Intelligent Agents

吉建民

USTC
jianmin@ustc.edu.cn

2024 年 3 月 3 日

Used Materials

Disclaimer: 本课件采用了 S. Russell and P. Norvig's Artificial Intelligence –A modern approach slides, 徐林莉老师课件和其他网络课程课件，也采用了 GitHub 中开源代码，以及部分网络博客内容

Table of Contents

Agents and Environments

PEAS (Performance measure, Environment, Actuators, Sensors)

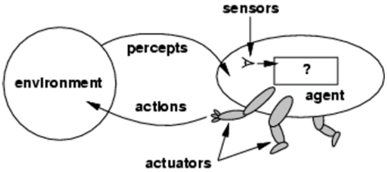
Environment Types

Agent Types

Agents

- An agent (智能体) is anything that can be viewed as perceiving its environment through sensors（传感器） and acting upon that environment through actuators（执行器）
- Agent 通过传感器感知环境并通过执行器对所处环境产生影响
- Agents include humans, robots, softbots（软件机器人）, thermostats（自动调温器）, etc.
  - 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

Agents and environments

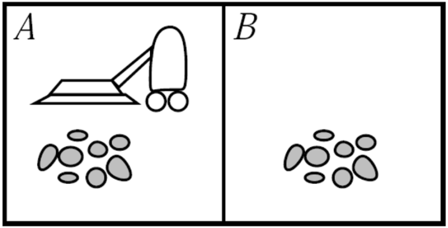


- The agent function maps from percept histories to actions:

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

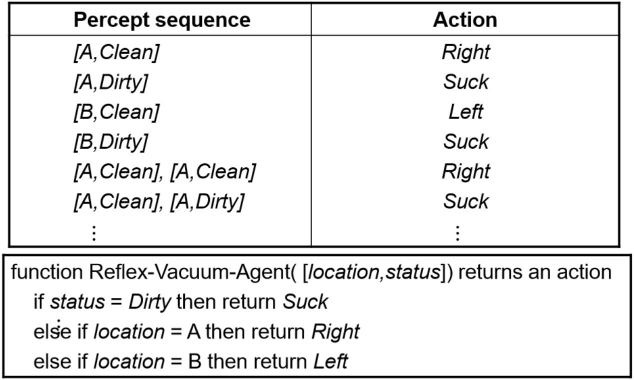
- The agent program runs on the physical architecture to produce f
- agent = architecture + program

Vacuum-cleaner world



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

A vacuum-cleaner agent



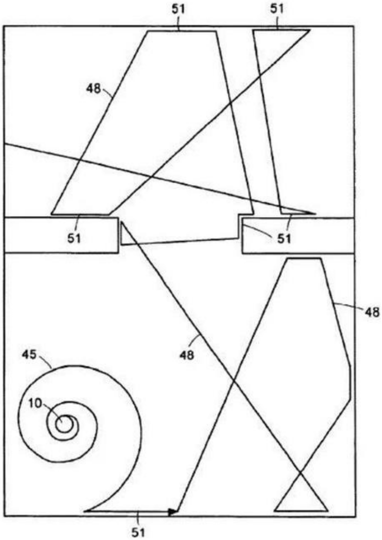
Percept sequence	Action
[A,Clean]	*Right*
[A,Dirty]	*Suck*
[B,Clean]	*Left*
[B,Dirty]	*Suck*
[A,Clean], [A,Clean]	*Right*
[A,Clean], [A,Dirty]	*Suck*
⋮	⋮

```
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
```

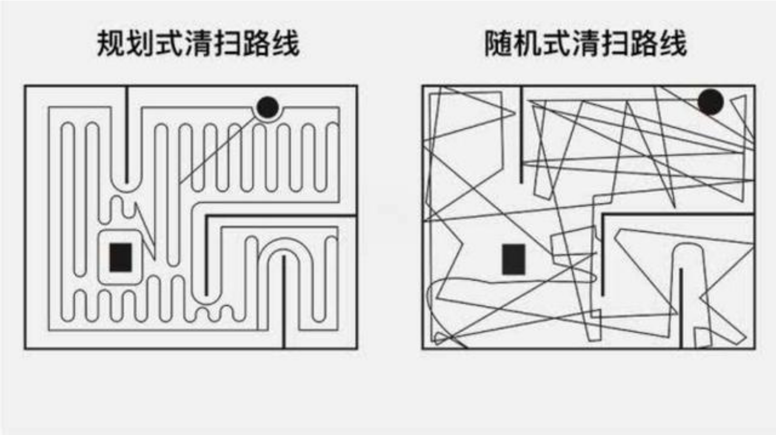
What is the right function?
Can it be implemented in a small agent program?

随机覆盖式扫地机器人

iRobot Roomba 3-8 系列机器人算法



规划式扫地机器人



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（**性能度量**）：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

- 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.
- 理性智能体：对于每一个可能的感知序列，理性智能体应该基于已知的感知序列提供的信息，和智能体已有的先验知识，选择能够使它的性能度量最大化的行为。

Rational agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge 全知的)
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
  - 为修改未来的感知信息而采取行动——信息收集
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)
  - 理性智能体应该能够尽可能地学习，以弥补不全面或不正确的先验知识。

Rationality

- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date
- Rational $\neq$ omniscient (全知的)
  — percepts may not supply all relevant information
- Rational $\neq$ clairvoyant (明察秋毫)
  — action outcomes may not be as expected
- Hence, rational $\neq$ successful
- Rational $\rightarrow$ exploration, learning, autonomy

Table of Contents

Agents and Environments

PEAS (Performance measure, Environment, Actuators, Sensors)

Environment Types

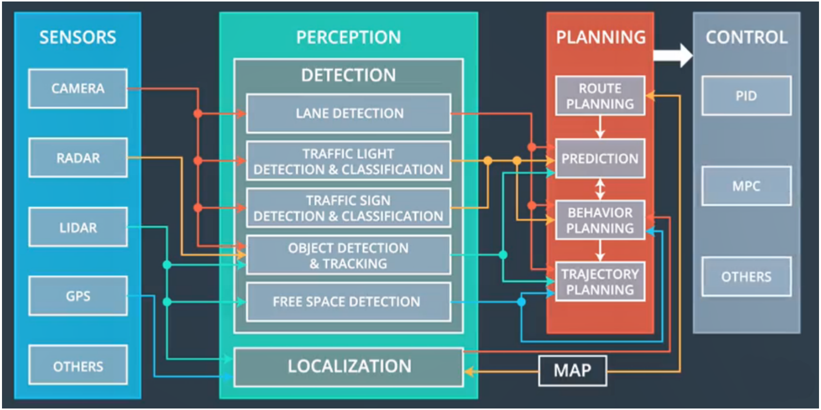
Agent Types

PEAS

Agent: automated taxi

- Performance measure? safety, destination, profits, legality, comfort, ...
- Environment? streets/freeways, traffic, pedestrians, weather, ...
- Actuators? steering, accelerator, brake, horn, speaker/display, ...
- Sensors? video, accelerometers, gauges, engine sensors, keyboard, GPS, ...

自动驾驶系统架构

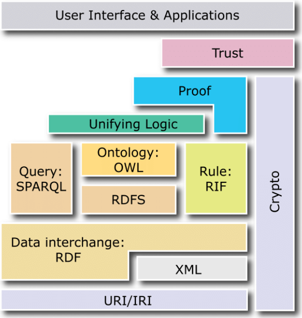

SENSORS
CAMERA
RADAR
LIDAR
GPS
OTHERS
PERCEPTION
DETECTION
LANE DETECTION
TRAFFIC LIGHT DETECTION & CLASSIFICATION
TRAFFIC SIGN DETECTION & CLASSIFICATION
OBJECT DETECTION & TRACKING
FREE SPACE DETECTION
LOCALIZATION
MAP
PLANNING
ROUTE PLANNING
PREDICTION
BEHAVIOR PLANNING
TRAJECTORY PLANNING
CONTROL
PID
MPC
OTHERS

PEAS

Agent: Internet shopping agent

- Performance measure? price, quality, appropriateness, efficiency
- Environment? current and future WWW sites, vendors, shippers
- Actuators? display to user, follow URL, fill in form
- Sensors? HTML pages (text, graphics, scripts)

Semantic web and web services



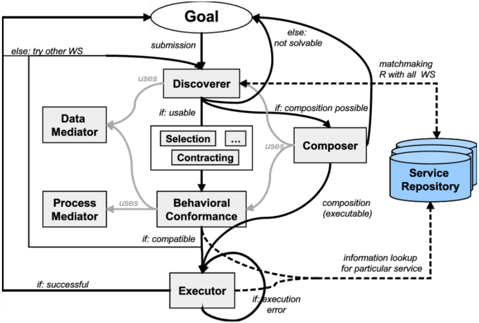


Table of Contents

Agents and Environments

PEAS (Performance measure, Environment, Actuators, Sensors)

Environment Types

Agent Types

Environment types

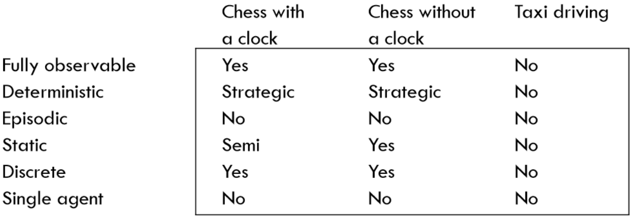
- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
- Episodic (vs. sequential) 片断式 (vs. 延续式): 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.

如装配线上检测次品零件的机器人只需要把每次决策建立在当前零件基础上，不用考虑以前的决策。

Environment types

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating (思考). (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. multi-agent): An agent operating by itself in an environment.

Environment types



	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Table of Contents

Agents and Environments

PEAS (Performance measure, Environment, Actuators, Sensors)

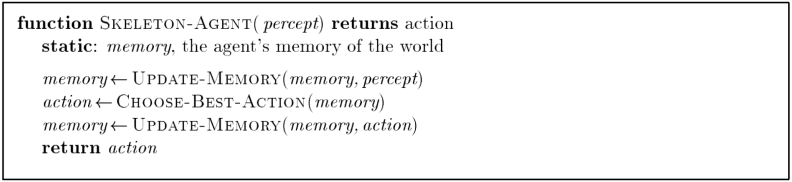
Environment Types

Agent Types

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational
  - Aim: find a way to implement the rational agent function concisely

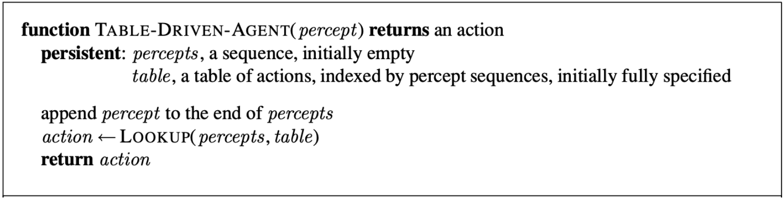
An agent program takes a single percept as input, keeps internal state:



```
function SKELETON-AGENT( percept) returns action
   static: memory, the agent's memory of the world

   memory ← UPDATE-MEMORY(memory, percept)
   action ← CHOOSE-BEST-ACTION(memory)
   memory ← UPDATE-MEMORY(memory, action)
   return action
```

Table-lookup agent



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

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

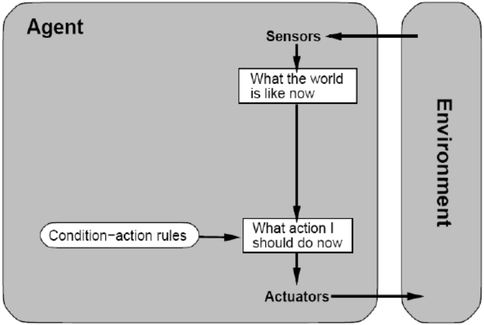
Drawbacks:

- 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 types

- Four basic types in order of increasing generality:
  - simple reflex agents
  - reflex agents with state
  - goal-based agents
  - utility-based agents
- All these can be turned into learning agents

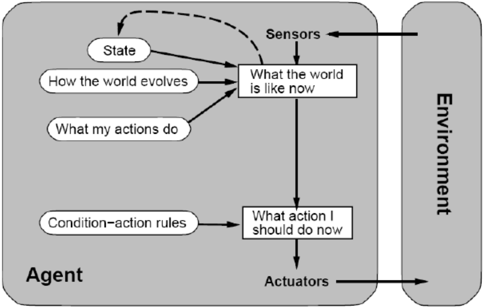
Simple reflex agents

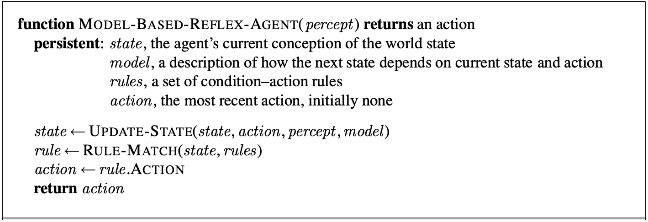


```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition–action rules

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

Reflex agents with state





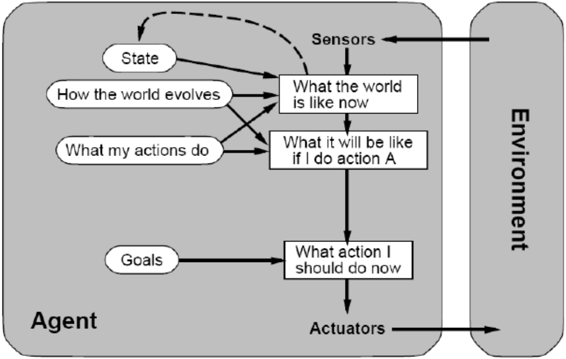
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action,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

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

Goal-based agents

Reasoning about actions

- Reflex agents only act based on pre-computed knowledge (rules)
- Goal-based (planning) agents act by reasoning about which actions achieve the goal
- Less efficient, but more adaptive and flexible


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

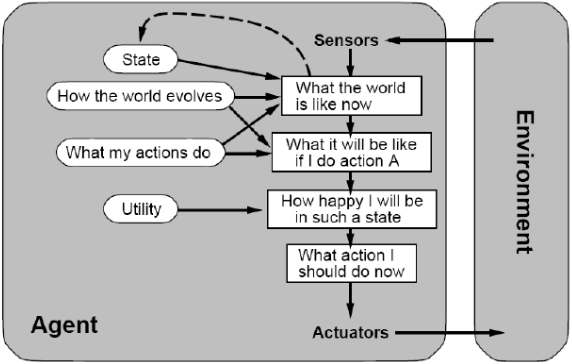
Goal-based agents (continued)

- Knowing current state is not always enough
    - State allows agent to keep track of unseen parts of world
    - Agent must update state based on changes and its actions
- Choose between potential states using goal
    - Can change goal without need to “reprogram” rules, for example a new destination for the taxi-driving agent
- Search and planning
    - concerned with finding sequences of actions to satisfy a goal.
    - contrast with condition-action rules: involves consideration of future “what will happen if I do ...” (fundamental difference)

Utility-based agents

Utility Function

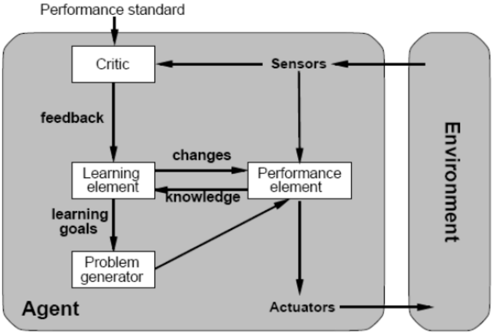
- A mapping of states onto real numbers
- Allows rational decisions in two kinds of situations
  - Evaluation of the tradeoffs among conflicting goals
  - Evaluation of competing goals
- Preferred world state has higher utility for agent


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Learning agents

Four main components

- Performance element: the agent function
- Learning element: responsible for making improvements by observing performance
- Critic: gives feedback to learning element by measuring agent's performance
- Problem generator: suggest other possible courses of actions (exploration)



Summary

- Agents interact with environments through actuators and sensors
- The agent function describes what the agent does in all circumstances
- The performance measure evaluates the environment sequence
- A perfectly rational agent maximizes expected performance
- Agent programs implement (some) agent functions
- PEAS descriptions define task environments
- Environments are categorized along several dimensions:
  - observable? deterministic? episodic? static? discrete? single-agent?
- Several basic agent architectures exist:
  - reflex, reflex with state, goal-based, utility-based

Next ...

- Main intelligent capabilities of agents
  - 问题求解/Search (chapters 3 $\sim$ 6)
  - 知识、推理与规划 / Logic, Knowledge, Reasoning, and Planning (chapters 7 $\sim$ 12)
  - 不确定知识与推理 / Uncertainty and Decision Making (chapters 13 $\sim$ 17)
  - 学习 / Learning (chapters 18 $\sim$ 21)
  - 应用 / NLP, Perception, and Robotics (chapters 22 $\sim$ 25)