7/12/22

Vaccum Cleaner agent.

Aim: To implement Vaccum cleaner agent program

Algorithm:

```
if location A = dirty
    cost + = 1
    clean A
    if location B is dirty:
        move to B
        cost + = 1
        clean B
        cost + = 1
    else:
        no action
```

```
if location A is clean:
    if location B = "dirty"
        cost + = 1
        move to B
        clean B
        cost + = 1
    else
        No action.
```

```
if location B = dirty.
    cost + = 1
    clean B
    if location A is dirty.
        move to A.
        cost + = 1
        clean A
        cost + = 1.
```

```
if location B is clean.
    if location A is dirty
        move to A
        cost + = 1
        clean A
        cost + = 1
    else
        No action
```

```
print (path cost).
print (goal state).
```

code:

```
def vaccum_world():
    goal_state = {'A':'0', 'B':'0'}.
    cost = 0
    location_input = input("Enter the location of the
                            vaccum : ")
    status_input = input("Enter the status of "+
                         location_input + " ")
    status_input_complement = input("Enter status of
                                     other room")

    if location_input == 'A':
        print("Vaccum is placed in location A")
        if status_input == 1:
            print("location A is dirty.")
            goal_state['A']='0'
            cost += 1
            print("cost for cleaning A "+str(cost))
            print("location A has been cleaned.")

        if status_input_complement == '1':
            print("location B is dirty")
            print("Moving right to location B")
            cost += 1
            print("cost for moving right "+str(cost))
            goal_state['B'] = '0'
            cost += 1
            print("cost for suck "+str(cost))
            print("location B has been cleaned.")

        else:
```

```
        print("No action" + str(cost))
        print("location B is already clean")

    if status_input == '0':
        print("location A is already clean")
        if status_input_complement == '1':
            print("location B is dirty.")
            print("moving right to location B")
            cost += 1
            print("cost for moving right" + str(cost))
            goal_state['B'] = '0'
            cost += 1
            print("cost for suck" + str(cost))
            print("location B has been cleaned")

else:
    print("vaccum is placed in location B")
    if status_input == '1':
        print("location B is dirty")
        goal_state['B'] = '0'
        cost += 1
        print("cost for cleaning" + str(cost))
        print("location B has been cleaned")

    if status_input_complement == '1':
        print("location A is dirty")
        print("Moving left to location A")
        cost += 1
        print("cost for moving left" + str(cost))
        goal_state['A'] = '0'
        cost += 1
        print("cost for suck" + str(cost))
        print("location A has been cleaned")
```

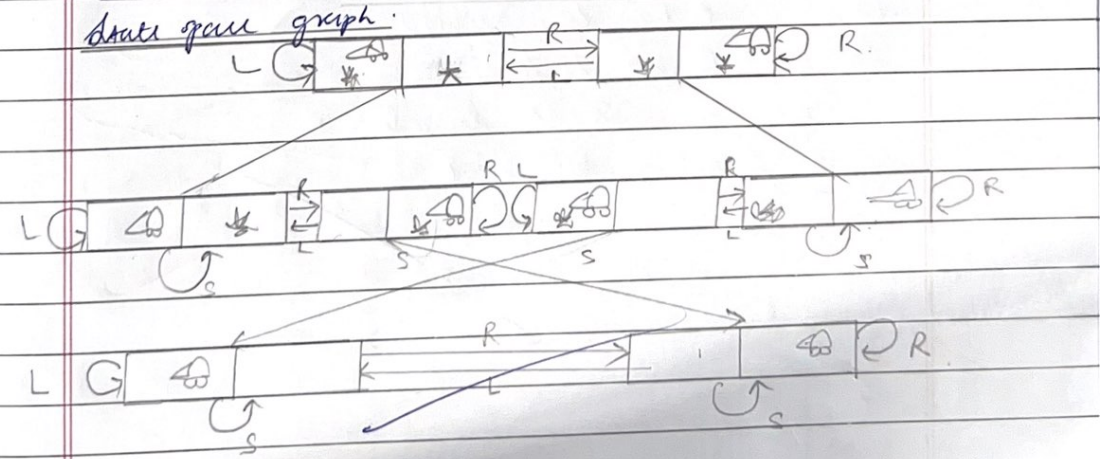
```
else:
    print (cost)
    print("location B is already clean")
    if status_input_complement == '1':
        print("location A is dirty")
        print("Moving left to location A")
        cost += 1
        print("cost for moving left "+str(cost))
        goal_state['A'] = '0'
        cost += 1
        print("cost for suck "+ str(cost))
        print("location A has been cleaned")

else:
    print("No action "+str(cost))
    print("location A is already clean")


print("goal state:")
print(goal_state)
print("performance measurement: "+str(cost))

vacuum_world()
```

State space graph:



Vacuum Cleaner O/p.

Enter the location of Vacuum:
A

Enter the status of A
1

Enter status of other room.
1

Vacuum is placed in location A.
location A is dirty.
cost for cleaning A : 1
location A has been cleaned.
location B is dirty
moving right to the location B.
cost for moving right : 2
cost for such : 3.
location B has been cleaned
Goal State :
{'A' : '0' : 'B' : '0'}
performance measurement : 3.