LAB 1: INTELLIGENT AGENTS AND ENVIRONMENTS

1.1 Aim

To understand the concept of intelligent agents, environments, PEAS descriptions, and agent architectures by implementing simple agent-based systems.

1.2 Theory Overview

An **agent** perceives its environment through sensors and acts upon that environment through actuators. A **rational agent** acts to maximize its expected performance measure.

Agent Types (AIMA Ch. 2):

- Simple Reflex Agent
- Model-Based Agent
- Goal-Based Agent
- Utility-Based Agent
- Learning Agent

Environment Properties:

- Fully / Partially Observable
- Deterministic / Stochastic
- Episodic / Sequential
- Static / Dynamic

Solved Example: Vacuum Cleaner Agent

A vacuum cleaner agent operates in a simple environment consisting of two rooms, labeled A and B, arranged linearly such that room A is to the left of room B.

Each room can be in one of two states: Clean or Dirty.

The agent is located in exactly one room at any time and is equipped with sensors that allow it to perceive:

- its current location (A or B), and
- the cleanliness status (Clean or Dirty) of the current room only.

The agent can perform the following actions:

- Suck: cleans the current room if it is dirty,
- Left: moves the agent from room B to room A,
- Right: moves the agent from room A to room B.

The environment is assumed to be fully observable, deterministic, static, discrete, and single-agent.

The goal of the agent is to clean all rooms while minimizing unnecessary movements.

PEAS Description:

- Performance: Cleanliness, minimal movement
- Environment: Two rooms
- Actuators: Left, Right, Suck
- Sensors: Location, Dirt

Solution:

```python
class VacuumAgent:
    def __init__(self):
        self.location = 'A'

    def act(self, percept):
        location, status = percept
        if status == 'Dirty':
            return 'Suck'
        elif location == 'A':
            return 'Right'
        else:
            return 'Left'


# Example percept
agent = VacuumAgent()
print(agent.act(('A', 'Dirty')))
```

Experiment 2: Simple Reflex Agent in a 4-Room Grid Environment

Problem Statement: Extend the Simple Reflex Vacuum Cleaner Agent to operate in a 4-room grid environment, arranged as a 2 × 2 grid with rooms labeled (0,0), (0,1), (1,0), (1,1).

Each room can independently be Clean or Dirty, with the initial dirt configuration generated randomly at the start of the simulation.

Percepts:

At each time step, the agent perceives:

- Its current grid position (x, y)
- Cleanliness status of the current room

Actions

The agent can perform:

- Suck
- Up, Down, Left, Right

(Movement actions are valid only if they keep the agent within grid boundaries.)

Task Requirements

1. Modify the reflex agent to operate in the grid environment.
2. When the current room is clean, the agent must select a random valid movement action.
3. Run the simulation for a fixed number of steps and record:
    - Number of rooms cleaned
    - Total number of movements

Constraints

- The agent must not maintain any internal representation of the environment.
- Decisions must depend only on the current percept**.**

Expected Outcome

The agent should eventually clean multiple rooms but may revisit already cleaned rooms due to the lack of memory.

Experiment 3: Model-Based Vacuum Cleaner Agent

Problem Statement: Design and implement a **Model-Based Vacuum Cleaner Agent** for the **4-room grid environment** described in Experiment 2. Unlike the simple reflex agent, this agent must maintain

an **internal model** of the environment that records the cleanliness status of rooms it has already visited.

Percepts

- Current location of the agent
- Cleanliness status of the current room

Actions

- Suck
- Up, Down, Left, Right

Task Requirements

1. Maintain an internal data structure that stores the cleanliness status of each visited room.
2. Update the internal model after each percept.
3. Use the model to:
    - Avoid revisiting rooms known to be clean when possible
    - Prefer movement toward unvisited or dirty rooms
4. Terminate the simulation when all rooms are known to be clean.

Constraints

- The agent may only update its model based on its own percepts.
- The environment remains deterministic and static.

Expected Outcome

The model-based agent should clean all rooms with **fewer redundant movements** compared to the simple reflex agent.

1. Why does the simple reflex agent perform redundant actions in the grid environment?
2. How does the internal model improve agent performance?
3. Compare the simple reflex and model-based agents in terms of:
    a. Rationality
    b. Efficiency
    c. Scalability
4. Would the model-based agent still work correctly in a partially observable environment? Justify.