MDL-Assignment : Search

Khushi Agarwal

2020101092

Ques 1.

a) Initial state: The initial configuration given to us is the initial state.

b) Goal test: The final goal configuration given to us is the goal test/state.

c) Actions: It is all the possible moves that the robot arm can perform.

Pickup(A) - picks up A (present on the table or on another block)

Pop(A, B) - remove block A from block B

Push(A,B) - Put block A on block B

Place(A) - Place A on the table

d) States: The possible config b/w two blocks or the block and the table.

InHand(A)- The robot has A in it's arms.

EmptyHand() - The robots arm is empty

Top(A, B) - Block A is on Block B

TableTop(A) - Block A is on table

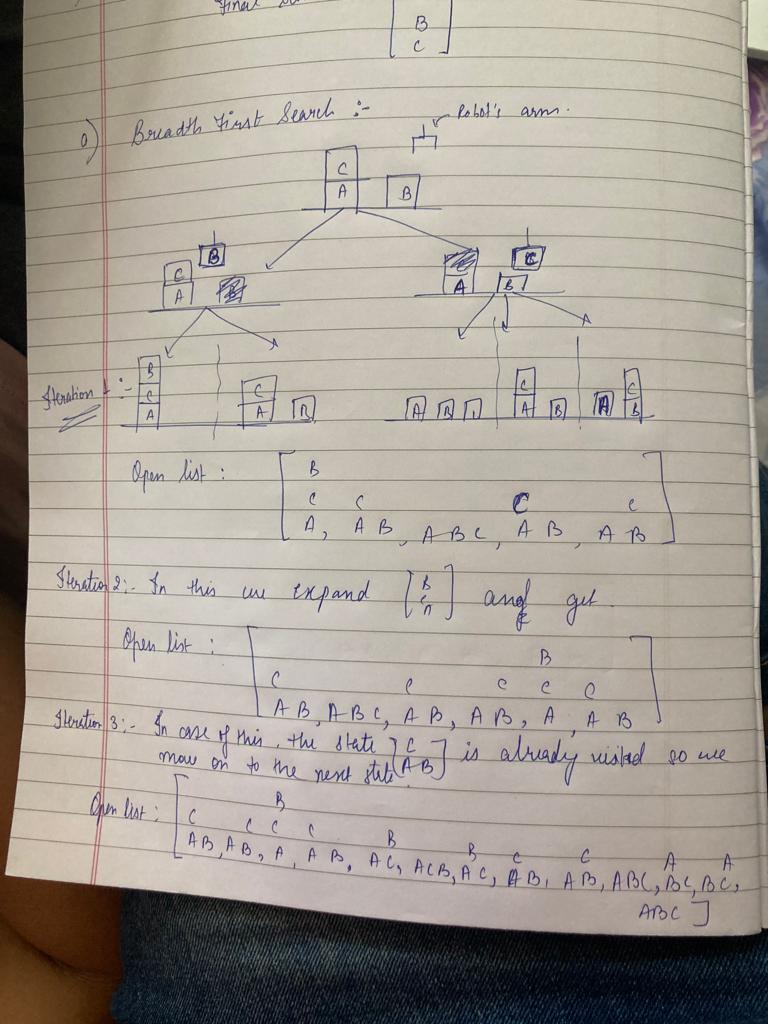
Clear(A) - Nothing is on Block A

e) Path cost:

The cost is 1 for every action i.e moving a block from its current position to another position

The optimal solution is the solution with the minimum cost (or else the minimum actions required to get from the initial state to the goal state).

Ques 2 .



B.

Depth-First Search:-

1. Depth-first
   1. First iteration

[B ]

C C C C

A, A B, A B C, A B, A B

* 1. Second Iteration

[ B ]

C C C C C

A, A B, A B, A B C, A B, A B

* 1. Third iteration

Because the configuration (node) [ B ] is already visited in our DFS in the

C

A

second iteration, we move on to the next node.

Now,

Because the configuration (node) [ C ] is already visited in our DFS in the

A B

Initial iteration, we skip this configuration and move on to the next node. This happens twice, so the next node becomes [A B C]

[ B, B C C A A C C]

A C, A C B, A C, A B, A B, A B C, B C, B C, A B C, A B, A B

C.

Uniform Cost Search:- It is the number of steps required to reach the configuration from the initial state.

i. First Iteration

[ A ]

C C C C

A B, A B, A B, B, C A B

Cost: 1 0 0 1 1

The configuration (node) [ C ] is already visited in

A B

first iteration, we skip this configuration and move on to the next node.

ii. Second Iteration

[ A A ]

C C C C C C C

A B C, A B, A B, A B, B, A B, A B, B, C A B

Cost: 1 0 1 1 1 0 0 1 1

The configuration (node) [ C ] is already visited in

A B

first iteration, we skip this configuration and move on to the next node.

Let us remove all the visited configurations (node) from the list:

[A ]

C

B, A B C

Cost: 1 1

iii. Third iteration

[A ]

C C

B, B A, A B C

Cost: 1 1 1

Ques 3.

3. Let us consider these configurations to understand both the examples:

Current state:

[ C ]

A B

Final state:

[ C ]

A

B

Heuristic 1: Count all the blocks which are not in their correct position (as compared to the goal test). The block which is in the arm of the robot will not be counted.

For the above example:

A and C are not in their correct position, so two is added to the heuristics.

Heuristic 2: Calculate the distance between the current state of a block and the final/goal state of the block. Details of each block will be looked at.

For the above example:

In the goal configuration A is supposed to be above B and below C, but in the current configuration B and C both are not in their correct places with respect to A and so we add 2 to the heuristics .

Similarly for C there is no block at the top which is same in initial and final and the block below C is different in both the cases , so only 1 is added.

For B only the blocks above it don’t match, so 1 is added.

Ques 4 :-

To show the open-list for 3 iterations of A\* search by using one of the admissible heuristics, we are using Heuristic 1.

We have taken the initial and final states to be the one in ques 2.

Initial State:

[C ]

A B

Final State:

[A]

B

C

1. First iteration

[B ]

C C C C

A, A B, A B C, A B, A B

Cost: 3, 3, 2, 3, 3

We will expand the node with cost 2 as in case of A\* search , we expand the node which has the minimum cost and the cost is found on the basis of the heuristic 1 as explained in part3.

b. Second iteration

[ B ]

B, B C C A A C C C C

A C, A C B, A C, A B, A B, A B C, B C, B C, A B C, A, A B, A B, A B

Cost: 1, 2 , 2, 3, 3, 2, 2, 2, 2, 3, 3, 3, 3

Now, we will expand the node with cost 1 and remove all the visited nodes.

c.

1. Third iteration

[ A B ]

B B B B C C A A C

A C B, A C, C, A C, A C, A B, A B, B C, B C, A

Cost: 2 1 0 1 2 3 3 2 2 3