Artificial Intelligence

Instructor: Kietikul Jearanaitanakij

Department of Computer Engineering

King Mongkut’s Institute of Technology Ladkrabang

- Textbook: Artificial Intelligence A Modern Approach, 3^{rd} Edition
  By Stuart Russell and Peter Norvig

- Grading Criteria:
  - Programming Assignments 50% (5 students per group)
    - Assignment 1 25%
    - Assignment 2 25%
  - Final Exam 50%

Lecture 1

- What is AI?
- Intelligent agents
- Agent program
- Properties of environments

What is Artificial Intelligence (AI) ?

- We call ourselves “Homo sapiens” – man the wise – because our intelligence is so important to us.
- AI encompasses a huge variety of subfields, ranging from the general to the specific, such as playing chess, proving math theorems, writing poetry, driving a car, and diagnosing diseases.

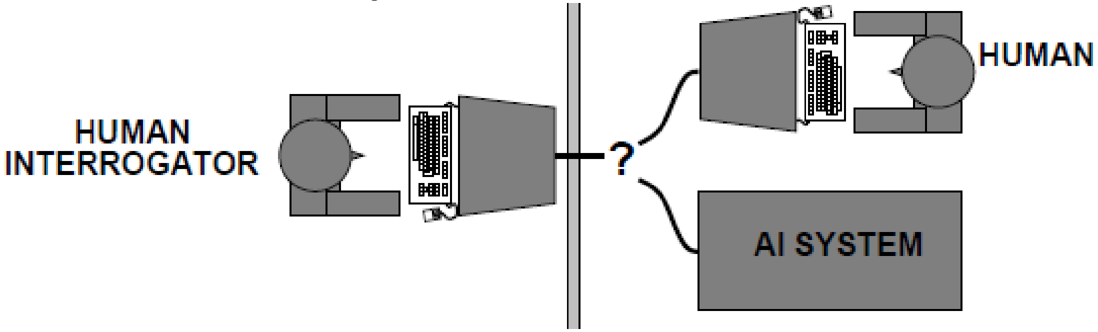
- Definition of AI can be organized into four categories.

Thinking Humanly	**Thinking Rationally**
Acting Humanly * Need the Turing test	**Acting Rationally**

Acting Humanly: The Turing Test approach

- The Turing Test, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence.

- A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.



To take the Turing Test, the computer needs the following capabilities:

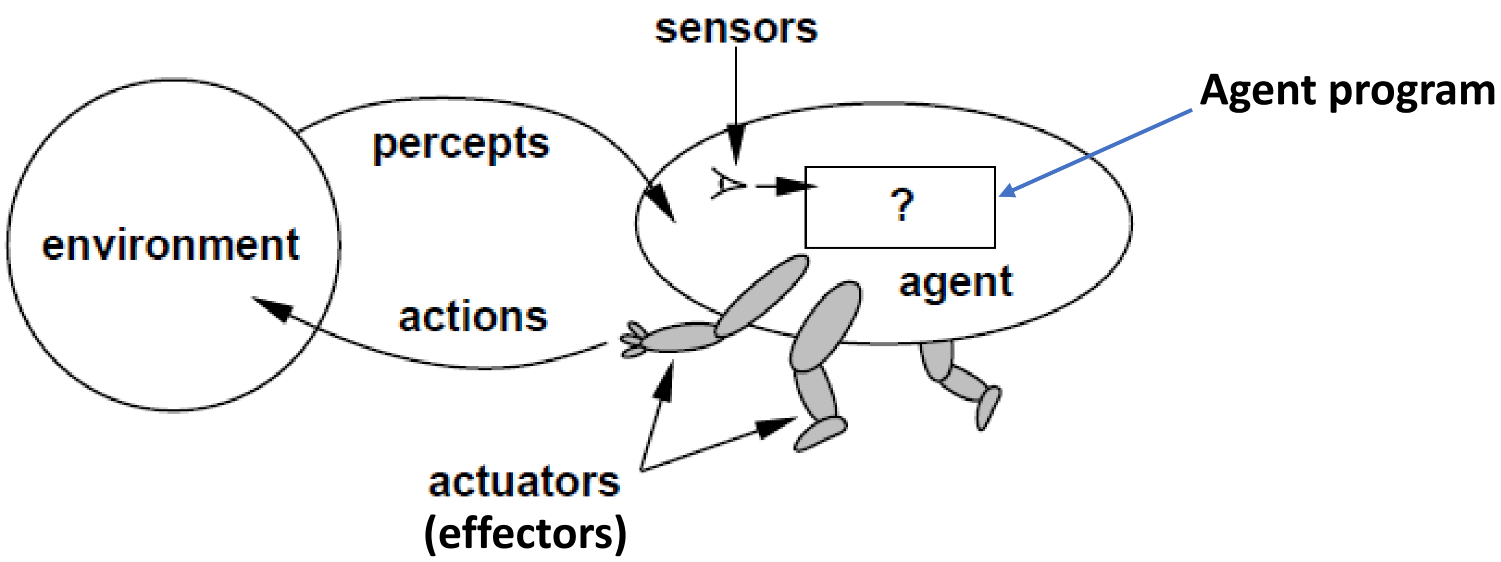
- Natural language processing
- Knowledge representation (transform fact to sentence)
  - Ex. Today is hot => IS(Today, Hot)
- Automated reasoning
  - Ex. Bigger(A, B) and Bigger(B,C) => Bigger(A,C)
- Machine learning

To pass the **total** Turing Test, the computer will need

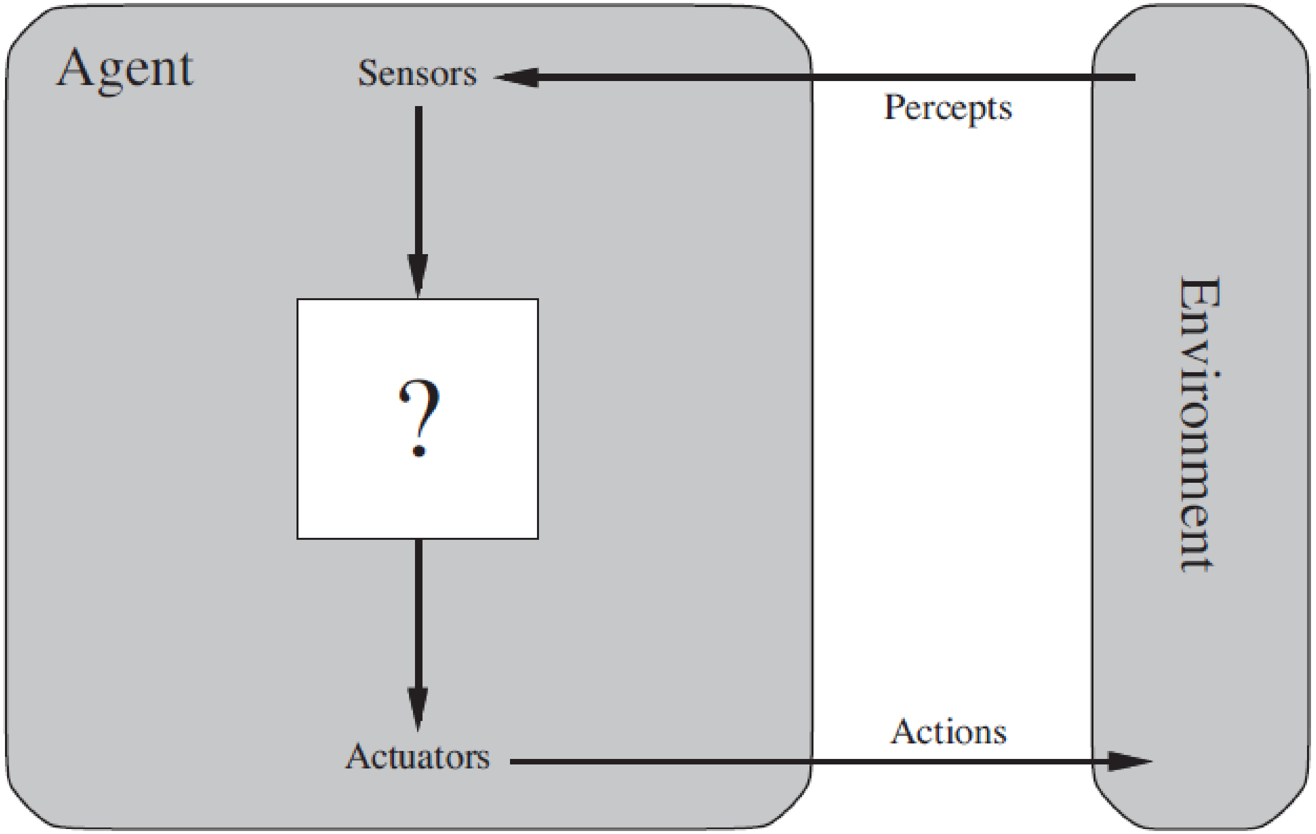
- Computer vision
- Robotics

Intelligence Agents

Agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.



Overview of Agents



- **Omniscience** VS **Rationality**

  Omniscience is the capacity to know everything that there is to know.

  Rationality is the quality or state of being rational.

- We would need our agent to be rational rather than omniscience.
- A rational agent is one that does the right thing (most successful).

- Rationality depends on four things
  1. Performance measure.
  2. Everything that the agent has perceived so far (**percept sequence**).
  3. What the agent knows about the environment.
  4. The actions that the agent can perform.

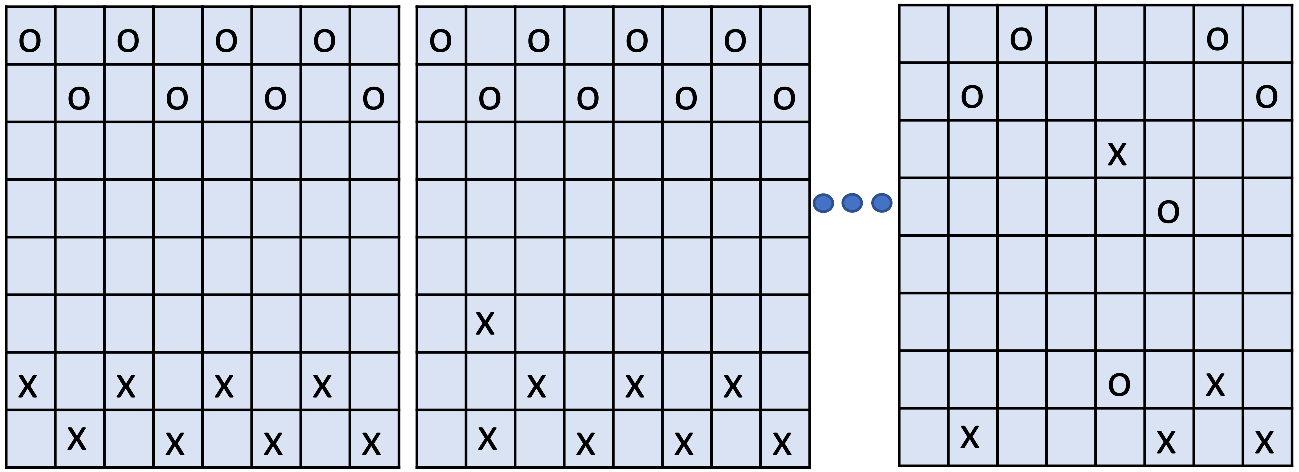
Agent Program

0) Table lookup (won't be counted as agent program)

Keeping in memory its entire percept sequence, and using it to index into table, which contains the appropriate action for all possible percept sequences.

Percept sequence	Action
P3, P1, P4, P2, P1, P4, P2, …, Pi	A2
P1, P4, P6, P2, P3, P5, P3, …, Pj	A4
P3, P1, P7, P10, P3, P4, P2, …, Pk	A1
P2, P6, P4, P2, P1, P4, P9, …, Pn	A3
…	…

Example: Checker



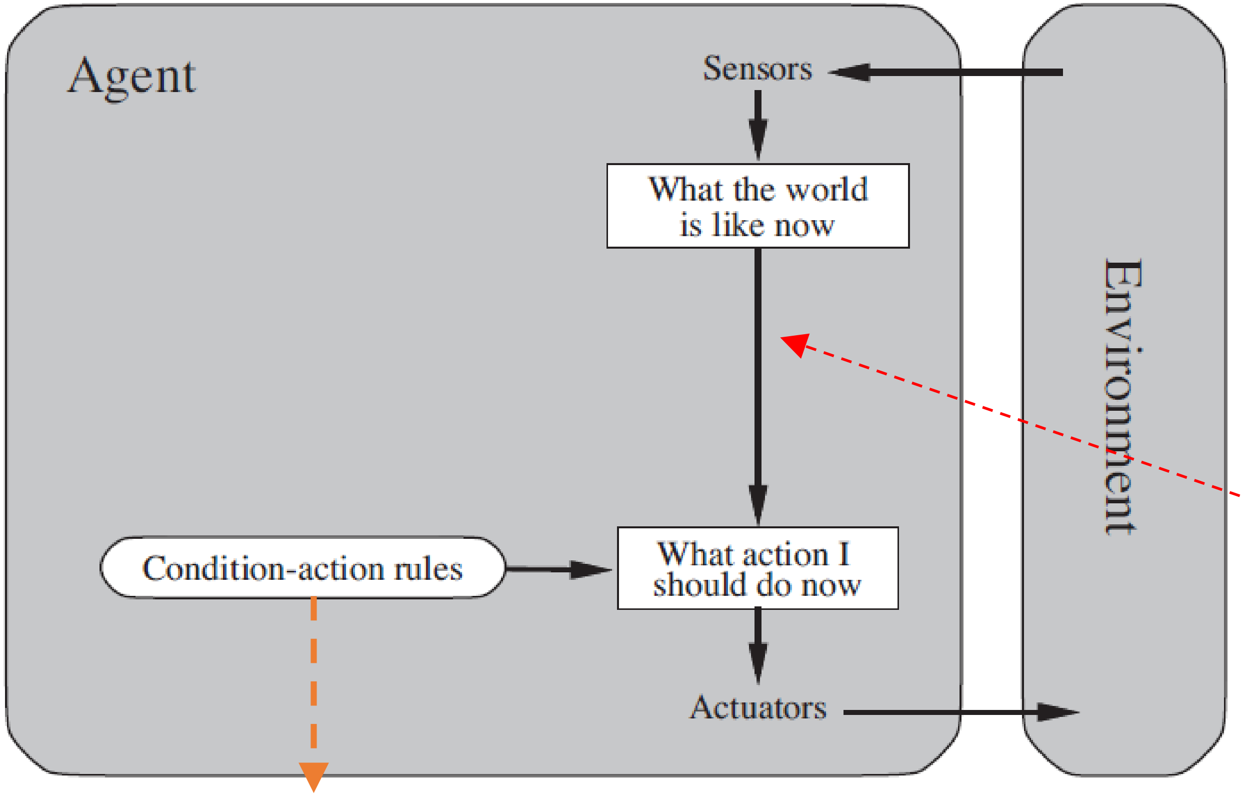
P1 P4 Pj

Disadvantages

1. Space (Ex: Chess need 35^{100} entries $\approx$ 2.55E+154)
2. Time (Construction & Execution)
3. No autonomy (If the percept sequence doesn't perfectly match, it take no action)
4. Take forever for agent to learn the table entries

Agent Program

1. Simple Reflex Agents



A simple reflex agent acts according to a rule whose condition matches the current state, as defined by the percept.

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

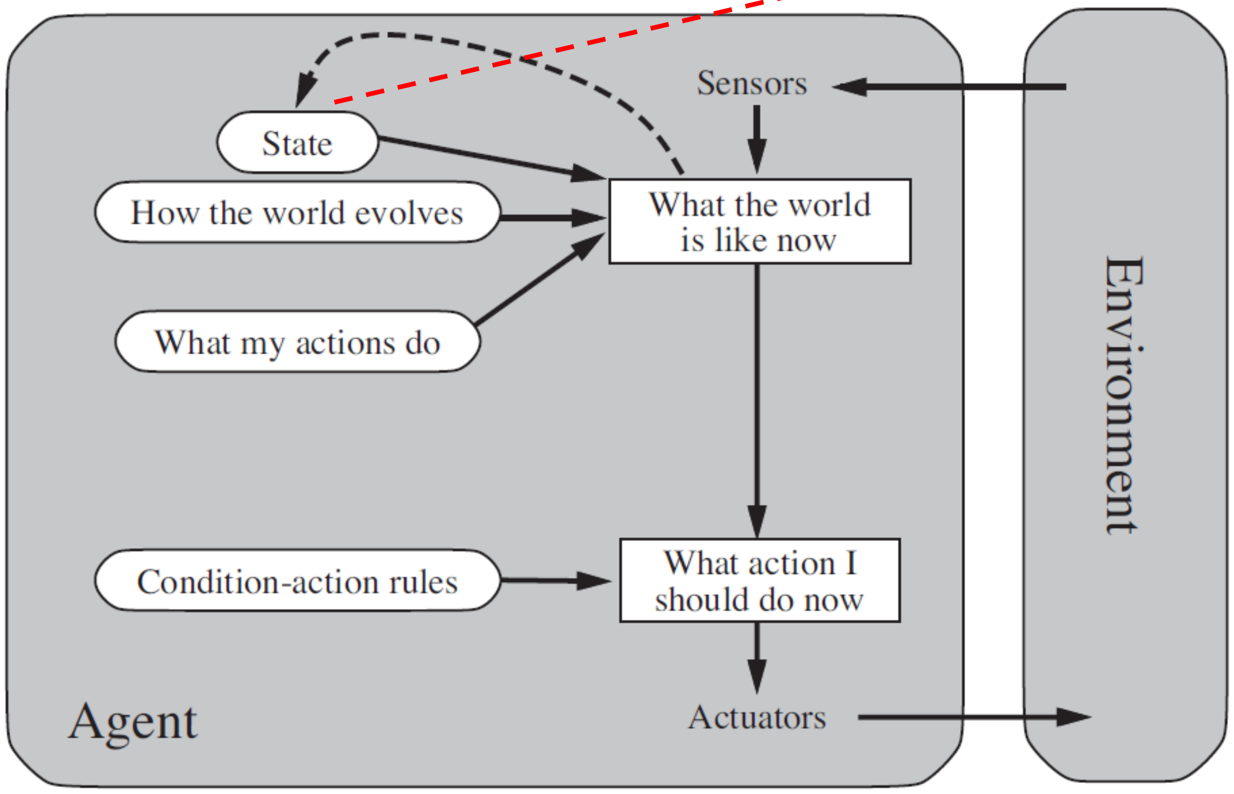
  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

IF car-in-front-is-braking **THEN** Initiate-braking

State (defined by percept)

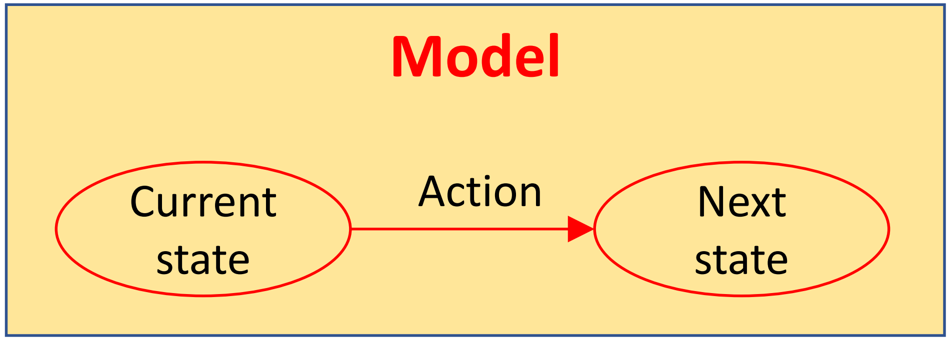
Agent Program

2. Model-based Reflex Agents



- Some problems need to handle partial observability from the past (**internal state**) which cannot see now.
- For the braking problem, we need the previous frame from the camera, allowing the agent to detect when two red breaking lights of the car go on/off simultaneously.

Agent Program



2. Model-based Reflex Agents (Cont.)

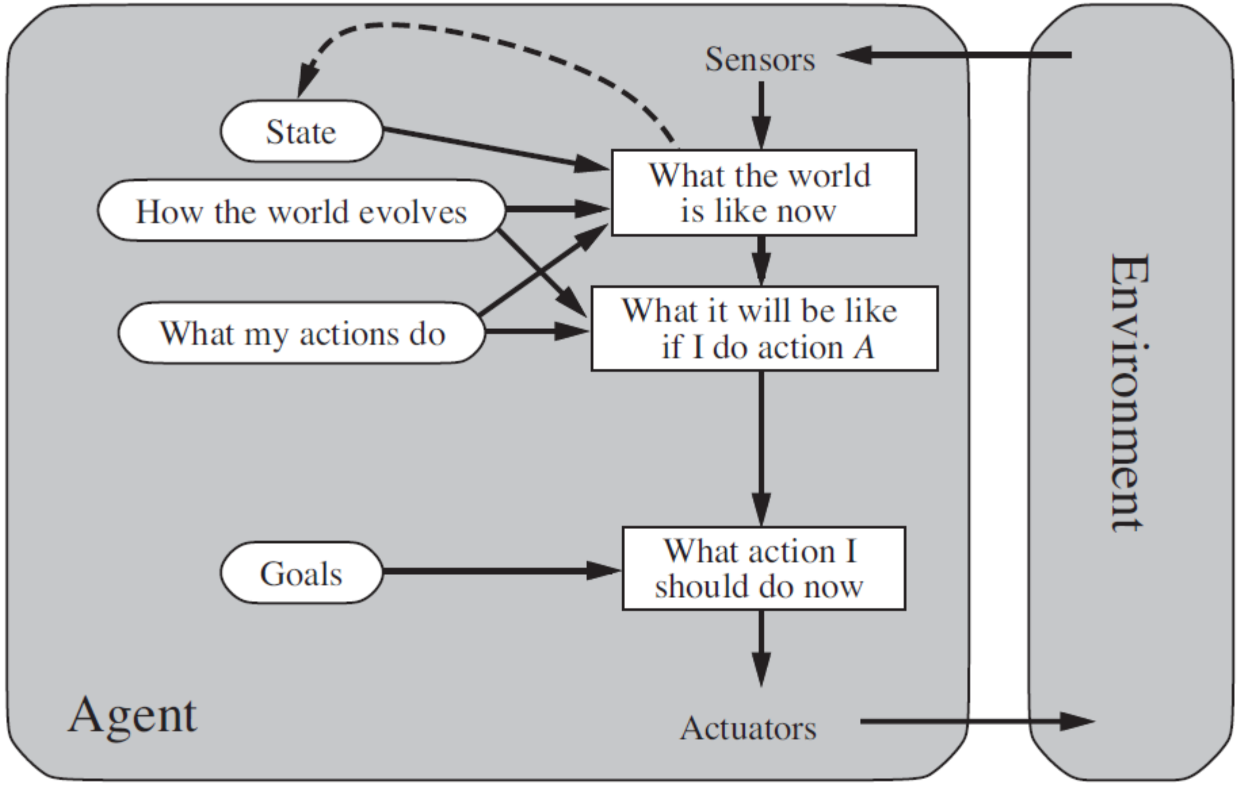
function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action, initially none

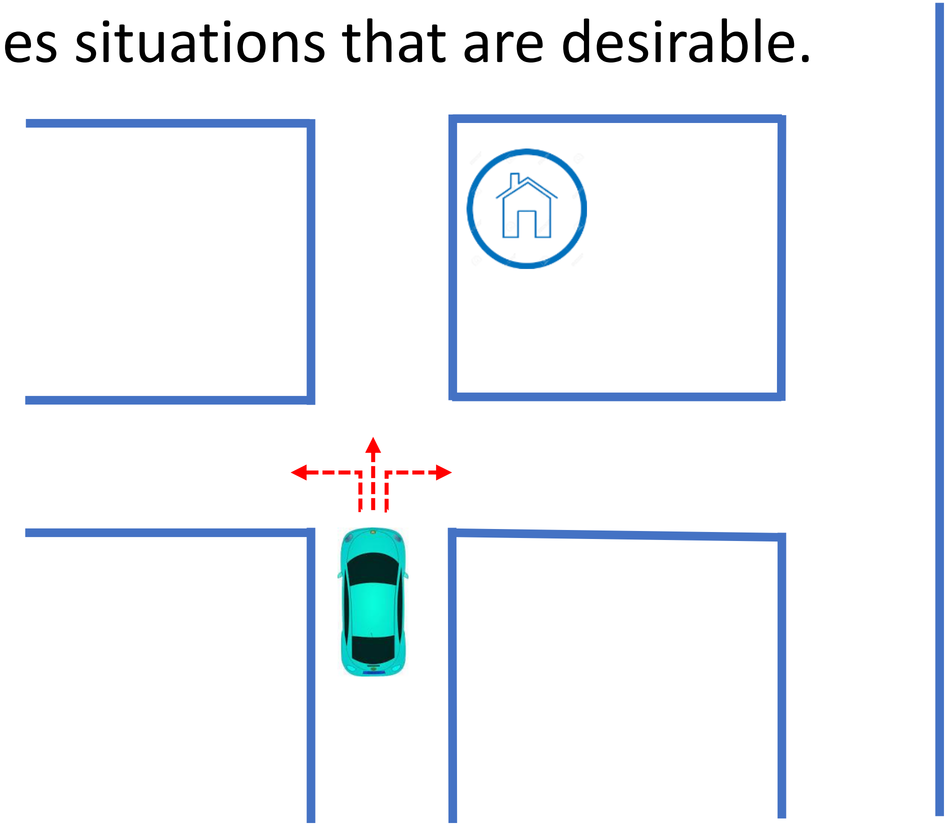
  state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Agent Program

3. Goal-based Agents

When the agent can do more than one action at the current state, it needs some sort of **goal** information that describes situations that are desirable.

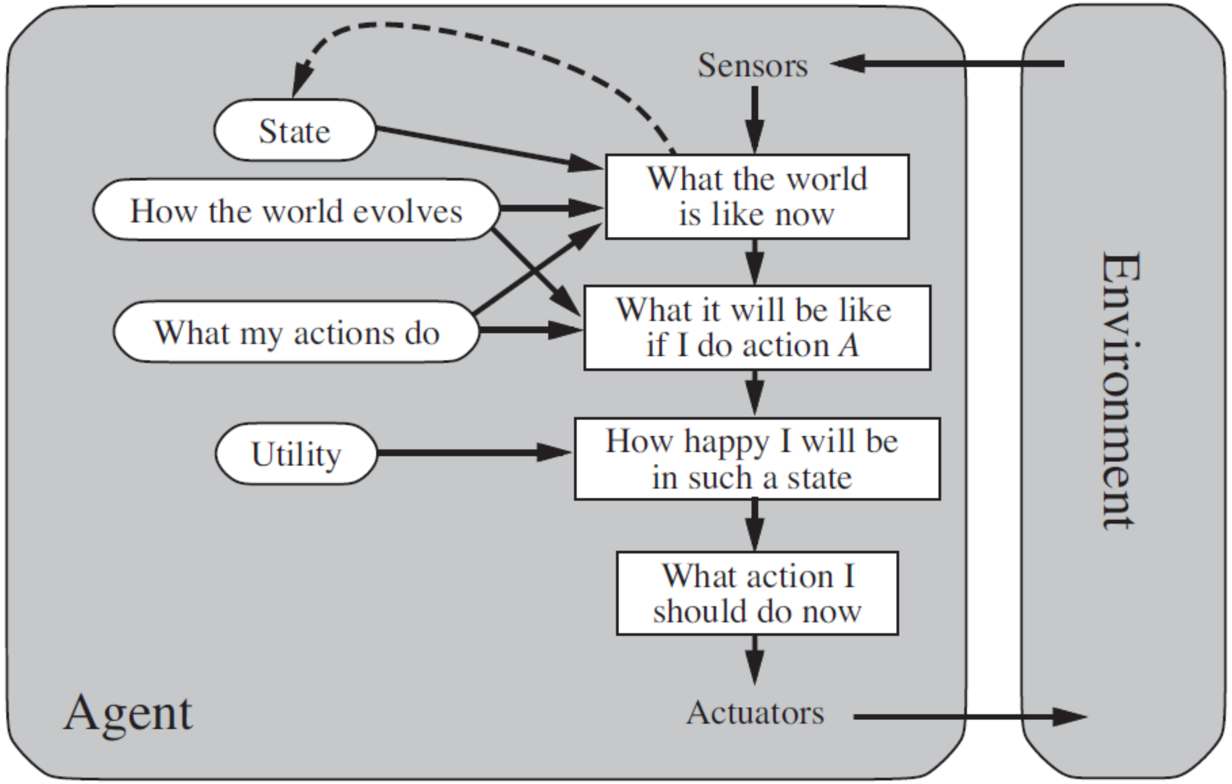


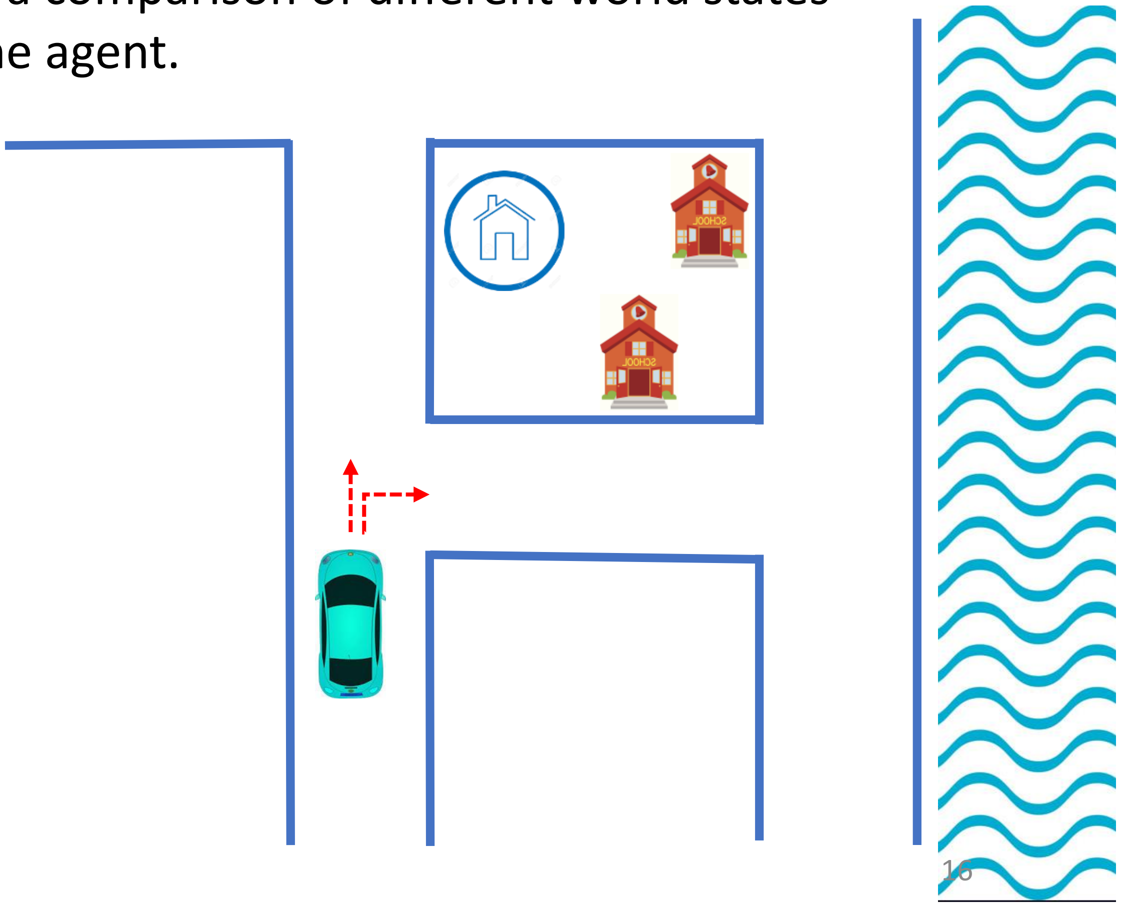


Agent Program

4. Utility-based Agents

A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent.





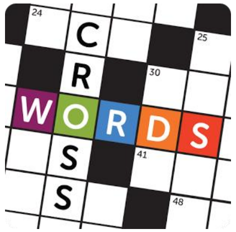
Properties of Environments

- **Fully observable** vs. **partially observable**

If an agent's sensors give it access to the complete state of the environment (that are relevant to the choice of action) at each point in time, then we say that the environment is **fully observable**.

An environment might be **partially observable** because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

- **Single agent** vs. **multiagent**

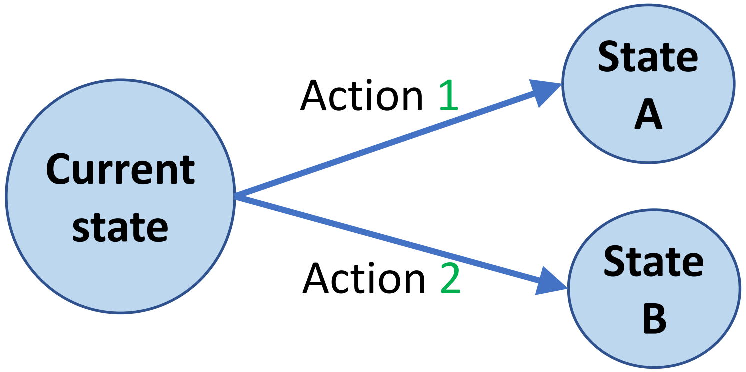


Properties of Environments

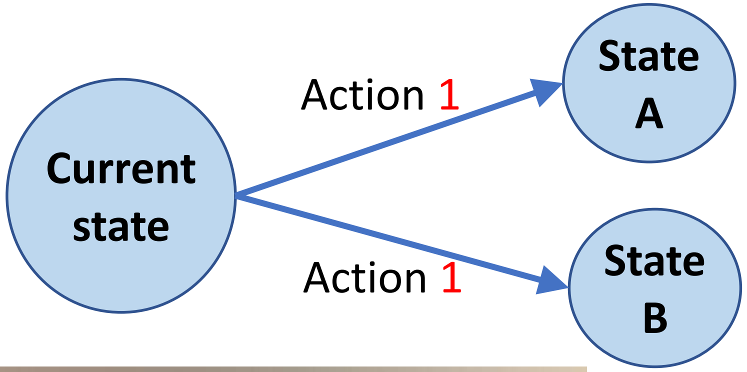
- **Deterministic** vs. **stochastic**

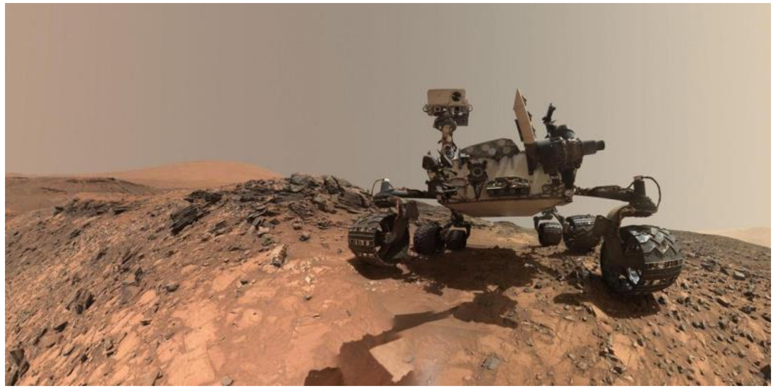
If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

Deterministic



NonDeterministic





Properties of Environments

- **Episodic** **vs.** **sequential**

Episodic: The agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action. Crucially, the next episode does not depend on the actions taken in previous episodes.

Episodic: Robot in assembly line

Sequential: Taxi driver

Properties of Environments

- **Static** vs. **dynamic**

If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

Static : Backgammon

Dynamic : Part picking robot

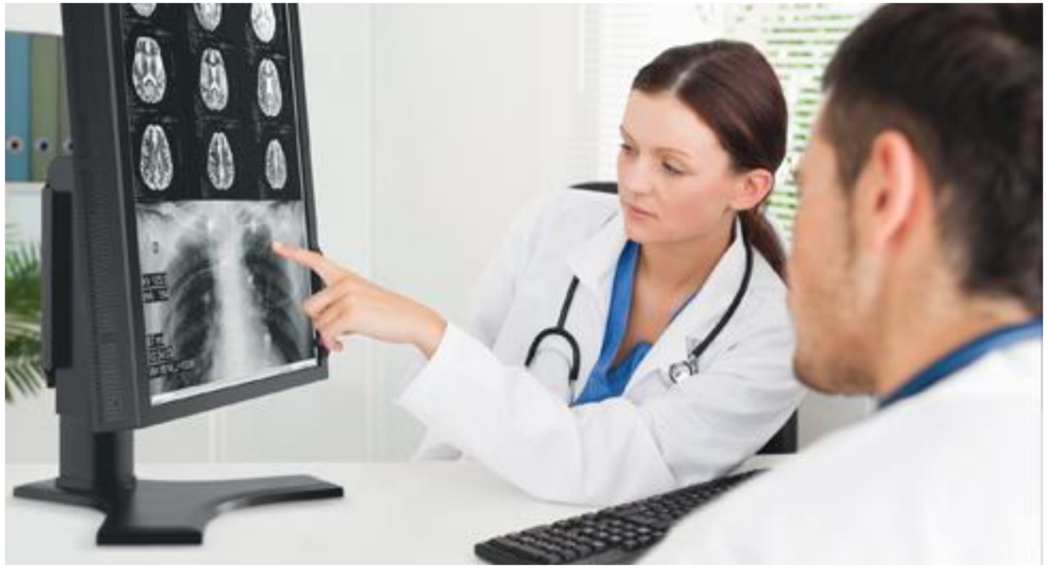
Properties of Environments

- **Discrete** vs. **continuous**

The environment is discrete if there are a limited number of distinct states, percepts, and actions. Otherwise, it is continuous.

Discrete

Continuous : Medical diagnosis



Properties of Environments

- **Known** vs. **unknown**

  In a known environment, the outcomes for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions.

Known : Checker

Unknown : Crazy machine game