Tutorial No:- 02

Name :- Ankita Shridhar Digaskar

Class :- BE - IT

Roll No :- 51 / I3

Sub :- IS LA1

RIGHT WAY
Date / /
Page No. /

Tutorial No:- 2

Tutorial 2 :- To understand state space Problems formulation.

Aim :- To understand state space based Problem formulation of AI Problems so that Problem Solving Agent can be applied.

Theory :- First we understand the Problem Solving agent. Algorithm shown in figure 3 shows agent program for Problem solving agent. Agent first formulates goal and Problem, the determines or rather searches an action.

```
Function SIMPLE-PROBLEM-SOLVING-AGENT (percept)
                                    return an action
  static : seq, an action sequence, initially empty
           state, some description of the current
           world state goal, a goal,
           initially null problem,
           a Problem formulation
  state ← UPDATES-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
```

Fig. Problem Solving Agent Architecture

Defining the problem is referred to as Problem formulation. It involves defining following five things.

Initial State : It is the starting state that the problem is in.

Actions : It defines all possible action available to the agent given it is in some state currently.

Transition model : also known as successor function which define state tend to move to when a particular action is executed by the agent. Successive application of transition model.

Goal test :- This act as a stopping condition when the state passed to this function.

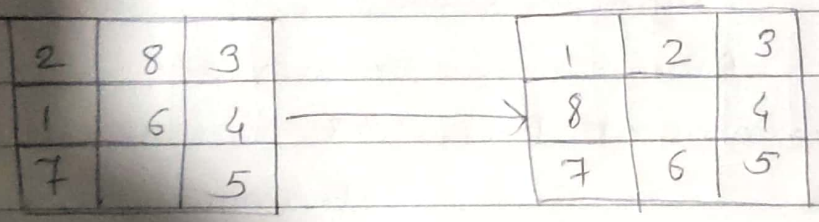
Path cost :- It is accumulated cost of performing certain sequence of action. transison model, goal test and Path cost.

In term of problem solving agent solution is the path from initial state.

Tutorial 2

8 Puzzle Problem:

The 8 Puzzle consists of eight numbered movable tiles set in a 3×3 frame. one cell of the frame is always empty thus making it possible to move an adjacent numbered tile into the empty cell. such a Puzzle is illustrated in following figure.



Initial state Goal state

fig. Example of 8 Puzzle

This Program is to change the initial configuration into the goal config. sequence of moves, such as "move file 5 to the right move file 7 to the left. move the tile 6 to the down" etc.

To solve a Problem, we must specify the global database the rules and the

control strategy for the 8 Puzzle problem that corresponds to 3 components.

These elements are the problem states, moves and goal. In this problem each tile configuration is a state. The set of all possible configuration in the problem space. consists of 9.62.880 different configurations of the 8 tiles and blank space.

For the 8-Puzzle a straight forward description is a 3×3 array of matrix of numbers. Initial global database is this description of the initial problem state. Virtually any kind of data structure can be used to describe states.

A move transforms one problem state into another state. the 8-puzzle is conveniently interpreted as having the following four moves:-
. move empty space (blank) to the left,
move blank up. move blank to the right
and move blank down.
. These move blank down modeled by production rules that operate on the state descriptions in the appropriate manner.

The goal condition from the basis for the termination. The control strategy repeatedly applies rules to state descriptions until a description of a goal state is produced. It also keeps tracks of rules that have been applied so that it can compose them into sequence.

representing the problem solution:

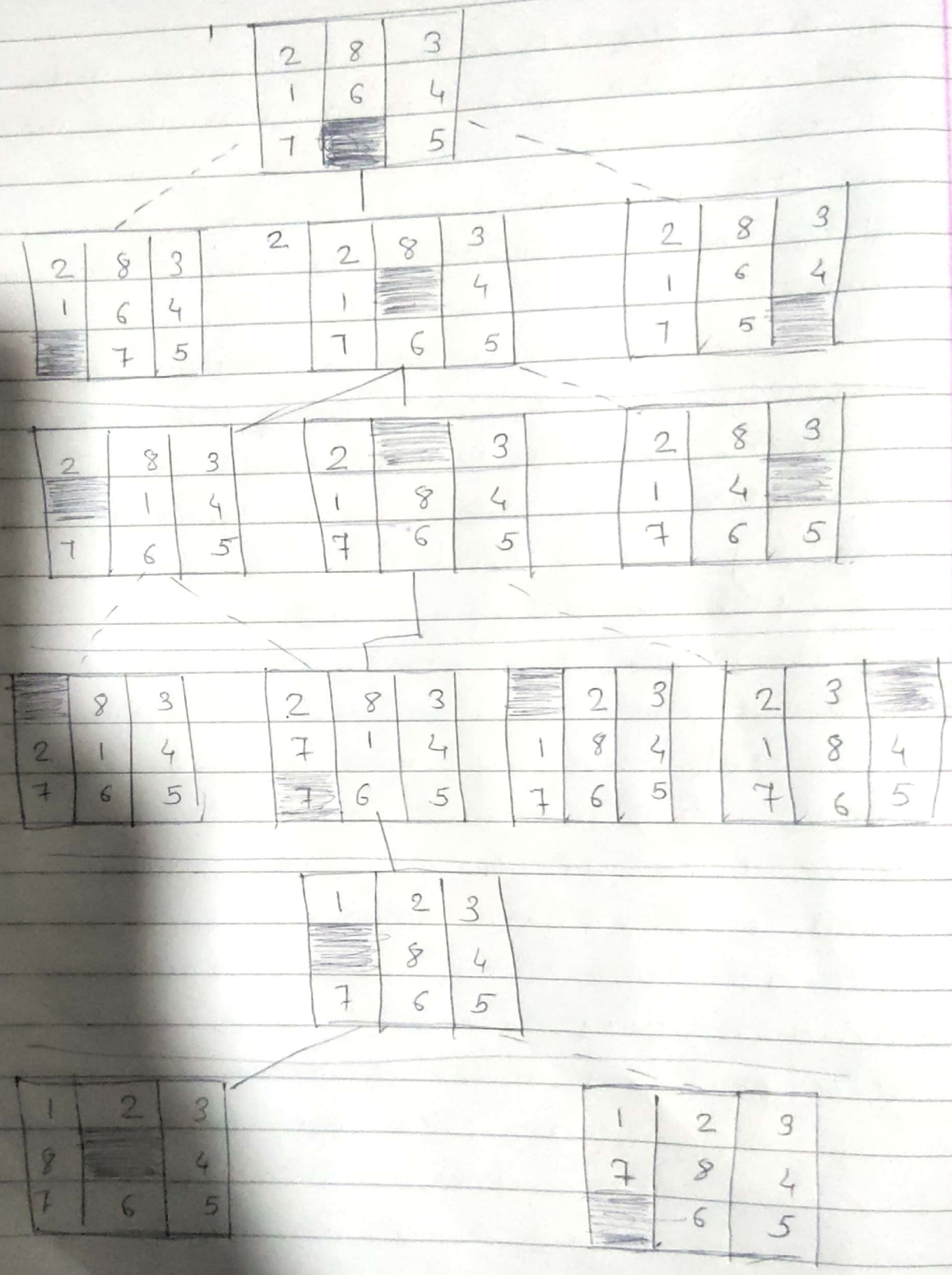


Fig Solution of 8 Puzzle problem

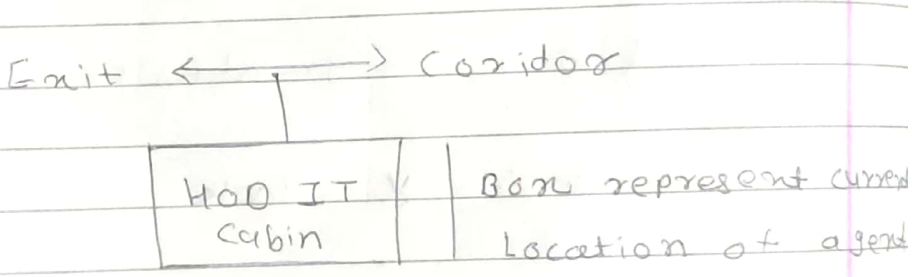
RIGHT WAY
Date / /
Page No. /

Tutorial 2

Navigate to KGKCE workshop from HOD IT Cabin with minimum number of moves. Moves can be Climbing or alighting Shareca turning left, right walking through a caridor.

States :- It can be represented as a top view of the agent along with arrows in directions left, right, farword and backwards. we use 'climb' and alight for moving through shareca se.

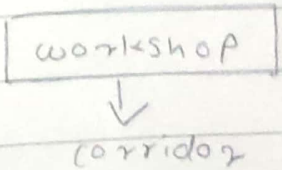
1) Initial State :-



2) Actions :- The agent moves in left, right, forward and backward direction along with alighting and climbing the stars (if any)

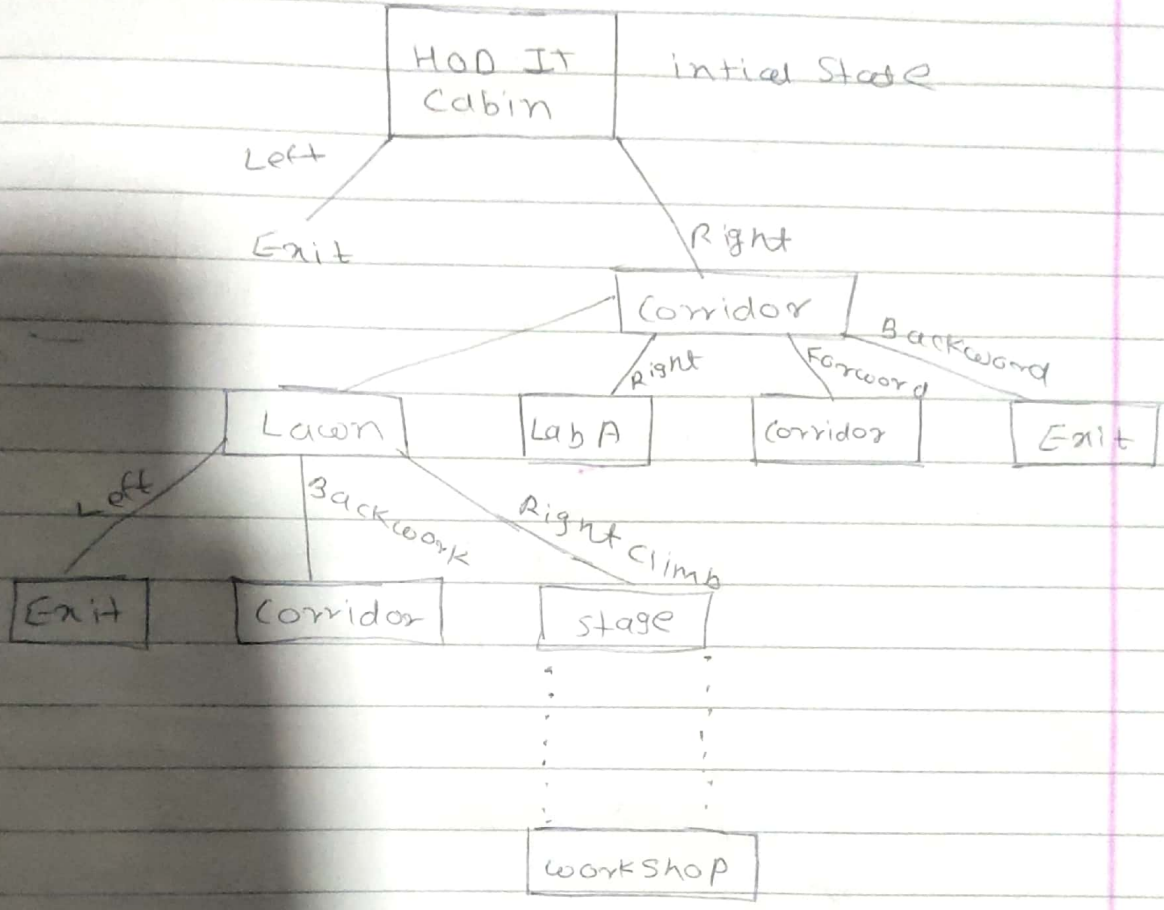
3. Successor function :- If we apply right operation to the start state, agent, enter the coridor - the first step toward goal state.

4. Goal test



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

State space :



HOD IT Cabin → KGCE workshop (solution)