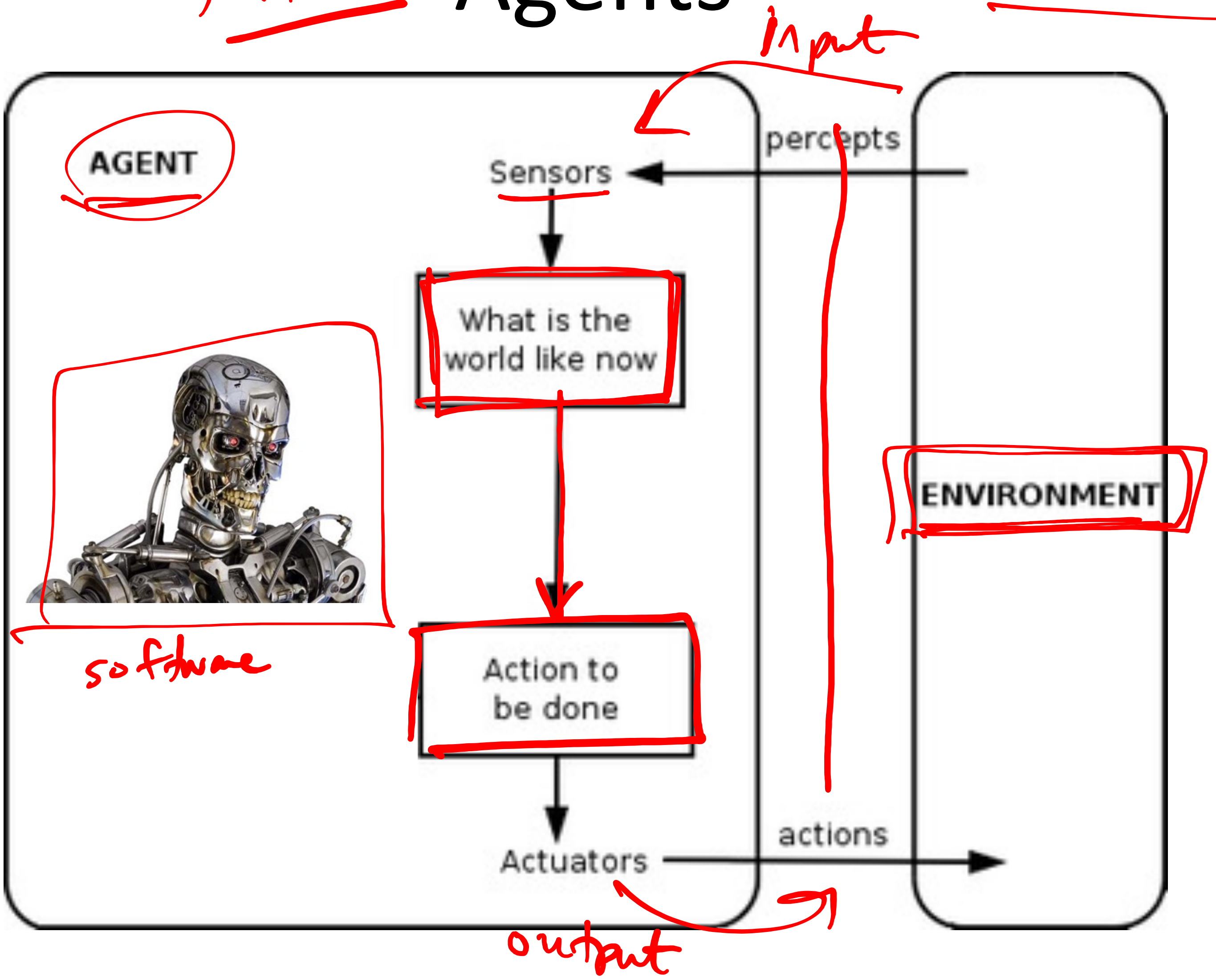
Rational Agents

"Act rationally"



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

Q: when the agent takes an action, does it know for certain what will happen?

→ stochastic

- Episodic vs sequential

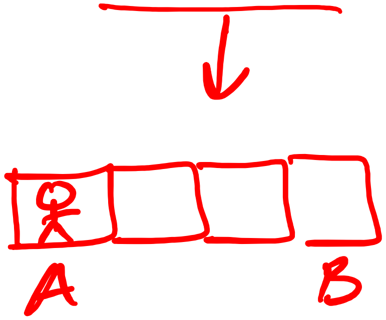
one episode doesn't affect the next one

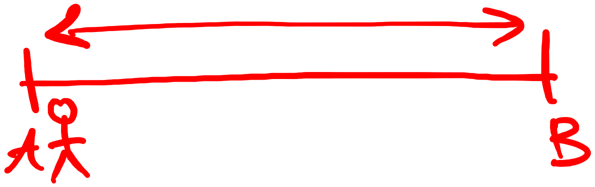
- Static or dynamic

env is changing

env is not changing

- Discrete or continuous





Ch 3

3.1 – 3.2

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)

GOAL
5
4
3
2
1
Agent

What is search?

Agent - trying to get to a goal

↳ Build a solution through constructing a sequence of steps.

Ex : Navigation problem

: Getting dressed

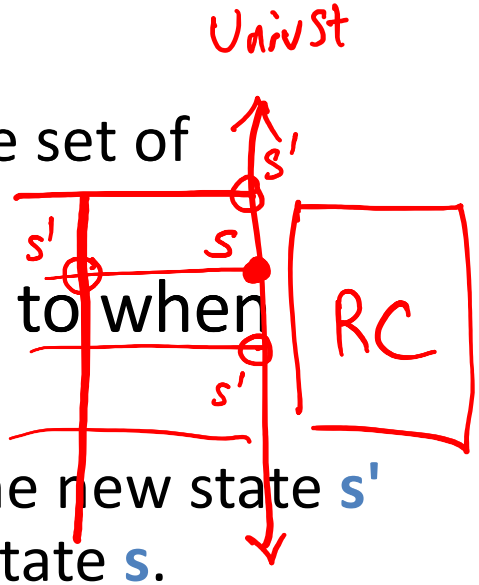
: Solitaire

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. stuff that is changing
  - Think of this as a *snapshot* of the world at a given point in time. → Think of the agent in a video game, state = everything that has to be saved to restore the game later.
  - 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.**

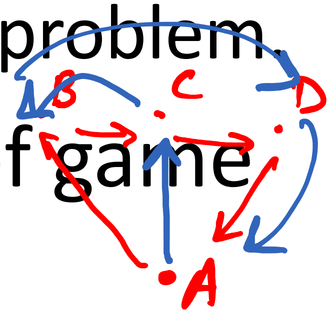
- How does an agent know what actions are possible in a state?
  - Imagine a function **ACTIONS(*s*)** that returns the set of actions possible in a state ***s***.
- How does an agent know what state they go to when they take an action?
  - Imagine a function **RESULT(s, a)** that returns the new state **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.

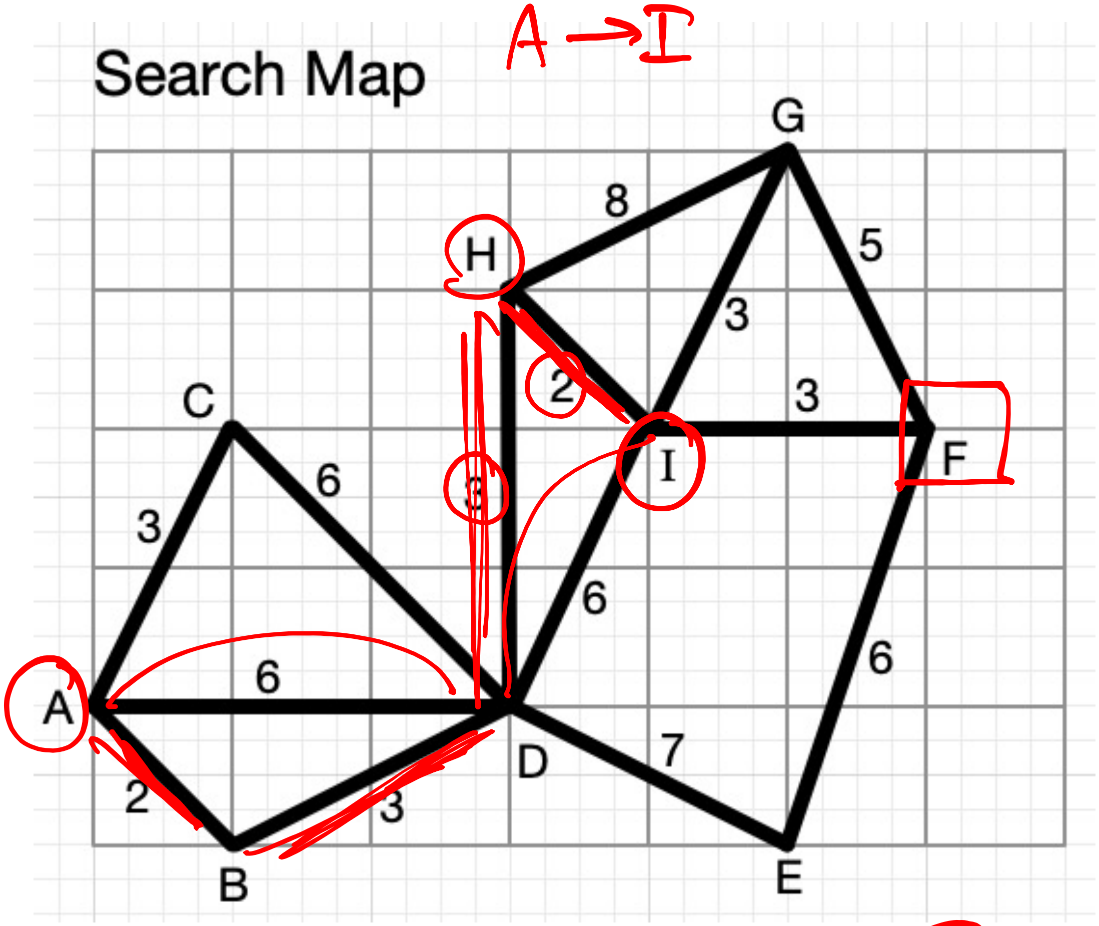


- Sliding block puzzle (almost any kind of game or puzzle can be formulated this way).
- Roomba problem.

	6	2
5	1	3
4	8	7

Formulate navigation problem

A → I



Env? Distances whole map

State? Where the agent is.

initial/goal state

A I

Actions? Go to any vertex connected to your state.

State space: 9

Formulate navigation problem

Formulate 8-puzzle problem

2	6	3
5	1	4
	7	8

initial state / goal state

1	2	3
4	5	6
7	8	

State?

where each # is.

→ Represent?

2D array

1D array

2	6	3	5	1	4	X	7	8

Map

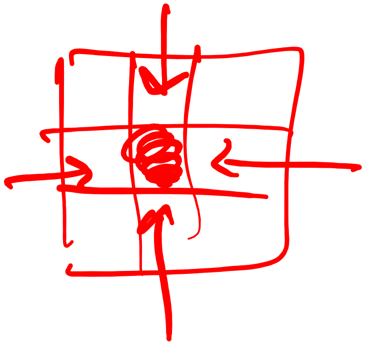
1 → (x,y)
2
3
:
:
8

Env?

3x3 grid w/ #s 1-8

Actions?

L, R, U, D

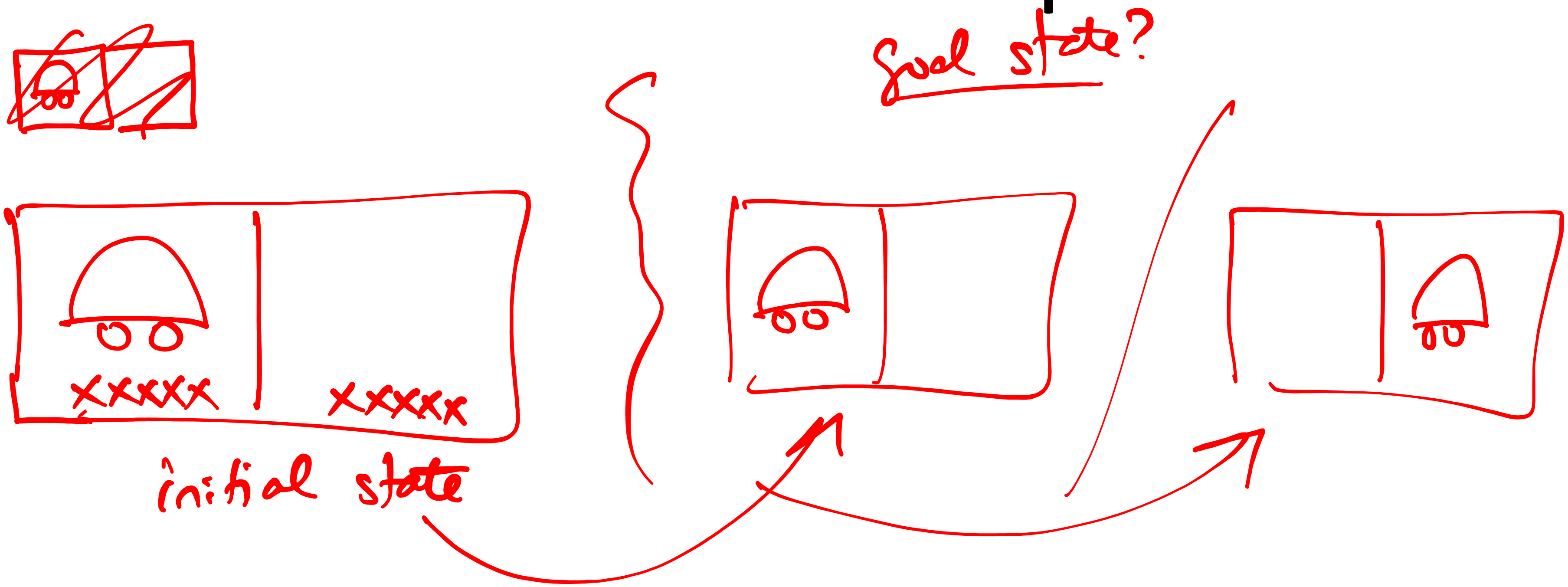


State space

?!

Formulate 8-puzzle problem

Formulate Roomba problem



Env

State

location of Roomba

which rooms are clean/dirty

Actions

CLEAN

MOVE-LEFT

MOVE-RIGHT