

# Artificial Intelligence (18CSC305J)

Faculty : Helen Victoria A

## Team Tesla 2.0

Name	Regno.	Email Id
Abhighyan B	RA1911033010002	ab2134@srmist.edu.in
Sanjana N B	RA1911033010016	sn8740@srmist.edu.in
Roehit Ranganathan	RA1911033010017	rr9344@srmist.edu.in
Venkata Naga Sai Ram Nomula	RA1911033010021	vn4903@srmist.edu.in
K. Dushyant Reddy	RA1911033010029	kr7119@srmist.edu.in

## Experiment 2 - Agent Function

### Problem Statement:

Write an Agent Program that perceives the Environment and reacts according to the perception with the help of Agent Functions.

---

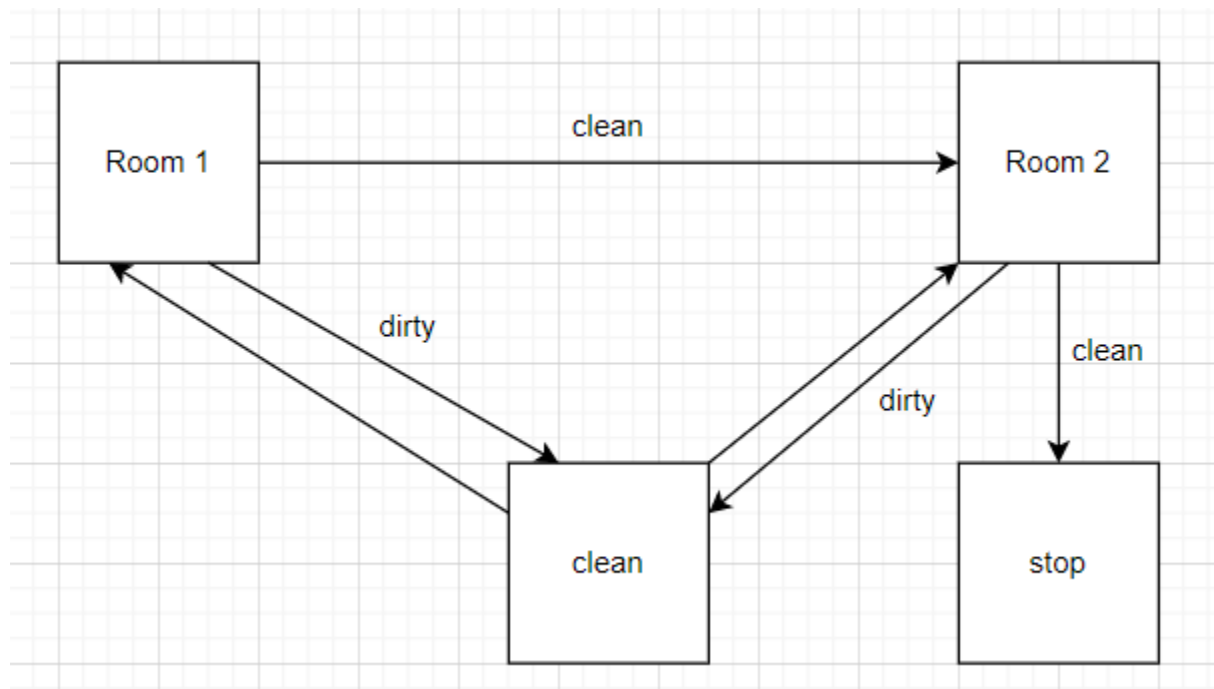
### Algorithm:

1. Get input for number of rooms
2. Get input whether the rooms are dirty/clean
3. Get initial position of the vacuum cleaner
4. Define forward function, which runs a loop with the condition that the number of rooms not equal to the starting position. If that room is dirty, it changes it to clean and prints the same. While incrementing the starting position by one.
5. Define Backward function, which runs a loop with the condition that the starting point is more than zero. Checks for the conditions of the room, and cleans it if necessary, and decrements the starting position.

- Both forward and backward functions are called and the rooms are cleaned.

---

### State space tree Diagram:



### Code:

```
def GoForward(rooms, pos):
    while pos != len(rooms):
        if rooms[pos] == False:
            rooms[pos] = True
            print("Room {} is dirty, Clean and moving forward".format(pos+1))
            pos += 1
        else:
            print("Room {} is clean, moving forward".format(pos+1))
            pos += 1
    print("Reached the End, so turning back")

def GoBackward(rooms, pos):
    while pos >= 0:
```

```

        if rooms[pos] == False:
            rooms[pos] = True
            print("Room {} is dirty, Clean and moving back".format(pos+1))
            pos -= 1
        else:
            print("Room {} is clean, moving back".format(pos+1))
            pos -= 1
        print("Reached the start, so turning back")

n = int(input("Enter Number of Rooms: "))
rooms = []
for i in range(n):
    x = int(input("Enter 1 for Clean and 0 for dirty for room {} :
".format(i+1)))
    if x == 1:
        rooms.append(True)
    else:
        rooms.append(False)
pos = int(input("Enter Initial position of the vacuum cleaner
(1-{})".format(n)))
pos -= 1
while rooms != [True for i in range(n)]:
    GoForward(rooms, pos)
    GoBackward(rooms, pos)
print("All Rooms are cleaned")

```

## Output:

```

Enter Number of Rooms: 5
Enter 1 for Clean and 0 for dirty for room 1 : 0
Enter 1 for Clean and 0 for dirty for room 2 : 1
Enter 1 for Clean and 0 for dirty for room 3 : 0
Enter 1 for Clean and 0 for dirty for room 4 : 1
Enter 1 for Clean and 0 for dirty for room 5 : 0
Enter Initial position of the vacuum cleaner (1-5)3
Room 3 is dirty, Clean and moving forward
Reached the End, so turning back
Room 4 is clean, moving forward
Reached the End, so turning back
Room 5 is dirty, Clean and moving forward
Reached the End, so turning back
Room 3 is clean, moving back
Reached the start, so turning back

```

```
Room 2 is clean, moving back  
Reached the start, so turning back  
Room 1 is dirty, Clean and moving back  
Reached the start, so turning back  
All Rooms are cleaned
```

**Type of Agent Used:** Simple Reflex Agent.

**Performance Measures:** of a vacuum cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc. It will work only if the correct decision can be made on the basis of the current perception – that is only if the environment is fully observable.

**Real World Solution:**

- Search in the Internet
- Finding routes  
Routing in networks
- Traveling Salesman Problem (TSP)
- VLSI Layout  
Chip layout
- Robot navigation  
Generalization of route-finding problem (continuous space)

**Result:** Vacuum Cleaner Agent problem has been implemented successfully.