IFA-305 Sistem Cerdas (Intelligent System)
Lecture 2

Intelligent Agent

Nur Uddin, PhD.

Program Studi Informatika

Universitas Pembangunan Jaya

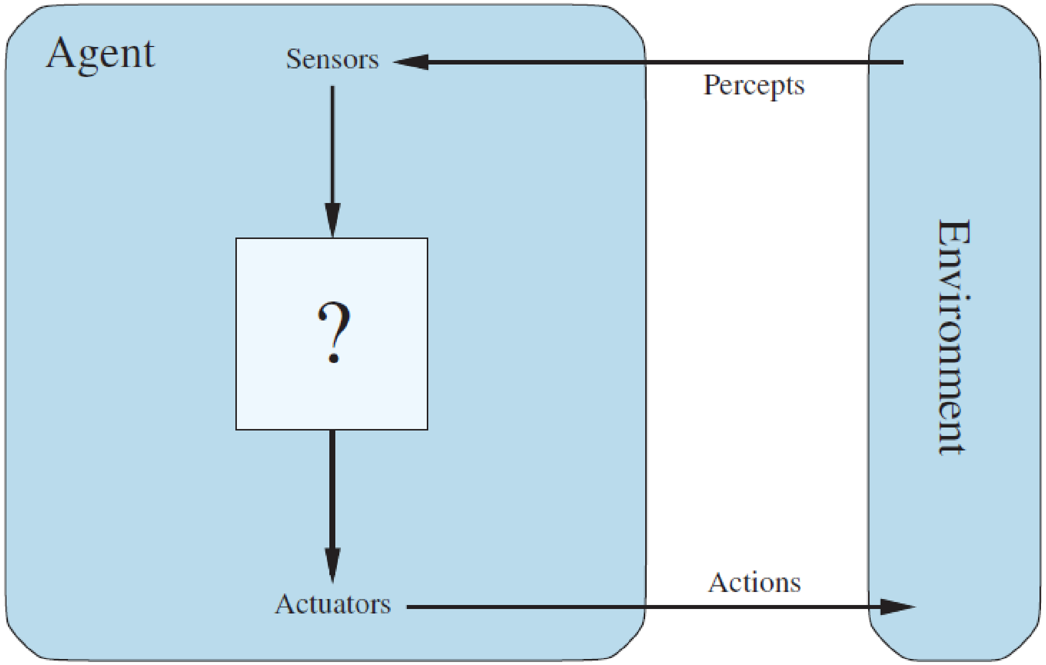
Tangerang Selatan

Outline

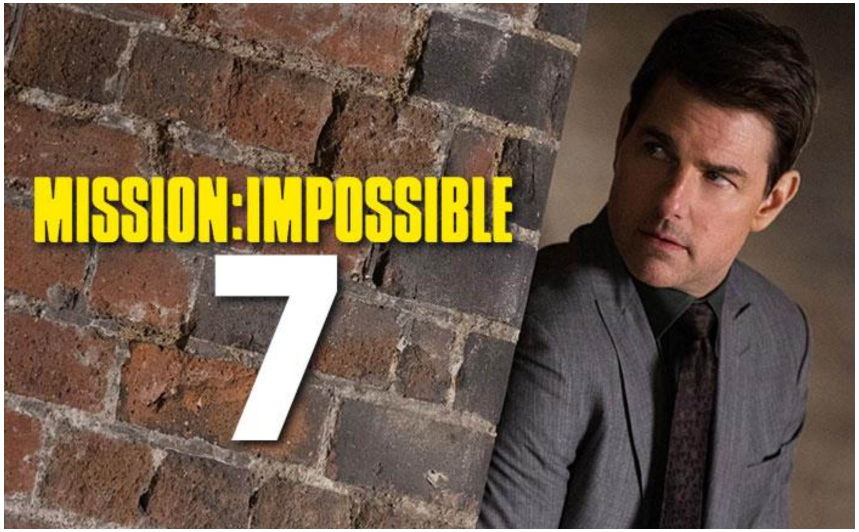
- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

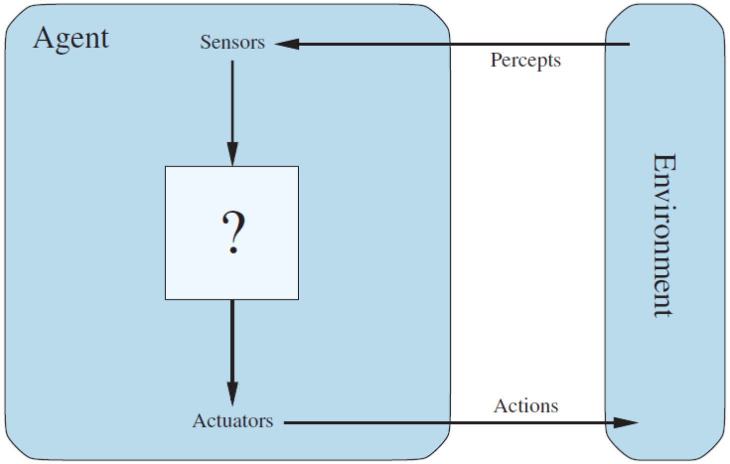
Agent

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



Human as Agent

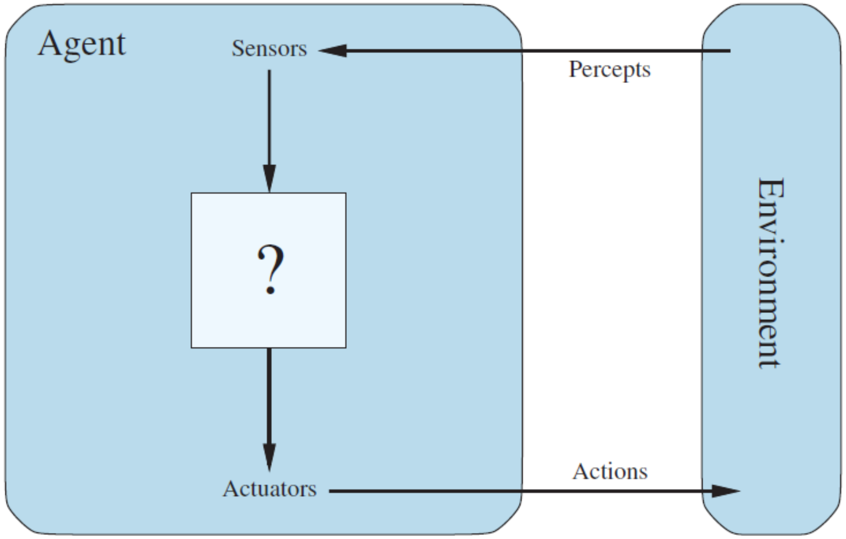




A human agent

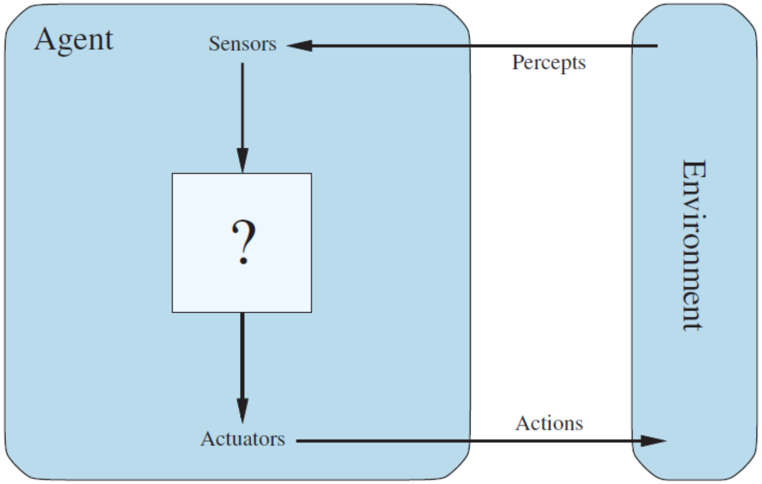
- Sensors: eyes, ears, and other organs for sensors
- Actuator: hands, legs, vocal tract, and so on for actuators.

Robot as Agent



A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators.

Automatic Door as Agent

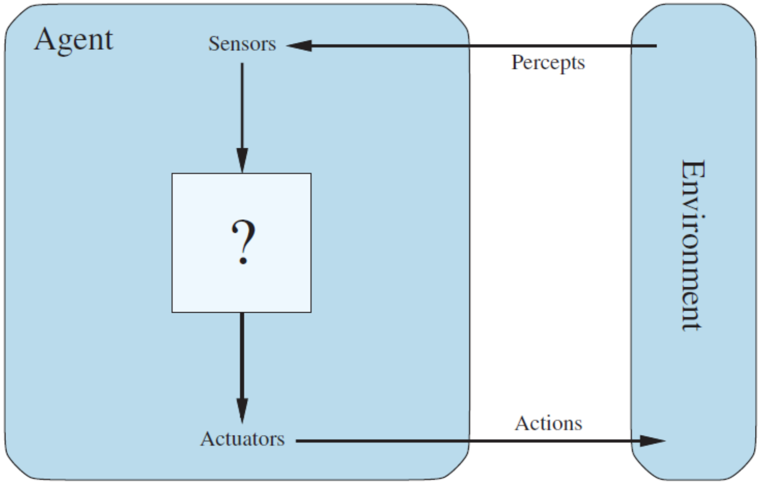


Sensors:

Actuator:

Gerbang Toll Otomatis (GTO) - #1

Is it an agent ?



Gerbang Toll Otomatis (GTO) - #2

Sensors:
NFC (Near Field Communication) card reader

Actuator: electric motor

Sensors:
RFID (Radio-frequency identification)

Actuator: electric motor

Software as Agent

- A software agent receives file contents, network packets, and human input (keyboard/mouse/touchscreen/voice) as sensory inputs and acts on the environment by writing files, sending network packets, and displaying information or generating sounds.

- The environment could be everything—the entire universe! In practice it is just that part of the universe whose state we care about when designing this agent—the part that affects what the agent perceives and that is affected by the agent's actions.

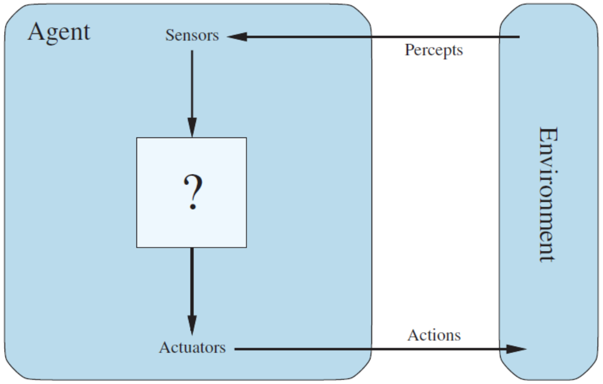
Percept

- We use the term percept to refer to the content an agent's sensors are perceiving.
- An agent's percept sequence is the complete history of everything the agent has ever perceived.
- In general, an agent's choice of action at any given instant can depend on its built-in knowledge and on the entire percept sequence observed to date, but not on anything it hasn't perceived.

Agent Behavior

- By specifying the agent's choice of action for every possible percept sequence, we have said more or less everything about the agent.

- Mathematically speaking, we say that an agent's behavior is described by the agent function that maps any given percept sequence to an action.



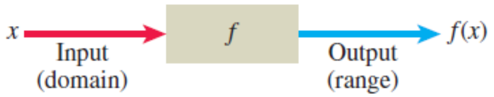


FIGURE 1.1 A diagram showing a function as a kind of machine.

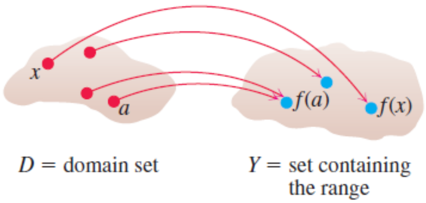
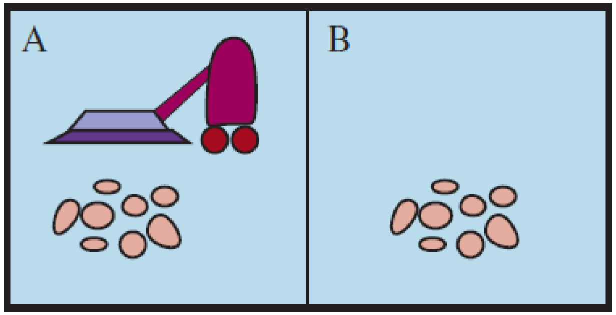


FIGURE 1.2 A function from a set D to a set Y assigns a unique element of Y to each element in D.

Agent Function

- We can imagine tabulating the agent function that describes any given agent; for most agents, this would be a very large table (infinite), unless we place a bound on the length of percept sequences we want to consider.
- Given an agent to experiment with, we can, construct this table by trying out all possible percept sequences and recording which actions the agent does in response.
- The table is an external characterization of the agent. Internally, the agent function for an artificial agent will be implemented by an agent program.
- It is important to keep these two ideas distinct. The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

Example: a robotic vacuum-cleaning (#1)

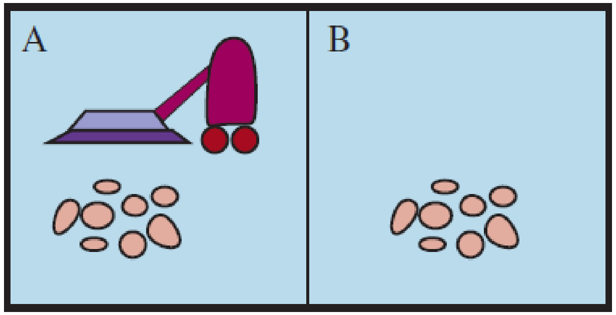


- The vacuum agent perceives which square it is in and whether there is dirt in the square.

- The agent starts in square A. The available actions are to move to the right, move to the left, suck up the dirt, or do nothing.

- One very simple agent function is the following:
  - if the current square is dirty, then suck;
  - otherwise, move to the other square.

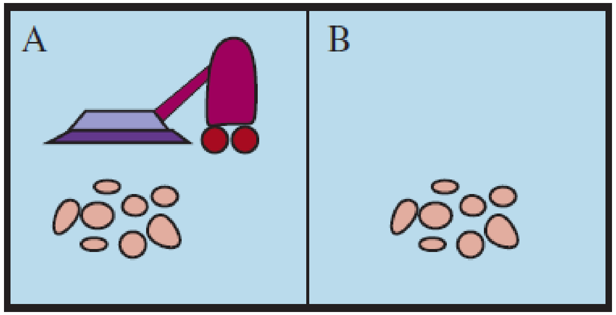
Example: a robotic vacuum-cleaning (#2)



- A partial tabulation of this agent function is shown as follows:

Percept sequence	Action
[A, *Clean*]	*Right*
[A, *Dirty*]	*Suck*
[B, *Clean*]	*Left*
[B, *Dirty*]	*Suck*
[A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮
[A, *Clean*], [A, *Clean*], [A, *Clean*]	*Right*
[A, *Clean*], [A, *Clean*], [A, *Dirty*]	*Suck*
⋮	⋮

Example: a robotic vacuum-cleaning (#3)



- Looking at previous table, we see that various vacuum-world agents can be defined simply by filling in the right-hand column in various ways.
- The obvious question, then, is this: What is the right way to fill out the table?
- In other words, what makes an agent good or bad, intelligent or stupid?

Good Behavior:
The Concept of Rationality

- A rational agent is one that does the right thing. Obviously, doing the right thing is better than doing the wrong thing, but what does it mean to do the right thing?

- Moral philosophy has developed several different notions of the “right thing,” but AI has generally stuck to one notion called consequentialism: we evaluate an agent’s behavior by its consequences.

Performance Measures (#1)

- When an agent is plunked down in an environment, it generates a sequence of actions according to the percepts it receives. This sequence of actions causes the environment to go through a sequence of states. If the sequence is desirable, then the agent has performed well.

- This notion of desirability is captured by a performance measure that evaluates any given sequence of environment states.

Performance Measures (#2)

- Humans have desires and preferences of their own, so the notion of rationality as applied to humans has to do with their success in choosing actions that produce sequences of environment states that are desirable from their point of view.

- Machines, on the other hand, do not have desires and preferences of their own; the performance measure is, initially at least, in the mind of the designer of the machine, or in the mind of the users the machine is designed for.

- We will see that some agent designs have an explicit representation of (a version of) the performance measure, while in other designs the performance measure is entirely implicit—the agent may do the right thing, but it doesn’t know why.

Performance Measures (#3)

- Recalling Norbert Wiener's warning to ensure that "the purpose put into the machine is the purpose which we really desire" (page 51), notice that it can be quite hard to formulate a performance measure correctly.

- As a general rule, it is better to design performance measures according to what one actually wants to be achieved in the environment, rather than according to how one thinks the agent should behave.

Rationality

What is rational at any given time depends on four things:

- The performance measure that defines the criterion of success.
- The agent's prior knowledge of the environment.
- The actions that the agent can perform.
- The agent's percept sequence to date.

Definition of rational agent:

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

Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful

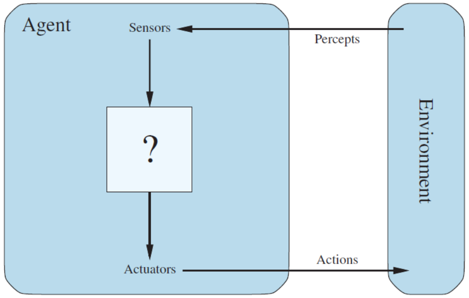
- Performance measure: An objective criterion for success of an agent's behavior.

- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge).
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration).
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt).

Environment



- Task environments are the “problems” to which rational agents are the “solutions.”

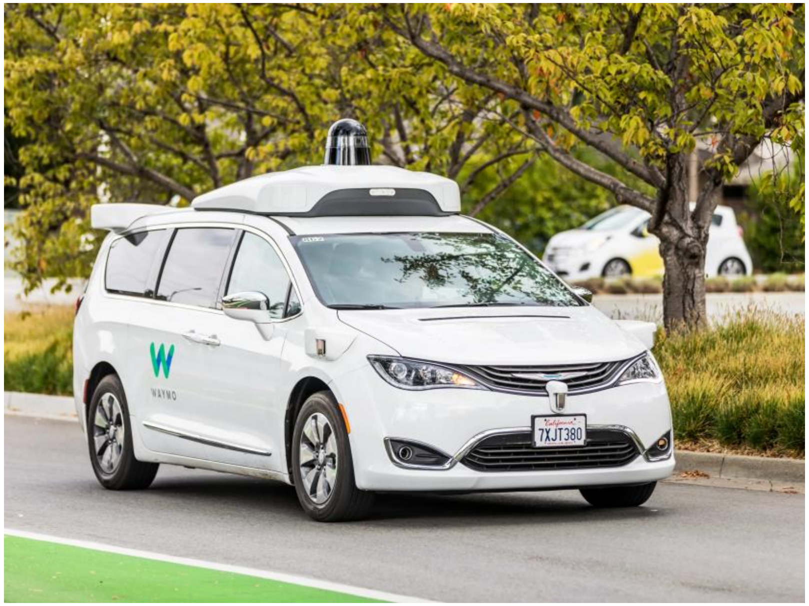
- In our discussion of the rationality of the simple vacuum-cleaner agent, we had to specify the performance measure, the environment, and the agent’s actuators and sensors.

- We group all these under the heading of the task environment. For the acronymically minded, we call PEAS this the PEAS (Performance, Environment, Actuators, Sensors) description.

- In designing an agent, the first step must always be to specify the task environment as fully as possible.

PEAS of automated taxi driver

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen



Environment types (#1)

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.

- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic).

- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Environment types (#2)

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating.
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- Single agent (vs. multiagent): An agent operating by itself in an environment.

Agent functions and programs

- An agent is completely specified by the <u>agent function</u> mapping percept sequences to actions.

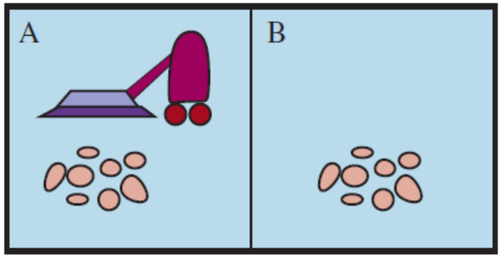
- One agent function (or a small equivalence class) is <u>rational.</u>

- Aim: find a way to implement the rational agent function concisely.

Table-lookup agent

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

Agent program for a vacuum-cleaner agent



Percept sequence	Action
[*A*, *Clean*]	*Right*
[*A*, *Dirty*]	*Suck*
[*B*, *Clean*]	*Left*
[*B*, *Dirty*]	*Suck*
[*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

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

Figure 2.8 The agent program for a simple reflex agent in the two-location vacuum environment. This program implements the agent function tabulated in Figure 2.3.