Discussion

CS 5/7320 Artificial Intelligence

Intelligent Agents
AIMA Chapter 2

Slides by Michael Hahsler
with figures from the AIMA textbook.

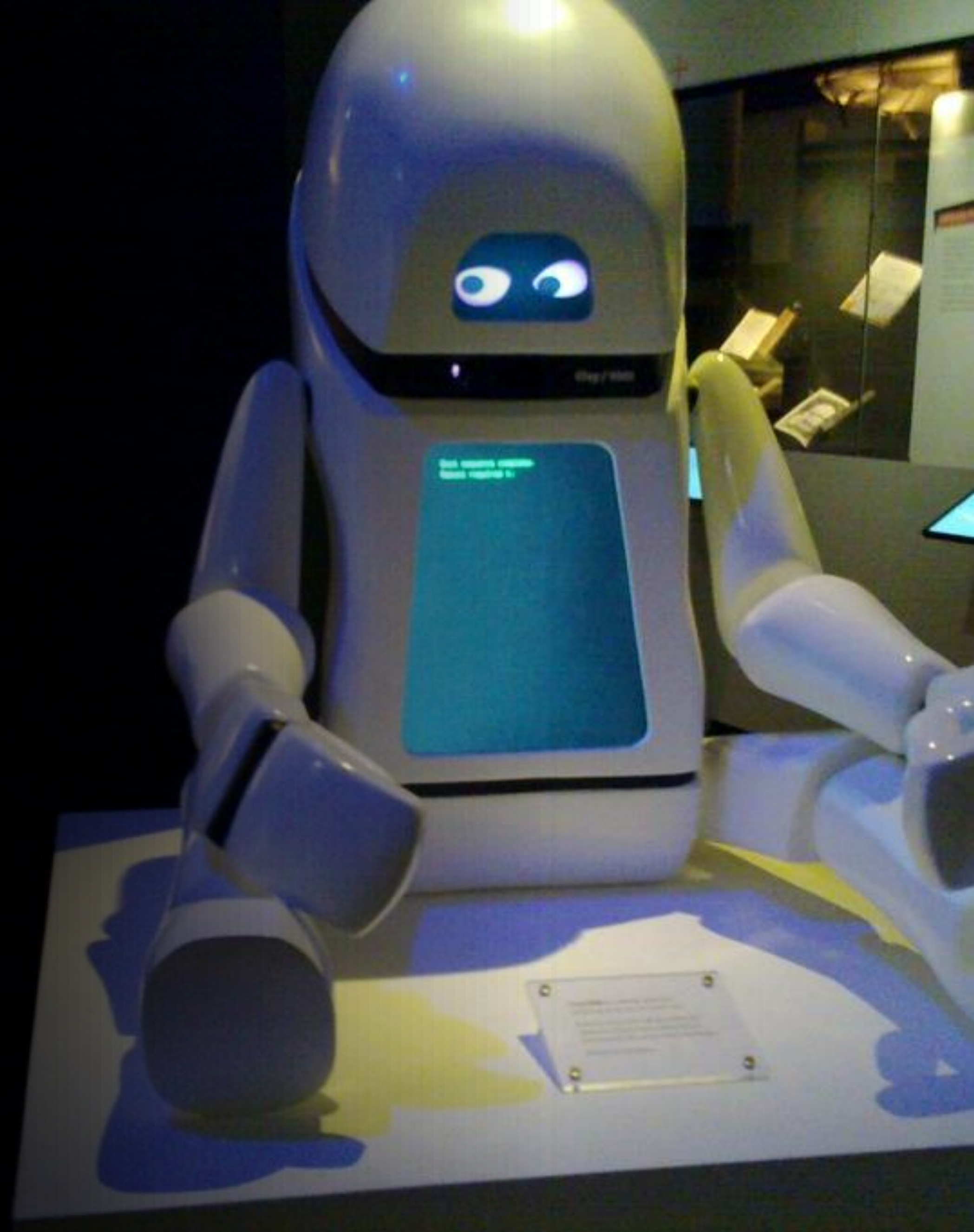


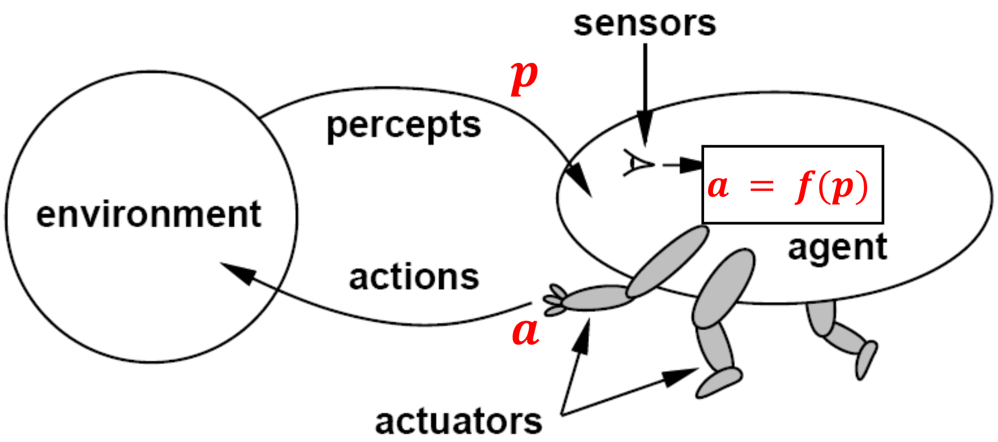
Image: "Robot at the British Library Science Fiction Exhibition" by BadgerGravling

Module Review 1

Agent Function and Agent Program

The agent function maps from the set of all possible *percept sequences* P^* to the *set of actions* A formulated as an abstract mathematical function.

$$f : P^* \rightarrow A$$



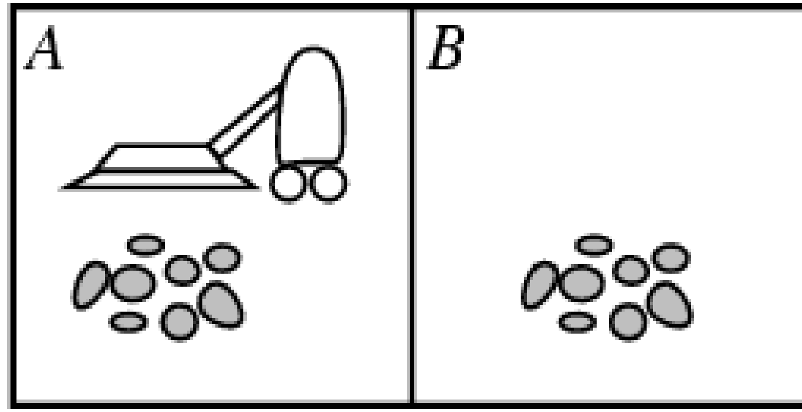
The agent program is a concrete implementation of this function for a given physical system.

Agent = architecture (hardware) + agent program (implementation of f)

- Sensors
- Memory
- Computational power

Example: Vacuum-cleaner World

- **Percepts:**
  Location and status,
  e.g., [A, Dirty]
- **Actions:**
  Left, Right, Suck, NoOp



Most recent Percept p

Agent function: $f : P^* \rightarrow A$

Percept Sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
…	
[A, Clean], [B, Clean]	Left
…	
[A, Clean], [B, Clean], [A, Dirty]	Suck
…	

Problem: This table can become infinitively large!

Implemented agent program:

function Vacuum-Agent([location, status])
returns an action a

```
if status = Dirty then return Suck
else if location = A then
                          return Right
else if location = B then
                          return Left
```

Rational Agents

Rule: Pick the action that maximize the expected utility

$$a = \operatorname{argmax}_{a \in \mathrm{A}} E(U \mid a)$$

This means:

- **Rationality is an ideal** – it implies that no one can build a better agent
- **Rationality ≠ Omniscience** – rational agents can make mistakes if percepts and knowledge do not suffice to make a good decision
- **Rationality ≠ Perfection** – rational agents maximize **expected** outcomes not actual outcomes
- **It is rational to explore and learn** – I.e., **use percepts** to supplement prior knowledge and become autonomous
- **Rationality is often bounded** by available memory, computational power, available sensors, etc.

Case Study: Self-Driving Cars

Self-driving Cars

SAE Automation Levels

- Level 1 - Driver Assistance (“hands on”)
- Level 2 - Partial Automation (“hands off”)
- Level 3 - Conditional Automation
- Level 4 - High Automation
- Level 5 - Full Automation (“steering wheel optional”)

Components

- Sensing
- Maps
- Path planning
- Controlling the vehicle

Why is this so hard?

A Self-Driving Car as a Rational Agents

Rule: Pick the action that maximize the expected utility

$$a = \operatorname{argmax}_{a \in \mathrm{A}} E(U \mid a)$$

Answer the following questions:

- If we have two cars and one provides more (expected) utility.
  Which car is rational?
- Can a rational self-driving car be involved in an accident?
- How would a self-driving car explore and learn?
- What does bounded rationality mean for a self-driving car?

PEAS Description of the Environment of a Self-Driving Car

Complete the PEAS description.

Performance measure	Environment	Actuators	Sensors

PEAS Description of the Environment of a Self-Driving Car

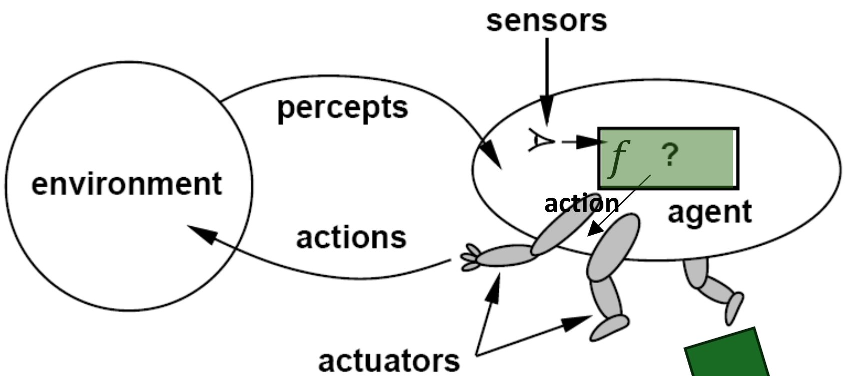
Performance measure	Environment	Actuators	Sensors
• Safe	• Roads	• Steering wheel	• Cameras
• fast	• other traffic	• accelerator	• sonar
• legal	• pedestrians	• brake	• speedometer
• comfortable trip	• customers	• signal	• GPS
• maximize profits		• horn	• Odometer
			• engine sensors
			• keyboard

Assignment 1

Introduction : Environment

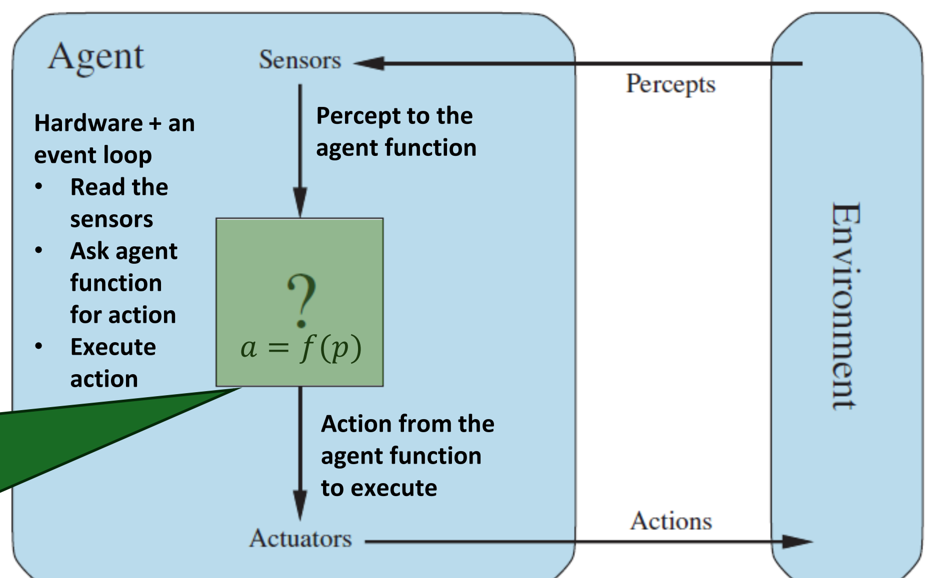
Module Review 2

Designing a Rational Agent



Remember the definition of a rational agent:

> *"For each possible percept sequence, a rational agent should select an **action** that **maximizes its expected performance measure**, given the evidence provided by the **percept sequence** and the **agent's built-in knowledge**."*



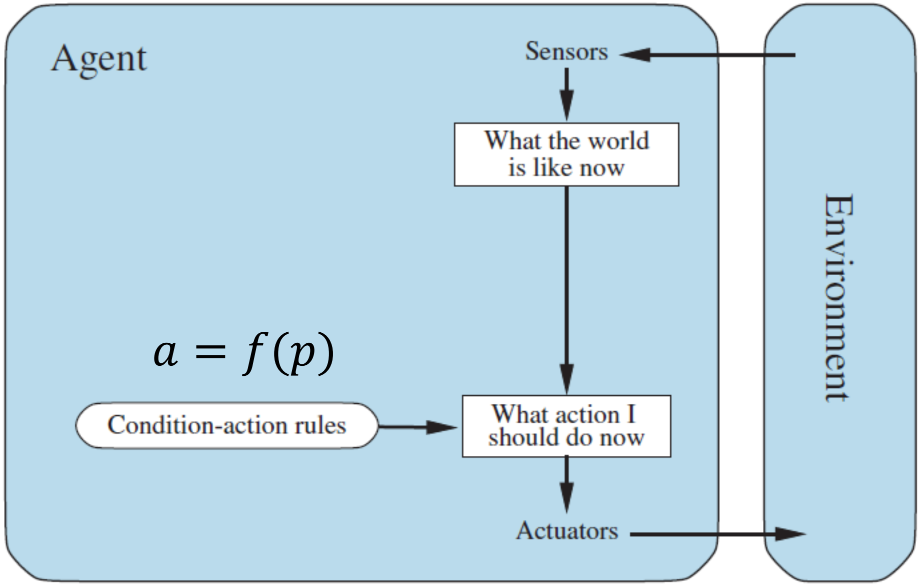
Agent Function

- Represents the "brain"
- Assess performance measure
- Remember percept sequence
- Built-in knowledge

Important: Everything outside the agent function represents the environment. This includes the physical robot, its sensors and its actuators, and event loop!

Simple Reflex Agent

- Uses only built-in knowledge in the form of **rules** that select action only **based on the current percept.** This is typically very fast!
- The **agent does not know about the performance measure**! But well-designed rules can lead to good performance.
- The agent needs no memory and ignores all past percepts.

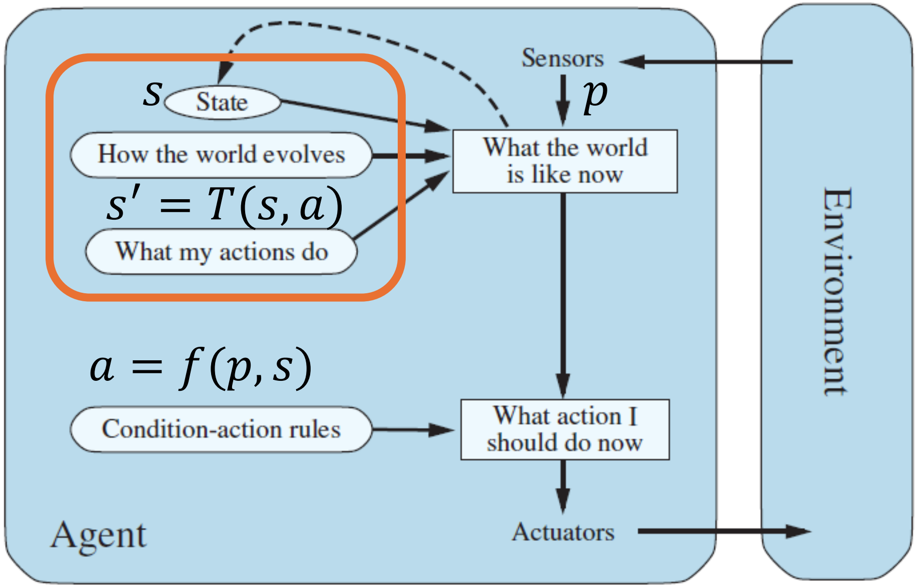


The interaction is a sequence: $p_0, a_0, p_1, a_1, p_2, a_2, \ldots p_t, a_t, \ldots$

Example: A simple vacuum cleaner that uses rules based on its current sensor input.

Model-based Reflex Agent

- Maintains a **state variable** to keeps track of aspects of the environment that cannot be currently observed. I.e., it has memory.
- It knows how the environment evolves over time given its last action. It updates the state using a **transition function** and the new percept.
- There is now more information for the **rules** to make better decisions.



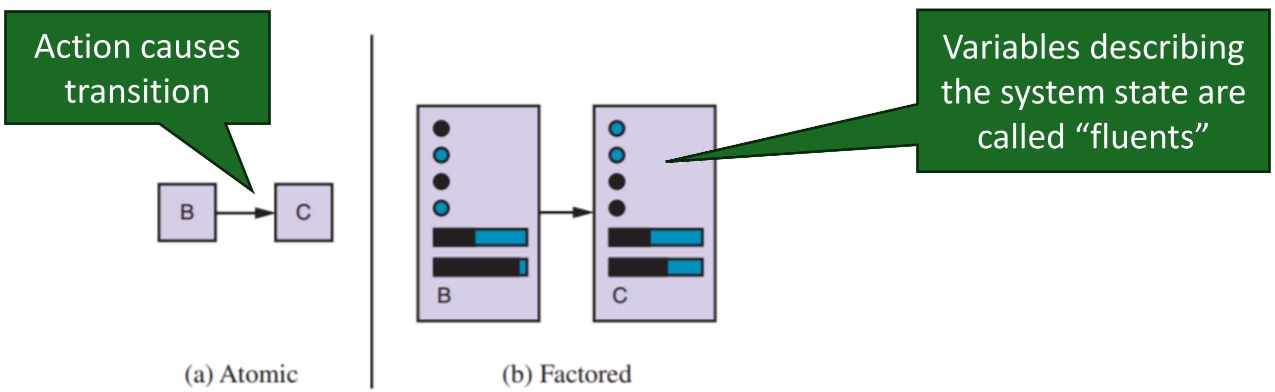
The interaction is a sequence: $p_0, s_0, a_0, p_1, s_1, a_1, p_2, s_2, a_2, p_3, \dots, p_t, s_t, a_t, \dots$

Example: A vacuum cleaner that remembers were it has already cleaned.

State Representation

States help to keep track of the environment and the agent in the environment. This is often also called the **system state**. The representation can be

- **Atomic**: Just a label for a black box. E.g., A, B
- **Factored**: A set of attribute values called fluents.
  E.g., [location = left, status = clean, temperature = 75 deg. F]

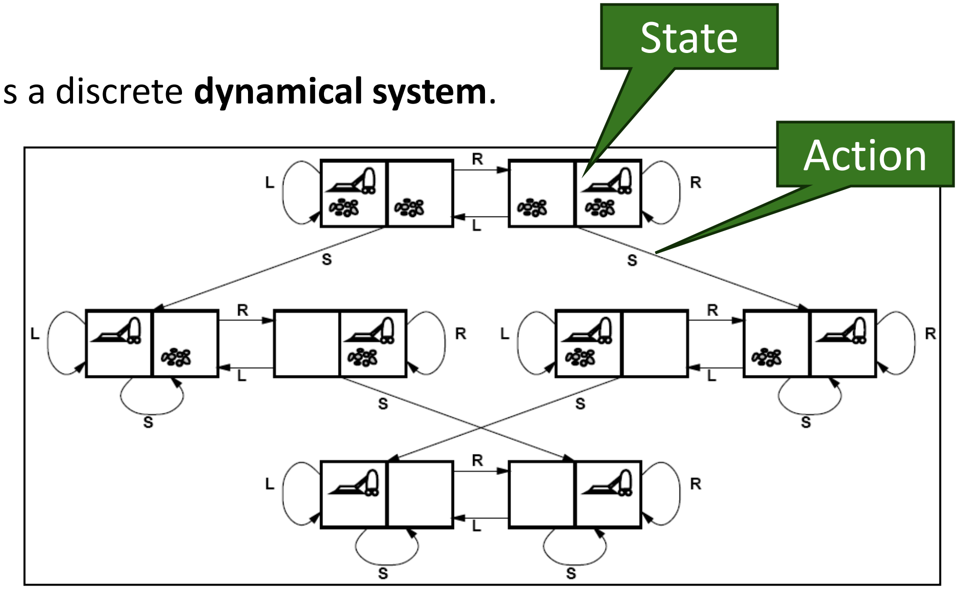


We often construct atomic labels from factored information. E.g.: If the agent’s state is the coordinate x = 7 and y = 3, then the atomic state label could be the string “(7, 3)”. With the atomic representation, we can only compare if two labels are the same. With the factored state representation, we can reason more and calculate the distance between states!

The set of all possible states is called the **state space S**. This set is typically very large!

Transition Function

- The environment is modeled as a discrete **dynamical system**.
- Example of a state diagram for the Vacuum cleaner world.



- States change because of
  a. System dynamics of the environment (the environment evolves by itself).
  b. The actions of the agent.

- Both types of changes are represented by the transition function written as

$$T: S \times A \rightarrow S \quad \text{or} \quad s' = T(s, a)$$

S ... set of states
A ... set of available actions
$a \in A$ ... an action
$s \in S$ ... current state
$s' \in S$ ... next state

Case Study: Self-Driving Cars

State Representation: Self-Driving Car

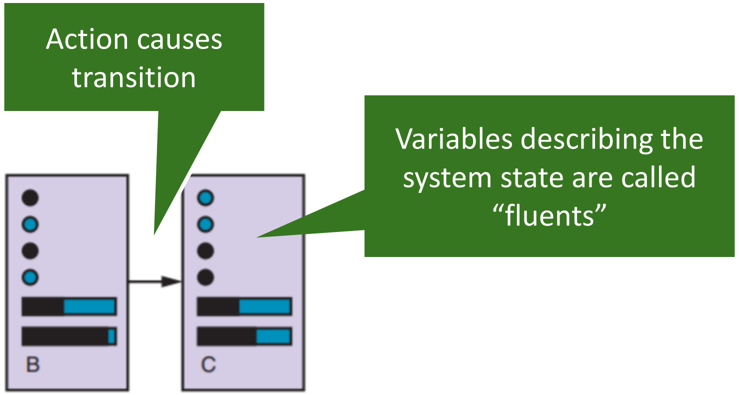
States help to keep track of the environment and the agent in the environment.

Design a structured representation for the state of a self-driving car.

a) What fluents should it contain?

b) What actions can cause transitions?

c) Draw a small transition diagram.



(b) Factored

Environment for a Self-Driving Car

☐ **Fully observable:** The agent has access to the complete current **state** of the environment.
Has deterministic percepts that are 100% reliable.

vs.

☐ **Partially observable:** The agent's sensors provide incomplete or noisy information about the **state** of the environment.
Noisy information means unreliable stochastic percepts (aka a stochastic sensor model)

☐ **Deterministic:**
Deterministic **transition function**: Changes in the environment are completely determined by the current state of the environment and the agent's action.

vs.

☐ **Stochastic:**
Stochastic **transition function:** leads to transition probabilities and a Markov process.

☐ **Known:** The agent knows the **transition function**.

vs.

☐ **Unknown:** The needs to **learn the transition function** by trying actions.

☐ **Static:** The environment is **not** changing while agent is deliberating.

vs.

☐ **Dynamic:** The environment is changing while the agent is deliberating.

☑ Check what applies and explain what it means for a self-driving car.

What Type of Intelligent Agent is a Self-Driving Car?

Is it learning? ☐

☐ Utility-based agents

Does it collect utility over time? How would the utility for each state be defined?

☐ Goal-based agents

Does it have a goal state?

☐ Model-based reflex agents

Does it store state information. How would they be defined (atomic/factored)?

☐ Simple reflex agents

Does it use simple rules based on the current percepts?

☑ Check what applies

Why is this so hard?

- Self-driving cars operate in a very complicated partially observable, stochastic, and dynamic environment.
- Can only use bounded rationality because of limits with sensors and computational power.
- Require a set of different agents that cooperate.

What You Should Know

- What an **agent function**
  $$action = f(percepts)$$
  is and how it interacts with the environment.
- What are **states** and what is the **transition function**?
- How **environments** differ in terms of observability, uncertainty (stochastic behavior), and if the transition function is known.
- How to identify different **types of agents**.

Assignment 1

Reflex Agents