**PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES**

**Department of Computer and Information Sciences (DCIS)**

**Dated: 19-07-2017**

**Assignment: Artificial Intelligence**

**Name: Muhammad Dawood**

**Registration Number: 30251**

# Assignment Title: Cleaning Up Using Search

## Using Robot for cleaning rooms:

## Description of the Robot:

An electric robot that can move left or right only using its electric battery. The robot can suck the dirt in the room it is currently in. Its battery capacity is 6 units. It can charge its battery by exactly one unit in a unit time step if it is connected to a battery charger and not doing anything else (i.e., moving or vacuuming). The robot's maximum battery capacity is 4 units and it cannot be charged beyond that. The robot can be charged irrespective of its current battery level but the minimum charging time is 1 time step. The robot can perform only one action at a time. Every movement of the robot requires 1 unit of electricity whereas performing suck requires 2 units. If the battery reaches 0 then the robot dies. The robot knows its location in the environment, i.e., it knows what room it is in.

## Environment:

A building with three rooms named Left (L), Middle (M) and Right (R). The left room is connected to the middle room which is connected to the right room and there is no way to move from L to R or vice-versa without having to going through M. Every room can be either dirty (D) or clean (C). A dirty room can be cleaned by vacuuming. Only Room L has a charging station for the robot and the charging station has infinite capacity, i.e., it can keep on charging for as long as required. If the robot tries to move left in room L it stays in L. If it tries to move right in room R, it stays in R. If the robot executes "Suck" in a clean room, the room remains clean.

## Goal:

Keep all rooms clean in minimum time.

**Starting State:**

Any state can be the initial state. The robot begins with a full charge.

**Required:**

Find a solution that reaches the goal from the initial state. As a test case, you can start off in room L with full battery with all three rooms dirty.

**Steps required:**

Use the pseudo-code of the goal-based agent in the slides or the AIMA book to guide you. Please choose a state representation and write the path cost and successor functions for all possible states. Construct the search tree.

**Questions:**

1. Will the robot always be able to clean all rooms? Why or why not?

2. Will the robot always be able to clean all rooms if we require that the robot end up in room L after cleaning up? Why or Why not?

3. What is the minimum battery capacity for the robot to be able to clean all rooms in the above two cases? Alternatively, by what minimum factor should the robot be made more power-efficient if it were to solve this problem?

4. Can you design an algorithm that cleans up the room with minimum battery power consumption?

5. What is the optimal place to put the charging station? Why?

## Solution to the cleaning problem

**Goal:**

Keep all rooms clean in minimum time.

**States:**

Total number of possible states we have are $3*2^3=24$

**Initial state:**

Any state can be initial robot is fully charged.

**Successor Function:**

S(In(R1,B,(D/C,D/C,D/C),t))={<L,R1,B=B-1,(D/C,D/C,D/C), t=t+1>, <R,R2, B=B-1,(D/C,D/C,D/C), t=t+1>, <C,R1, B=B+1,(D/C,D/C,D/C),1>,<S,R1, B=B-2,(C,D/C,D/C), t=t+1> }

S(In(R2,B,(D/C,D/C,D/C),t)))={<L,R1, B=B-1,(D/C,D/C,D/C), t=t+1>, <R,R3, B=B-1,(D/C,D/C,D/C), t=t+1>, <S,R2, B=B-2,(C,C,D/C), t=t+1> }

S(In(R3,B,(D/C,D/C,D/C),t)))={<L,R2, B=B-1,(D/C,D/C,D/C), t=t+1>, <R,R3,5,(D/C,D/C,D/C), t=t+1>, <S,R3, B=B-2,(C,C,D/C), t=t+1> }

S (In (R1, B, (C, D/C, D/C), t)) = {<L,R1, B=B-1,(C,D/C,D/C),t=t+1>, <R,R2, B=B-1,(C,D/C,D/C),2>, <C,R1,B+1,(D/C,D/C,D/C),t+1> }

S (In (R2, B, (C, D/C, D/C), t)) = {<L,R1, B=B-1,(C,D/C,D/C), t=t+1>, <R,R3, B=B-1,(C,D/C,D/C), t=t+1>,<S,R2, B=B-2,(C,C,D/C), t=t+1> }

S (In (R3, B, (C, D/C, D/C), t)) = {<L,R2, B=B-1,(C,D/C,D/C),t=t+1>, <R,R3, B=B-1,(C,D/C,D/C),t=t+1>,<S,R3, B=B-2,(C,D/C,C),t=t+1> }

S (In (R3, B, (C/D, C/D, C), t)) = {<L,R2, B=B-1,(C/D,D/C,C), t=t+1>, <R,R3, B=B-1,(C/D,D/C,C), t=t+1> }

S (In (R2, B, (C/D, C/D, C), t)) = {<L,R1, B=B-1,(C/D,D/C,C), t=t+1>, <R,R3, B=B-1,(C/D,D/C,C), t=t+1>,<S,R2, B=B-2,(C/D,C,C), t=t+1>, }

S (In (R1, B, (C/D, C/D, C), t)) = {<L,R1, B=B-1,(C/D,D/C,C), t=t+1>, <R,R2, B=B-1,(C/D,D/C,C), t=t+1>,<S,R1, B=B-2,(C,C/D,C), t=t+1>,<C,R1, B=B=1,(C/D,D/C,C), t=t+1> }

S (In (R2, B, (C/D, C, D/C), t)) = {<L,R1, B=B-1,(C/D,C,D/C), t=t+1>, <R,R3, B=B-1,(C/D,C,D/C), t=t+1> }

S (In (R1, B, (C/D, C, D/C), t)) = {<L,R1, B=B-1,(C/D,C,D/C), t=t+1>, <R,R2, B=B-1,(C/D,C,D/C), t=t+1> , <S,R1, B=B-2,(C,C,D/C), t=t+1>,<C,R1, B=B=1,(C/D,D/C,C), t=t+1> }

S (In (R3, B, (C/D, C, D/C), t)) = {<L,R2, B=B-1,(C/D,C,D/C), t=t+1>, <R,R3, B=B-1,(C/D,C,D/C), t=t+1>,<S,R3, B=B-2,(D/C,C,C), t=t+1>, }

S (In (R1, B, (C, C, D/C), t)) = {<L,R1, B=B-1,(C,C,D/C), t=t+1>, <R,R2, B=B-1,(C,C,D/C), t=t+1>,<C,R1, B=B+1,(C/D,D/C,C), t=t+1> }

S (In (R2, B, (C, C, D/C), t)) = {<L,R1, B=B-1,(C,C,D/C), t=t+1>, <R,R3, B=B-1,(C,C,D/C), t=t+1>}

S (In (R3, B, (C, C, D/C), t)) = {<L,R2, B=B-1,(C,C,D/C), t=t+1>, <R,R3, B=B-1,(C,C,D/C), t=t+1>,<S,R3, B=B-2,(C,C,C), t=t+1> }

S (In (R1, B, (D/C, C, C), t)) = {<L,R1, B=B-1,(D/C,C,C), t=t+1>, <R,R2, B=B-1,(D/C,C,C), t=t+1>,<C,R1, B=B+1,(C/D,C,C), t=t+1>,

<S,R1, B=B-2,(C,C,C), t=t+1> }

S (In (R2, B, (D/C, C, C), t)) = {<L, R1, B=B-1, (D/C,C,C), t=t+1>, <R,R3, B=B-1,(D/C,C,C), t=t+1>}

S (In (R3, B, (D/C, C, C), t)) = {<L,R2, B=B-1,(D/C,C,C), t=t+1>, <R,R3, B=B-1,(D/C,C,C), t=t+1>}

S (In (R1, B, (C, D/C, C), t)) = {<L,R1, B=B-1,(C,D/C,C), t=t+1>, <R,R2, B=B-1,(C,D/C,C), t=t+1>,<C,R1, B=B+1,(C,D/C,C), t=t+1>,}

S (In (R2, B, (C, D/C, C), t)) = {<L, R1, B=B-1, (C,D/C,C), t=t+1>, <R,R3, B=B-1,(C,D/C,C), t=t+1>,<S,R2, B=B-2,(C,C,C), t=t+1>}

S (In (R3, B, (C, D/C, C), t)) = {<L, R2, B=B-1, (C,D/C,C), t=t+1>, <R,R3, B=B-1,(C,D/C,C), t=t+1>}

**Goal test:**

Check wither all rooms are clean or not.

**Path Cost:**

Time taken to clean all the room. 1 unit for all operation i.e: Suck, Left, Right.

**Conditions:**

Battery Capacity is 6 volt.

Battery consumption is 1 unit for Left, Right movement, 2 unit for Suck operation.

Battery can only be charged at left room. At a time we perform only one operation.

**Search tree:**

Figure 1 Show the Search tree of Vacuum cleaner.

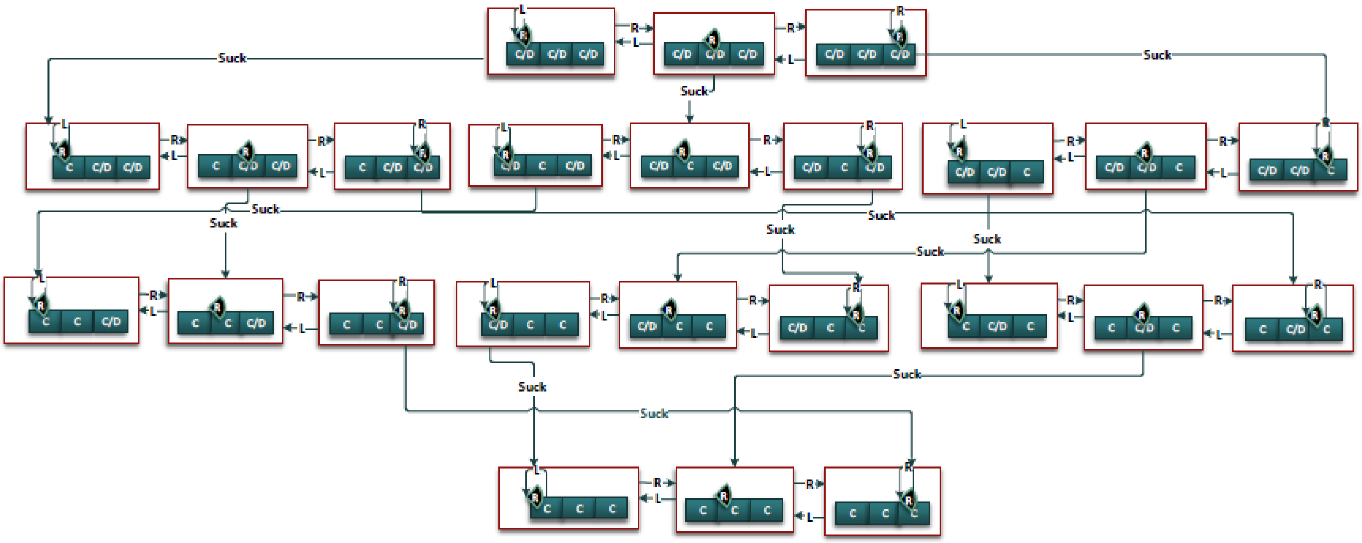


Figure 1 Search tree Vacuum Cleaner

**Questions:**

**Q1. Will the robot always be able to clean all rooms? Why or why not?**

The cleaning of all the room depends on the action sequence the robot actually execute. For finding this action sequence we are using search techniques. If we select proper search techniques robot will always be able to clean the room.

Figure 2 Show a simple scenario in which the robot stuck and not able to clean all the rooms. The reason for this is that robot is taking greedy choice at each stage robot try to reach the goal and try path which give height advantage. As a result due to this greedy choice robot stuck.

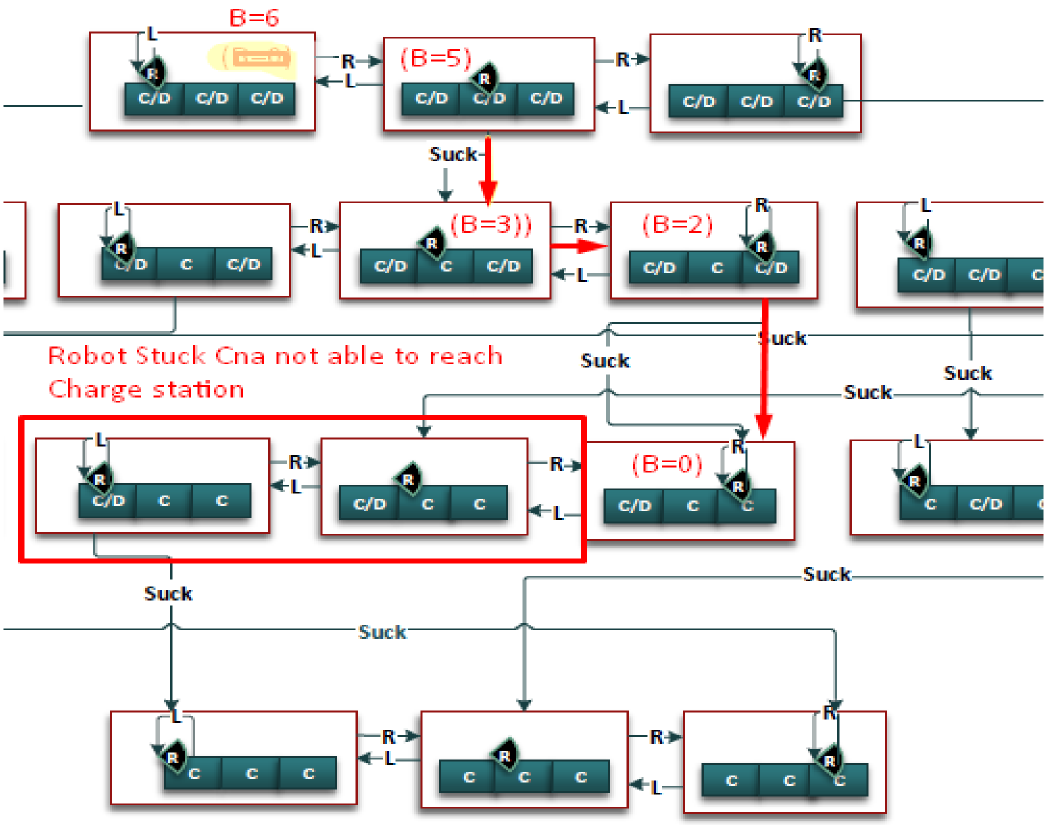


Figure 2 Robot Stuck

**Q 2. Will the robot always be able to clean all rooms if we require that the robot end up in room L after cleaning up? Why or Why not?**

Yes, this will always be the case for robot to reach to room L1 after cleaning all room, if we charge robot again and again. **Error! Reference source not found.** show a simple case in which the robot reach to Middle room and battery empty in this case the battery is not charged for second time, path is shown by the red arrow.

After charging battery for second time the robot end up in room L as shown by green line in **Error! Reference source not found.**. But charging battery again and again will be time consuming as a result cleaning all room will take a lot of time.

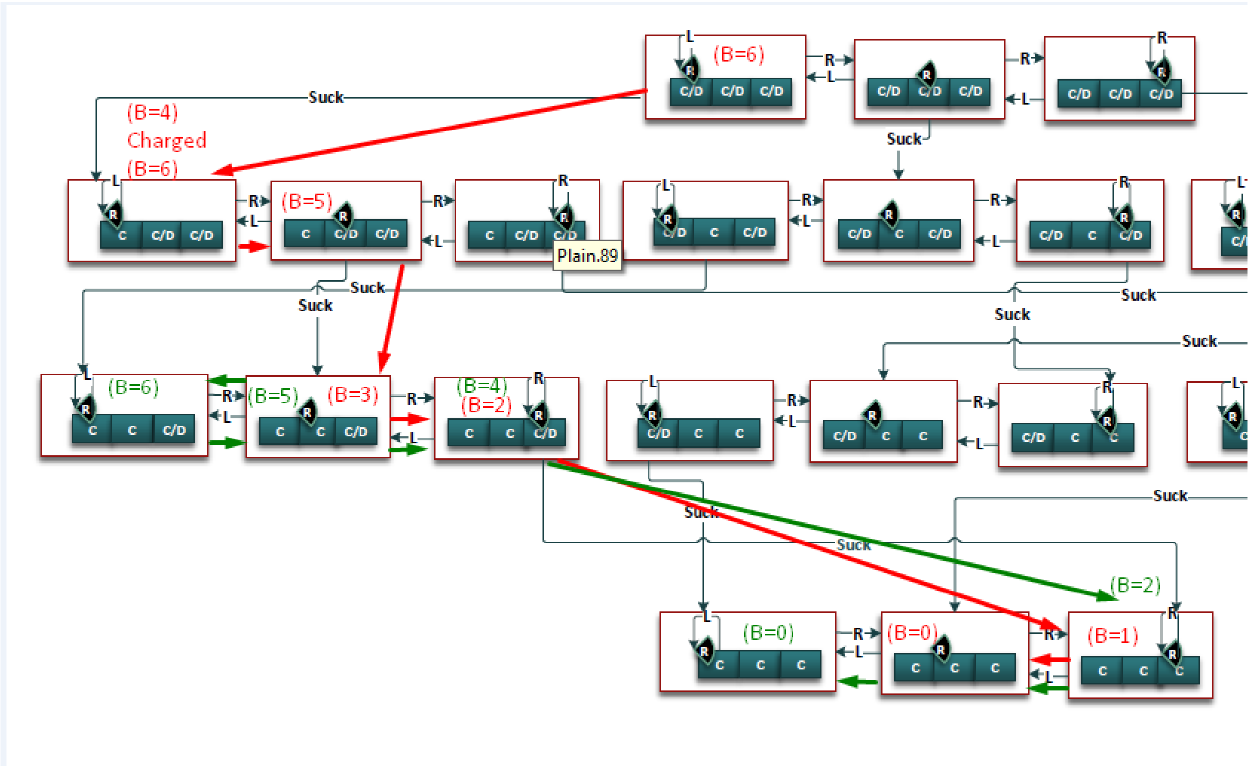


Figure 3 Reaching or not Reaching L1 after cleaning

**Q 3. What is the minimum battery capacity for the robot to be able to clean all rooms in the above two cases? Alternatively, by what minimum factor should the robot be made more power-efficient if it were to solve this problem?**

In the search tree we have two optimal solution to the vacuum cleaning problem. Both of these path result in 7 unit of time shown by green and blue. If we consider red path after cleaning R and Middle rooms, robots run out of battery as a result if wait two unit time to reach battery level at least 2 to clean the Left room as a result the time become 7 unit. If we increase the battery to 8 volt all room will take only 5 unit time clean, if we start the robot cleaning the room from right.

Same is the case when robot start cleaning the room from left to right. In this case robot after cleaning Left room recharge the battery to 6 unit again, it have to wait 2 unit for battery to charge, after that cleaning the rooms. But in this case problem is that if the robot start cleaning the room for next time it have to increase battery to 10 volts to get robot back to room Left.

**Q 4. Can you design an algorithm that cleans up the room with minimum battery power consumption?**

The algorithm which I prefer for this problem is enumerative techniques.

**Pseudo-code:**

```
Clean Room Right.

Move to the left 1 step.

Clean room Middle.

Move to left 1 step.

Charge battery.

Clean room Left.
```

In this case the battery power is minimum, we just use the actual power for cleaning rooms, no extra penalty.

**Q 5. What is the optimal place to put the charging station? Why?**

The optimal place for charging station is Left or Right. Because in both the cases we have no penalty for getting back to charge station. But in case of placing charge station at center we have some penalty for getting back to charge station and path is traversed again.

In case of charging station at left or right the cost is same. If at one time charge station is at left, starting from right after robot reach Left it will charge battery and will clean the Left room. Next time the robot will end’s up cleaning room task at right room we have to pay penalty to get robot back to Left room.

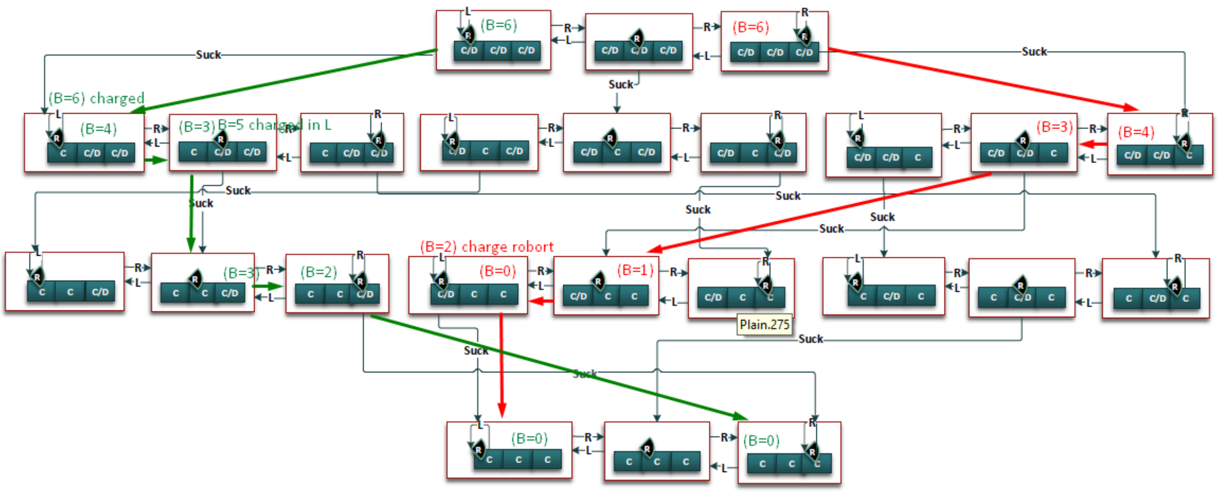

Figure 4  minimum battery