

# Artificial Intelligence

## AI 2002

### Lecture 2

Mahzaib Younas

Lecture Department of Computer Science

FAST NUCES CFD

# Agent Types

- Five Types of Agents
  - Simple reflex agents
  - Reflex agents with state
  - Goal-based agents
  - Utility-based agents
  - Learning agent.

# Simple Reflex Agents

- Simple reflex agents act only on; -
  - the basis of the current percept, ignoring the rest of the percept history.
- The agent function is based on the condition-action rule:
  - if condition then action.

This agent function only succeeds

- when the environment is fully observable.

# Simple Reflex Agent (Cont....)

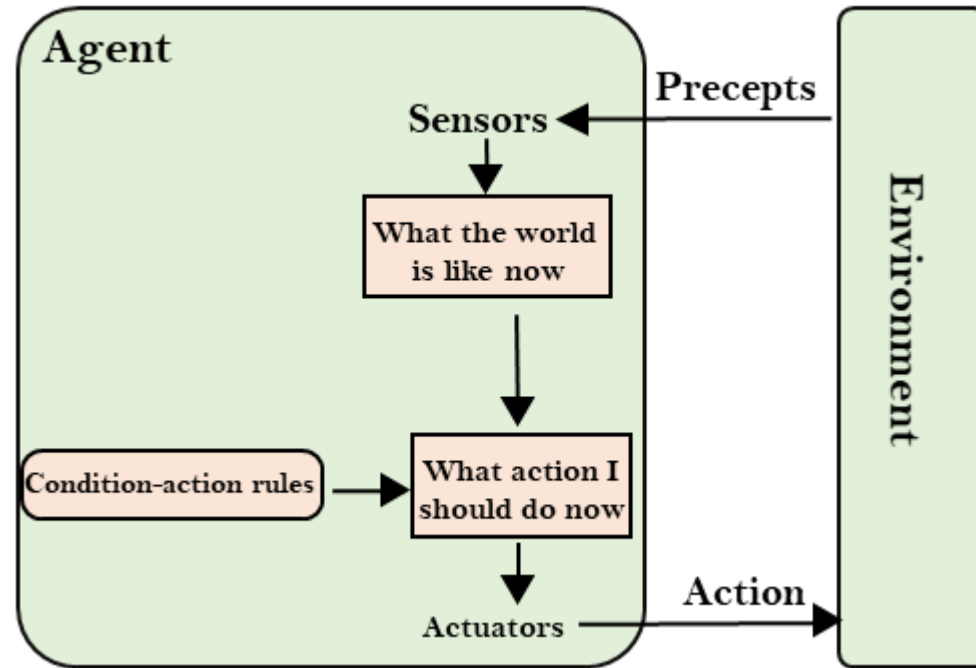
SimpleReflexAgent(percept)

state = InterpretInput(percept)

rule = RuleMatch(state, rules)

action = RuleAction(rule)

Return action



# Example:

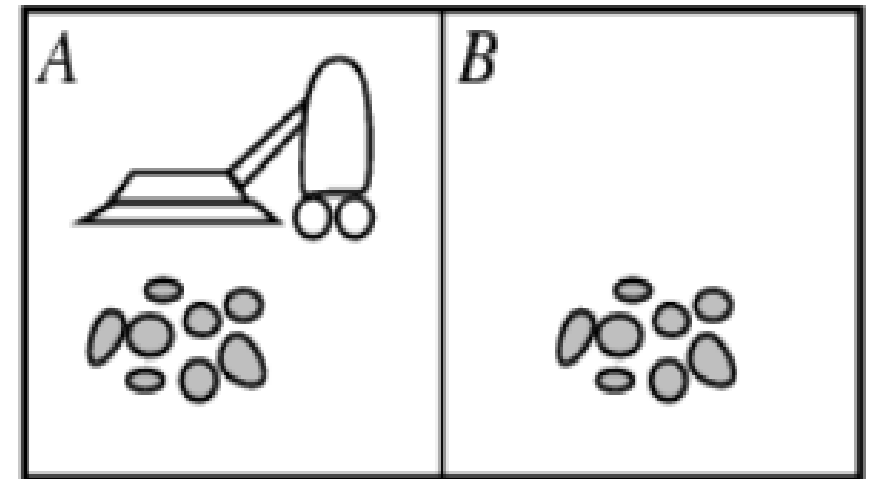
- A thermostat that turns on the air conditioner when the current temperature exceeds a certain threshold is a simple reflex agent.

# Example:

- The vacuum agent is a simple reflex agent because the decision is based only on the current location, and whether the place contains dirt.
- A simple reflex agent comprises the following parts:
  - **Agent:** The agent is the one who performs actions on the environment.
  - **Sensors:** Sensors are the things that sense the environment. They are devices that measure physical property.
  - **Actuators:** Actuators are devices that convert energy into motion.
  - **Environment:** The environment includes the surroundings of the agent.

# Example:

- Performance?
  - point for each square cleaned in time
  - clean squares per time step - #moves per time step?
- Environment:
  - vacuum, dirt, multiple areas defined by square regions.
- Actions:
  - Left
  - Right
  - Suck
  - idle
- Sensors: location and contents.
  - RuleMatch[A, dirty] then RuleAction[suck]



# Explanation

- The actions are taken depending upon the condition. If the condition is true, the relevant action is taken. If it is false, the other action is taken.
- The agent takes input from the environment through sensors, and delivers the output to the environment through actuators.
- The colored rectangles denote the current internal state of the agent's decision process.
- The ovals represent the background information used in the process.



# Advantages

- The advantage of simple reflex agents are
  - Easy to design and implement, requiring minimal computational resources
  - Real-time responses to environmental changes
  - Highly reliable in situations where the sensors providing input are accurate, and the rules are well designed
  - No need for extensive training or sophisticated hardware

# Drawbacks

- Problems for the simple reflex agent design approach:
  - They have very limited intelligence
  - They do not have knowledge of non-perceptual parts of the current state
  - Mostly too big to generate and to store.
  - Not adaptive to changes in the environment.

# Reflex Agents with state

- A model-based agent has two important factors:
  - **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
  - **Internal State:** It is a representation of the current state based on percept history.
- Store previously-observed information. Can reason about unobserved aspects of current state

# Reflex Agents with state (Cont...)

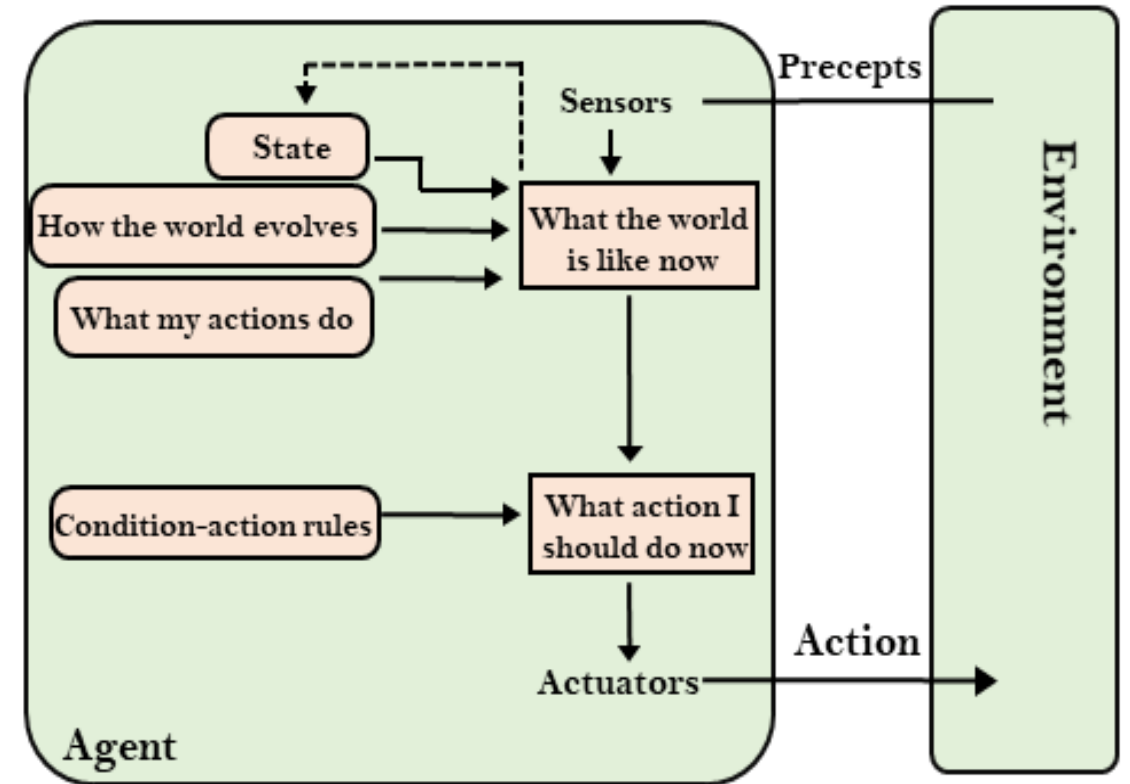
- These agents have the model, "which is knowledge of the world" and based on the model they perform actions.
- Updating the agent state requires information about:
  - How the world evolves
  - How the agent's action affects the world.
- A model-based reflex agent follows the condition-action rule, which specifies the appropriate action to take in a given situation. But *unlike a simple reflex agent, a model-based agent also employs its internal state to assess the condition during the decision and action process.*

# Representation:

ReflexAgentWithState(percept)

- `state = UpdateDate(state, action, percept)`
- `rule = RuleMatch(state, rules)`
- `action = RuleAction(rule)`

Return action



# How does the Reflex Agents with state Work

The model-based reflex agent operates in four stages:

## **1.Sense**

It perceives the current state of the world with its sensors.

## **2.Model:**

It constructs an internal model of the world from what it sees.

## **3.Reason:**

It uses its model of the world to decide how to act based on a set of predefined rules or heuristics.

## **4.Act:**

The agent carries out the action that it has chosen.

# Example:

A chess-playing AI that considers the history of moves and the current board state to decide the next move is a model-based agent.

# Advantages

- The advantage of simple reflex agents with state are
  - Quick and efficient decision-making based on their understanding of the world
  - Better equipped to make accurate decisions by constructing an internal model of the world
  - Adaptability to changes in the environment by updating their internal models
  - More informed and strategic choices by using its internal state and rules to determine the condition



# Drawbacks

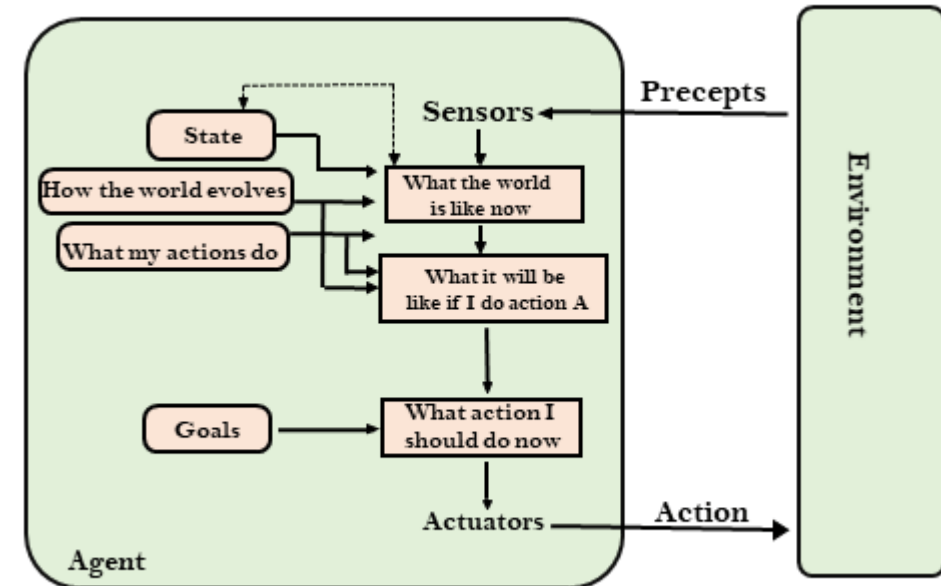
- Problems for the simple reflex agent with state design approach:
  - Building and maintaining models can be computationally expensive
  - The models may not capture the real-world environment's complexity very well
  - Models cannot anticipate all potential situations that may arise
  - Models need to be updated often to stay current
  - Models may pose challenges in terms of interpretation and comprehension

# Goal Based Agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.

# Goal Based Agents (Cont....)

- Goal reflects desires of agents.
- May project actions to see if consistent with goals.
- Takes time, world may change during reasoning



# How does it work:

The working pattern of the goal-based agent can be divided into five steps:

## **1. Perception:**

The agent perceives its environment using sensors or other input devices to collect information about its surroundings.

## **2. Reasoning:**

The agent analyzes the information collected and decides on the best course of action to achieve its goal.

## **3. Action:**

The agent takes actions to achieve its goal, such as moving or manipulating objects in the environment.

## **4. Evaluation:**

After taking action, the agent evaluates its progress towards the goal and adjusts its actions, if necessary.

## **5. Goal Completion:**

Once the agent has achieved its goal, it either stops working or begins working on a new goal.

# Example:

- We can say that Google Bard is a goal-based agent. No doubt, it is also a learning agent.
- As a goal-based agent, it has a goal or objective to provide high-quality responses to user queries. It chooses its actions that are likely to assist users in finding the information they seek and achieving their desired goal of obtaining accurate and helpful responses.

# Advantages

- The advantage of Goal Based Agents are
  - Simple to implement and understand
  - Efficient for achieving a specific goal
  - Easy to evaluate performance based on goal completion
  - It can be combined with other AI techniques to create more advanced agents
  - Well-suited for well-defined, structured environments
  - It can be used for various applications, such as robotics, game AI, and autonomous vehicles.

# Drawbacks

- Problems for the goal based approach design approach:
  - Limited to a specific goal
  - Unable to adapt to changing environments
  - Ineffective for complex tasks that have too many variables
  - Requires significant domain knowledge to define goals

# Utility Based Agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.

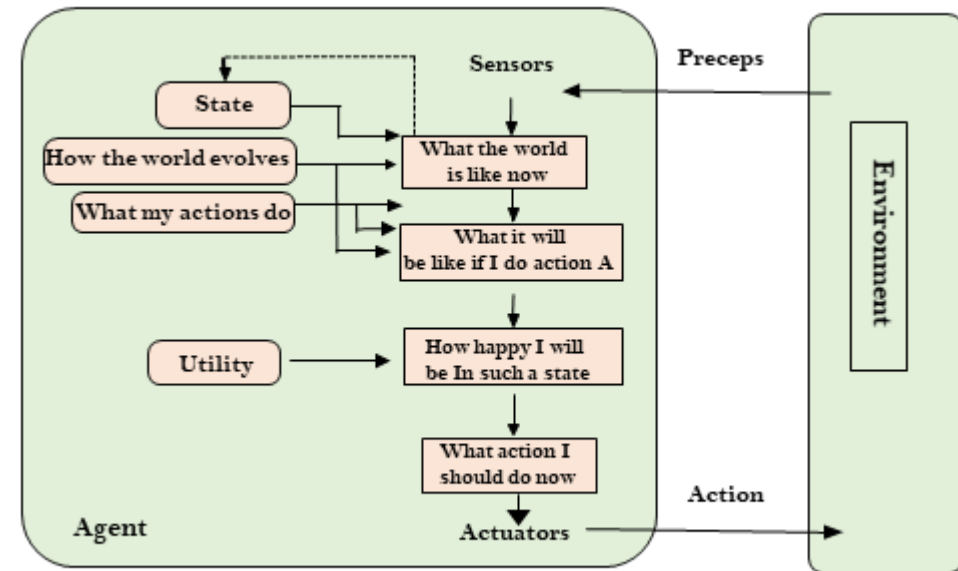


# Utility Based Agents (Cont...)

- Utility is therefore a function that maps a state onto a real number;
- Two kind of cases in Utility;

**when there are conflicting goals. (for example; speed or safety). - the utility function specifies the appropriate trade-off.**

**when there are several goals that the agent can aim for, none of which can be achieved with certainty. - utility function provides a way in which the likelihood of success can be weighed up against the importance of the goals**



# How does it Work

- A utility-based agent aims to choose actions that lead to a high utility state. To achieve this, it needs to model its environment, which can be simple or complex.
- Then, it evaluates the expected utility of each possible outcome based on the probability distribution and the utility function.
- Finally, it selects the action with the highest expected utility and repeats this process at each time step

# Advantages

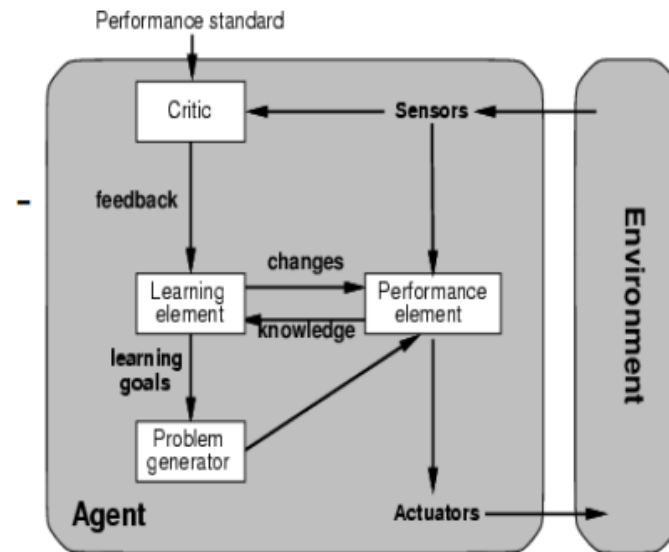
- The advantage of utility Based Agents are
  - Handles a wide range of decision-making problems
  - Learns from experience and adjusts their decision-making strategies
  - Offers a consistent and objective framework for decision-making

# Drawbacks

- Problems for the goal based approach design approach:
  - Requires an accurate model of the environment, failing to do so results in decision-making errors
  - Computationally expensive and requires extensive calculations
  - Does not consider moral or ethical considerations
  - Difficult for humans to understand and validate

# Learning Based Agents

- Learning agent aim to learn new actions as it goes about its business.



**Figure:** procedure for learning-based agent.

# Learning Based Agents

- A learning agent is comprised of (4) components:
  1. Learning Element:

It is responsible for learning and making improvements based on the experiences it gains from its environment.
  2. Critic:

It provides feedback to the learning element by the agent's performance for a predefined standard.
  3. Performance Element:

It selects and executes external actions based on the information from the learning element and the critic.
  4. Problem Generator:

It suggests actions to create new and informative experiences for the learning element to improve its performance.

# How does it Work?

- **Observation:** The learning agent observes its environment through sensors or other inputs.
- **Learning:** The agent analyzes data using algorithms and statistical models, learning from feedback on its actions and performance.
- **Action:** Based on what it has learned, the agent acts in its environment to decide how to behave.
- **Feedback:** The agent receives feedback about their actions and performance through rewards, penalties, or environmental cues.
- **Adaptation:** Using feedback, the agent changes its behavior and decision-making processes, updating its knowledge and adapting to its environment.

# Example:

A spam filter that learns to identify new types of spam emails based on user feedback is a learning agent.



# Advantages

- The agent can convert ideas into action based on AI decisions
- Learning intelligent agents can follow basic commands, like spoken instructions, to perform tasks
- Unlike classic agents that perform predefined actions, learning agents can evolve with time
- AI agents consider utility measurements, making them more realistic

# Drawbacks

- Prone to biased or incorrect decision-making
- High development and maintenance costs
- Requires significant computing resources
- Dependence on large amounts of data
- Lack of human-like intuition and creativity

# 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, Learning Based