**Experiment-2**

Write a program to control the VACUUM Cleaner moves (Intelligent systems design process)

**Duration: 2 hours**

**Objective:**

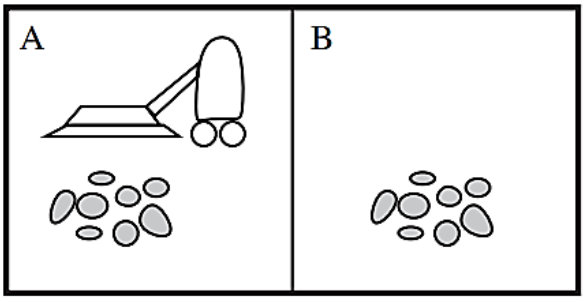
Designing a program to control the movements of a vacuum cleaner involves creating an intelligent system with predefined objectives. Here's how the intelligent systems design process can be structured for such a program.

**Objectives:**

1. **Efficient Cleaning**: Cover the entire area with minimal redundant paths to save time and energy.
2. **Obstacle Avoidance**: Detect and avoid obstacles, ensuring smooth operation without damage.
3. **Adaptive Navigation**: Adapt to different room layouts and dynamically plan the path based on the environment.

**The Problem:**

Consider a Vacuum cleaner world



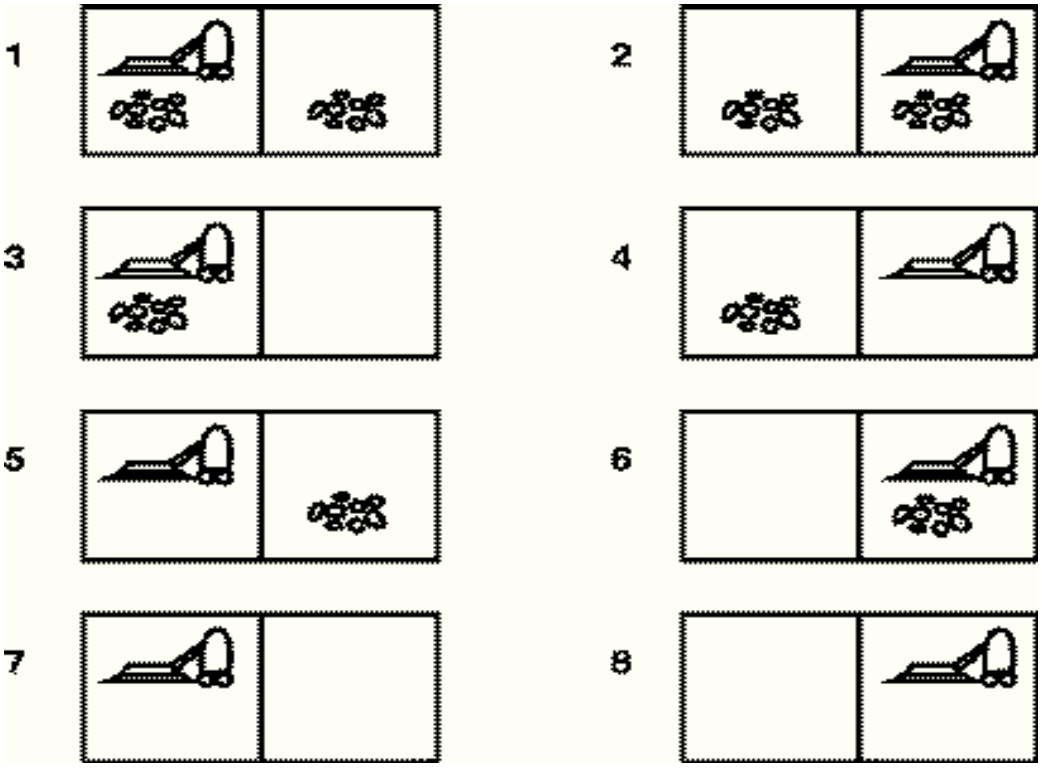

- There are 2 tiles: Tile A and Tile B. The pile of dust could be on any, on both or on neither of the two tiles.
- Your vacuum cleaner is on one of the tiles, and it can sense and clean only one tile at a time.

- The cleaner could move left, move right, or clean the tile it rests on.

The percept sequence consist of the history of percepts the vacuum cleaner senses, and the agent function maps each percept of the sequence to an action.
The agent function for this vacuum cleaning agent is:

**States:**

**So, there are eight possible states possible in our vacuum cleaner problem**




States 1 and 2 are our initial states and state 7 and state 8 are our final states (goal states). This means that, initially, both the rooms are full of dirt and the **vacuum cleaner** can reside in any room. And to reach the final goal state, both the rooms should be clean and the **vacuum cleaner** again can reside in any of the two rooms.

The **vacuum cleaner** can perform the following functions: move left, move right, move forward, move backward and to suck dust. But as there are only two rooms in our problem, the vacuum cleaner performs only the following functions here: move left, move right and suck.

Here the performance of our agent (vacuum cleaner) depends upon many factors such as time taken in cleaning, the path followed in cleaning, the number of moves the agent takes in total, etc. But we consider two main factors for estimating the performance of the agent. They are:

1. **Search Cost:** How long the agent takes to come up with the solution.
2. **Path cost:** How expensive each action in the solution are.
3. World state space: **2 positions, dirt or no dirt 8 world states**
4. Actions: **Left (L), Right (R), or Suck (S)**
5. Goal: **no dirt in the rooms**
6. Path costs: **one unit per action**

**Agent function** = {([A: Clean], Right), ([A: Dirty], Clean), ([B: Clean], Left), ([B: Dirty], Clean), ([A: Clean, B: Clean], Stop), ([A: Clean, B: Dirty], Clean), …}

1. **Possible States**

- Both tiles are dirty: [A:Dirty,B:Dirty]
- Only Tile A is dirty: [A:Dirty,B:Clean]
- Only Tile B is dirty: [A:Clean,B:Dirty]
- Both tiles are clean: [A:Clean,B:Clean]

## Actions Required

The agent can start on **Tile A** or **Tile B**. Actions will vary based on where it starts and the dirt configuration:

***Case 1: Both tiles are dirty [A:Dirty,B:Dirty]***

- **Starting on Tile A:**
    1. Clean Tile A: Clean
    2. Move to Tile B: Right
    3. Clean Tile B: Clean
    - **Total steps: 3**
- **Starting on Tile B:**
    0. Clean Tile B: Clean
    1. Move to Tile A: Left
    2. Clean Tile A: Clean

- **Total steps: 3**

***Case 2:*** ***Only Tile A is dirty [A:Dirty,B:Clean]***

- **Starting on Tile A:**
  1. Clean Tile A: Clean
  - **Total steps: 1**
- **Starting on Tile B:**
  0. Move to Tile A: Left
  1. Clean Tile A: Clean
  - **Total steps: 2**

***Case 3:*** ***Only Tile B is dirty [A:Clean,B:Dirty]***

- **Starting on Tile A:**
  1. Move to Tile B: Right
  2. Clean Tile B: Clean
  - **Total steps: 2**
- **Starting on Tile B:**
  0. Clean Tile B: Clean
  - **Total steps: 1**

***Case 4:*** ***Both tiles are clean [A:Clean,B:Clean]***

***The agent will immediately stop.***

- **Total steps: 0**

**3. Total Steps Across All Scenarios**

Summing up the steps for all possible initial states:

1. Both dirty: 3+3=63 + 3 = 63+3=6
2. Only Tile A dirty: 1+2=31 + 2 = 31+2=3
3. Only Tile B dirty: 2+1=32 + 1 = 32+1=3
4. Both clean: 000

---

**Final Total: 6+3+3+0=126 + 3 + 3 + 0 = 126+3+3+0=12 Steps**

This represents the total steps across all scenarios for one cleaning cycle in the vacuum cleaner world.

## TASK-3

1. Write a program to control the VACUUM Cleaner moves (Intelligent systems design process) based on different different case studies.
2. Write the program based on proper understanding of the case studies.

**Answers : Program**

**Answers : Output**

**Name:**

**Reg_no:**

**Date:**