Ajay Kumar Garg Engineering College, Ghaziabad

Department of CSE

Model Solution- ODD Semester (2017-18)

Sessional Test -2

Subject Code	:	NCS-702
Subject Name	:	Artificial Intelligence
Names of Faculty Teaching with Signature	:	Dr. Sunita Yadav Mr. Ajay Kumar Ms. Anupama Sharma

Mamta Bhusry

Name and Signature of Hod: Prof. Mamta Bhusry

Divya Gupta

A. Attempt all the parts. (5x2=10)

(1) Define heuristic function. Also write heuristic function for Best first search.

Ans A heuristic function h(n) provides an estimate of the cost of the path from a given node to the closest goal state. It must be zero if the node represents a goal state.

Best first Search uses the hueristic function -

Cost/distance from current node to goad node.

(2) Name the difficulties of hill climbing search.

Ans: The difficulties which arise in case of hill climbing search are:

Local Maximum: A state better than all its neighbors but is not better than some other states farther away.

Plateau: A flat area of the search space in which neighboring states have the same value. On a plateau its not possible to get best direction to move.

Ridge: An area ~~in~~ which is higher than the surrounding areas but can not be searched in a simple move.

(3) List various issues in knowledge representation

Ans: The issues that arise while KR are:

1) Important attributes : Any attribute of objects so basic that they occur in almost every problem domain.

2) Relationship among attributes : Any important relationship that exists among object attributes.?

3) Choosing Granularity : At what level of detail should the knowledge be presented.

4) Set of Objects : How set of objected be represented?

5) Finding Right Structure : Given a large amount of knowledge stored, how can relevant part be accessed.

(4) What are the Limitations in using propositional Logic to represent the knowledge base?

Ans: Limitations of propositional Logic are:

① It is unable to express the properties that the subject of a statement can have.

② It is unable to express the quantity of objects in the statement.

(5) Prove that Breadth first search is a special case of Best first search.

Ans: Breadth first search is a special case of Best first search with evaluation function

$$f(n) = depth(n).$$

B. Attempt all the parts. (5x5=25)

(6) Explain the effect of over-estimation and under-estimation on A* algorithm.

Ans: A* search finds optimal solution to problems as long as the heuristic is admissible which means it never overestimates the cost of the path from any given node.

A* maintains a priority queue of options that it is considering, ordered by how good they might be. It keep searching until it finds a route to the goal that is so good that none of the other options could possibly make it better. How good an alternative might be is based on the heuristic and on actual costs found in the search so far.

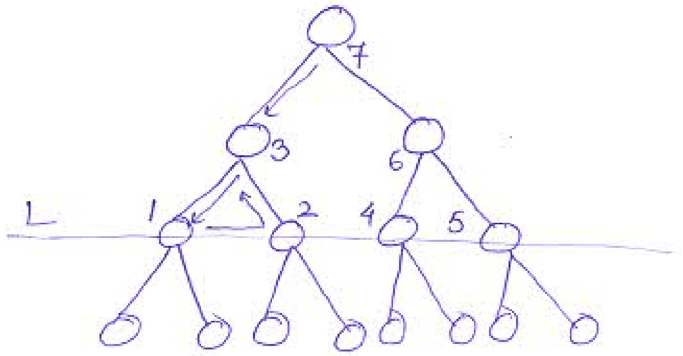
If the heuristic underestimates, the other options will look better than they really are. The algorithm checks them out considering those might improve the route. If the heuristic only underestimates by a little bit, may be some of those routes will turn out to be useful.

If the heuristic overestimates, A* may consider that the alternative to the route are comparatively not good, so won't look at them. But because the heuristic overestimates, they might be much better than they seem.

(7) Write a short note on depth limited search.

Ans: To avoid the infinite depth problem of DFS, we can decide to only search until depth L, i.e. we don't expand beyond depth L. This is Depth-Limited Search.

Depth Limited Search is most useful if the maximum depth of the solution is known. But if wrong branch expanded it may not terminate.



Depth Limited search will always terminate.
It will find solution if there is one in the depth bound.
Too small depth bound misses solutions and too large depth bound may find poor solutions when there are better ones.

Time complexity : $O(b^L)$ where L is maximum depth selected.

Space complexity : $O(bL)$

(8) Convert the following sentences in FOPL:

(i) John Likes all kind of food.

Ans: $\forall x \; food(x) \rightarrow Likes(John, x)$.

(ii) Apples are food.

Ans $food(Apple)$

(iii) Coconut is a biscuit.

Ans $Biscuit(Coconut)$

(iv) Mary is a child who takes Coconut.

Ans: $Child(Mary) \land takes(Mary, Coconut)$

(v) John loves children who take biscuits.

Ans $\forall x ((Child(x) \land \exists y (takes(x,y) \land Biscuit(y))) \rightarrow Loves(john, x)$

(9) Consider the following axioms:

(i) P
(ii) $(P \land Q) \rightarrow R$
(iii) $(S \lor T) \rightarrow Q$
(iv) T

Prove that R is true by resolution

Ans: Converting them in CNF form:

(i) P
(ii) $\neg(P \land Q) \lor R \equiv \neg P \lor \neg Q \lor R$

(iii) $\neg(S \vee T) \vee Q \equiv (\neg S \wedge \neg T) \vee Q \equiv (\neg S \vee Q) \wedge (\neg T \vee Q)$

Can be splitted in two clauses as:
(AND Elimination)

(iii) a $\quad \neg S \vee Q$

(iii) b $\quad \neg T \vee Q$

(iv) T

For proving by resolution start with $\neg R$

$\neg R \qquad \neg P \vee \neg Q \vee R$ (ii)

$\neg P \vee \neg Q \qquad \neg T \vee Q$ (iii) b

$\neg P \vee \neg T \qquad T$ (iv)

$\neg P \qquad P$ (i)

$\square$

Hence by resolution it is proved that R is true.

(10) Explain Hidden Marcov model in detail.

Ans: Let set of states be $\{S_1, S_2, -- S_N\}$.

The process moves from one state to another generating a sequence of states: $S_{i1}, S_{i2}, -- S_{ik}$--

According to Marcov chain property, probability of each subsequent state depends only on what was the previous state:

$$P(S_{ik} | S_{i1}, S_{i2}, -- S_{ik-1}) = P(S_{ik} | S_{ik-1})$$

In Hidden Marcov model, states are not visible, but each state randomly generates one of M observations (or visible states $(V_1, V_2, -- V_M)$.

To define Hidden Marcov Model, the following probabilities have to be specified:

matrix of transition probabilities $A = (a_{ij})$,

where $a_{ij} = P(S_i | S_j)$

matrix of observation probabilities $B = (b_i(V_m))$

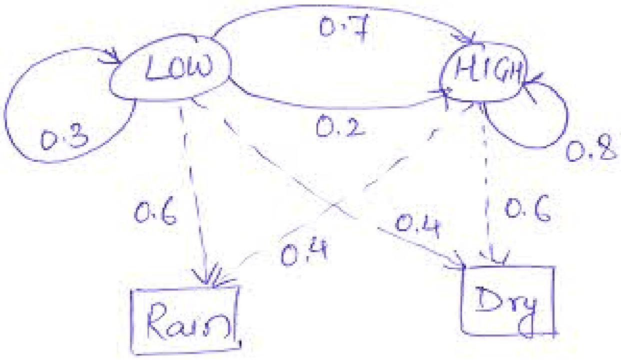
where $b_i(V_m) = P(V_m | S_i)$

and a vector of initial probabilities $\pi = (\pi_i)$,

where $\pi_i = P(S_i)$

Model is represented by $M(A, B, \pi)$.

Example:



States : 'Low', 'High' atmospheric pressure
Observations : Rain and Dry

Suppose we want to calculate a probability of a sequence of observations in {Dry, Rain}.
Consider all possible hidden state sequences:

$P(\{Dry, Rain\}) = P(\{Dry, Rain\}, \{Low, Low\}) + P(\{Dry, Rain\}, \{Low, High\}) + P(\{Dry, Rain\}, \{High, Low\}) + P(\{Dry, Rain\}, \{High, High\})$

First term is :

$P(\{Dry, Rain\}, \{Low, Low\}) =$

$P(\{Dry, Rain\} \mid \{Low, Low\})\, P(\{Low, Low\}) =$

$P(Dry \mid Low)\, P(Rain \mid Low)\, P(Low)\, P(Low \mid Low)$

$= 0.4 \times 0.4 \times 0.6 \times 0.4 \times 0.3$

Likewise other terms can be calculated.

(11) Discuss Min-Max search procedure with Alpha-Beta cutoff in detail.

Ans: Min-Max search is used in case of game-playing. The two-players are called MIN and MAX. MAX is playing a strategy for maximizing its utility and MIN is trying to minimize MAX's utility. MIN has impact on the moves of MAX.

An optimal strategy is determined by examining 'minmax' value of each node. Minmax value is computed by the following function

MINMAX-VALUE(n) = Utility(n) if n is a terminal state

$max_{s \in Successors(n)}$ MINMAX-VALUE(s) if n is a MAX node

$min_{s \in Successors(n)}$ MINMAX-VALUE(s) if n is a MIN node

MAX moves to states with highest minimal values and MIN moves to states with lowest maximal values.

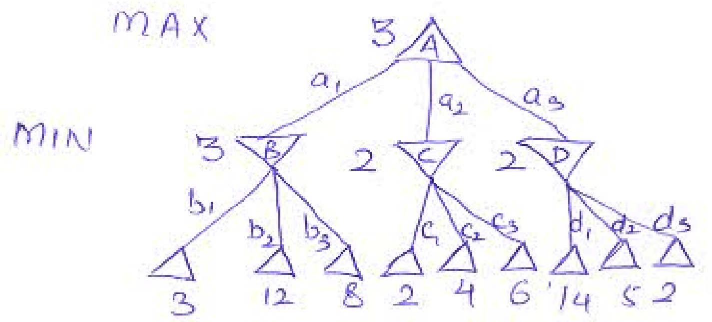


Figure 1

Alpha-Beta prunning:

Basic idea of α-β prunning is to eleminate nodes which will never reached in the actual play.

In figure 1,

$$\text{MINMAX-VALUE(root)} = \max(\min(3,12,8), \min(2,x,y), \min(14,5,2))$$

$$= \max(3, \min(2,x,y), 2)$$

$$= \max(3, z, 2) \text{ where } z \leq 2$$

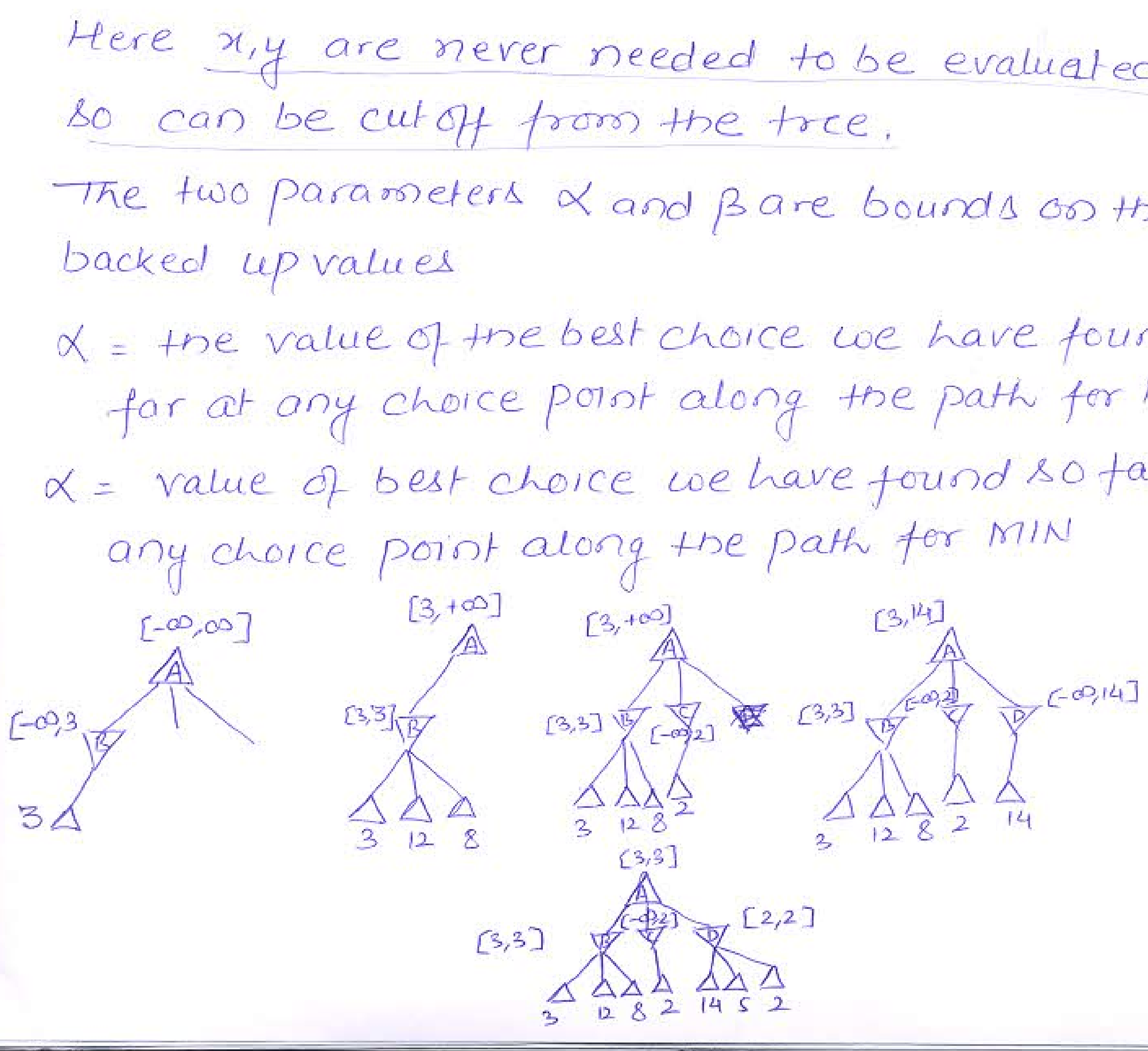
$$= 3$$

Here x,y are never needed to be evaluated, so can be cut off from the tree.

The two parameters α and β are bounds on the backed up values

α = the value of the best choice we have found so far at any choice point along the path for MAX

α = value of best choice we have found so far at any choice point along the path for MIN



(12) Explain the difference between working of forward chaining and backward chaining with the help of an example.

Ans: Forward Chaining starts with available facts and attempts to draw conclusions about the goal. Backward chaining starts from goal and prove some fact to be true.

Forward chaining:

① It is a data driven method of deriving a particular goal from a given knowledge base and set of inference rules.

② Inference rules are applied by matching facts to antecedents of consequence relations in the KB.

③ The application of inference rules results in new knowledge, which is then added to KB.

Backward chaining:

① It is a goad driven method of reasoning.

② Here we start with the goal which is the hypothesis we wish to prove and we aim to show how conclusions can be reached from the rules and facts in the database.

③ This type of method is used when the goal is clearly mentioned.

* Backward chaining is more efficient than forward chaining in ~~case~~ terms of set of rules fire to reach the conclusion.
* Backward chaining is more focused and streamlined.
* In case of large KB backward chaining is more prefered.

<u>Examples:</u>

Forward chaining:

KB is A, $A \rightarrow B$

Using Modus ponen $\dfrac{\begin{array}{c} A \\ A \rightarrow B \end{array}}{B}$

Backward Chaining: Resolution

If we want to prove B

Start with $\neg B$, Given A, $A \rightarrow B \equiv \neg A \vee B$

$$
\begin{array}{ccc}
\neg B & & \neg A \vee B \\
& \neg A & & A \\
& & \square &
\end{array}
$$

By resolution B is proved.