Planning And Automated Reasoning

Alessio Gjergji

Contents

Chapter 1

Action Ragionally

1.1 Agents and environments

Agents include humans, robots, softbots, thermostats, etc. The agent function maps from percept histories to actions:

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

The agent program runs on the physical architecture to produce f.

Example: Vacuum-cleaner world

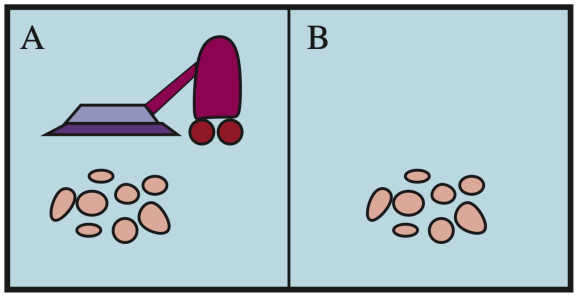

Figure 1.1.1: A vacuum-cleaner world with just two locations. Each location can be either clean or dirty. The agent perceives the location and the status of the location (clean/dirty). The agent can move left, right, suck, or do nothing.

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
⋮	⋮

What is the right function f?

Agent Programs vs Agent functions

Agent program is the implementation of the agent function. If an agent has $|\mathcal{P}|$ possible perceptions, the entries in the table after T time steps will be $\sum_{t=1}^{T} |\mathcal{P}|^t$

The artificial intelligent goal is to design small agent programs that can represent large agent functions. A possible agent program for the vacuum-cleaner world is:

```
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
```

1.2 Rationality

Fixed **performance measure** evaluates the environment sequences

- one point per square cleaned up in the timer T?
- one point per clean square per time step, minus one point per move?
- penalizefor $> k$ dirt squares?

A ragional agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date. but we have to know that **rational $\neq$ omniscient**. Percepts may not supply all relevant information. We have also the problem of **rational $\neq$ clairvoyant**. Actions outcomes may not be as expected. And then **Hence, rational $\neq$ successful**.

Rational means exploration, learning, and autonomy.

1.3 PEAS (Performance measure, Environment, Actuators, Sensors)

To design a rational agent, we must specify the task environment in which it is to operate. Lets consider the example of a taxi driver.

- **Performance measure**: safe, fast, legal, comfortable trip, maximize profits
- **Environment**: roads, other traffic, pedestrians, customers, weather, etc.
- **Actuators**: steering wheel, accelerator, brake, signal, horn, display, etc.
- **Sensors**: cameras, sonar, speedometer, GPS, odometer, engine sensors, etc.

1.4 Environment types

The environments types are:

- **Fully observable** vs **partially observable**
- **Deterministic** vs **stochastic**
- **Episodic** vs **sequential**
- **Static** vs **dynamic**
- **Discrete** vs **continuous**
- **Single-agent** vs **multi-agent**

The **environment type largely determines the agent design**. The real world is **partially observable**, **stochastic**, **sequential**, **dynamic**, **continuous**, and **multi-agent**.

1.5 Agent types

We can define an agent:

$$agent = architecture + program$$

The general Skeleton for a program is:

- input: current perception
- output: next action

Four basic types in order of increasing generality:

- **Simple reflex agents**
- **Model-based reflex agents**
- **Goal-based agents**
- **Utility-based agents**

All these can be turned into **learning agents**.

1.5.1 Simple reflex agents

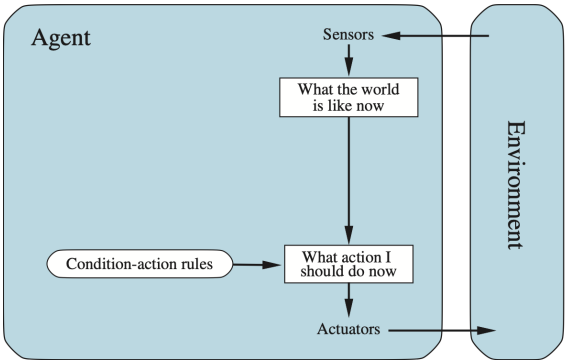

Figure 1.5.1: Simple reflex agent

```
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
```

1.5.2 Reflex agents with state

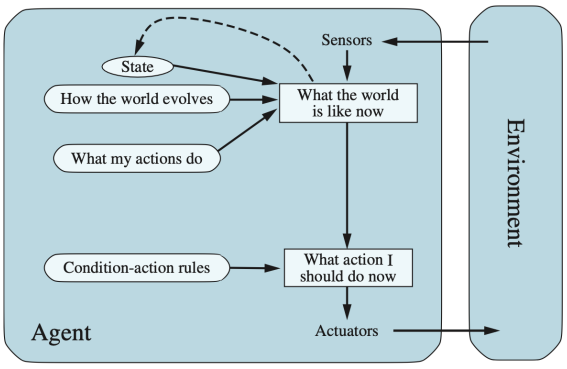

Figure 1.5.2: Reflex agent with state

```
function Reflex-Vacuum-Agent([location, status]) returns an action
    static: current-location, current-status,
    current-action = none,
    next-action = none

    current-location = Update-State(current-location, current-action)
        if status = Dirty then next-action = Suck
```

```
else if current-locondition = A then next-action = Right
else if current-condition = B then next-action = Left
current-action = next-action
return current-action
```

1.5.3 Goal-based agents

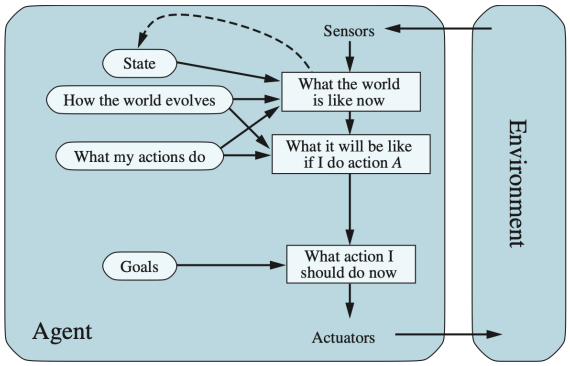

Figure 1.5.3: Goal-based agent

1.5.4 Utility-based agents

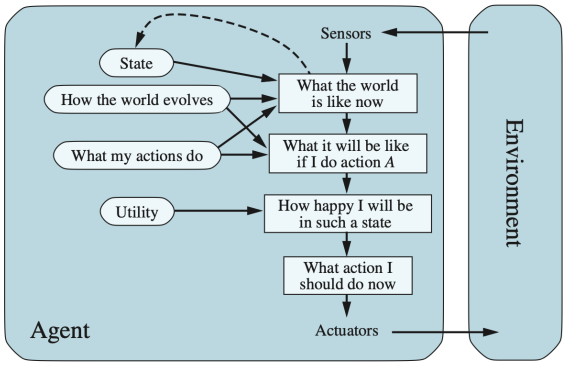

Figure 1.5.4: Utility-based agent

1.5.5 Learning agents

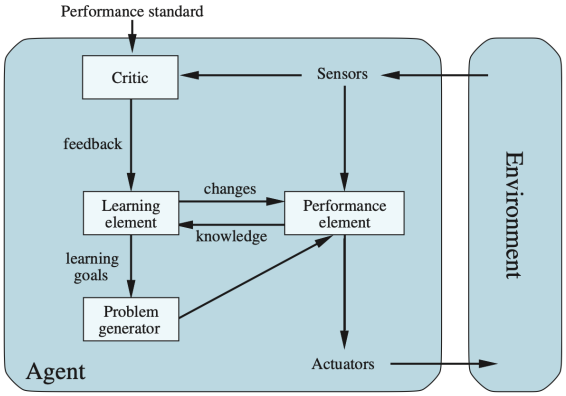

Figure 1.5.5: Learning agent

1.5.6 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