K.G.C.E.
Karjat - Raigad

Page No. :
Date :

Tutorial 2 :

KGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCEKGCE

Tutorial 2 : To understand state space problem formation.

Name : Rahul Patil

Class : BE - IT

Roll no : 47

Subject : IS Lab.

DOP	DOA	Remark	Sign.

Tutorial 2.

Aim: To understand state space based problem formulation for AI problems so that problem solving Agent can be applied.

Theory:

First we understand, the problem solving agent. Agent first formula goal and problem, then determines or rather searches an action sequence after which it returns the next action to be executed in a sequential manner.

Defining the problem is referred to as Problem Formulation. It involves defining five things:

1) Initial state: It is the starting state that the problem is in.

2) Action: It defines all possible actions available to the agent, given it is in some state & currently. It is a function action(x) that returns list of all possible actions.

3) Transition model: Also known as successor function which define which state the system tend to move to when a particular action is executed by the agent. Successive application of transition model gives rise to what is known as state space.

Problem Solving Agent Architecture.

```
function SIMPLE : PROBLEM-SOLVING-AGENT (percept)
    returns an action.

    static : seq, an action sequence, initially empty
             state, some description, initially empty.
             goal, a goal, initially null
             problem, a problem formulation.

    state ← UPDATE-STATE (state, percept)

    if seq is empty then do
        goal ← Formulate-GOAL (state)
        problem ← Formulate-problem (state, goal)
        seq ← search (problem)

    action ← First (seq)
    seq ← REST (seq)
    return action.
```

4) Goal Test : This act as a stopping conclusion when the state passed to this function is goal state it will return true and searching would stop.

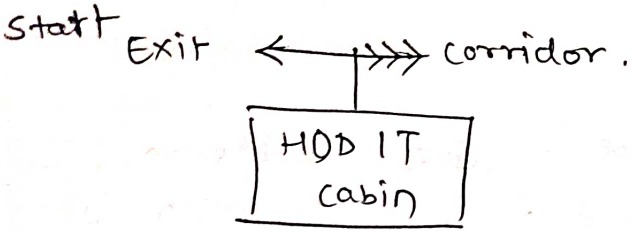
5) Path Test : It is accumulated cost of performing certain sequence of actions. This can help in determining weather the action sequence under consideration is optimal.

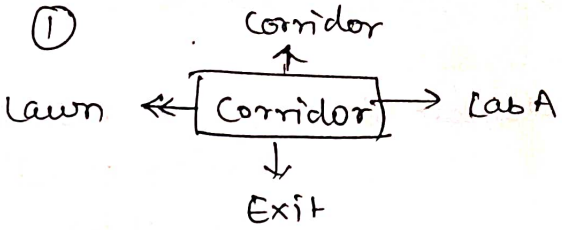
* Working : Based on understanding of problem formulation students need to formulate following problems. They will clearly show state space up to depth level 3 or till goal node which ever is shallowest

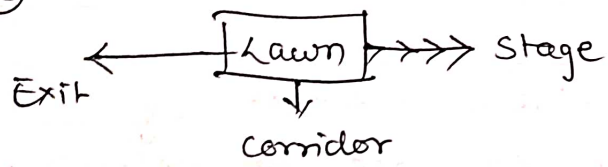
1) Navigate to KGCE Workshop from HOD IT Cabin with minimum number of moves, moves can be climbing or alighting staircase, turning left, right, walking through a corridor.

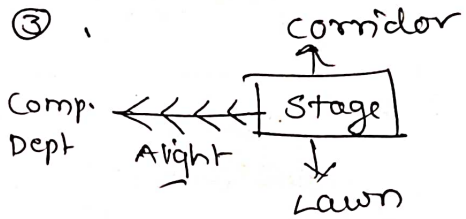
→ states : It can be represented as a top view of the agent along with arrows in directions left, right, forward and backwards. We use 'climb' and 'alight' for moving through staircases.

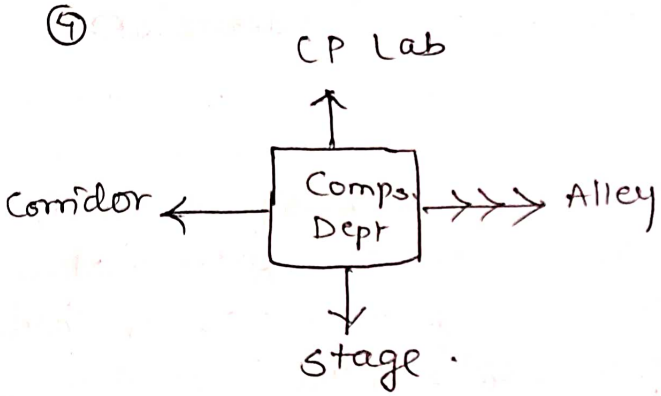
HOD IT Cabin → KGCE Workshop (Solution)

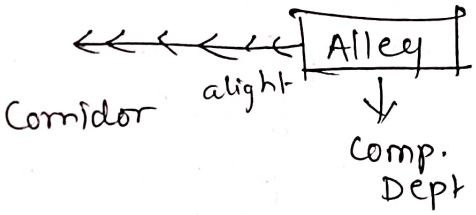


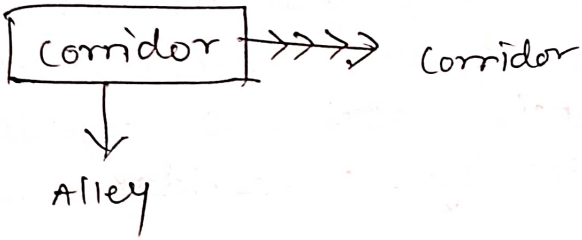
① 

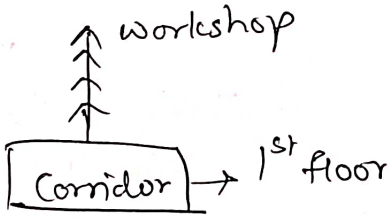
② 

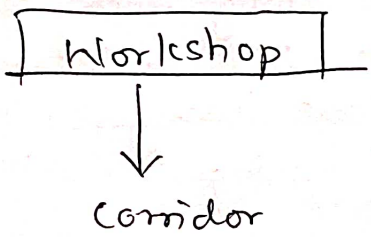
③ 

④ 

⑤ 

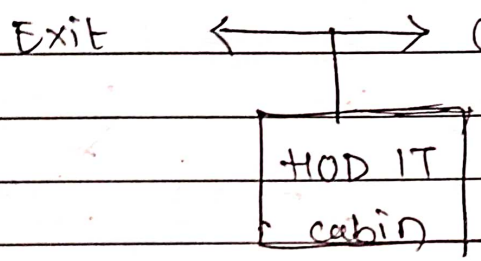
⑥ 

⑦ 

⑧. 

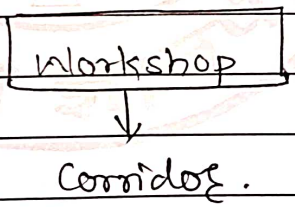
Goal state.

1. Initial state :

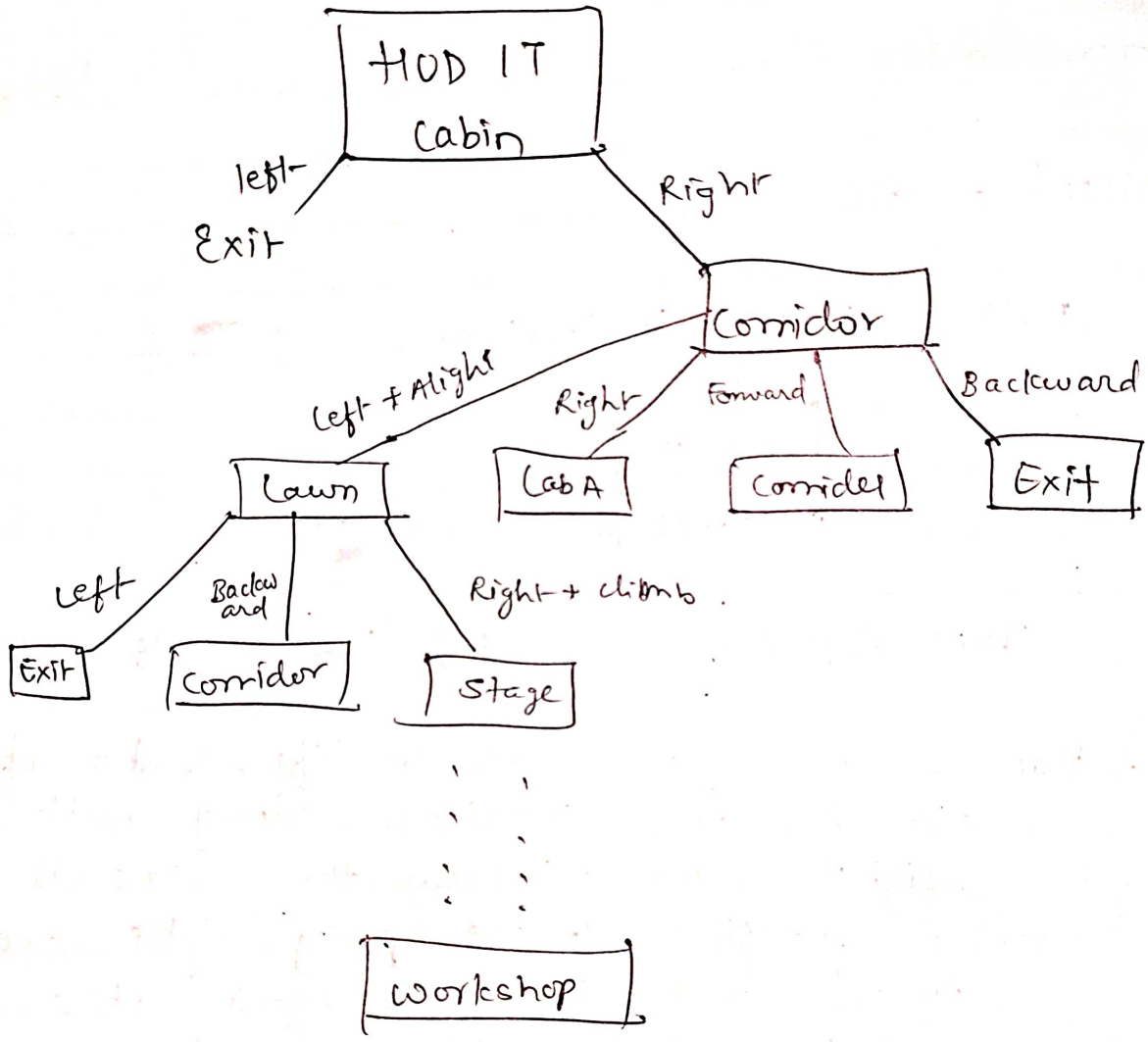


Box represents current location of agent.

2. Actions : The agent moves in left, right, forward and backward directions along with alighting and climbing the stairs (if any).
3. Successor function : If we apply 'right' operation to the start state, the agent enters the corridor the corridor - the first step towards goal state.
4. Goal Test :



5. Path cost : No. of actions to reach the workshop path cost = 8 directions + 4 staircases.
= 12.



goal state.

fig :: state space.

2) 8-puzzle problem:

→ The problem can be formulated as:

- States: states can be represented by a 3x3 matrix data structure with blanks denoted by an underscore:

1. Initial state: {{1,2,3}, {4,8,_}, {7,6,5}}
2. Actions: The blank space moves in left, right, up and down direction specifying the actions.
3. Successor function: If we apply 'down' operation, to the start state, the next state has '5' and '_' switched.
4. Goal test: {{1,2,3}, {4,5,6}, {7,8,_}}.
5. path test: No. of tests to reach to the final state.

Solution:

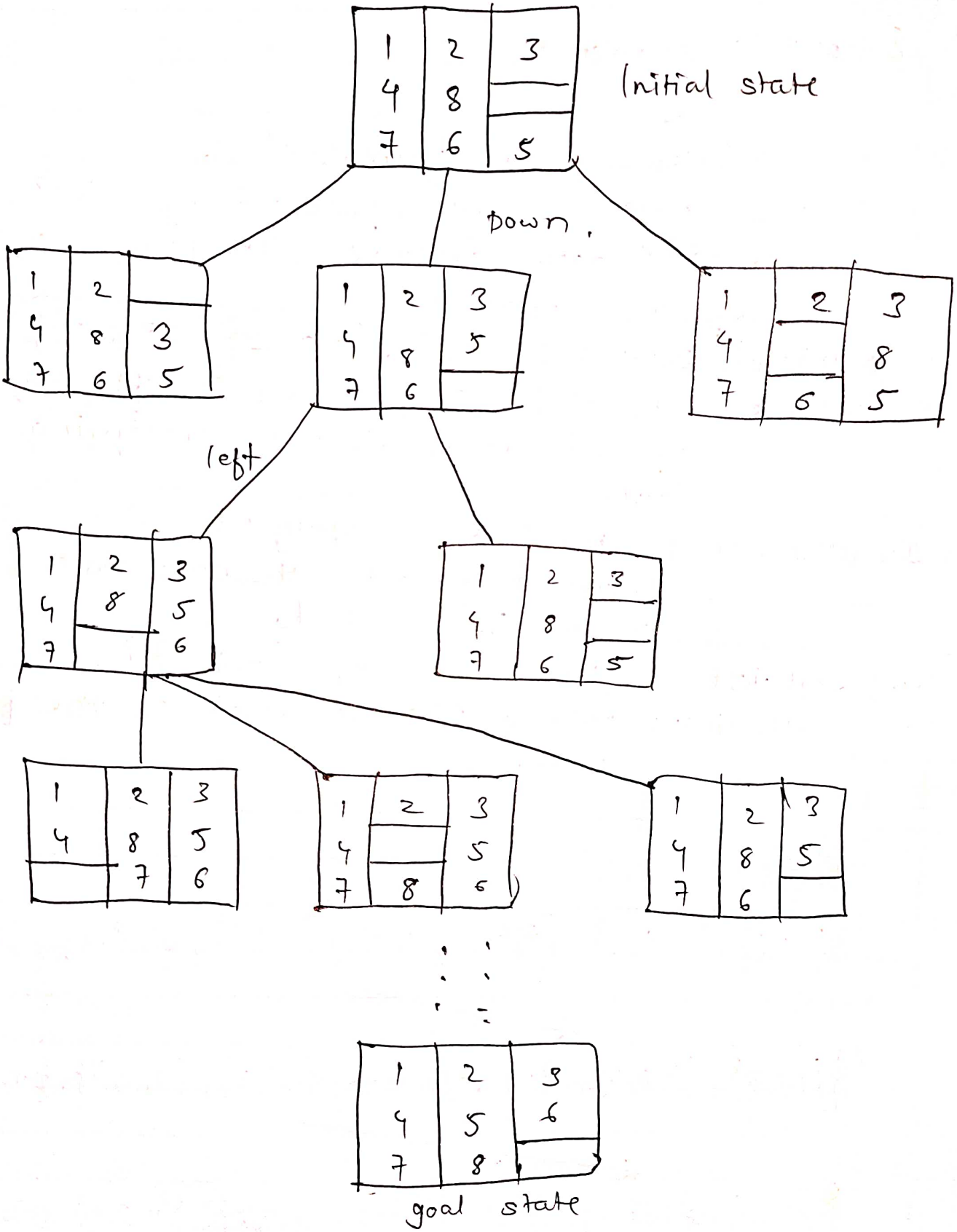
{{1,2,3}, {4,8,_}, {7,6,5}} → {{1,2,3}, {4,8,5}, {7,6,_}}

{{1,2,3}, {4,8,5}, {7,_,6}} → {{1,2,3}, {4,_,5}, {7,8,6}}

{{1,2,3}, {4,5,_}, {7,8,6}} → {{1,2,3}, {4,5,6}, {7,8,_}}

Path cost = 5 steps.

8 puzzle problem (State space)



3) The missionaries and cannibals problem.

States : State can be data structure having triplet (i,j,k) representing the number of missionaries, cannibals and canoes on the left bank of the river respectively.

1. Initial state : (3,3,1) as all missionaries, cannibals and canoes are on the left bank.
2. Actions : Take x number of missionaries & number of cannibals.
3. Successor function : If we take one missionary, one cannibal the other side of the river will have two missionaries and cannibals left.
4. Goal Test = (0,0,0).
5. Path Test : Number of crossings to attain goal state.

Solution :

(3,3,1) → (2,2,0) → (3,2,1) → (3,0,0) → (3,1,1) → (1,1,0) → (2,2,1) → (0,2,0) → (0,3,1) → (0,1,0) → (0,2,1) → (0,0,0).

Cost = 11 crossing.