CS 232: Artificial Intelligence

Spring 2024

Prof. Carolyn Anderson
Wellesley College

Reminders

- HW 1 due today
- Set up Python 3.8 virtual environment (I recommend using Anaconda)
- I have help hours today from 3:30-4:30 (4th floor W)
- Lepei has help hours this week on Wednesday from 12-2pm on Zoom (usually: Thursdays 3:45-5:45)
- Reading for Tuesday: YLLATAILY Chapters 1 & 2

Jordan (2019)

Jordan (2019)

- Intelligent Infrastructure (II)
- Intelligence Augmentation (IA)
- Human-Imitative Artificial Intelligence

Jordan (2019)

- Intelligent Infrastructure (II): "*a web of computation, data, and physical entities [...] that makes human environments more supportive, interesting, and safe*"
- Intelligence Augmentation (IA): "*computation and data are used to create services that augment human intelligence and creativity*"
- Human-Imitative Artificial Intelligence: "*realizing in software and hardware an entity possessing human-level intelligence*"

Task-based AI

Task-based AI

Our goal is to write programs that can solve tasks. This is sort of the goal of all computer science.

In AI, though, the tasks we focus on are ones that seem to require human intelligence. This is a moving standard- what seems impossible for a computer to solve one day may eventually become very easy.

AI Tasks

Almost all AI tasks can be grouped into one of three main categories:

- Search
- Classification
- Generation

Search

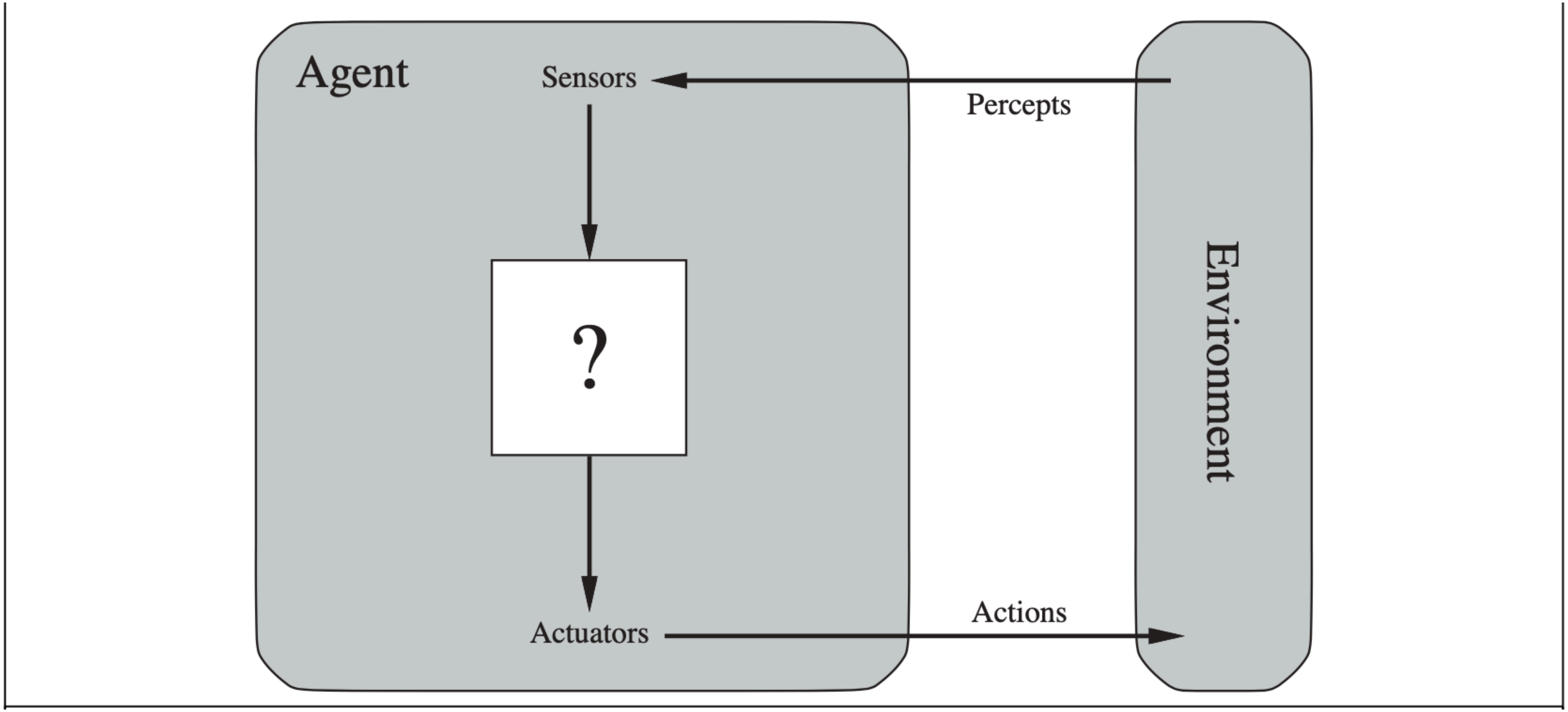
Agents

An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.

This is **extremely broad**!

Agents

An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.



Russell & Norvig ed. 3

Defining an environment

Making problems out of the real world

If our goal is to tackle hard real-world problems, we will need a way of describing them.

Agents exist in **environments**. How do we define environments?

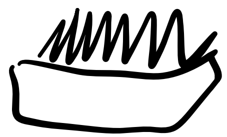
Defining the environment

Consider my cat Captain Haddock as an agent.

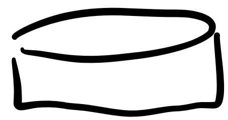
Defining the environment

We could define his environment very simply. Here are two possible states:

State 1: foodInBowl = true



State 2: foodInBowl = false



Defining the environment

Here are Captain Haddock's actions in each state:

State	Actions
1: foodInBowl = true	eat
2: foodInBowl = false	wait

Defining the environment

If we represent more information about the environment, we can make more nuanced choices:

State 1: humansHome = true, foodInBowl = true

State 2: humansHome = true, foodInBowl = false

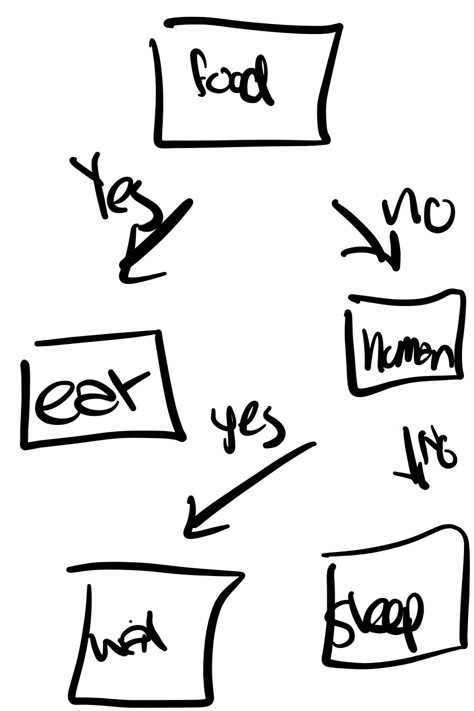
State 3: humansHome = false, foodInBowl = true

State 4: humansHome = false, foodInBowl = false

Defining the environment

Here are Captain Haddock's actions in each state:

State	Actions
1: humansHome = true, foodInBowl = true	→ eat
2: humansHome = true, foodInBowl = false	→ wail
3: humansHome = false, foodInBowl = true	→ eat
4: humansHome = false, foodInBowl = false	→ sleep



Rationality

Rational agents

A rational **agent** is an entity that acts to optimize a desired outcome (accomplish a task).

Rational agents

Given a goal, an AI agent must decide what the best action to take is in order to reach this goal.

For complex tasks, this can mean:

- gathering information
- coming up a set of possible actions
- weighing the best action
- acting
- updating and adapting based on changes to the environment

Defining Goals

In a search problem, our first step is to **define a goal**.

Captain Haddock has two goals in life:

1) eat

2) sleep

Actions

Given a goal, an AI agent must decide which actions to take.

Captain Haddock has several available actions:

- eat
- sleep
- wail
- stare at birds

Actions

Not all of these are possible in every environment:

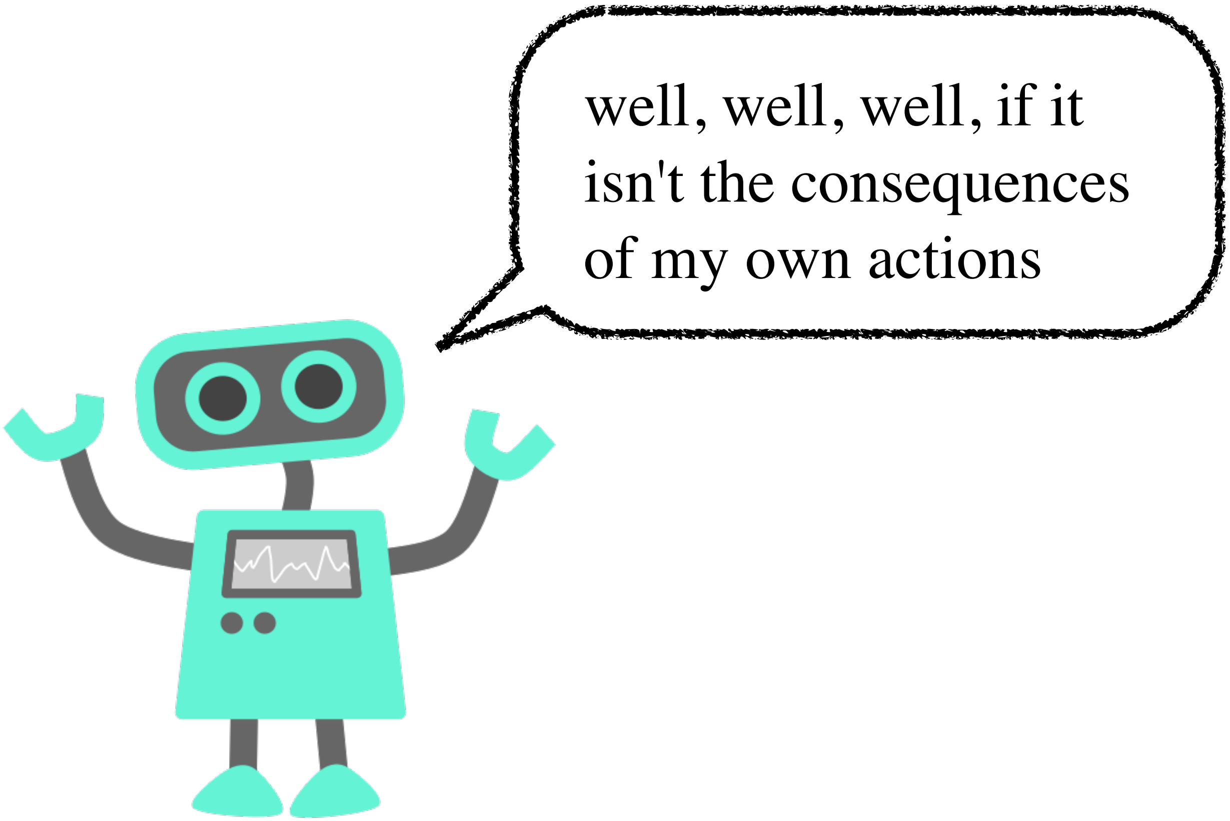
- eat - successful if food in bowl
- sleep - always successful
- wail - always successful
- stare at birds - successful if birds

Success = possible in the state

Optimal = best action to take

Assessing Consequences

Given a goal and a set of actions, an AI agent must decide which action to take by weighing their **consequences**.



Assessing Consequences

Some actions have costs as well as consequences!

eat: - achieves goal of eating
- updates environment by removing food

sleep: - achieves secondary goal
- replenishes energy

wail: - chance of humans adding food if humans present
- cost: loss of dignity

stare at bird: - absolutely no effect

Assessing Consequences

Some actions have costs as well as consequences!

eat:
- achieves goal of devouring food
- updates environment by removing food

sleep:
- achieves goal of sleeping
- replenishes energy

wail:
- chance of human adding food to the bowl, if human is available and sufficiently convinced
- cost: lack of dignity

stare at bird:
- absolutely no consequences
- cost: some energy

Transitions

When an agent takes an action, they may transition to a new state (the environment may change).

Solutions

- A **solution** is a sequence of actions from the initial state to a goal state.
- A solution is **optimal** if no other solution has a lower cost.

Agent Complexity

Reflex agent: decides what to do based only on current state.

Agent Complexity

Problem-solving agent: capable of considering a sequence of actions that form a path to a goal state (planning ahead to find a solution).

Defining Search Problems

Formal Definition

1. States: a set S
2. An initial state $s_i \in S$
3. Actions: a set A

   $\forall s$ Actions(s) = the set of actions possible in s.
4. Transition Model: $\forall s\ \forall a \in$ Actions(s) Result(s, a) $\longrightarrow s_r$

   s_r : a successor of s
5. Path Cost: must be additive

   $c(s_1, a, s_2)$: cost of going from s_1 to s_2 by action a
6. Goal test: Goal(s)

Slides adapted from Chris Callison-Burch

Formal Definition

1. ***States***: a set S
2. *An initial state* $s_i \in S$
3. ***Actions:*** *a set* A

   $\forall s$ $Actions(s)$ = *the set of actions that can be executed in* s*, that are applicable* in s.
4. ***Transition Model:*** $\forall s \forall a \in Actions(s)$ $Result(s, a) \rightarrow s_r$

   s_r *is called a successor of* s

   $\{s_i\} \cup Successors(s_i)^*$ *= state space*
5. ***Path cost*** *(Performance Measure)*: Must be additive, e.g. sum of distances, number of actions executed, …

   $c(x,a,y)$ is the step cost, assumed ≥ 0

   - (where action a goes from state x to state y)
6. ***Goal test:*** $Goal(s)$

   Can be implicit, e.g. $checkmate(s)$

   s is a *goal state* if $Goal(s)$ is *true*

Slides adapted from Chris Callison-Burch

Vacuum World

States: A state of the world says which objects are in which cells.

In a simple two cell version,

- the agent can be in one cell at a time
- each cell can have dirt or not

Slides adapted from Chris Callison-Burch

Vacuum World

Goal states: States where everything is clean.

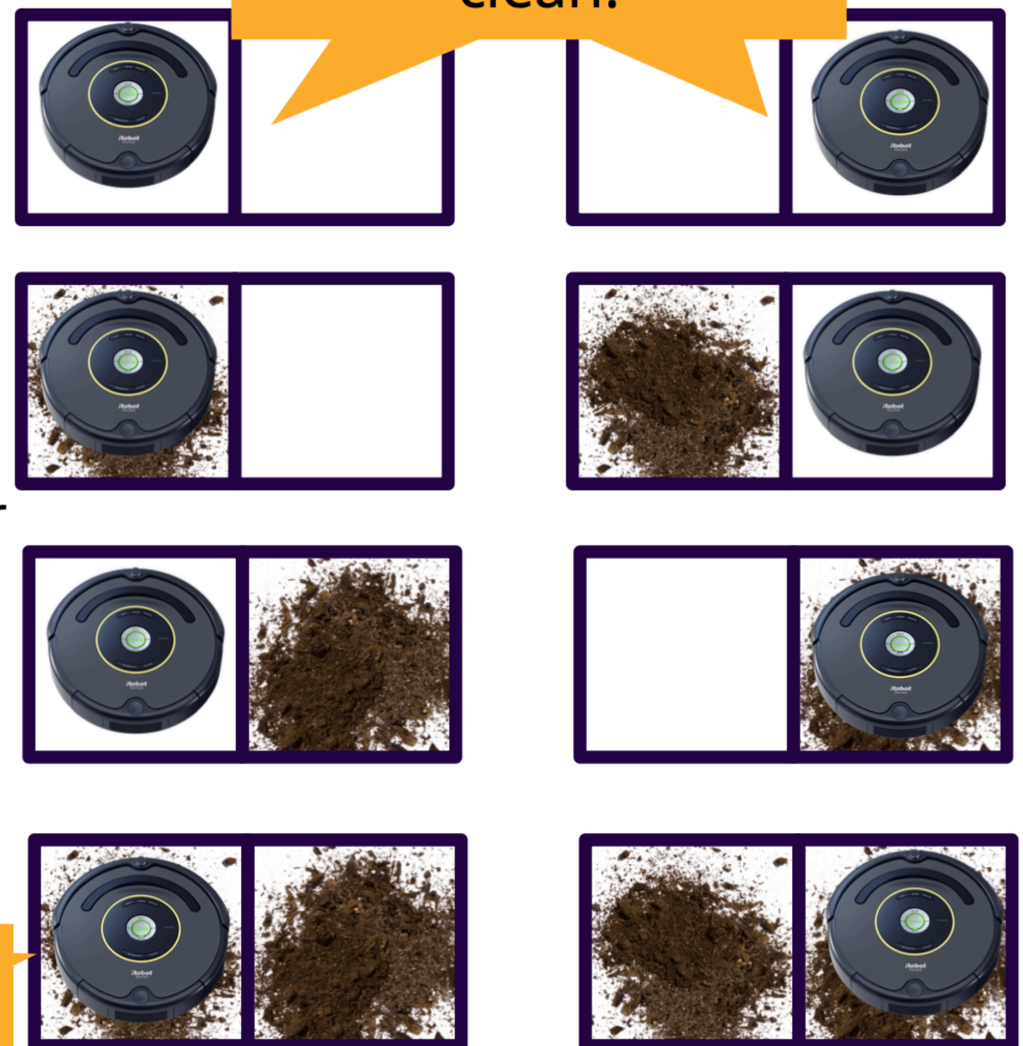
States: A state of the world says which objects are in which cells.

In a simple two cell version,

- the agent can be in one cell at a time
- each cell can have dirt or not

2 positions for agent * 2^2 possibilities for dirt = 8 states.

With n cells, there are $n*2^n$ states.



One state is designated as the **initial state**

Vacuum World

Start State

Actions:

Move Right
Move Left
Clean

Transition:

> succeed if no wall & moves Roomba

- succeeds if dirt puts us in the clean version of that state

Vacuum World

Actions:

- *Clean*
- *Move Left*
- *Move Right*
- *(Move Up)*
- *(Move Down)*

Transition:

Clean – removes dirt

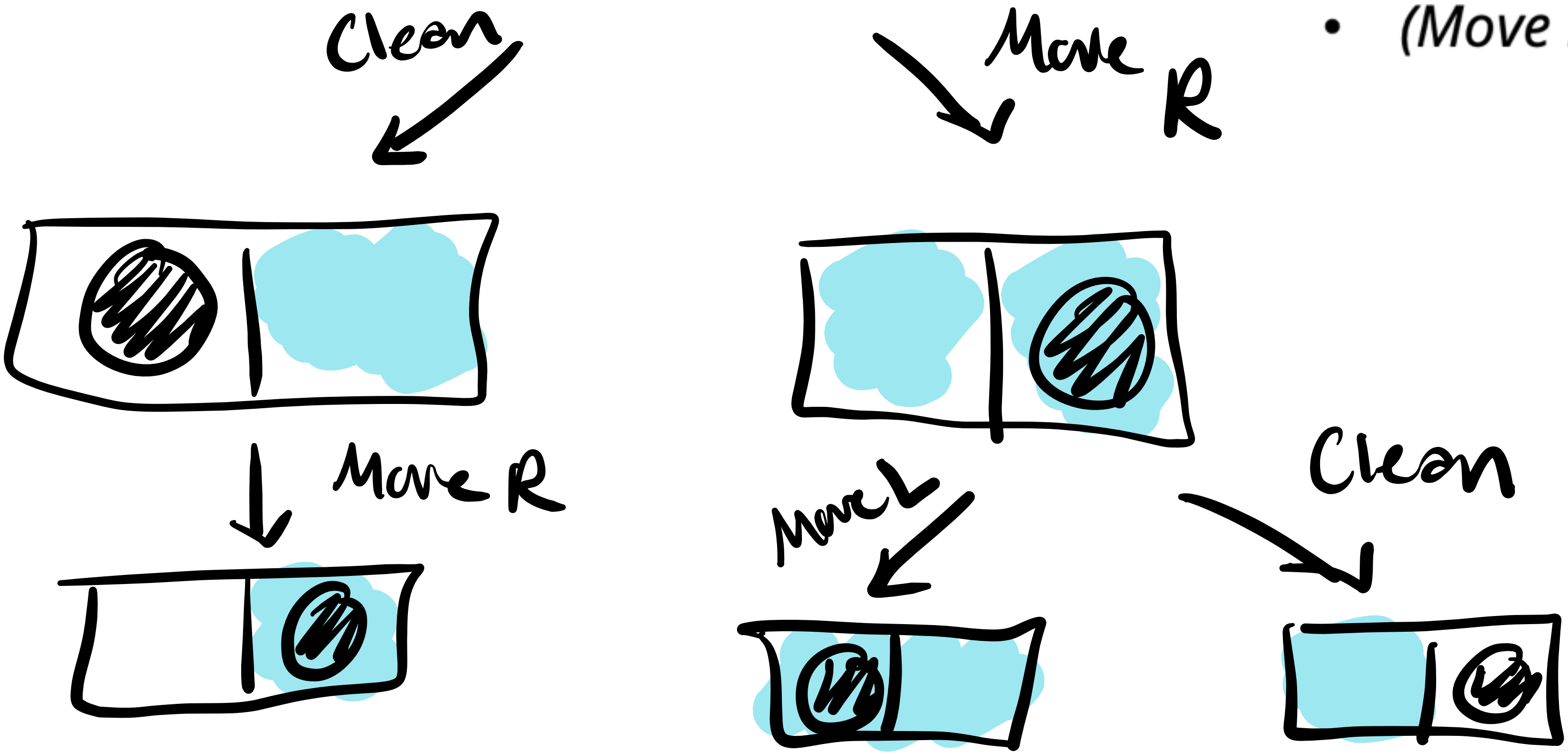
Move – moves in that direction, unless agent hits a wall, in which case it stays put.

Slides adapted from Chris Callison-Burch

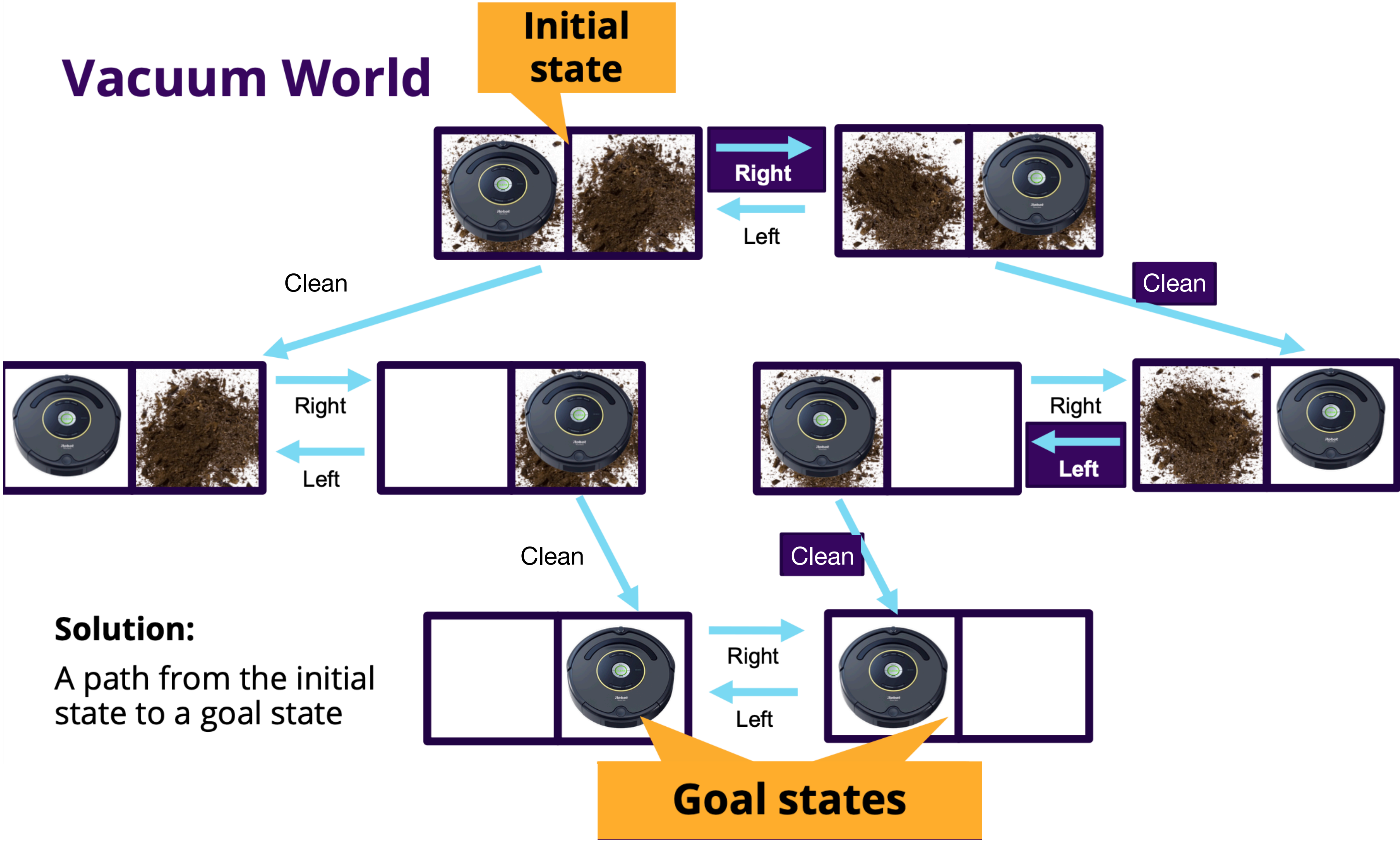
Vacuum World

Actions:

- *Clean*
- *Move Left*
- *Move Right*
- *(Move Up)*
- *(Move Down)*



Vacuum World



Solution:

A path from the initial state to a goal state

Art: Formulating a Search Problem

Decide:

Which properties matter & how to represent

- *Initial State, Goal State, Possible Intermediate States*

Which actions are possible & how to represent

- *Operator Set: Actions and Transition Model*

Which action is next

- *Path Cost Function*

Formulation greatly affects combinatorics of search space and therefore speed of search

Useful Concepts

State space: the set of all states reachable from the initial state by *any* sequence of actions

- *When several operators can apply to each state, this gets large very quickly*
- *Might be a proper subset of the set of configurations*

Path: a sequence of actions leading from one state s_j to another state s_k

Solution: a path from the initial state s_i to a state s_f that satisfies the goal test

Search tree: a way of representing the paths that a search algorithm has explored. The root is the initial state, leaves of the tree are successor states.

Frontier: those states that are available for *expanding* (for applying legal actions to)