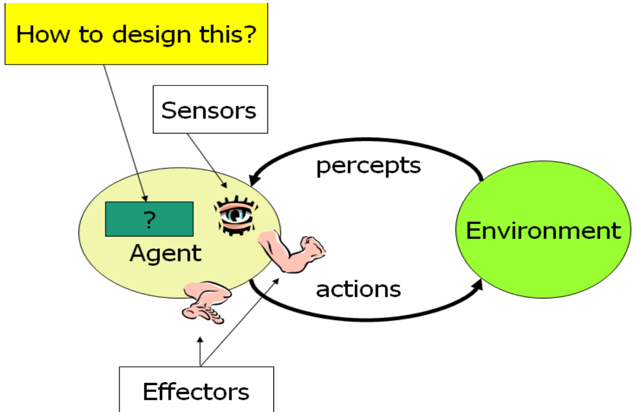
Intelligent Agents

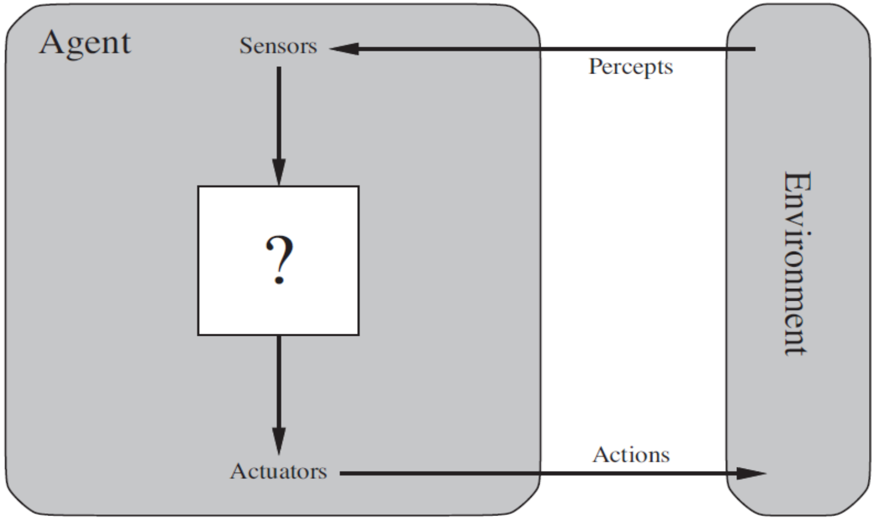
Learning Goals

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

Agents and Environments

Agents





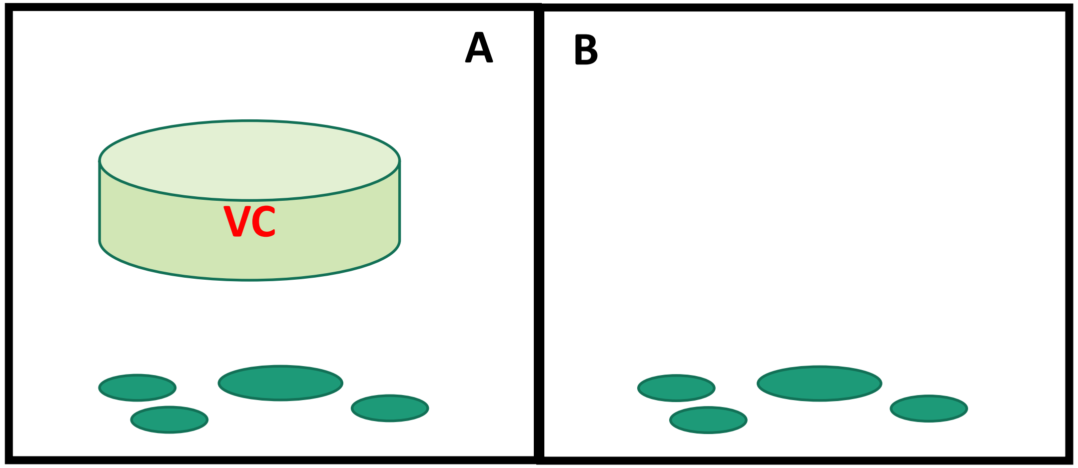
- An **agent (代理人)** is anything that can be viewed as **perceiving(覺察)** its **environment** through **sensors (感測器)** and **acting** upon that environment through **actuators(作用器)**
- **Human agent:**
  - eyes, ears, and other organs for sensors;
  - hands, legs, mouth, and other body parts for actuators
- **Robotic agent:**
  - cameras and infrared range finders for sensors;
  - various motors for actuators
- **Software agent:**
  - receives keystrokes, file contents, and network packets as sensory inputs and
  - displaying on the screen, writing files, and sending network packets.

http://aima.cs.berkeley.edu/figures.html

Agents

- percept sequence
  - the complete history of everything the agent has ever perceived
  - an agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived.
- agent function
  - maps any given percept sequence to an action.
  - *tabulating* the agent function
  - very large table—infinite
  - abstract mathematical description
- agent program
  - concrete implementation

Vacuum-cleaner world



http://aima.cs.berkeley.edu/figures.html

- **geography of the environment (two locations A and B)**
- **Percepts: location and contents, e.g., [A, Dirty]**
- **Actions: *Left, Right, Suck, NoOp***
- **agent function: if the current square is dirty, then suck; otherwise, move to the other square.**
- **The performance measure awards one point for each clean square at each time step, over a "lifetime" of 1000 time steps.**

Simple Agent Function for Vacuum Cleaner World

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], [B,Clean]	Left
[A,Clean], [B,Dirty]	Suck
[A,Dirty], [A,Clean]	Right
[A,Dirty], [A,Dirty]	Suck
[A,Clean], [A,Clean], [A,Clean]	Right

http://aima.cs.berkeley.edu/figures.html

Good Behavior: The Concept Of Rationality (理性)

- **An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform by considering the *consequences* of the agent's behavior.**
- **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**
- **When to use performance measure? (long period of time)**

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 environment.
  - The actions that the agent can perform.
  - The agent's percept sequence(感知序列) to date.
- 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.

Definition: Rational agents

- **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.**

- **A better agent for this case would do nothing once it is sure that all the squares are clean, if dirty then move and clean, …**

Omniscience (全知的)

- omniscience
  - all-knowing with infinite knowledge
  - an omniscient agent knows the *actual* outcome of its actions and can act accordingly; it is impossible in reality
  - Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration, learning)
- Rationality ≠ perfection.
  - Rationality maximizes *expected* performance
  - Perfection maximizes *actual* performance.
  - if we expect an agent to do what turns out to be the best action after the fact, it will be impossible to design an agent to fulfill this specification
  - depends only on the percept sequence *to date*.
  - haven’t inadvertently allowed the agent to engage in decidedly under intelligent activities.

Autonomy (自 治)

- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt) and prior knowledge.
- A rational agent should be autonomous—it should learn what it can to compensate for partial or incorrect prior knowledge.
- Agent must learn from what it perceives
  - initial configuration reflects prior knowledge of environment
  - agent gains experience so that knowledge of environment may be modified/augmented

The Right Thing = The Rational Action

- Rational Action: The action that **maximizes the expected value** of the performance measure <u>given the percept sequence to date</u>

  - Rational = Best Yes, to the best of its knowledge
  - Rational = Optimal Yes, to the best of its abilities
    (incl. its constraints)
  - Rational ≠ Omniscience (全知)
  - Rational ≠ Clairvoyant (有洞察力)
  - Rational ≠ Successful

The Nature of Environment

- Task Environment: PEAS
  - Performance measure
  - Environment
  - Actuators
  - Sensors

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, engine sensors, keyboard

http://aima.cs.berkeley.edu/figures.html

Examples of Agent Types and their Descriptions

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

http://aima.cs.berkeley.edu/figures.html

Environment types

- **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)**
  - **stochastic/uncertain/nondeterministic**
- **Episodic(情節不連貫的,插曲) (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.**
  - **Sequential: Game playing/driving**

Environment types

- **Static (vs. dynamic):** The environment is unchanged while an agent is deliberating. (The environment is **semidynamic** if the environment itself does not change with the passage of time but the agent's performance score does)
- **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.
  - Competitive
  - Cooperative
- Know vs. unknown

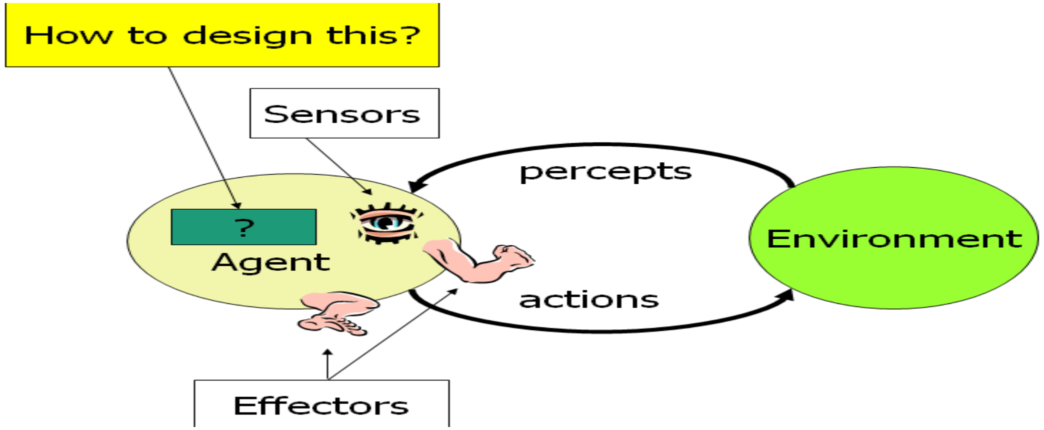
Examples of task environments and their characteristics.

Task Environment	**Observable**	**Agents**	**Deterministic**	**Episodic**	**Static**	**Discrete**
Crossword puzzle	fully	single	deterministic	sequential	static	discrete
Chess with a clock	fully	multi	deterministic	sequential	semi	discrete
Poker	partially	multi	stochastic	sequential	static	continuous
Taxi driving	Partially	multi	stochastic	sequential	dynamic	Continuous
Image analysis	fully	single	deterministic	episodic	dynamic	Continuous

http://aima.cs.berkeley.edu/figures.html

The structure of Agents

- The job of AI is to design an agent program that implements the agent function-the mapping from percepts to actions.
- We assume this program will run on some sort of computing device with physical sensors and actuators—we call this the architecture.
- Agent = architecture + program
- 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



Agents Programs vs. Agent functions

- **Agent program takes the current percept as input from the sensors and return an action to the actuators.**
- **Agent program, which takes the current percept as input (with or without perceptions memory)**
- **Agent function, which takes the entire percept history.**
- **Agent program: the implementation of $f : P^* \rightarrow A$, the agent's perception (P*)-action (A) mapping**

Table-Driven Agent

Table-Driven Agent

```
function TABLE-DRIVEN-AGENT (percept) returns action
  static: percepts, a sequence, initially empty
          table, a table, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```

https://github.com/aimacode/aima-pseudocode/blob/master/aima3e-algorithms.pdf

- **Construct action table**
- **Drawbacks:**
  - **Huge table, storage size/how to construct**
  - **Take a long time to build the table**
  - **No autonomy,**
  - **Even with learning, need a long time to learn the table entries**

Table-Driven agent

- P = the set of possible percepts
- T= lifetime of the agent
  - The total number of percepts it receives
- Size of the look up table $\sum_{t=1}^{T} |P|^t$
- Consider playing chess
  - P =10, T=150
  - Will require a table of at least 10^{150} entries

Table-Driven agent

- Despite of huge size, look up table does what we want.
- The key challenge of AI
  - Find out how to write programs that, to the extent possible, produce rational behavior
    - From a small amount of code
    - Rather than a large amount of table entries
  - E.g., a five-line program of Newton's Method
  - v.s. huge tables of square roots, sine, cosine, …

Agent types

Agent types

- Five basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents
  - Learning Agents

Reflex-Vacuum-Agent

- **The simplest kind of agent is the simple reflex agent.**
- **These agents select actions on the basis of the *current* percept, ignoring the rest of the percept history.**

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** $Left$

https://github.com/aimacode/aima-pseudocode/blob/master/aima3e-algorithms.pdf

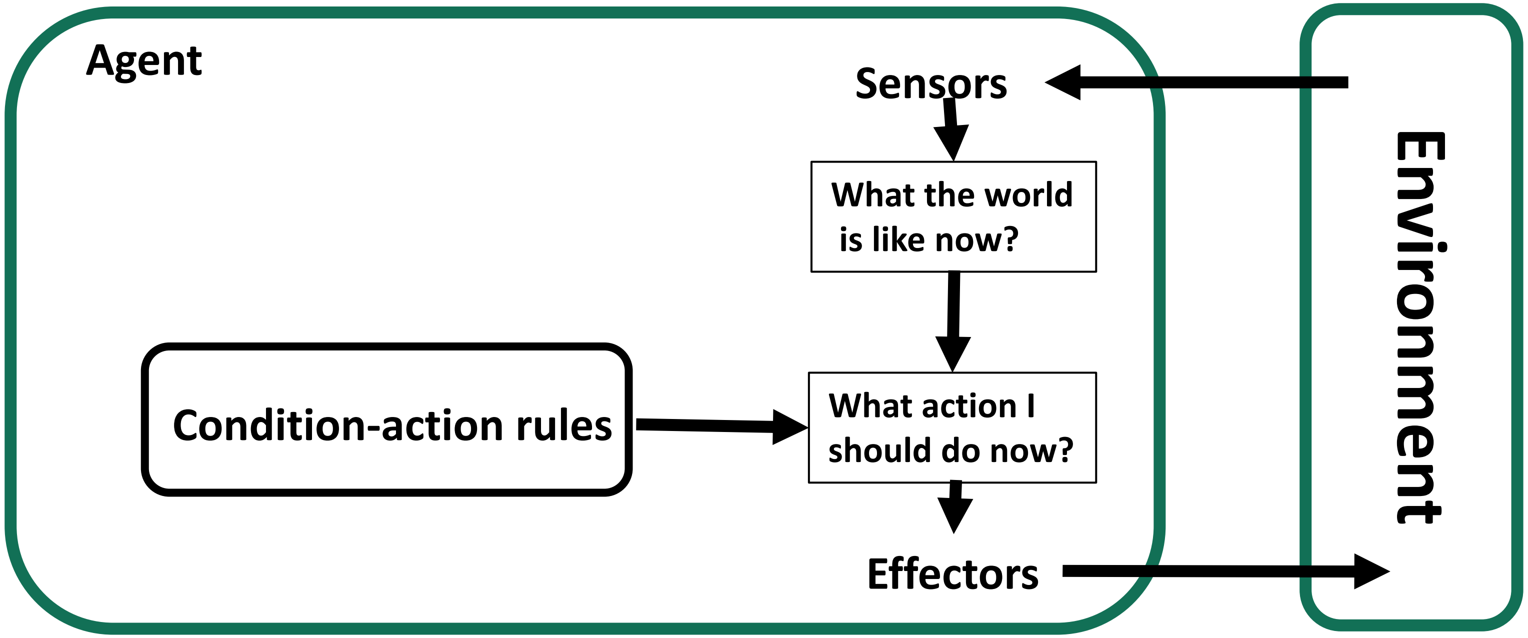
- **because its decision is based only on the current location and on whether that location contains dirt (status).**

Table size 4^T

Simple-reflex-Agent

- **Select actions on the basis of *current* percept, ignoring the rest of the percept history.**
- **Eg.**
  - **Vacuum Agent**
  - **Car driver (barking light).**
- **Condition-action rule written as**
  - **if *car-in-front-is-braking* then *initiate-braking***
- **Humans also have many such connections, some of which are learned responses (as for drive and some of which are innate reflexes (such as, blinking when something approaches the eye).**

Simple-reflex-Agent



http://aima.cs.berkeley.edu/figures.html

function REFLEX-AGENT-With-STATES (*percept*) returns action
static: *rules*, a set of condition-action rules
***state* ← INTERPRET-INPUT (*percept*)**
***rule* ← RULE-MATCH (*state, rules*)**
***action* ← *rule*.ACTION**
return *action*

https://github.com/aimacode/aima-pseudocode/blob/master/aima3e-algorithms.pdf

Simple-reflex-Agent

- Reactive agents do not have internal symbolic models.
- Act by stimulus-response to the current state of the environment.
- Each reactive agent is simple and interacts with others in a basic way.
- The agent will work *only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable.*
- Even a little bit of unobservability can cause serious trouble (eg. car braking rule).
- Complex patterns of behavior emerge from their interaction.
- Benefits: robustness, fast response time
- Challenges: scalability, how intelligent? and how do you debug them?

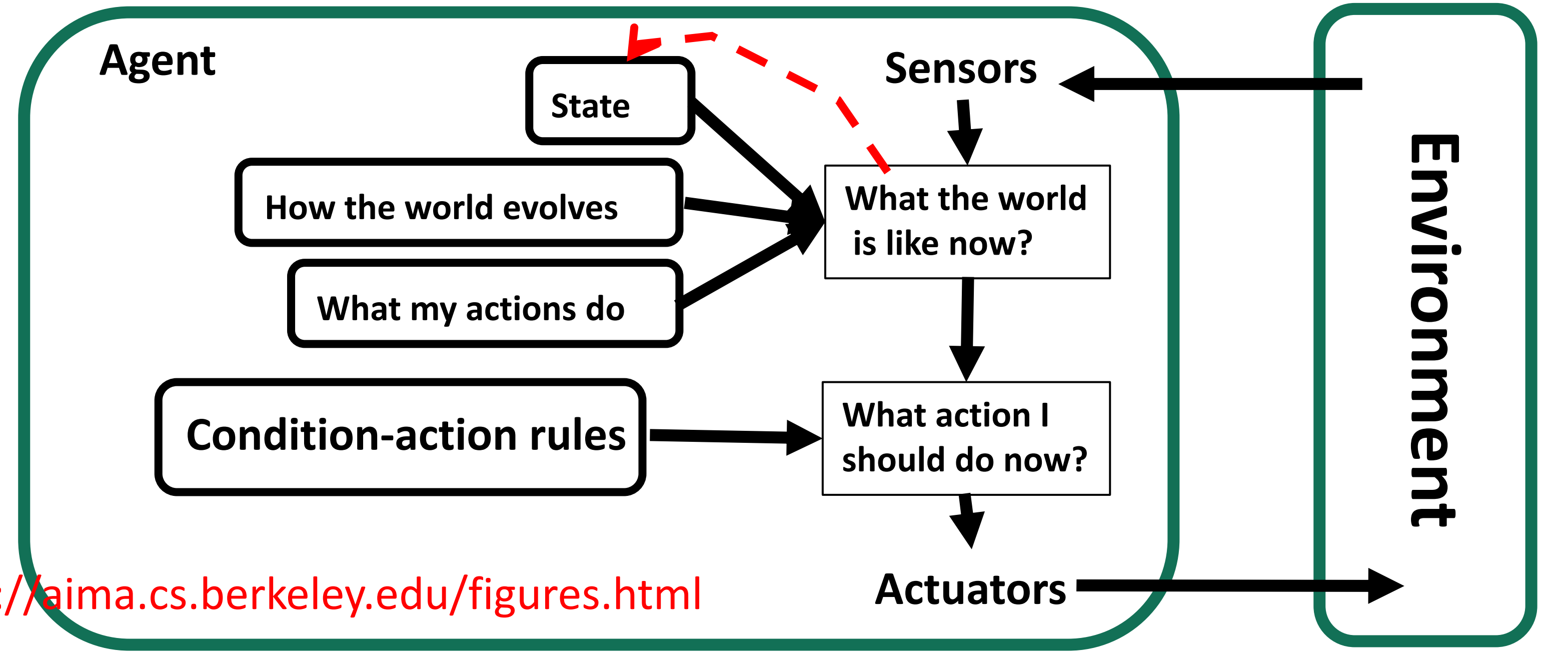
Problem?

- Remove location sensor from Vacuum agent?
- Different environment?
- Escape from infinite loops is possible if the agent can randomize its actions.
- Hence, a randomized simple reflex agent might outperform a deterministic simple reflex agent.

A Model-based Reflex agent

- Thus, even for the simple braking rule, our driver will have to maintain some sort of internal state in order to choose an action.
- Here, the internal state is not too extensive—it just needs the previous frame from the camera to detect when two red lights at the edge of the vehicle go on or off simultaneously.
- requires two kinds of knowledge to be encoded in the agent program.
  - how the world evolves independently of the agent
  - how the agent's own action affect the model of the world
- Word of model

A model-based reflex agent



http://aima.cs.berkeley.edu/figures.html

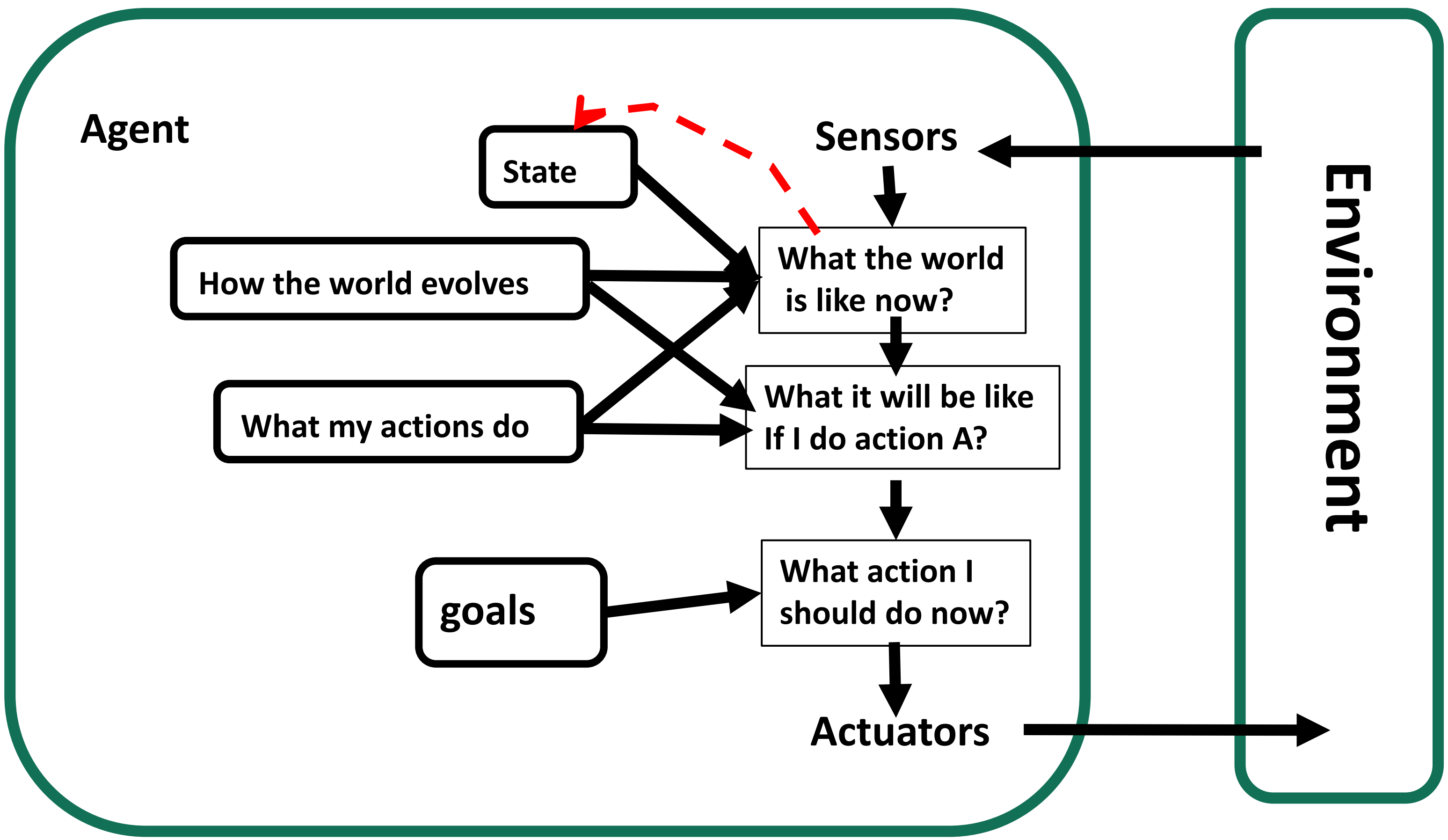
function MODEL-BASED-REFLEX-AGENT(percept) **returns** an action
persistent: *state*, the agent's current conception of the world state
model , a description of how the next state depends on current state and action
rules, a set of condition–action rules
action, the most recent action, initially none
State ← UPDATE-STATE(state, action, percept, model)
Rule ← RULE-MATCH(state, rules)
action ← rule.ACTION
return action

https://github.com/aimacode/aima-pseudocode/blob/master/aima3e-algorithms.pdf

Goal-based Agent

- The agent needs some sort of goal information, which describes situations that are desirable.
  - Taxi: the passenger's destination.
- The agent program can combine this information about the results of possible actions (the same information as was used to internal state in the reflex agent) in order to choose actions that achieve the goal.
- Sometime this will be simple, when goal satisfaction results immediately from a single action; sometime it will be more tricky, when the agent has to consider long sequences of twists and turns to.

Goal-based agents



http://aima.cs.berkeley.edu/figures.html

Goal-based agents

- Conclusion
  - Goal-based agents are less efficient
  - but more flexible
    - Agent ← Different goals ← different tasks
  - Search and planning
    - two other sub-fields in AI
    - to find out the action sequences to achieve its goal

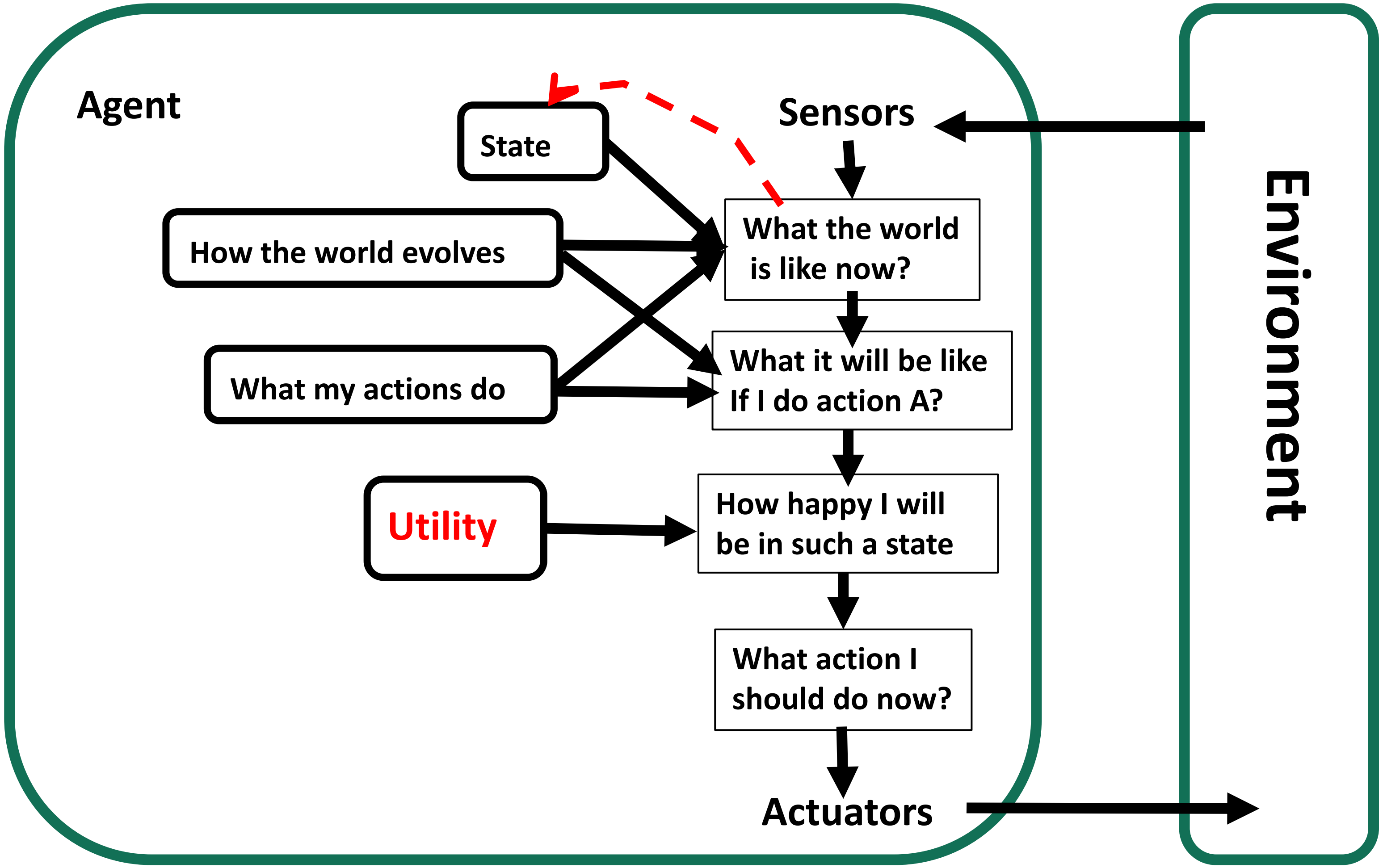
Utilities-based agents

- Goals alone are not really enough to generate high-quality behavior.
- There are action sequences that will get the taxi to its destination, thereby achieving the goal, but so are quicker, safer, more reliable, or cheaper than others.
- Goals just provide a crude distinction between "happy" and "unhappy" states, whereas a more general performance measure should allow a comparison of different world states (or sequences of states) according to exactly how happy they would make the agent if they could be achieved.
- Utility function (效用函數) map a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.

Utility-based agents

- Goals alone are not enough
  - to generate high-quality behavior
  - E.g. meals in Canteen, good or not ?
- Many action sequences → the goals
  - some are better and some worse
  - If goal means success,
  - then utility means the degree of success (how successful it is)
- it is said state A has higher utility
  - If state A is more preferred than others
- Utility is therefore a function
  - that maps a state onto a real number
  - the degree of success

Utility and Model-based agents



http://aima.cs.berkeley.edu/figures.html

Utility-based agents

- **Utility has several advantages:**
  - **When there are conflicting goals (speed/safe),**
    - **Only some of the goals but *not* all can be achieved**
    - **utility describes the appropriate trade-off**
  - **When there are several goals**
    - **None of them are achieved <u>certainly</u>**
    - **utility provides a way for the decision-making**

Learning Agents

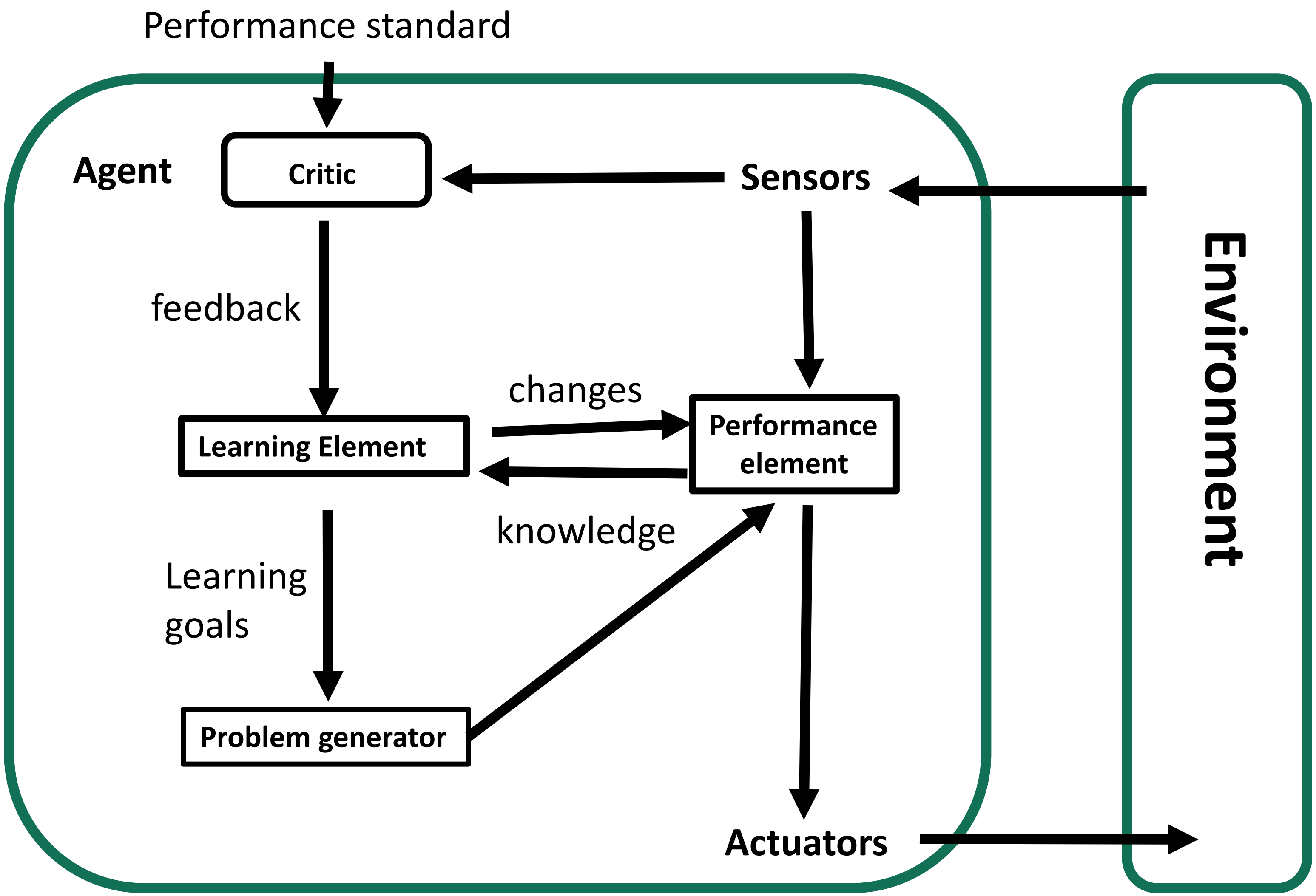
- After an agent is programmed, can it work immediately?
  - No, it still need teaching
- In AI,
  - Once an agent is done
  - We teach/train it by giving it a set of examples
  - Test it by using another set of examples
- We then say the agent learns
  - A learning agent

Learning Agents

- Four conceptual components
  - Learning element
    - Making improvement
  - Performance element
    - Selecting external actions
  - Critic (批評者)
    - Tells the Learning element how well the agent is doing with respect to fixed performance standard.

    (Feedback from user or examples, good or not?)
  - Problem generator
    - Suggest actions that will lead to new and informative experiences.

Learning agents



http://aima.cs.berkeley.edu/figures.html

Summary

- **Intelligent Agents:**
  - **Anything that can be *viewed as* perceiving its environment through sensors and acting upon that environment through its effectors to maximize progress towards its goals.**
  - **PAGE (Percepts, Actions, Goals, Environment)**
  - **Described as a Perception (sequence) to Action Mapping:**

  $f : \mathcal{P}^* \rightarrow \mathcal{A}$

  - **Using look-up-table, closed form, etc.**
- **Agent Types: Reflex, state-based, goal-based, utility-based**
- **Rational Action: The action that maximizes the expected value of the performance measure <u>given the percept sequence to date</u>**