ARTIFICIAL INTELLIGENCE

LAB 5

Submitted by

Adil Hussain Mughal (12084)

BS (SE-5th) MORNING

Date: 21 May 2021

Submitted to: Sir Mr. Faiq Ahmed

DEPARTMENT OF ENGINEERING NATIONAL UNIVERSTIY OF MODERN LANGUAGES, ISLAMABAD

LAB TASK 1:

Consider the vacuum world shown in the figure below: This particular world has just two locations: squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt, or do nothing. One very simple agent function is the following: if the current square is dirty, then suck, otherwise move to the other square. Write a simple reflex agent for the vacuum cleaner. (Hint: Agent has no initial states knowledge) If the current square is dirty, then suck; otherwise, move to the other square.

Pseudocode to the task is as follows;

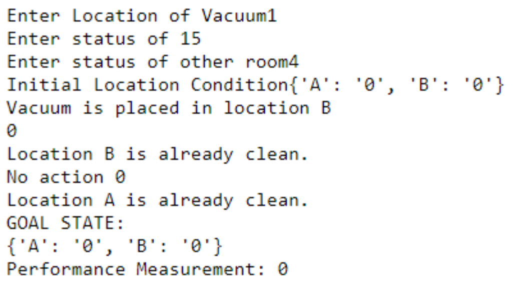
function Reflex-Vacuum-Agent([location,status]) returns an action if status = Dirty then return Suck else if location = A then return Right else if location = B then return Left

Code:

```
#INSTRUCTIONS

#Enter LOCATION A/B in captial letters

#Enter Status O/1 accordingly where 0 means CLEAN and 1 means DIRTY


def vacuum_world():

        # initializing goal_state

        # 0 indicates Clean and 1 indicates Dirty

    goal_state = {'A': '0', 'B': '0'}

    cost = 0


    location_input = input("Enter Location of Vacuum") #user_input of location vacuum is placed

    status_input = input("Enter status of " + location_input) #user_input if location is dirty or clean

    status_input_complement = input("Enter status of other room")

    print("Initial Location Condition" + str(goal_state))


    if location_input == 'A':

        # Location A is Dirty.

        print("Vacuum is placed in Location A")

        if status_input == '1':

            print("Location A is Dirty.")
    # suck the dirt  and mark it as clean
    goal_state['A'] = '0'
    cost += 1                      #cost for suck
    print("Cost for CLEANING A " + str(cost))
    print("Location A has been Cleaned.")

    if status_input_complement == '1':
        # if B is Dirty
        print("Location B is Dirty.")
        print("Moving right to the Location B. ")
        cost += 1                      #cost for moving right
        print("COST for moving RIGHT" + str(cost))
        # suck the dirt and mark it as clean
        goal_state['B'] = '0'
        cost += 1                      #cost for suck
        print("COST for SUCK " + str(cost))
        print("Location B has been Cleaned. ")
    else:
        print("No action" + str(cost))
        # suck and mark clean
        print("Location B is already clean.")

if status_input == '0':
    print("Location A is already clean ")
    if status_input_complement == '1':# if B is Dirty
        print("Location B is Dirty.")
        print("Moving RIGHT to the Location B. ")
        cost += 1                      #cost for moving right
        print("COST for moving RIGHT " + str(cost))
```

```
            # suck the dirt and mark it as clean
            goal_state['B'] = '0'
            cost += 1                       #cost for suck
            print("Cost for SUCK" + str(cost))
            print("Location B has been Cleaned. ")
        else:
            print("No action " + str(cost))
            print(cost)
            # suck and mark clean
            print("Location B is already clean.")

else:
    print("Vacuum is placed in location B")
    # Location B is Dirty.
    if status_input == '1':
        print("Location B is Dirty.")
        # suck the dirt  and mark it as clean
        goal_state['B'] = '0'
        cost += 1  # cost for suck
        print("COST for CLEANING " + str(cost))
        print("Location B has been Cleaned.")

        if status_input_complement == '1':
            # if A is Dirty
            print("Location A is Dirty.")
            print("Moving LEFT to the Location A. ")
            cost += 1  # cost for moving right
            print("COST for moving LEFT" + str(cost))
            # suck the dirt and mark it as clean
            goal_state['A'] = '0'
            cost += 1  # cost for suck
            print("COST for SUCK " + str(cost))
            print("Location A has been Cleaned.")

    else:
        print(cost)
        # suck and mark clean
        print("Location B is already clean.")

        if status_input_complement == '1':  # if A is Dirty
            print("Location A is Dirty.")
            print("Moving LEFT to the Location A. ")
            cost += 1  # cost for moving right
            print("COST for moving LEFT " + str(cost))
            # suck the dirt and mark it as clean
            goal_state['A'] = '0'
            cost += 1  # cost for suck
            print("Cost for SUCK " + str(cost))
            print("Location A has been Cleaned. ")
        else:
            print("No action " + str(cost))
            # suck and mark clean
            print("Location A is already clean.")

# done cleaning
print("GOAL STATE: ")
print(goal_state)
print("Performance Measurement: " + str(cost)
vacuum_world()
```

Output:



```
Enter Location of Vacuum1
Enter status of 15
Enter status of other room4
Initial Location Condition{'A': '0', 'B': '0'}
Vacuum is placed in location B
0
Location B is already clean.
No action 0
Location A is already clean.
GOAL STATE:
{'A': '0', 'B': '0'}
Performance Measurement: 0
```

For viewing code, please link on the link below

https://github.com/Enggadil/-AI-LAB-_BSSE-5-M-