**AIML 1 – Agents 1**

***The elementary definition***

The agent can be defined as an entity in a program which has **non-trivial decision making** skills using **computational resources**

This can be visualized in multiple ways,

***The mathematical way***

The agent is a function that, given a universe of percepts/stimuli, and the universe of actions that can be done on that precise percept:

Text, letter

Description automatically generatedf partially maps a **set** of percepts to an action

A “good” agent can be described by the rules:

* **Consequentialism**

Whether something is good or bad depends on its outcomes

* **Performance measure**

The process used to assess efficiency and effectiveness

* **Reward**

The reward system of an agent basically awards a prize to it whether a “good” action is performed

It will therefore act to obtain more rewards

The **rational agent** is a type of agent which strives for optimality, given the following conditions:

* A good **performance measure**
* The agent’s **prior knowledge** of the environment
* The agent’s **universe of actions**
* The agent’s **universe of percepts**

In this case, the optimality is that **the agent is expected to take an action that will certainly maximize its performance measure**

Moreover, the agent must have a **learning** component, meaning that it’s got to have **autonomy**

**This does not mean that the agent is omniscient**

**PEAS – The good agents**

* Performance measure
* Environment
* Actuators
* Sensors

Table

Description automatically generated

***Focusing on the task environment***

We now distinguish the types of environments with definitions:

* **Observability**

The environment is **fully observable** if the sensors provide a complete state of it at each point in time (Observability implies Accessibility)

* **Multiplicity**

Single vs. multi-agent environment

In the case of multi-agent environment, we further distinguish in:

* + Competitive; aim to weaken one of the agents
  + Cooperative; maximize every agent performance
* **Deterministic, non-deterministic and stochastic**

Deterministic (for an algorithm) means that when you re-run the algorithm with the same input, you get the same answer. Non-deterministic means the answer can change, and one way to do this is to use randomization (i.e., stochastics).

Unobservability **may** imply non-determinism

* **Episodic, sequential**

In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action (keyword: atomic actions)

However, in Sequential environment, an agent requires memory of past actions to determine the next best actions; actions **will change the environment**

* **Static, dynamic and semi-dynamic**
  + An environment is dynamic if it changes while an agent is in the process of responding to a percept sequence
  + It is static if it does not change while the agent is deciding on an action (i.e. the agent does not to keep in touch with time)
  + An environment is semi-dynamic if it does not change with time but the agent's performance score does
* **Discrete, continuous**

If the number of percepts and actions in the environment is limited and distinct then the environment is said to be discrete (e.g., A chess board)

* **Known, Unknown**

In a known environment, the output for all probable actions is given

In case of unknown environment, for an agent to make a decision, it has to gain knowledge about how the environment works (i.e. must have full observability)