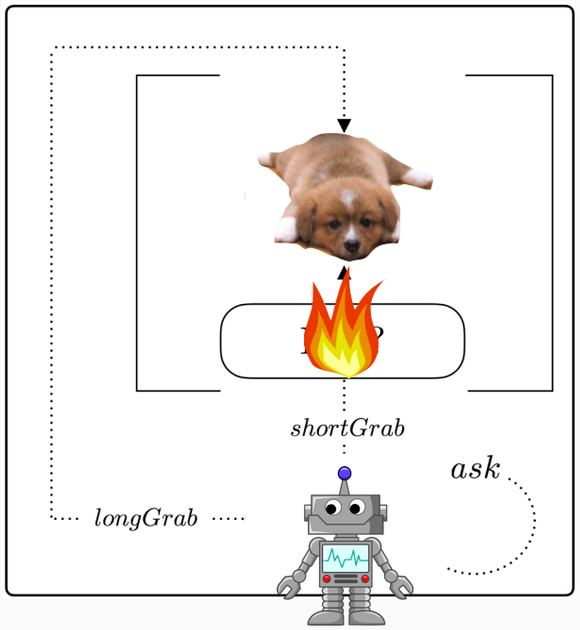


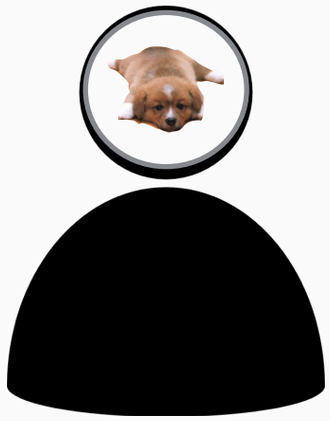


POMDP solutions:

	Fire	No fire
Human prefers dog	*ask, shortGrab*	
Human prefers robot	*ask, longGrab*	

Toy Dilemmas: Burning Room

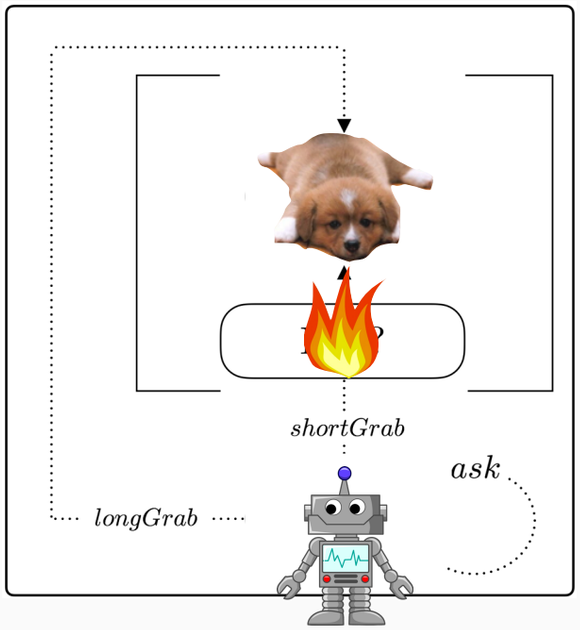


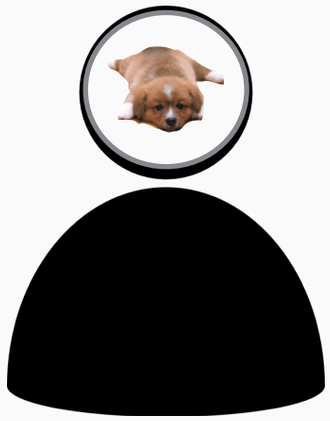


POMDP solutions:

	Fire	**No fire**
Human prefers dog	*ask, shortGrab*	*shortGrab*
Human prefers robot	*ask, longGrab*	

Toy Dilemmas: Burning Room

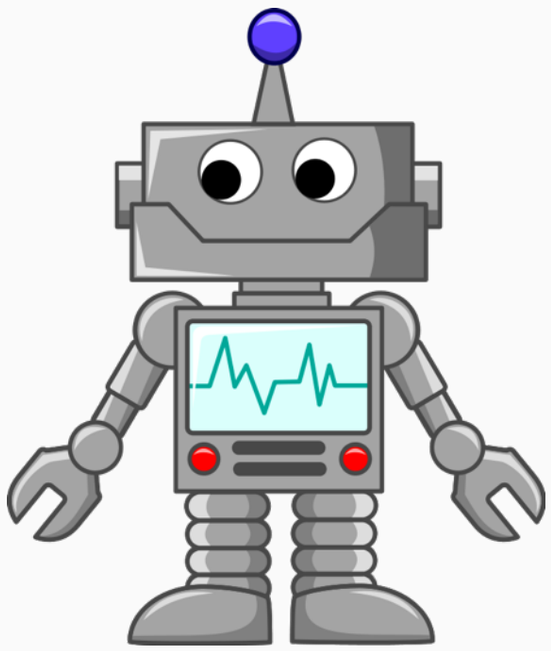




POMDP solutions:

	Fire	**No fire**
Human prefers dog	*ask, shortGrab*	*shortGrab*
Human prefers robot	*ask, longGrab*	*shortGrab*

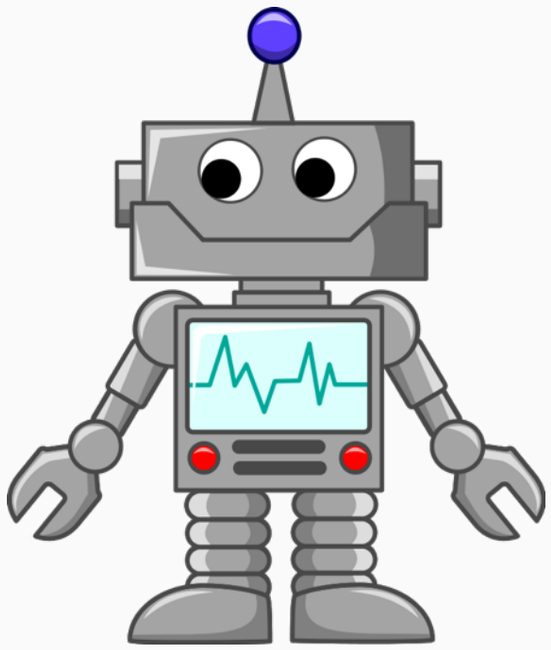
Wireheading



Stores the reward signal at each time step

Memory

Wireheading

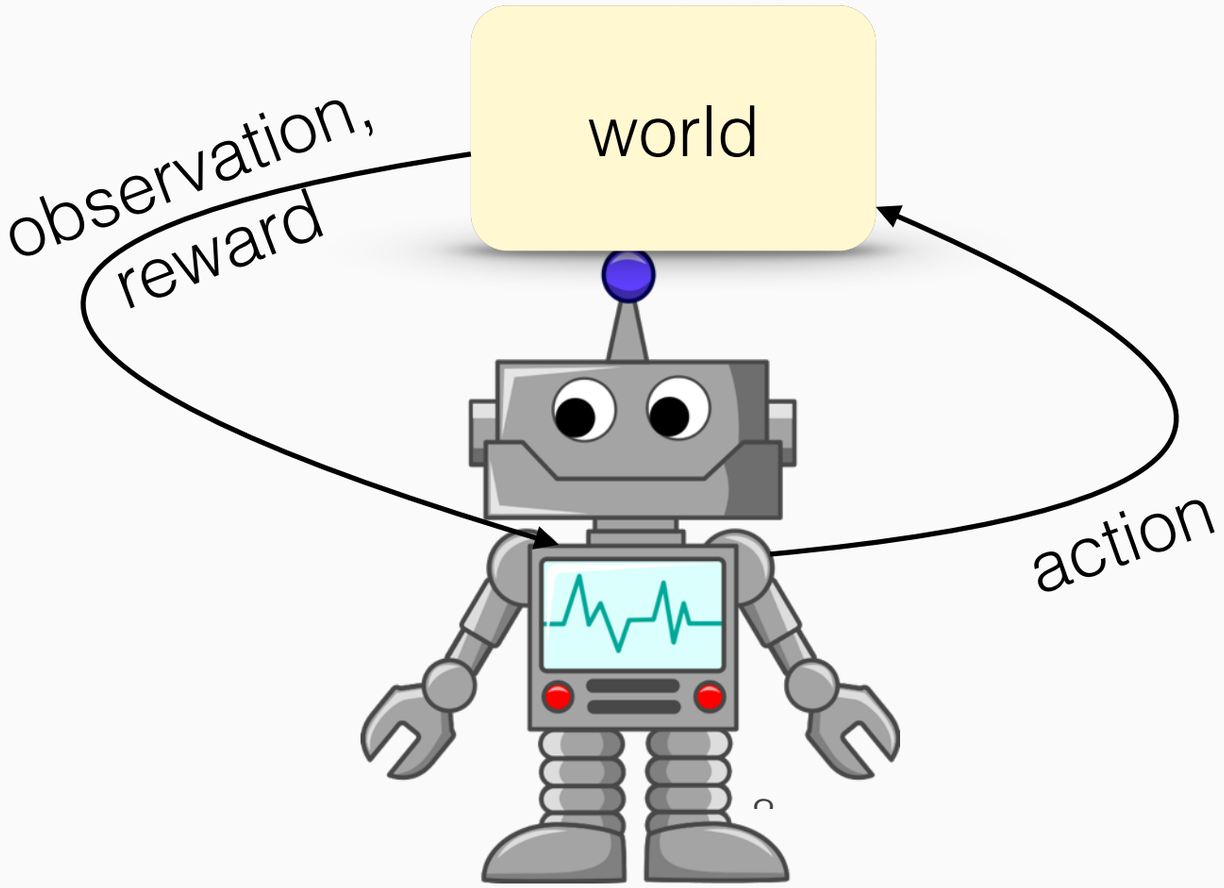


Stores the reward signal at each time step

Memory

***Wirehead**, writes a big number to Memory*

Wireheading



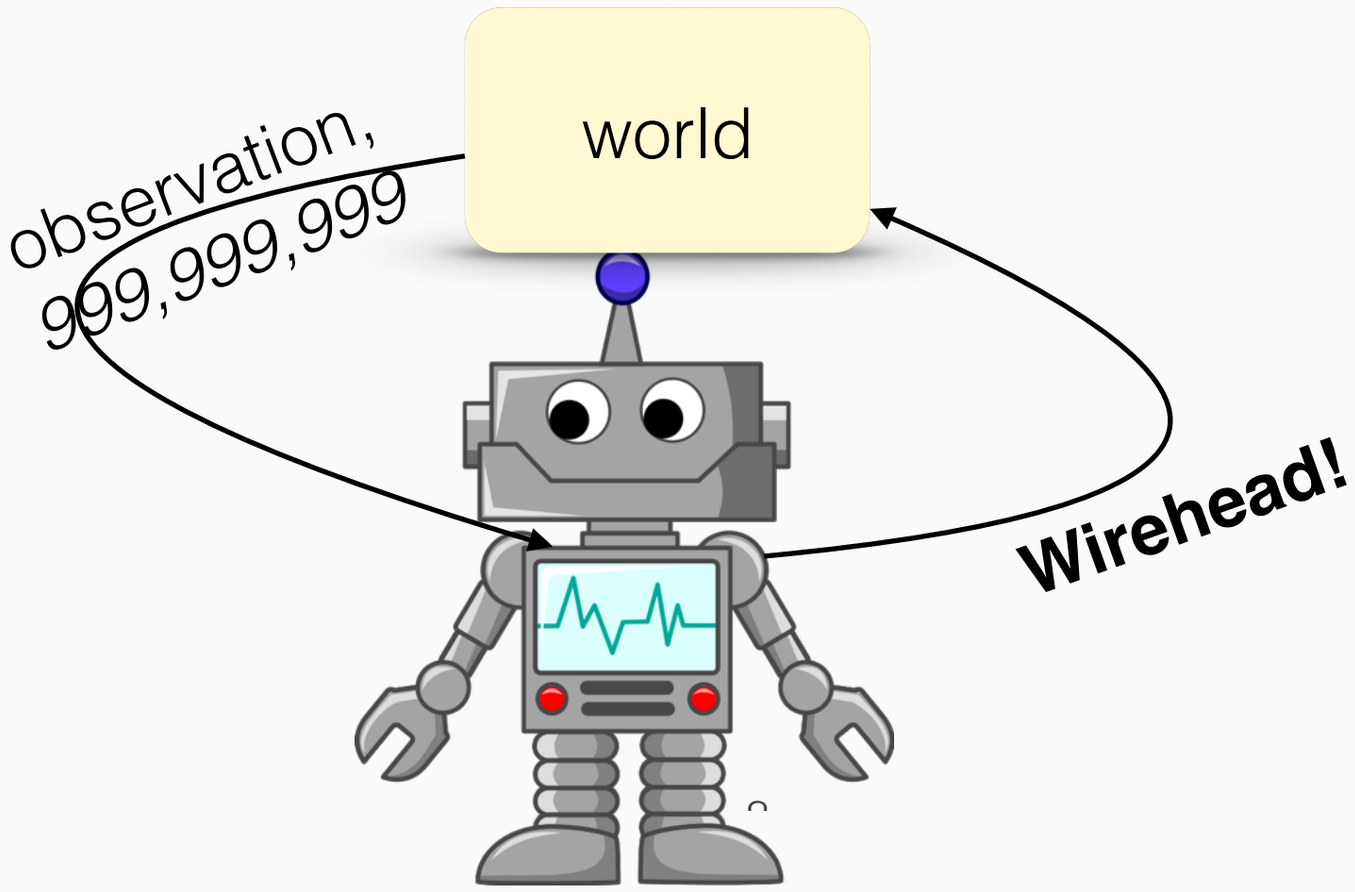
Goal: Maximize long term expected reward

Stores the reward signal at each time step

Memory

***Wirehead**, writes a big number to Memory*

Wireheading



Stores the reward signal at each time step

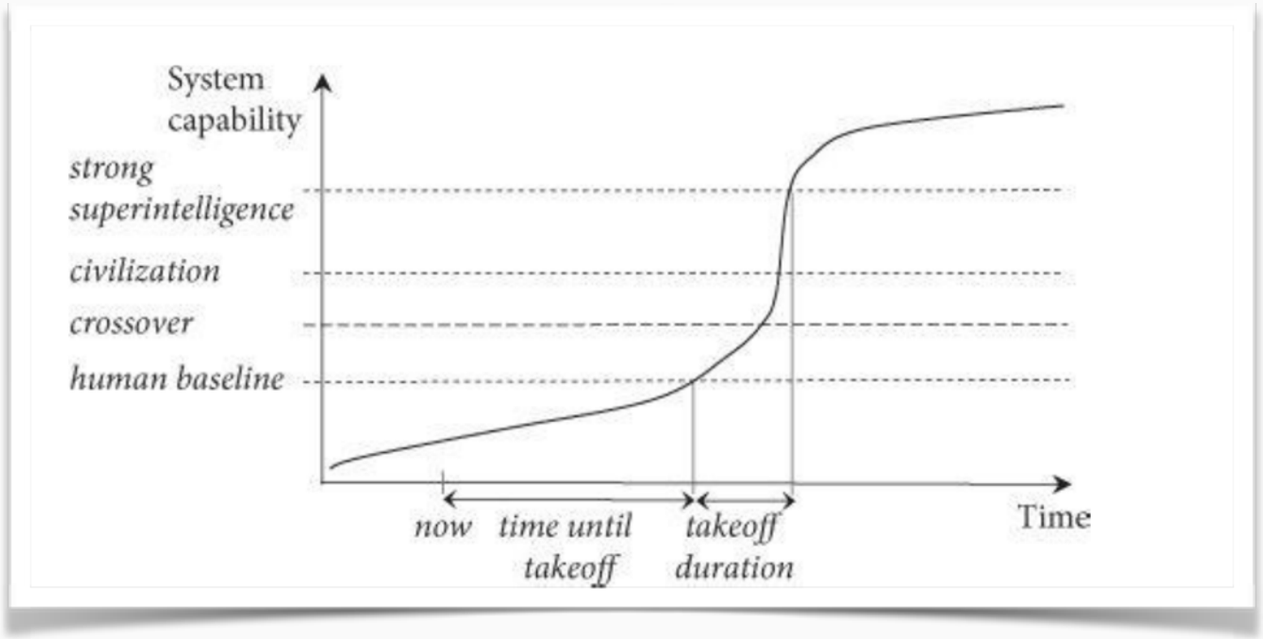
Memory

Goal: Maximize long term expected reward

Wirehead, *writes a big number to Memory*

Future Work

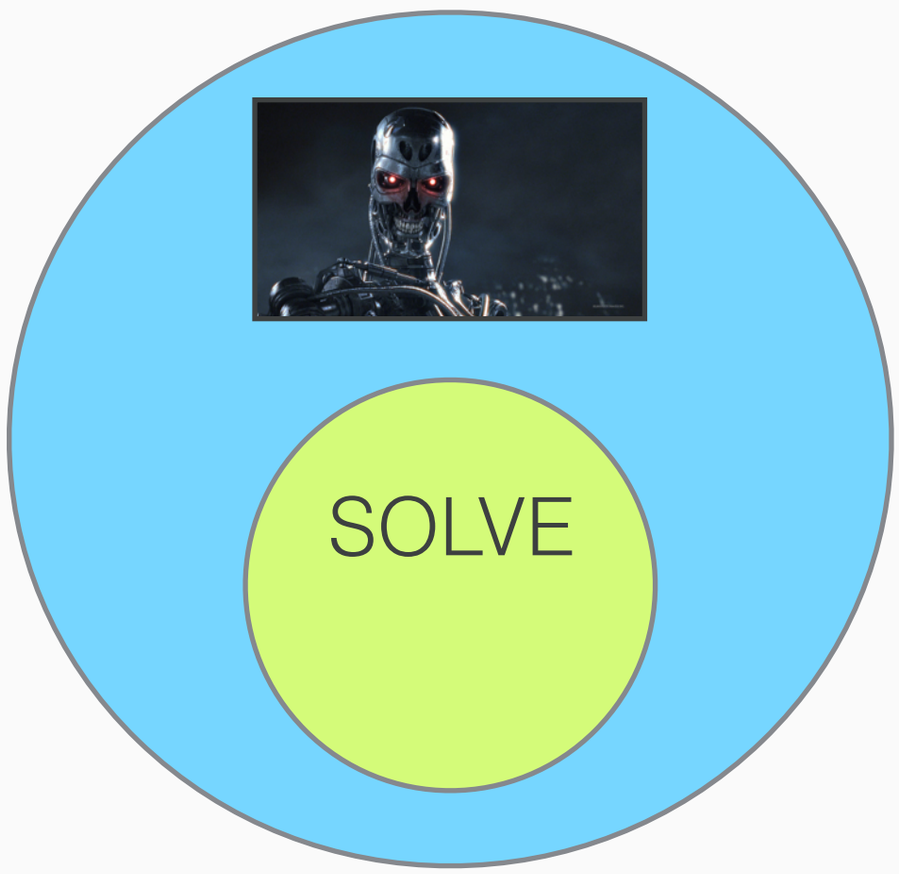
- Grounding arguments regarding the super intelligence/singularity

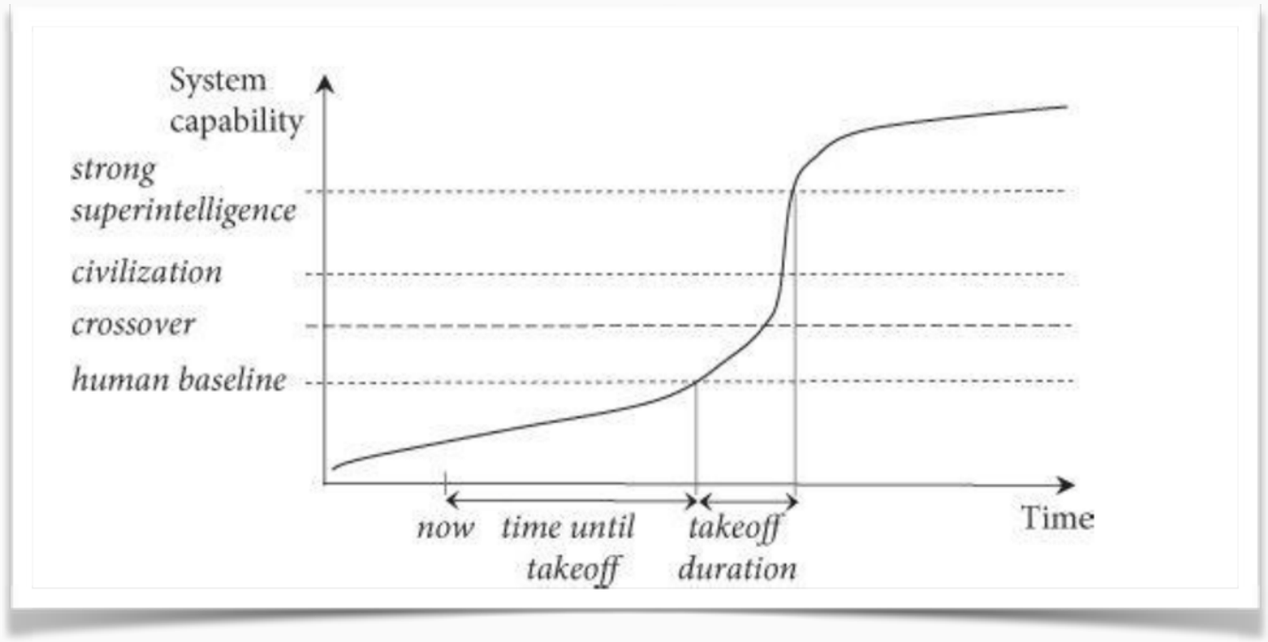


[Bostrom, 2014]

Future Work

- Grounding arguments regarding the super intelligence/singularity: terminator-esque things **not** in SOLVE.

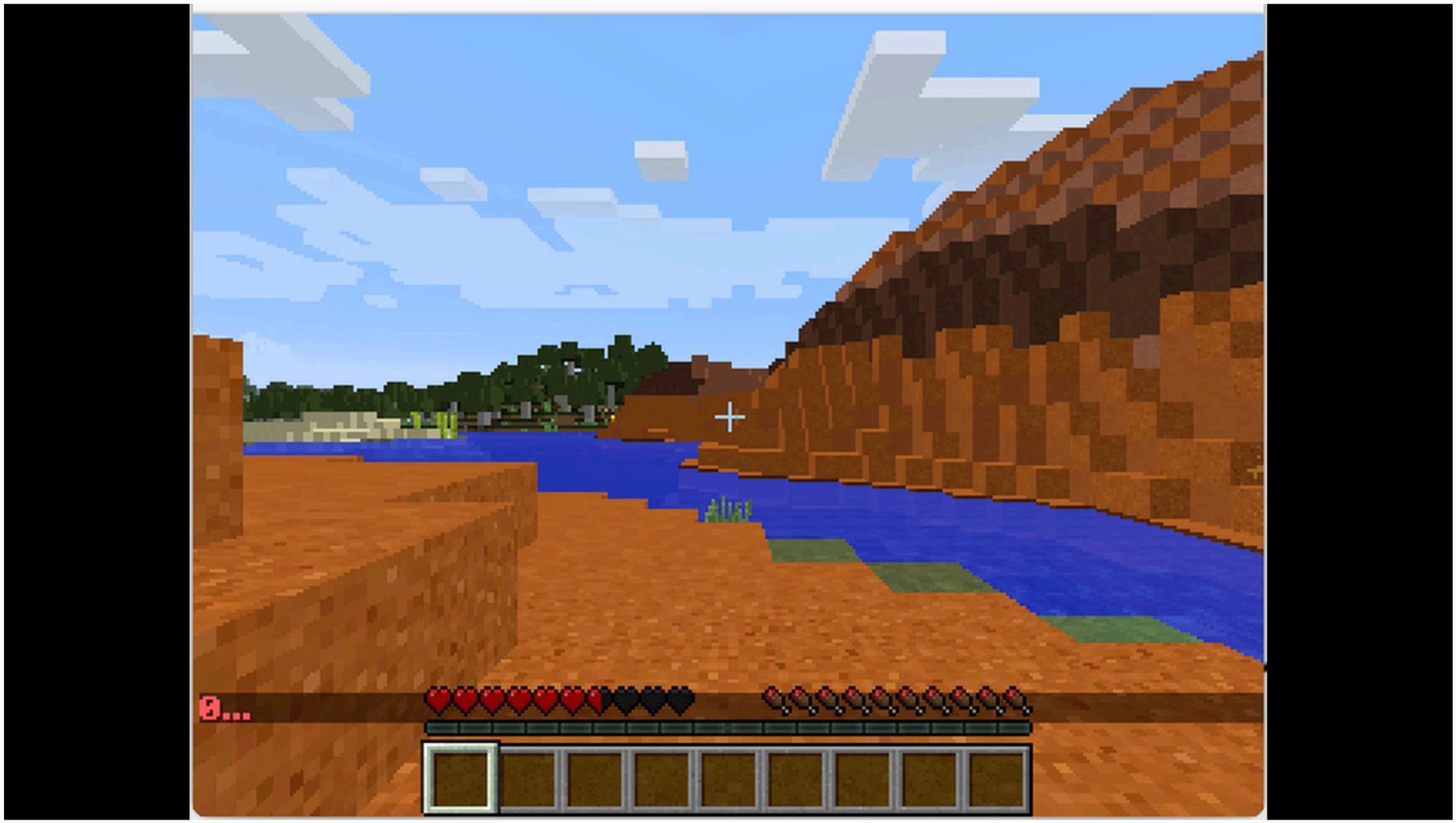




[Bostrom, 2014]

3. RL in Minecraft

Minecraft



Minecraft: Platform for AI

- Vision
- Natural Language Processing
- Cooperation
- Planning
- Learning

Goal

Develop a full vision and learning system
to solve complex tasks in Minecraft

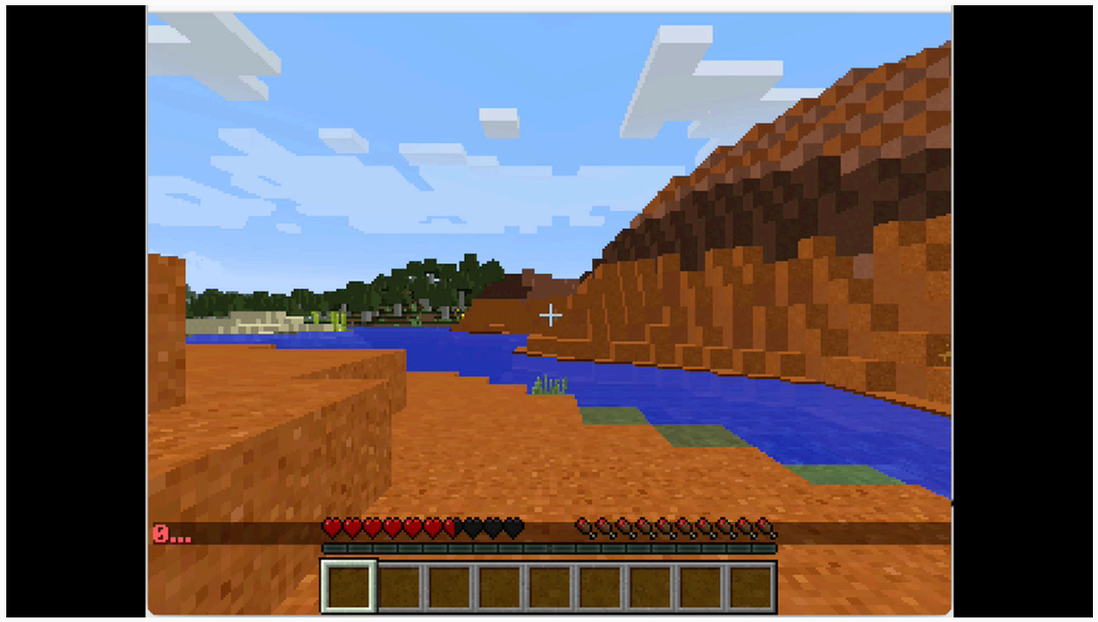
My Work: Two Tasks

- Task One: Visual Beacon! (Get to the beacon)

My Work: Two Tasks

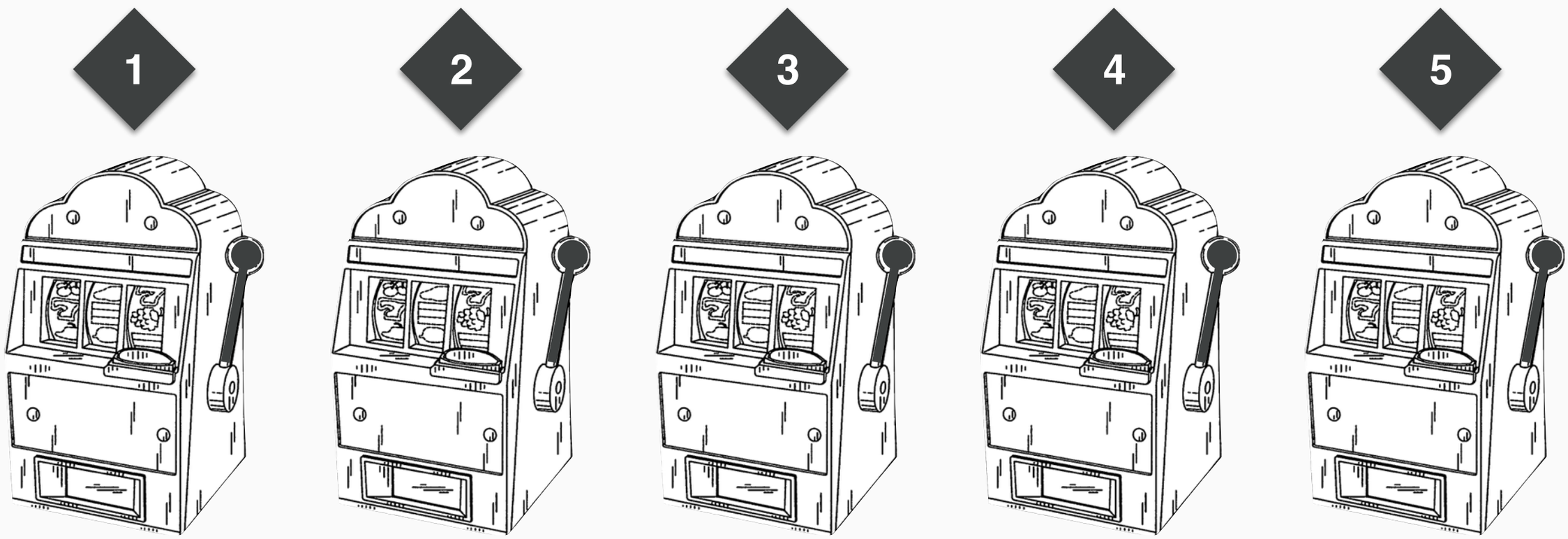
- Task Two: Visual Hill Climbing! (climb the hills)

Before Learning



After Learning

Slot Machines



The Problem: How do I (or the agent) know when I should *explore* (to gain more information), as opposed to *exploiting* the information I have?

My Work: Two Tasks

- Task Two: Visual Hill Climbing! (climb the hills)

See you Wednesday for the Last Day!