

# 1 Introduction to Artificial Intelligence

We can describe Artificial Intelligence (AI) as any task performed by a program or a machine. If a human carried out the same activity, we would say the human had to apply intelligence to accomplish the task. The natural question that comes to our mind is *what should we aim to do in AI?*

To answer this question, let us go back to 120 years ago when we were trying to create an object(airplane) that can fly like a bird. At that time, we had two choices.

1. Design objects that look like birds.
2. Design objects that can fly without necessarily looking like birds.

The initial efforts were towards the first route, but we settled for the second route, and we created airplanes later in time. Hence, we should aim to design objects that can act rationally, and we call such objects rational agents.

In this course, we will learn different techniques to design rational agents. Before going into details of it, first, address *how to determine whether or not these objects(rational agents) are capable of thinking like human beings.*

In 1950, Turing proposed that let humans and AI systems communicate through some medium. If an AI system can convince a human that he/she is talking to another human, not an AI system, then we can say that the AI system can think like a human being. This test to determine whether the AI system can fool humans is commonly known as the Turing test. Nowadays, before logging into many websites, we have to prove that we are not a robot by answering questions or entering some captcha, is an example of Turing test.

Now, let us discuss Rational Agent Design (RAD) in detail.

## 1.1 Rational Agent Design

An agent is an entity that perceives its environment through its sensors. Figure 1<sup>1</sup> explains the setup, it has environment sensors for perception, an *agent function* to map the history of precepts to action, and actuators to perform the action.

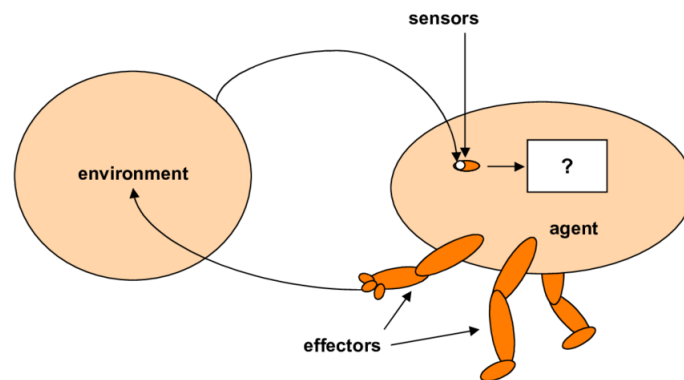


Figure 1: AI setup.

---

<sup>1</sup>The image is taken from [https://www.tutorialspoint.com/artificial\\_intelligence/artificial\\_intelligence\\_agents\\_and\\_environments.htm](https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm)

**Agent Function:** It is a function that maps percepts histories/sequences to actions. Let  $P$  be the sequence of percepts and  $A$  be the actions, then function  $f : P^* \mapsto A$ .

**Agent Program:** It run on physical architecture to perform agent function  $f$ .

Let us try to understand this setting through an example of a mopping robot, and we can call this as MopBot. Let us say there are two locations A and B, and the precepts of MopBot allow it to figure out the location and the state of the location, i.e., whether the location is dirty or clean.

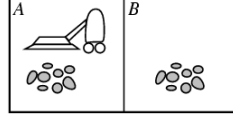


Figure 2: Mopping Bot Setup.

**Environment** - The location of MopBot and status of the each location.

**Percepts** - Current location and the status of location.

**Actions** - Left, Right, Clean, Idle.

Percepts Sequence	Action
[A, clean]	Right
[A, dirty]	Clean
[B, clean]	Left
[B, dirty]	Clean
[A, clean] [A, clean]	Right
[A, clean] [A, dirty]	Clean

Table 1: MopBot functions. Let us assume that location A is on left location B. Percept sequence represents [Location, Status].

Table 1 represents the action functions for MopBot agent. But, the question arise here is *what is a rational agent*. Before formally answering this question, let us introduce agent function for same MopBot agent. In Table 2, we can see that there are two set of actions are described for a percept sequence. Let say agent function  $f_1$  maps percepts to action<sub>1</sub>, and agent function  $f_2$  maps percepts to action<sub>2</sub>.

Percepts Sequence	Action <sub>1</sub>	Action <sub>2</sub>
[A, clean]	Right	Mop
[A, dirty]	clean	Right
[B, clean]	Left	Mop
[B, dirty]	Clean	Left
[A, clean] [A, clean]	Right	Clean
[A, clean] [A, dirty]	Clean	Right

Table 2: MopBot functions. Let us assume that location A is on left location B. Percept sequence represents [Location, Status].

Now, the question is *What is the difference between a system with agent function  $f_1$ , and a system with agent function  $f_2$ .* As we can see, with agent function  $f_2$ , the system is not MopBot. With the agent function  $f_2$ , the system is not doing the *right thing*; it is not behaving rationally.

**Rational Agent** An agent should strive to *do the right thing* based on what it can perceive and the actions it can perform. Note that rationality is not all-knowing with infinite knowledge.

## 1.2 Task Environment

How to check if an agent is doing the right thing or not; for that, we should have objective criteria for measuring the success of an agent's behavior, i.e., we should have a performance measure. For example, in the case of vacuum-cleaner agents, we can check the amount of dirt cleaned, time taken to clean, electricity consumed, noise generated as criteria for measuring performance.

To create a rational agent, we need to specify the task environment. The task environment consists of the following four things:

- Performance Measure
- Environment
- Actuators
- Sensors

Let us discuss the different properties of the task environment.

**Fully observable vs. Partially observable** If sensors provide access to the complete state of the environment at each point in time, it is fully observable; else, it is partially observable

**Deterministic vs. Stochastic** If the next state of the environment is completely determined by the current state and the action executed by the agent, it is deterministic, else stochastic.

**Episodic vs. Sequential** If the choice of current action does not depend on actions in past episodes, it is episodic, else sequential.

**Static vs. Dynamic** If the environment is unchanged while an agent is deliberating, it is static, else it is dynamic.

**Discrete vs. Continuous** If it has a finite number of distinct states, percepts, and actions, it is discrete, else continuous.

**Single agent vs. Multi-agent** If an agent is operating by itself in an environment, it is a single agent, else multi-agent.

Properties of task environment largely determine agent design. The world is partially observable, stochastic, sequential, dynamic, continuous, and multi-agent.

### 1.3 Agent Function

Let us again visit the definition of agent function: It maps percepts histories/sequences to actions. Let  $P$  be the sequence of percepts and  $A$  be the actions, then function  $f : P^* \mapsto A$ .

Suppose we have a table-lookup definition for agent function, i.e., a table where for every percept, there is action.

---

**Algorithm 1** Table-Driven-Agent(*percept*)

---

- 1: Static: *percepts*: a sequence  
          *table*: a table of actions, indexed by percept sequences,  
          fully specified.
  - 2: *percepts*  $\leftarrow$  *precepts*  $\cup$  *percept*.
  - 3: *action*  $\leftarrow$  LOOKUP(*percept*, *table*)
  - 4: **return** *action*
- 

Algorithm 1 represents the table driven agent function definition. Algorithm 1 has the following drawbacks:

- Huge table to store.
- Take a long time to build the table.
- So many entries to get correct

There are four types of agent functions: Simple Reflex, Model-Based Reflex, Goal based, Utility based. Table 3 shows all four types of agent functions in detail.

The goal-based agent function tries to achieve a respecified goal, and the utility-based agent functions tries to maximize utility. The performance measure is what others care about, while the utility is what the agent cares about. Ideally, the two should match, but it is not always the case.

In addition to that, there are two more types of agents.

**Learning agent** The agent can learn about the performance/environment measure.

**Autonomous agent** An agent can update its agent program based on its experiences.

We will not discuss Reflex/ Model-Based-Reflex agents, and we will start our discussion with goal-based agents.

Table 3: Four basic kinds of agent program

Agent Program	Description	Example
<b>Simple Reflex</b>	<ul style="list-style-type: none"> <li>• Passive.</li> <li>• State depends only on percept</li> <li>• The agent follows simple action-condition rules.</li> </ul>	Infant: <ul style="list-style-type: none"> <li>• <i>Hungry</i> <math>\rightarrow</math> <i>Cry</i></li> <li>• <i>Happy</i> <math>\rightarrow</math> <i>Sleep</i></li> </ul>
<b>Model-Based Reflex</b>	<ul style="list-style-type: none"> <li>• Passive</li> <li>• Agent maintains some sort of internal state based on percept history that reflects some of the unobserved aspects of the world.</li> <li>• State depends on the model of the world, percept and current state</li> </ul>	Kid: <ul style="list-style-type: none"> <li>• Want Ice Cream &amp; didn't Receive <math>\rightarrow</math> Ask for Ice Cream</li> </ul>
<b>Goal-Based</b>	<ul style="list-style-type: none"> <li>• Active</li> <li>• Plan actions depending on current state, percepts and model of the world to achieve a desirable goal.</li> </ul>	Teenager: <ul style="list-style-type: none"> <li>• Want to get into NUS <math>\rightarrow</math> Plan Study and Playing based on the Goal.</li> </ul>
<b>Utility-Based</b>	<ul style="list-style-type: none"> <li>• Actions based on a weighted function of multiple goals</li> <li>• Goals may be inconsistent or conflicting</li> <li>• Useful when unclear which goal is most important and a weighted balance is needed</li> </ul>	Adult: <ul style="list-style-type: none"> <li>• <math>\alpha\text{Job} + \beta\text{Partner} + \gamma\text{Health}</math></li> </ul>