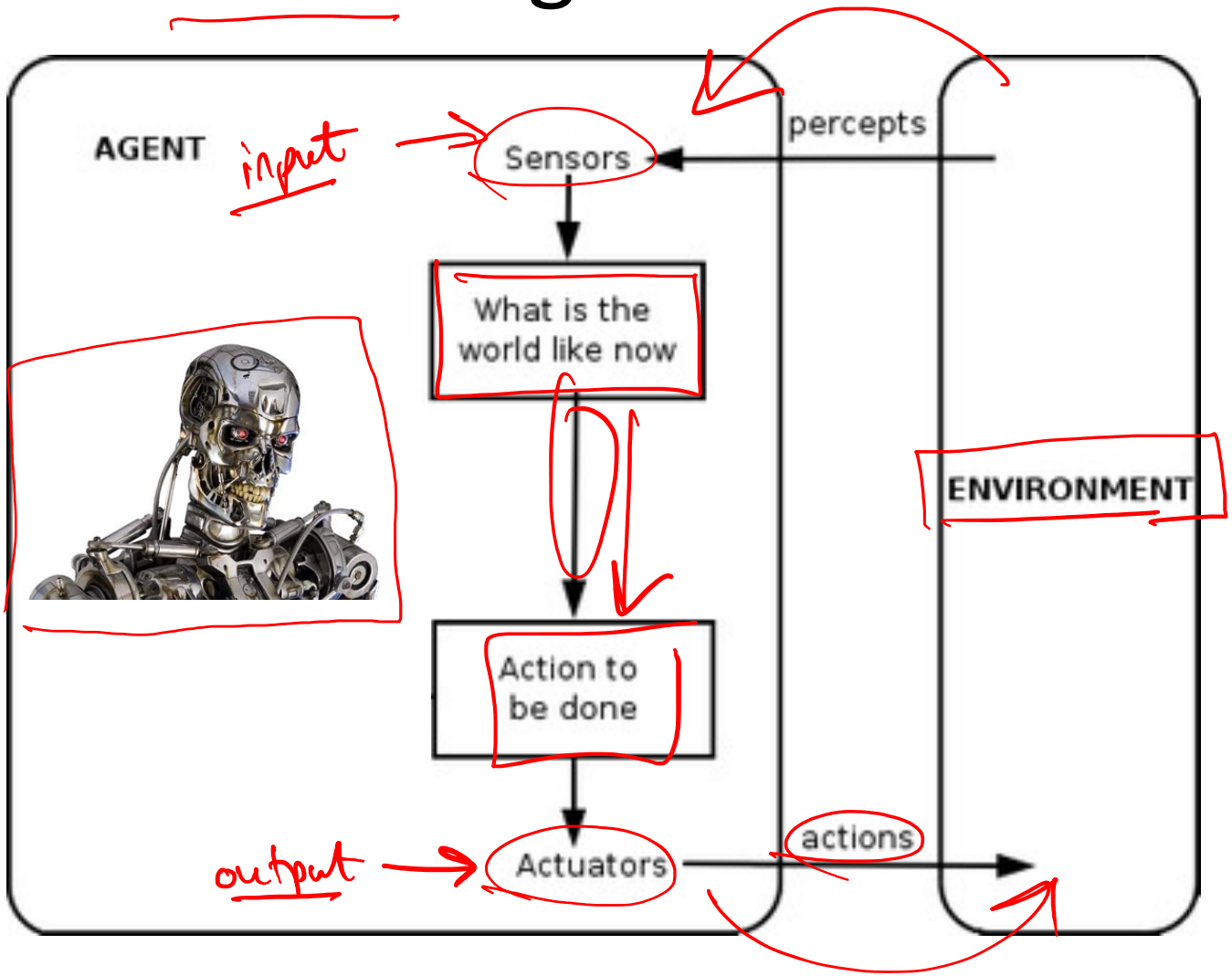
Rational Agents



Agents interact with their environment through sensors and actuators.

Rational Agents

- Rational agent:
  - For every possible percept sequence, a rational agent should
  - select an action that is expected to maximize its performance measure,
  - given evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational Agents

- Rational agent:
  - For every possible percept sequence, a rational agent should
  - select an action that is **expected to maximize its performance measure**,
  - given evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rational Agents

- Rational agent:
  - For every possible percept sequence, a rational agent should
  - select an action that is expected to maximize its performance measure,
  - given **evidence provided by the percept sequence and whatever built-in knowledge the agent has**.

Rational Agents

- Rational agent:
  - **For every possible percept sequence**, a rational agent should
  - select an action that is expected to maximize its performance measure,
  - given evidence provided by the percept sequence and whatever built-in knowledge the agent has.

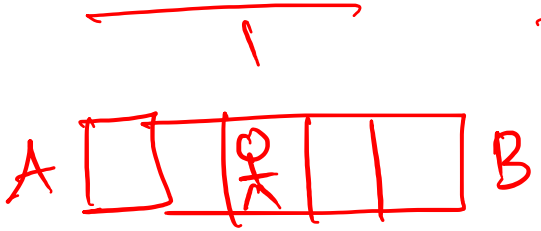
Environments

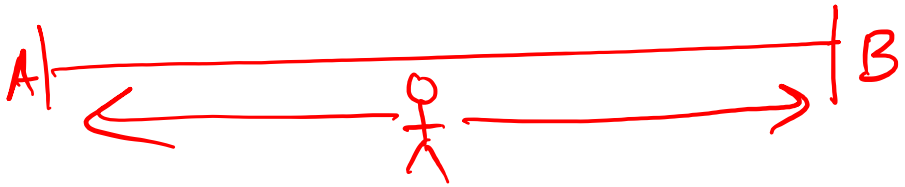
Chess/checkers Poker

- Fully-observable vs partially-observable
- Single agent vs multiple agents
- Deterministic vs non-deterministic (stochastic)
- Episodic vs sequential
- Static or dynamic
- Discrete or continuous

When the agent takes an action, is it 100% certain what will happen?

Does the environment change while the agent is ~~working~~ interacting w/ it.





3.1 - 3.3

State Space Search

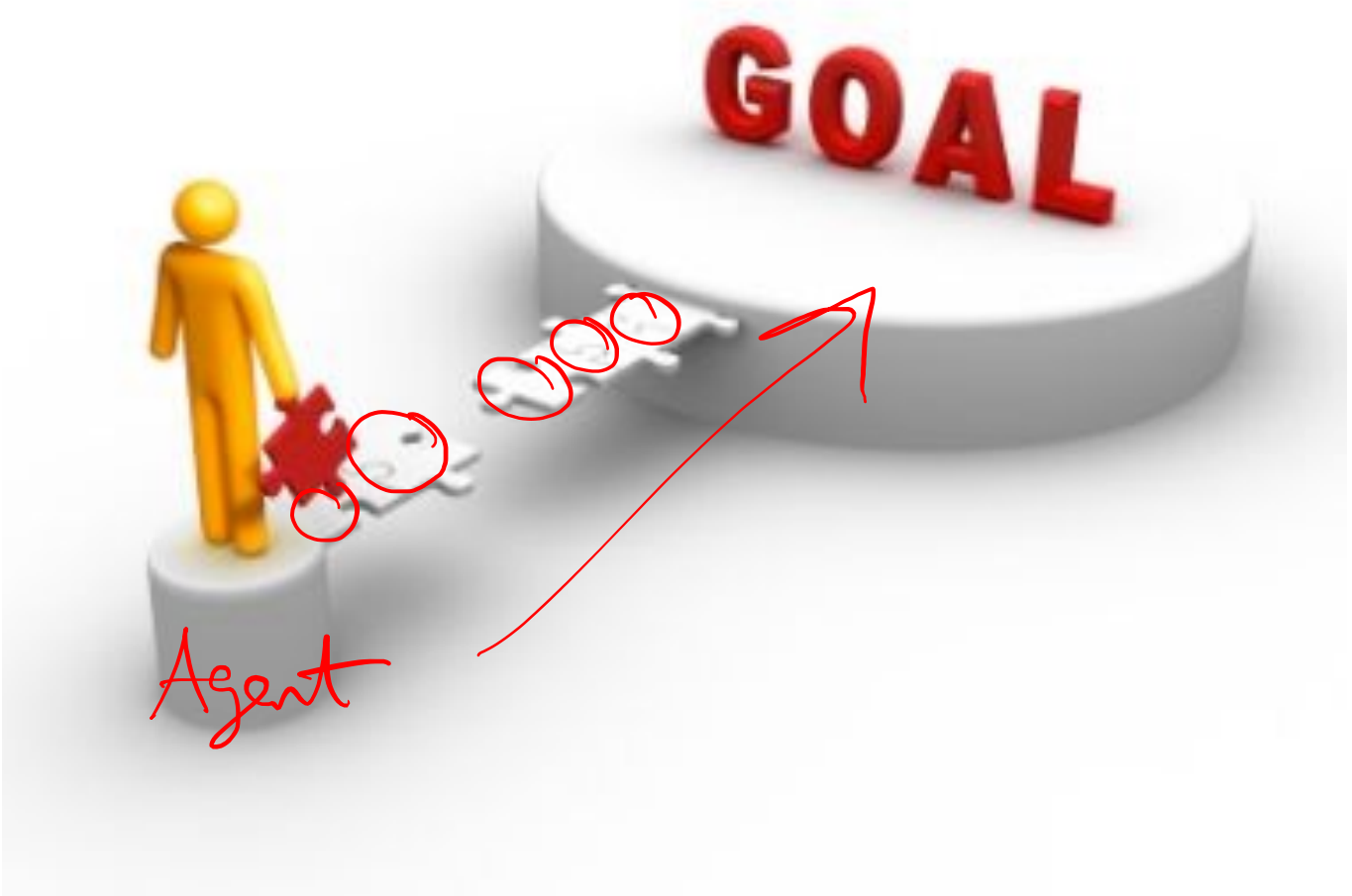
Environments

- Fully-observable vs partially-observable
- Single agent vs multiple agents
- Deterministic vs stochastic
- Episodic vs sequential
- Static or dynamic
- Discrete or continuous

Overview

- Problem-solving as search
- How to formulate an AI problem as search.
- Uninformed search methods

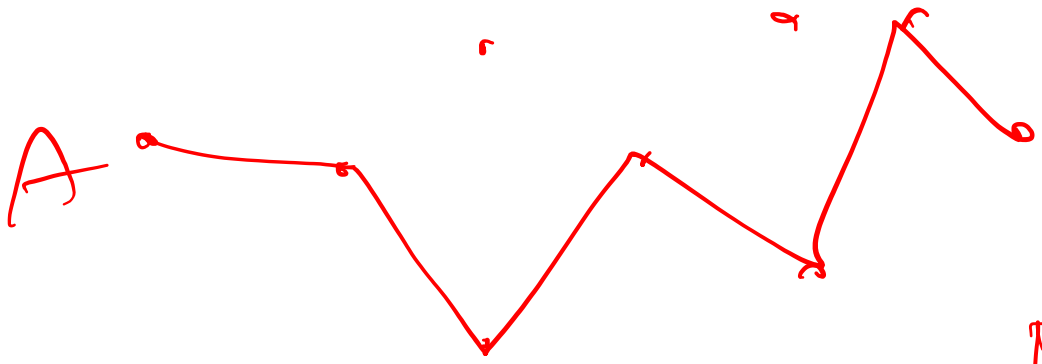
What is search? (3.1)

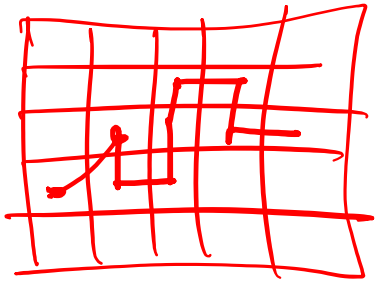


What is search?

Agent → take a discrete sequence of steps to solve a problem.

Usually used when the solution to the problem is the sequence of steps itself





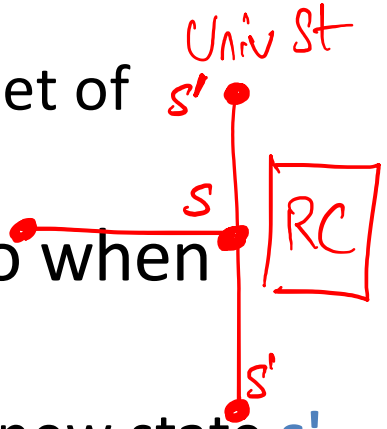
Environmental factors needed

- **Static** — The world does not change on its own, and our actions don't change it.
- **Discrete** — A finite number of individual states exist rather than a continuous space of options.
- **Observable** — States can be determined by observations.
- **Deterministic** — Action have certain outcomes.

- The **environment** is all the information about the world that remains constant while we are solving the problem.
- A **state** is the set of properties that define the current conditions of the world our agent is in.
  - Think of this as a *snapshot* of the world at a given point in time.
  - The entire set of possible states is called the **state space**.
- The **initial state** is the state the agent begins in.
- A **goal state** is a state where the agent may end the search.
- Agents move from state to state by taking **actions.** Moving from state to state has an associated **cost.**

If this were a video game, what info would it have to save to restore the game later?

- How does an agent know what actions are possible in a state?
  - Imagine a function **ACTIONS(*s*)** that returns the set of actions possible in a state ***s***.
- How does an agent know what state they go to when they take an action?
  - Imagine a function **RESULT(s, a)** that returns the new state **s'** that you end up in when taking action **a** from state **s**.
- How does an agent know when they have reached a goal state?
  - Imagine a function **IS-GOAL(s)** that returns true/false.
- How does an agent know the cost of moving from one state to another?
  - Imagine a function **ACTION-COST(s, a, s')** which returns the cost of taking action **a** in state **s** and moving to state **s'**.

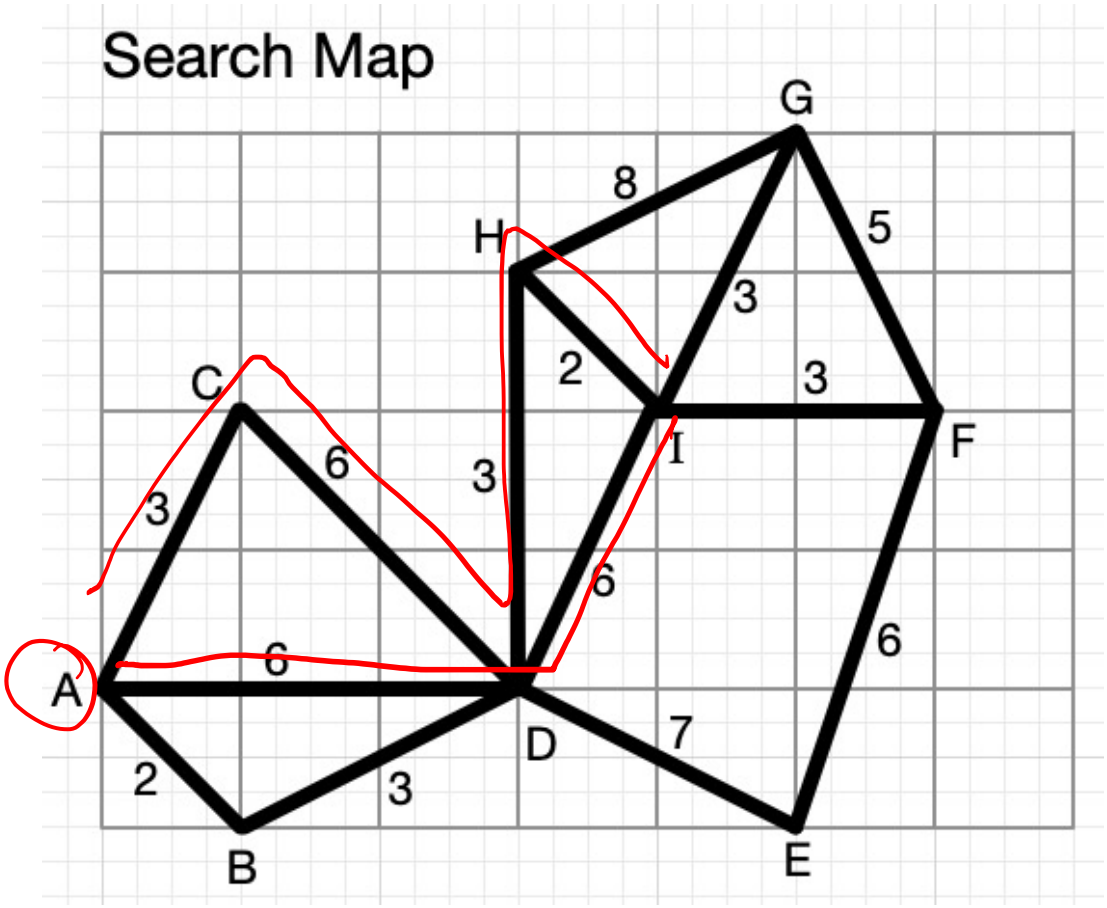


Formulating problems as search (3.2)

- *Canonical problem*: route-finding//Navigation

  – Route-finding with traveling salesperson problem.
- Sliding block puzzle (almost any kind of game or puzzle can be formulated this way).
- Roomba problem.

Formulate navigation problem



Agent: I → A

Env: everything about the graph

State: current pos'n of the agent

Actions: go to a neighboring vertex

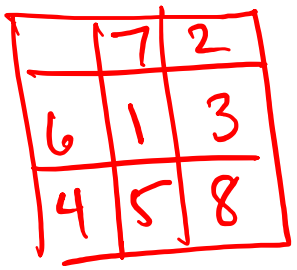
initial state - I

final state - A

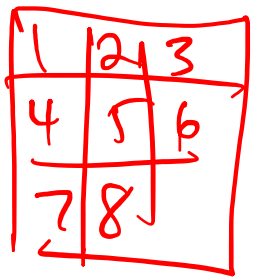
cost? - number on the graph

Size of the state space? 9

Formulate navigation problem



Formulate 8-puzzle problem



State

→ location of each number

- 2D array
- 1D | X | 7 | 2 | 6 | 1 | 3 | 4 | 5 | 8 |

1 (x,y)
2 (x,y)
3
4
|
8

Size of state space?

9!

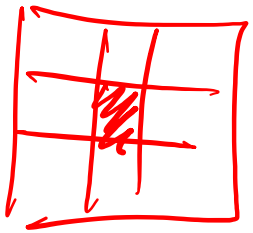
Env

3×3 grid

Action

- Slide a number

L, R, U, D



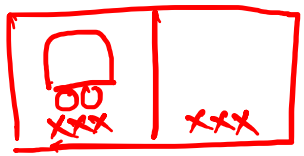
Initial state – starting configuration

goal state(s)

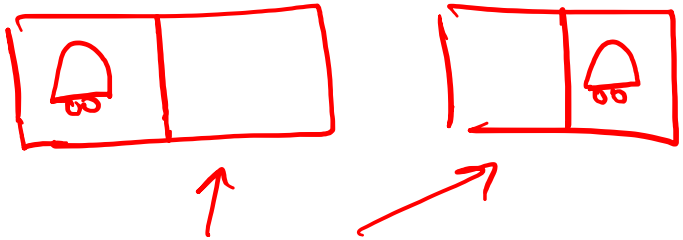
Cost? 1

Formulate 8-puzzle problem

Formulate Roomba problem



initial state



goal state?(s)

Actions: CLEAN
MOVE LEFT
MOVE RIGHT

state representation

→ current loc'n of Roomba

→ ~~#~~ which rooms are clean/dirty

| true | false |