Name : Subhendu Maji
Roll : 18ETCS002121

Subject : Artificial Intelligence
Course code : 19CSC312A

CSE - C Section.

Date : 7 June - 2021

(1)

TT-2

3.

a) Knowledge Base in Logic agents

An intelligent agent needs knowledge about the real world for taking

Knowledge base is a central component of a knowledge based agent, it is also known as KB. It is a collection of sentences.

These sentences are expressed in a language which is called a knowledge representation language. The knowledge base of KBA stores fact about the world.

Knowledge base is required for updating knowledge for an agent to learn with experiences & take action as per the knowledge.

Capabilities of KB agents

- a agent should be able to represent states, actions etc.
- a agent should be able to incorporate new percepts.

Name : Subhendu Maji Reg. No. : 18ETCS002121

(2)

(3).

- an agent can update the internal representation of the world.
- an agent can deduce the internal representation of the world.
- an agent can deduce appropriate actions.

b) Logical Entailment

Logics are formal languages for representing information such that conclusion can be ~~drawn~~ drawn.

Entailment means that one thing follows from another : $KB \models \alpha$

Knowledge base KB entails sentence α if and only if α is true in all worlds where KB is true.

eg. the KB containing "the Giants won" & "the Reds won" ~~not~~ entails "Either the Giants won or the Reds won".

eg. $x + y = 4$ entails $4 = x + y$

entailment is a relationship b/w sentences (i.e. syntax) that is based on semantics.

Name: Subhendu Maji Reg. no. : 18ETCS002121

4. a)

(i) $P \Rightarrow Q$ (is P implies Q.)

A sentence such as $P \Rightarrow Q$, is called an implication.

Implication are also known as if-then ~~rule~~ rules.

eg.

If it is raining, then the street is wet.

here, let P = It is raining

Q = Street is wet

so, it can represented as $P \Rightarrow Q$.

(ii) $P \Leftrightarrow Q$ (Biconditional)

A sentence such as $P \Leftrightarrow Q$ is a biconditional sentence.

(also known as if and only if rules).

eg,

If I am breathing, then I am alive.

here, let P = I am breathing

Q = I am alive

It can represented as $P \Leftrightarrow Q$.

Name : Subhendu Maji Reg. : 18ETCS002121

(4)

4.b)

The knowledge based agent takes percept as input & returns an action as output. The agent maintains the knowledge base, KB, & it initially has some background knowledge of the real world. It also has a counter to indicate the time for the whole process, & this counter is initialized with zero.

each time the function is called, it performs -

- firstly, it Tells the KB what it perceives
- secondly, it asks KB what action it should take
- third agent program tells that which action was chosen.

function KB-agent (percept):

persistant : KB - a knowledge base

t, a counter, initially 0, indicating time

Tell (kb, make-percept-sentence (percept, t))

action = ask (kb, make-action-query (t))

Tell (kb, make-action-sentence (action, t))

t = t+1

return action.

Name : Subhendu Maji

Reg. No : 18ETCS002121

(5)

1.

a) Zero-Sum of Perfect Information.

A zero-sum game is one in which no wealth is created or destroyed. So, in a two player zero-sum game, whatever one player wins, the other loses. Therefore, the players share no common interests.

In a game with perfect information, every player knows the results of all previous moves. Such game eg. are tic-tac-toe & Nim.

In game of perfect information, there is at least one 'best' way to play for each player. The best ~~strategy~~ strategy does not necessarily allow him to win but will minimize this loses, but there is no strategy that will allow you to always win.

b) Mini-max is a backtracking algorithm that is used in decision making & game theory to find the optimal move for a player, assuming that your opponent also plays optimally.

It is used in two-player turn base games

eg. Tic-tac-toe, chess etc.

Time complexity is $O(b^m)$

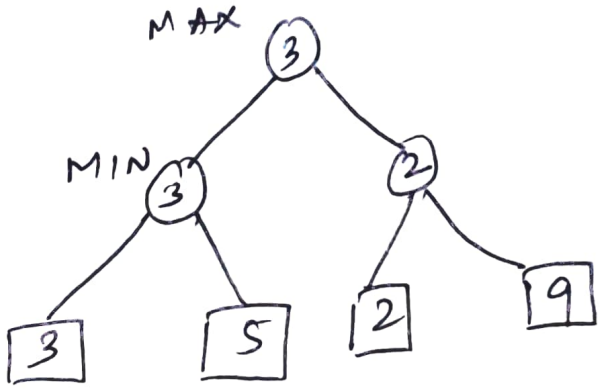
m = depth of tree

b = legal moves at each point.

Name : Subhendu Maji Reg.no : 18ETCS002121 (6)

In minimax, one player is called Maximizer, & other is minimizer.
The maximizer tries to get the highest possible while the minimizer tries to do the opposite & get the lowest score possible.

E.g.
consider a game, having 4 final states & paths to reach final state are from root to 4 leaves of a perfect binary tree.

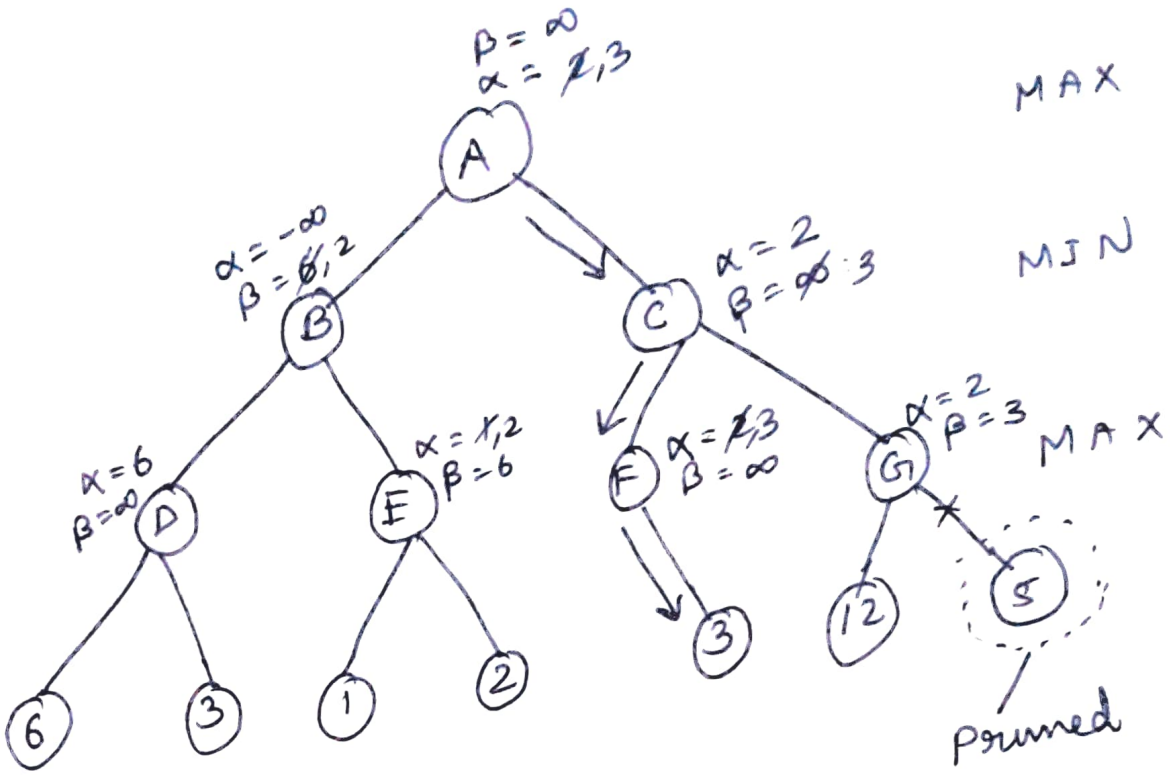


assuming you are the maximizing player getting first chance to move (i.e. you are root).
Since this is backtracking based algo., it tries all possible move, the backtracks & makes a decision.

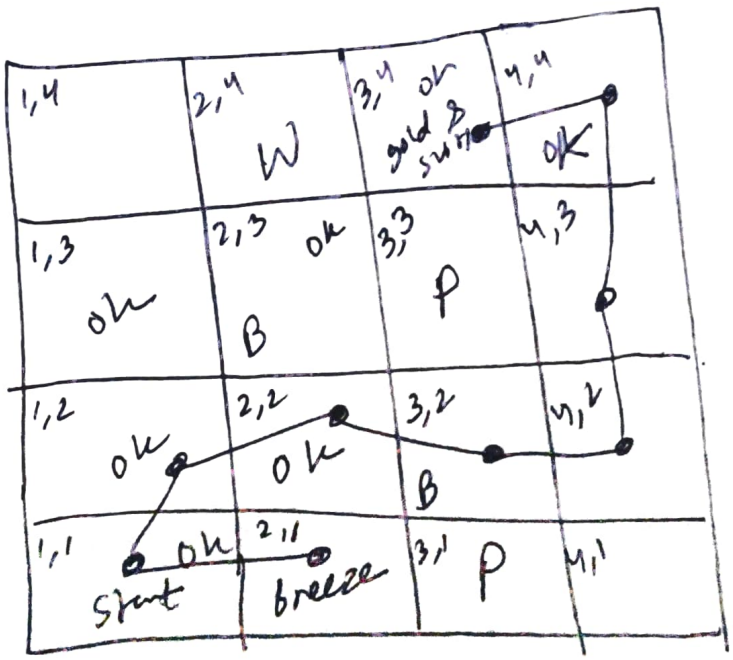
Name : Subhendu Maji Reg. No. : 18ETCS002121

(7)

2



5



(1,1) is ok so that adjacent rooms are also ok.

agent 1st step :

It moves to the (2,1). Agent feels breeze so he will come back to 1,1

agent 2nd step:

agent moves to the room (1,2) & there is no breeze or stench, so therefore adjacent rooms are Ok (1,3) & (2,2) are Ok.

agent 3rd step:

agent moves to room (2,2)

there is no breeze or stench. Therefore, adjacent rooms are (2,3) → (3,0). Ok

agent step 4

agent moves to room (3,2) & then he moves to the next room (4,2).

there is no breeze in room (4,2) therefore room (4,3) is also Ok.

agent step 5

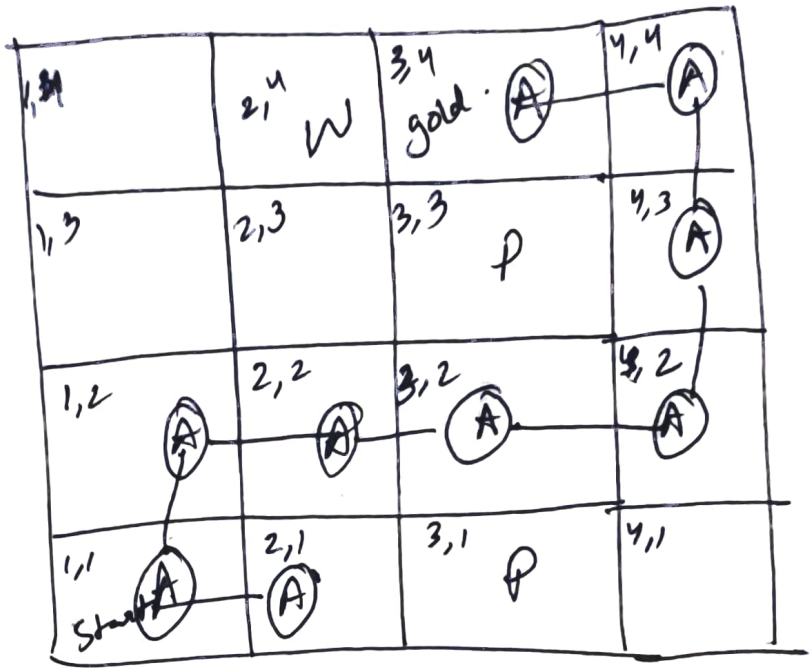
agent moves to the room 4,3) ; he feels breeze. on the room (4,3). therefore, room (3,3) is not Ok.

agent step 6

Agent moves to the room (4,4). there is no breeze or stench, therefore adjacent rooms (3,4) & (4,3) are Ok.

agent step 7

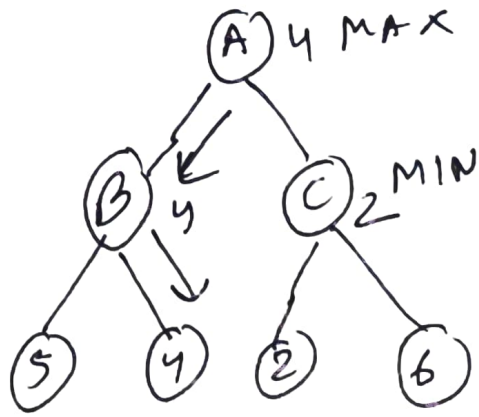
Agent moves to the room (3,4), where there is glitter (3,4). he digs up the gold & exits from the cave.



1. b) Cont.d.

the minimax. algo. computes the minimax. decision from the current state.

It uses a simple recursive computation of the minimax. values of each successor state.



In this game max moves first, which move is most optimal for max., we can ~~backtrack~~ backtrack & find

A → B → 4