3. Intelligent agent(IA) types

By: Alex S.

Feb, 2024

1. Intelligent agent as a function

As was described in previous lections we can describe intelligent agent as a function:

$$f : P \to A$$

where P is a perception set and A is an action set.

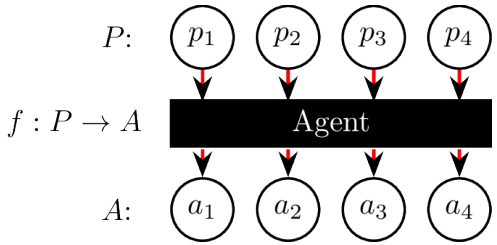

Figure 1: Agent as a function can be seen as a black box

Agent is a function that is a black box, i.e. we don't know how it produces its results. An agent receives information(perception P) from sensors, processes them via function(f) and returns back an action(A) to do.

The common algorithm of intelligent agent can be described as:

function $f : P \to a \in A$ `is`
  let `Knowledge Base : Memory(`M`)`
  `update :` $(M, P) \to M'$
  `action :` $M' \to a$
  `update :` $(M', a) \to M''$
return a

2. Intelligent agent types

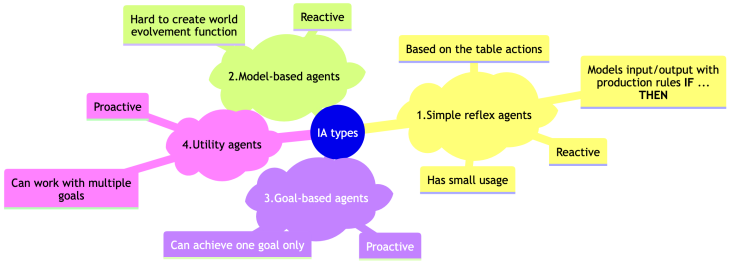

Figure 2: IA types

3. Simple reflex agents

Algorithm can be described the following way:

function $f_r : P \to a \in A$ `is`
  let R `= production rules - IF condition THEN action`
  interpret input : $P \to s_i \in S$[state]
  rule search : $(s_i, R) \to r_i \in R$
  rule burning : $r_i \to a$
return a

Illustration of the simple reflex agents working principle in Figure 3.

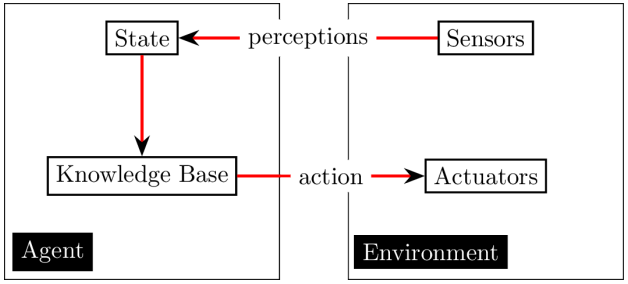

Figure 3: Simple reflex agents working principle

Simple reflex agents uses *Knowledge Base* with production rules.

Example:

R1: **IF** a driving car ahead has both stop signals turned on(*N*) **THEN** a driving car ahead is stopping (*C1*)

R2: **IF** a driving car ahead is stopping(*C1*) **THEN** a driving car ahead is slowing down (*C2*)

R3: **IF** a driving car ahead is slowing down(*C2*) **THEN** need to start to slow down (*C3*)

R4: **IF** need to start to slow down (*C3*) **THEN** press a braking pedal(*A*)

Where *R* is a production rule, *C* - conclusion, *N* - condition, *A* - action.

By using an **inference** mechanism, we can make a conclusion, that if we start with the production rule *N*, then we have the following chain: $N \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow A$ and that we should press the braking pedal.

4. Model-based agents

1. Follows how changes the environment, the state. Knows the previous state.
2. Environment changes regardless of agent.
3. Need to understand the consequences from actions.

All 3 actions describe the **inner state** of the agent(Figure 4).

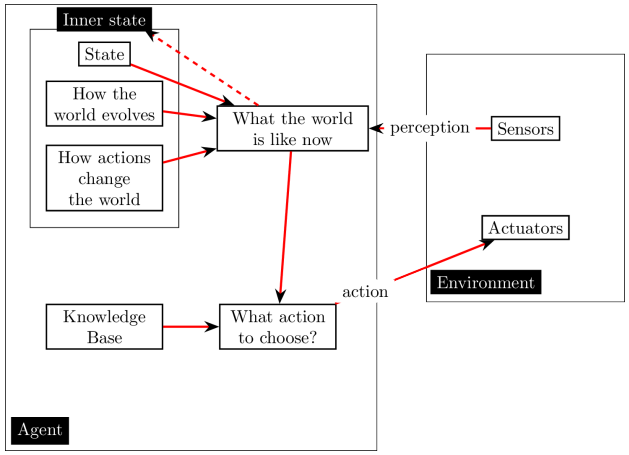

Figure 4: Model-based agents working principle

5. Goal-based agents

- Actions are dependent on what the goal is needed to be achieved by the agent.
- Can be effectively used with the search or planning(for robots) algorithms.

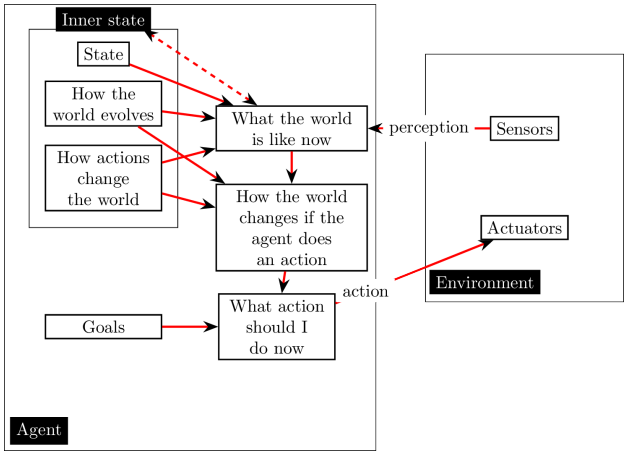

Figure 5: Goal-based agents working principle

6. Utility agents

- Can be applied when the problem cannot be solved with the goal-based agent, e.g.
  - More than 1 goal
  - And goals may be conflicting, therefore some compromise should be found
- Goals might be achieved using probabilities(e.g. games of chance)
- **Utility function**(U) is a measurement that allows to compare different states, i.e. given two states($s_1, s_2 \in S$) and give a numeric

value(v) of how these states differ: $U : (s_1, s_2) \rightarrow v$

- Agent can be based on the **utility theory**.

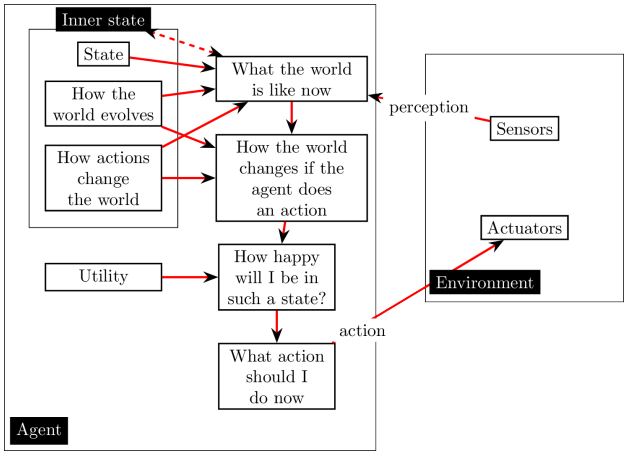

Figure 6: Utility agents working principle

7. Learning agents

- The agent has a capability to learn taking into account external performance standard

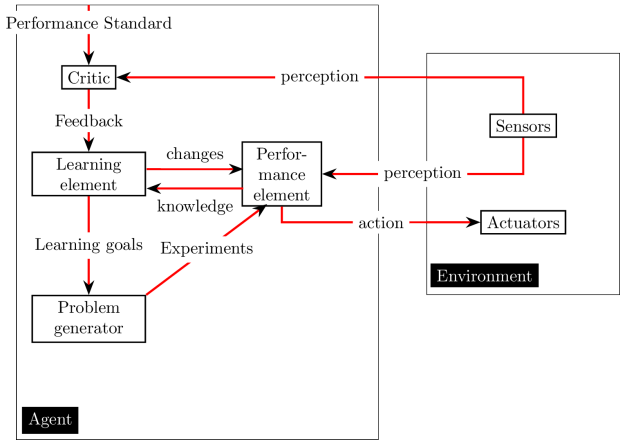

Figure 7: Learning agents working principle

8. Methodologies

- Not sure if something serious uses these tools, but somebody might use:
  - **FIPA** (Foundation for Intelligent Physical Agents)
  - **BDI** (behavior - desire - intent) agents
- Not sure if somebody will ever use some of these methodologies in practice:
  - **PROMETHEUS** - iterative waterfall model
  - **Gaia**
  - **MaSE**
  - **MASITS** - local RTU development for multi-agents that helps with tutoring