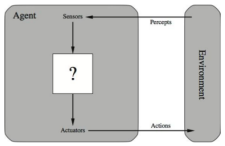
Agents and Environments

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



- An agent program runs in cycles of: **(1) perceive, (2) think, and (3) act.**
- **Agent** = **Architecture** + **Program**
  complementary & compatible

感知机 { 感知：传感器 → 环境；思考；行为：驱动器 → 环境

- **Human agent:**
  - Sensors: eyes, ears, and other organs.
  - Actuators: hands, legs, mouth, and other body parts.
- **Robotic agent:**
  - Sensors: Cameras and infrared range finders.
  - Actuators: Various motors.

- **Agents everywhere!**
  - Thermostat
  - Cell phone
  - Vacuum cleaner
  - Robot
  - Self-driving car
  - Apple Siri
  - Human
  - etc.

Well-behaved 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.*"

Rationality

- Rationality is relative to a **performance measure**.
- Judge rationality based on:
  - The **performance** measure that defines the criterion of success.
  - The agent prior knowledge of the **environment**.
  - The possible **actions** that the agent can perform.
  - The agent' s **percept** sequence to date.

理性与性能度量相关.

PEAS

PEAS

- When we define a rational agent, we group these properties under **PEAS**, the problem specification for the task environment.
- The rational agent we want to design for this task environment is the solution.
- PEAS stands for:
  - Performance
  - Environment
  - Actuators
  - Sensors

Problems (PEAS)
↓找到
Solutions (Agent)

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)
- **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.
- **Static (vs. dynamic):** The environment is unchanged while an agent is deliberating. (The environment is semi-dynamic 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. E.g., checkers is an example of a discrete environment, while self-driving car evolves in a continuous one.
- **Single agent (vs. multi-agent):** An agent operating by itself in an environment.
- **Known (vs. Unknown):** The designer of the agent may or may not have knowledge about the environment makeup. If the environment is unknown, the agent will need to know how it works in order to decide.

完全/部分观测：传感器是否能感受到环境的完整状态.

确定/随机：环境的下一状态是否由当前状态确定.

片断/连续：感知机的经历是否分成片断，每一片断中的行为都仅依赖于该片断.

静态/动态

离散/连续

单agent/多agent

已知/未知

Example

Environment	Observable	Agents	Deterministic	Static	Discrete
8-puzzle	Fully	Single	Deterministic	Static	Discrete
Chess	Fully	Multi	Deterministic	(Semi)Static	Discrete
Poker	Partially	Multi	Stochastic	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Static	Discrete
Car	Partially	Multi	Stochastic	Dynamic	Continuous
Cleaner	partially	Single	Stochastic	Dynamic	Continuous

Agent Types

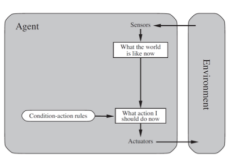
- Four basic types in order of increasing generality:
  - **Simple reflex agents**
  - **Model-based reflex agents**
  - **Goal-based agents**
  - **Utility-based agents**
- All of which can be generalized into **learning agents** that can improve their performance and generate better actions.

反射型类似于查表.

都可以泛化成学习智能体.

Simple reflex agents

- Simple reflex agents select an action **based on the current state only** ignoring the percept history.
- Simple but limited: can only work if the environment is **fully observable**, that is the correct action is based on the **current** percept only.

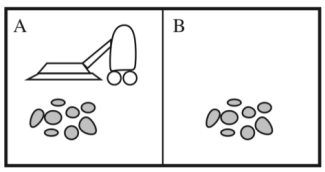


只能在完全可观测的环境中.

Example

Vacuum (reflex) agents

- Let' s write the algorithm for the Vacuum cleaner...
- Percepts: location and contents (location sensor, dirt sensor)
- Actions: Left, Right, Suck, NoOp



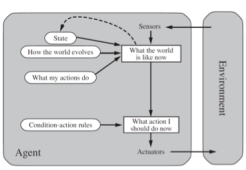
Percept	Action
[A, clean]	Right
[A, dirty]	Suck
[B, clean]	Left
[B, dirty]	Suck

Function Vacuum_agent (location, content)
return action
if content = **Dirty** then return **Suck**
else if location = **A** then return **Right**
else return **Left**

What if the vacuum agent is deprived of its location sensor?

Model-based reflex agents

- **Handle partial observability** by keeping track of the part of the world it can' t see now.
- Internal state depending on the percept history (**best guess**).
- Model of the world based on (1) how the world evolves independently from the agent, and (2) how the agent actions affects the world.

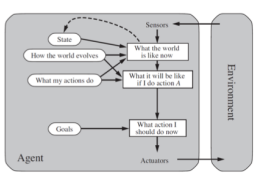


会基于历史信息(有时会对未知信息作猜测)

上面的reflex型都强依赖于lookup table
冗余、难修改、目标被隐藏了.

Goal-based agents

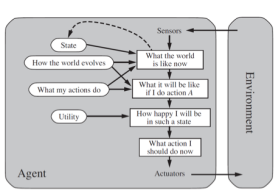
- Knowing the current state of the environment is not enough. The agent needs some **goal information**.
- Agent program combines the goal information with the environment model to choose the actions that achieve that goal.
- Consider the future with "What will happen if I do A?"
- Flexible as knowledge supporting the decisions is explicitly represented and can be modified.



不需要查找表.
更灵活.

Utility-based agents

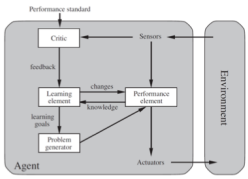
- Sometimes achieving the desired goal is not enough. We may look for quicker, safer, cheaper trip to reach a destination.
- Agent happiness should be taken into consideration. We call it **utility**.
- A utility function is the agent' s performance measure.
- Because of the uncertainty in the world, a utility agent choses the action that maximizes the expected utility.



最大化效应的期望.

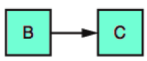
Learning agents

- Programming agents by hand can be very tedious. "Some more expeditious method seem desirable" Alan Turing, 1950.
- Four conceptual components:
  - Learning element: responsible for making improvements.
  - Performance element: responsible for selecting external actions. It is what we considered as agent so far.
  - Critic: How well is the agent is doing w.r.t. a fixed performance standard.
  - Problem generator: allows the agent to explore.



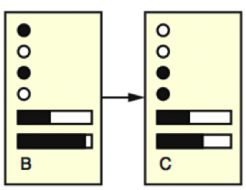
Agent states

- a) **Atomic Representation**: Each state of the world is a **black box** that has no internal structure.
  - E.g., finding a driving route, each state is a city.
  - AI algorithms: search, games, Markov decision processes, hidden Markov models, etc.



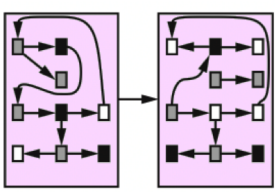
状态就是黑盒，不关心内部有什么。

- b) **Factored Representation:** Each state has some **attribute-value properties**.
  - E.g., GPS location, amount of gas in the tank.
  - AI algorithms: constraint satisfaction, and Bayesian networks.



考虑状态中的属性

- c) **Structured Representation:** Relationships between the objects of a state can be explicitly expressed.
  - AI algorithms: first order logic, knowledge-based learning, natural language understanding.



考虑状态中对象之间的关系

Intelligent agents

- The concept of intelligent agent is central in AI.
- AI aims to design intelligent agents that are useful, reactive, autonomous and even social and proactive.
- An agent perceives its environment through percept, and acts through actuators.
- A performance measure evaluates the behavior of the agent.
- An agent that acts to maximize its expected performance measure is called a rational agent.
- PEAS: A task environment specification that includes Performance measure, Environment, Actuators and Sensors.

Agent = **Architecture** + **Program**

- Four types of agents: Reflex agents, model-based agents, goal-based agents, and utility-based agents.
- Agents can improve their performance through learning. This is a high-level present of agent programs.
- States representations: atomic, factored, structured. Increasing expressiveness power.