4/8/22

M T W T F S S
Page No.:
Date:
YOUVA

— ARTIFICIAL INTELLIGENCE —

ARTIFICAL → man made
INTELLIGENCE → ability to think / learn

- We are putting human intelligence into machines with sets of rules.
- AI is a method of making a computer / machine intelligent.

→ ANN → Artificial Neural Network.

Input → [AI / ML System] → Outputs

- Data acquitation
- Pre-processing
- Feature extraction
- Feature selection
- Classification / Regression
- Training.
- Testing.

2/8/22

ARTIFICIAL INTELLIGENCE DETAILS —

- Artificial intelligence are man made thinking power.
- AI is defined as a branch of CS by which we can create intelligent machines which can behave like a human, think like human and able to make decisions

→ Main Goals :- (AI):

- To replecate human intelligence
- Solve knowledge intensive task
- Intelligent connection of perception and action
- Building machine which can create or perform task which require human intelligence
  1. Proving theorems
  2. Playing some game
  3. Planning surgical operation
  4. driving car
  5. Prediction & analysis
  6. Recommendations (creating).

- Creating system which exhibit intellegent behaviour, learn new things by itself, demonstrate, explain and can advise its user.

- AI is not just of computer science, it is so vast and requires lots of other factors for creating AI.

- Intelligence is combination of learning, reasoning, logic building, problem solving, language understanding and decision making.

→ Advantages of AI :-

1. High accuracy with less errors.
2. High speed (fast process)
3. High reliability (trust)
4. Useful for risky areas (remote areas)
5. digital assistance
6. useful for public units.

→ Disadvantages -
- High cost
- Increases dependency on machine
- Can't think out of ~~mind~~ box.
- No feeling & emotion
- Decreases job (employment)
- Work 24/7 without rest.
- It Easily handles representative task.
- ~~wor~~

3/8/22

→ LAB :-

- COMPUTERS VS NEURAL NETWORKS

"Standard" computers	Neural Networks
• One CPU	• Highly parallel processing
• Fast processing units	• slow processing unit
• reliable unit	• unreliable units
• Static infrastructure	• dynamic infrastructure

→ WHY ARTIFICAL NEURAL NETWORK?
- There are two basic reasons why we are interested in building artificial neural network (ANNs)

- Technical viewpoint -
Some problems such as character recognition or the prediction of future states of a system requires massively parallel and adaptive processing

- **Biological viewpoint -**
ANN's can be used to replicate and simulate components of the human (or animal) brain, thereby giving us insight into natural information processing.

→ ANN -

- The "building blocks" of neural networks are the neurons.
- In technical systems, we also refer to them as units or nodes.

- Basically, each neuron
  → recieves input from many other neurons.
  → changes its internal state (activation) based on the current input.
  → sends one ~~output signal~~ signals to output signal to many other neurons, possibly including its input neurons (recurrent network)

- Information is transmitted as a series of electric impulses, so called spikes.
- The frequency and phase of these spikes encodes the information.
- In biological systems, one neuron can be connected to as many as 10,000 other neuron.

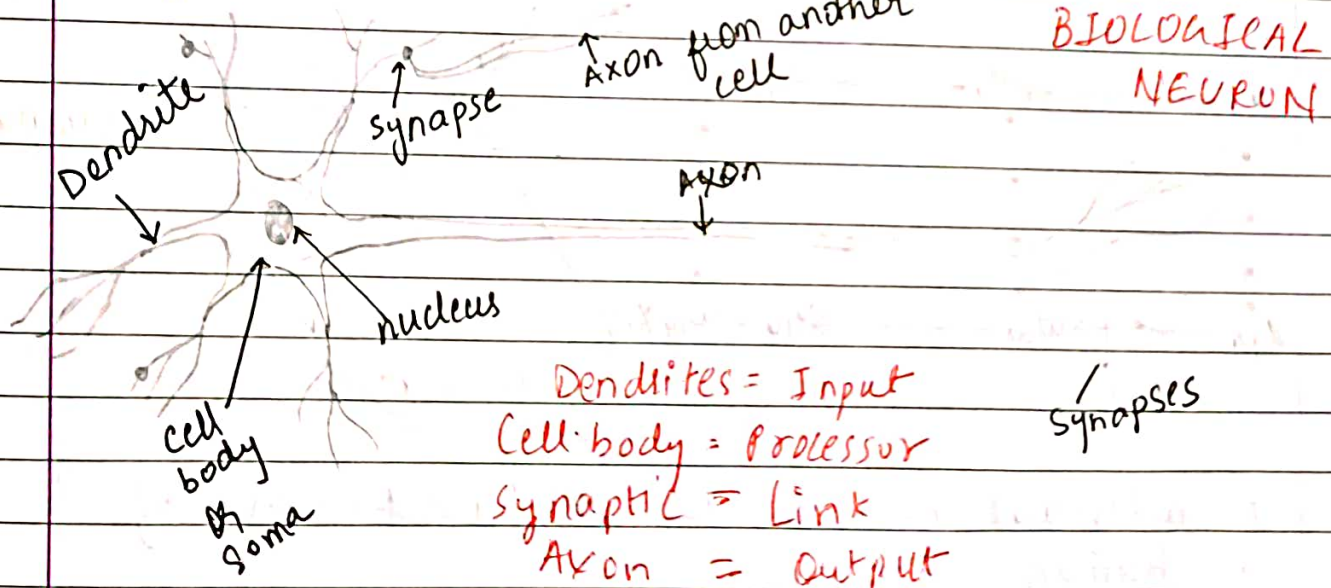
- Usually, a neuron recieves its information from other neurons in a confined area, its so called receptive field.

→ How do ANN work -

- An artificial neural network is either a hardware implementation or a computer program which strives to simulate the information processing capabilities of its biological exemplar. ANN are typically composed of a great number of interconnected artificial neurons.
- The artificial ~~not~~ neurons are simplified models of their biological counterparts.
- ANN is a ~~teorica~~ technique for solving problems constructing software that works like our brains.

→ How do our brain work?

- The Brain is a massively parallel information processing system.
- Our brain are a huge network of processing elements. A typical brain contain a network of 10 billion neurons.



- Once input exceeds a critical level, the neuron discharges a spike an electrical pulse that travels from the body, down the axon, to the next neuron(s).

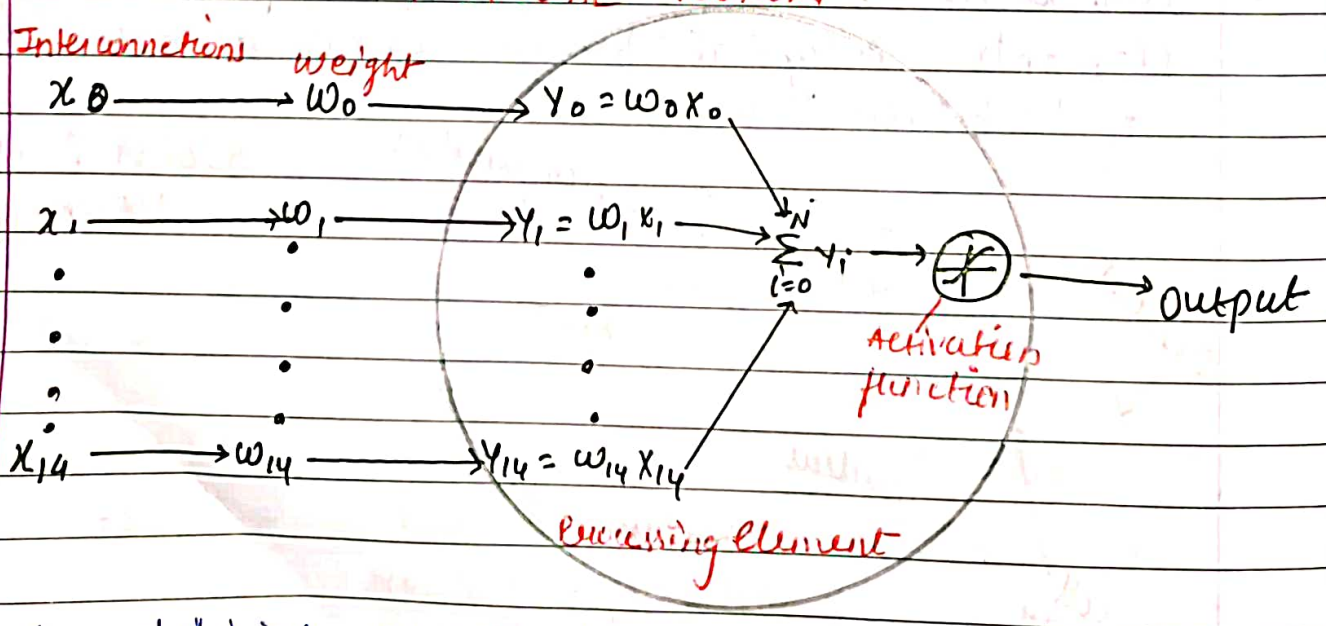
- The axon ending almost touch the dendrites or cell body of the next neuron.

- Transmission of an electrical signal from one neuron to the next is effected by neurotransmitter

- Neurotransmitters are chemicals which are released from the first neuron and which bind to the second.

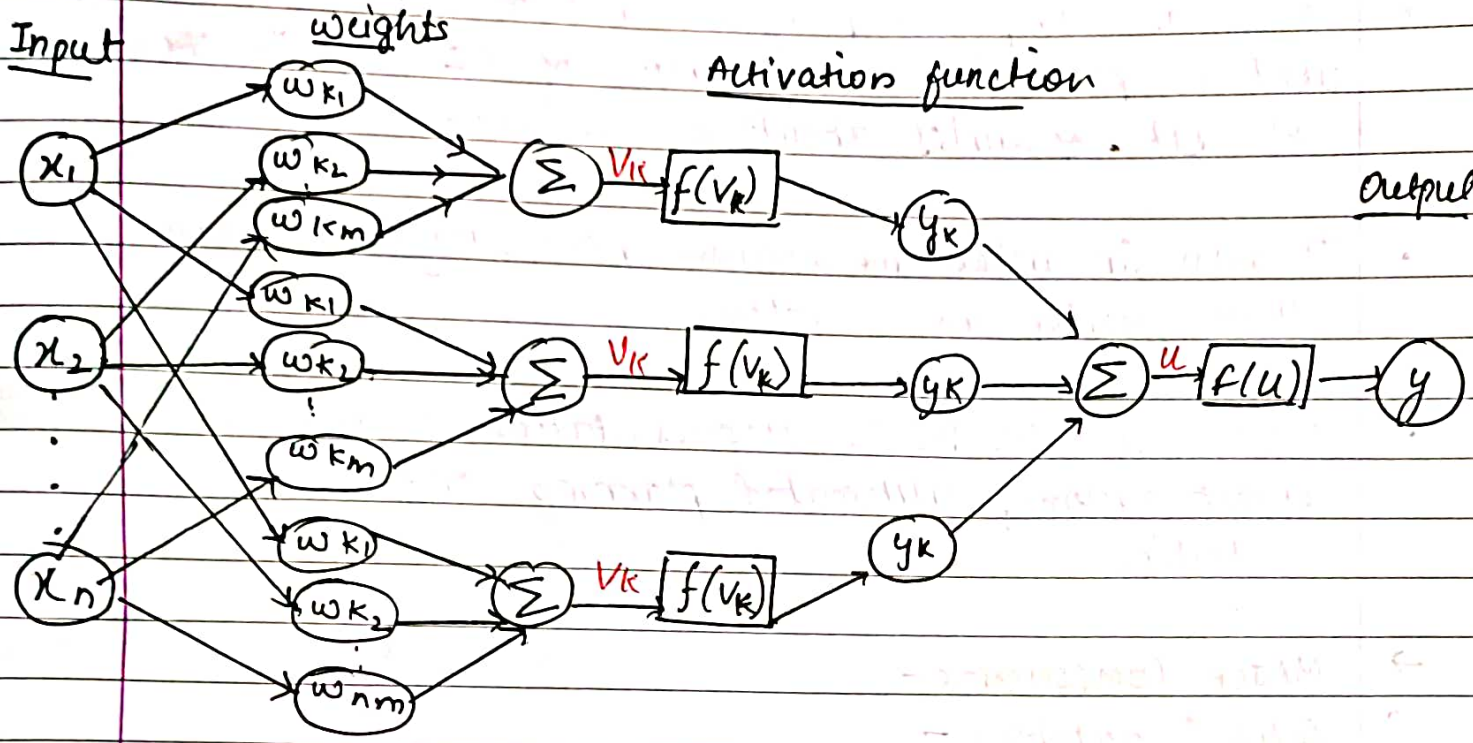
- The link is called synapses. The strength of the signal that reaches the next neuron depends depends on factors such as the amount of neurotransmitter available.

— ARTIFICIAL NEURON —



An artificial neuron is an imitation of a human neuron.

→ The output is a function of the input, that is affected by the weights, and the transfer functions.



→ TRANSFER FUNCTION:-

- **Linear:**
The output is proportional to the total weighted input

- **Threshold:**
The output is set at one of two values, depending on whether the total weighted input is greater than or less than some threshold values

- **Non-linear:**
The output varies continously but not linearly as the input changes.

3/8/22

→ Class Work:

→ Production System -

- It is a computer program which is typically used to provide some form of AI which consist of set of rules about behaviours.
- It also includes mechanism necessary to follow those rules as system.
- These rules are basic representation helpful in expert systems, automated planning and action selection.

→ MAJOR COMPONENTS -

① Global database -
The global database is the central data structure used by the production system in AI.

② Set of production rule -
- The production rule operates on the global database
- Each rule has a pre condition, i.e, either satisfied or not by the global database. If the precondition is satisfied the rule is usually applied.
- The application of rule changes database

③ Control system -

The control system then chooses which applicable rule should be applied. and stops computation when a termination condition on a database is ~~st~~ satisfied.

If multiple rules are applied at same time the control system resolves the conflict.

→ The main feature of production System are -

• Simplicity -

→ The structure of each sentence in production System is unique as they use "if-then' structure

→ This structure provides simplicity in knowledge representation and improves the ~~re~~ ~~creditability~~ creditability.

→ This means the production rule code & the knowledge available in discrete piece so information can be added or deleted from the system without any effect (side) to the whole system.

④ Modifiability -

• This means facility for modifying rules is available. ~~skeletal form first and then it is o~~

• It allows the development of production rule in a skeletal form first and then it is accurate/ modified to suit a specific application.

(5) Knowledge Intensive-

- Knowledge base of production systems stores pure knowledge
- This part doesnot contain any type of control or programming information.

5/8/22

→ CLASSES OF PRODUCTION SYSTEM :-

There are 4 types of production system :-

1) monotonic production System-

- In this type of production system, the rules can be applied simultaneously as the use of one production system does not prevent the involvement of another rule that is selected at the same time.

2) Non-monotonic production system-

- This type of production system increases Efficiency in solving problems. The Implementation of these system doesnot require back tracking to correct the previous incorrect rules. The non-monotonic production system are necessary for the implementation point of view to find and efficient solutions.

- This production system increases efficiency since it is not necessary to keep track of the changes made in the search process.

3) Partially Commutative Production System -

- This class of production System helps to create a production system (P.S) that & can give results even by interchanging the state of rules.
- If using a set of rules ~~to form~~ transform state A to state B, then multiple combinations of those rules will be capable to convert state B to state A.

4) Commutative System -

- Commutative system are helpful where order of an operation is not important
- Also, problems where changes are reversible user commutative systems.
- In compairing to partially commutative where changes are irreversible. As, order of process is important to get correct results.
  Ex: Chemical process.

→ Advantage of Production System in AI :-

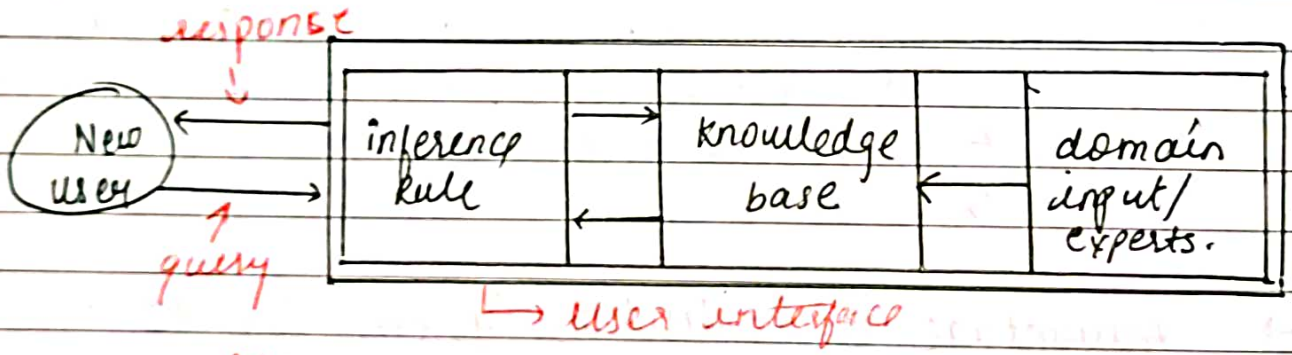
- Offers modularity that is rules can be added, deleted and modified individually.
- Separate Knowledge system and central system.
- excellent and feasible model that imitates human problem solving skills.
- Beneficial in real life application.

- It is very difficult to analysis the flow of control within production systems.
- It describes the operation that can be performed in the search research of a problem
- Their is absence of learning due to rule based production system that doesnot store the result of the problem for future use.

8/8/22 →

EXPERT SYSTEM:-

- knowledge base has many rules and information through domain experts.
- Domain experts has knowledge in particular field.



ARCHITECHTURE OF EXPERT SYSTEM

- An expert system consist of 3 components -
  1. inferenence rule
  2. User Interface
  3. knowledge base.

- EXAMPLES -

(1) DENDRAL -

- All the reactions of organic chemistry.

- It was an AI project that was made as a chemical analysis expert system.
- Used in organic chemistry to detect unknown organic ~~chemistry~~ molecules with help of their mass spectra and knowledge base of chemistry.

(2) MYCIN –

- It was designed to find bacteria causing infections like bacteremia and negitics.
- Used for recommendation of antibiotics and diganysis of blood clotting diseases

(3) PXDES –

- It is an expert system that is used to determine the type and level of lung cancer
- Takes picture of upper body and diognise it.

(4) CaDet –

- The cadet expert system is a diogonostic support system that can detect early stage cancer

→ USER INTERFACE:-

- It is an interface which ~~her~~ help a non-expert user to communicate with expert system to find the solution.

→ INFERENCE RULE:-

- CPU (Brain), It is brain of expert system as it is the main processing unit of system.

★ factual - Book, journals

★ Heuristic - rules of judgmental/analysis or kind of experimental knowledge

- It applies interface rules to the knowledge base to derive new information / conclusion

→ There are 2 types of interface engine - inference

① Deterministic inference engine -

- The conclusion drawn from this type of inference system are assumed to be true.

- Based on facts and rules.

② Probabilistic inference engine -

- This type of inference engine contains uncertainty, i.e. it is based on probability.

10/8/22

→ KNOWLFDGE BASE :-

- It contains domain specific information.
- Knowledge is required to gain intelligence or expertese.
- The Sucess of any expert system depends upon collection of accurate and precise knowledge.
- Knowledge base consist of factual and ~~heuristic~~ heuristic knowledge.
- It is about practice, judgment, once ability & of evaluation and guessing

★ Heuristic -

- Method of organise and formalise the knowledge in the knowledge base.

★ rule consist of if-then.
↳ production rule.

M T W T F S S
Page No.: YOUVA
Date:

→ EXPERT SYSTEM -

- It is in the form of if-then-else rule

If - condition, then - action. - else.

→ KNOWLEDGE ENGINEER -

- The knowledge engineer is the person who creates knowledge base, he acquires information from domain expert by recording, interviewing and observing his work, He then categorises and organises the information in a meaningful way in form of if-then-else rule.
- The knowledge engineer also monitors development of expert system.

→ USE OF INFERENCE ENGINE:

- Mechanism to derive the new knowledge from the knowledge base and the information provided by user.
- To recommend the use of efficient procedures and uses by the inference engine is essential is for getting a correct solution.
- To recommend a solution, inference engine uses following strategies:-

1. Forward chaining.
2. Backward chaining.

→ Development of expert system-

① MYCIN:

- To create knowledge base, eg - Mycin
- ES should be fed with expert ~~system~~ knowledge
- In case of mycin, experts, doctors (human) specilized in medical field of bacteria infection.
- Provide information about the causes, symptoms and other information in that domain.
- The knowledge base of Mycin is updated successfully in order to test it doctors provide a new problem ~~its~~ ~~which~~ to it.
- The problem is to identify the presence of bacteria and input details of patient. including symptons, medical history, current status.
- The expert system needs ~~at~~ a questionnaire to be ~~field~~ filled by the patient to know the genral info of patient such as age, gender, weight, hieght etc.
- Now, the System ~~symptom~~ has collected all the information, so it will find the output and solutions for problem by applying ~~the~~ if-then-else rule using inference engine and the facts / data stored in the knowledge base

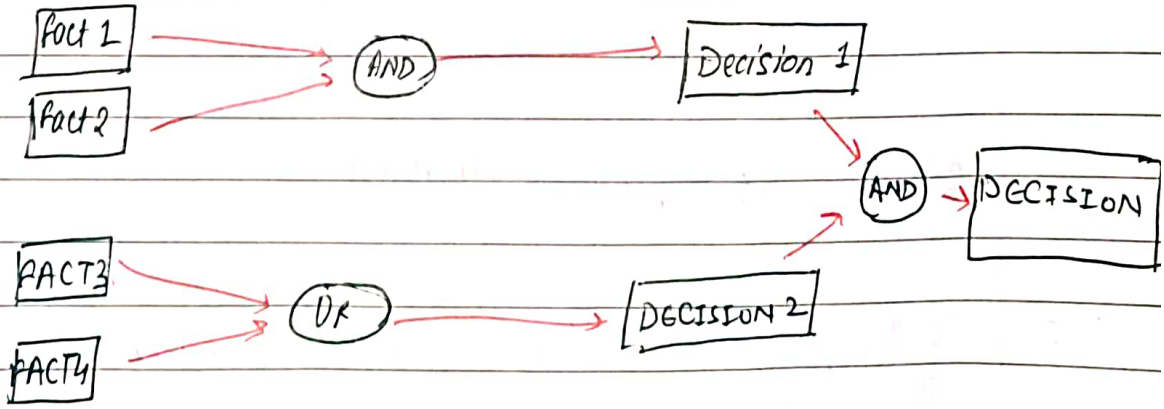
- In the end, user will get response by using the user interface.
- ~~In~~

★ Participants in development of expert system:-
- Knowledge engineer.
- domain experts.

→ Advantage of expert systems:-
- It is Not affected by emotions.
- It has constant performace
- It has high ~~seq~~ security
- Considers all the facts / data.
- Regular update in the system improves the performance.

26/8/22

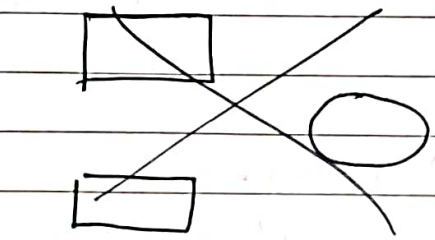
→ INFERENCE ENGINE:- (1) FORWARD
- Here the inference ~~it~~ engine follows the chain of variation and considers all the facts, rules before concluding to the solution.
- The strategy ~~control~~ is followed for working on the conclusion, result or effect.
- Eg - Prediction of share market status, as an effect of change in intrest rate.

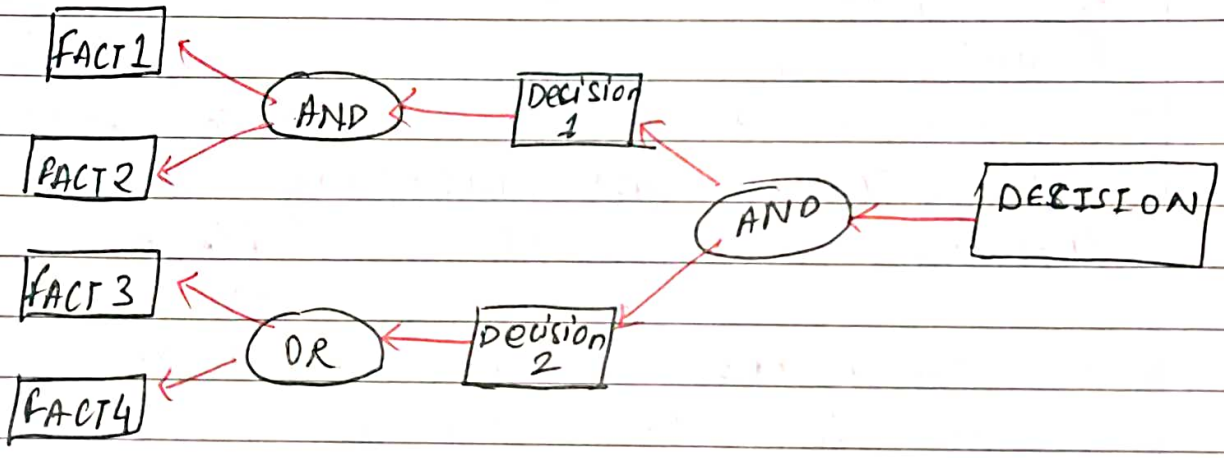


(2) BACKWARD –

This expert system

- answers to the system why it happens on the basis of what has already happened, the inference engine tries to find out which condition could have happened for this result.
- The strategy is followed for finding out cause or reason.
- Eg → diagnosys of blood cancer





1) Identify the problem domain.

2) To design the system. (Inference, rule design) or to integrate multiple source system data

diff. b/w forward & backward chaining / reasoning

M T W T F S S
Page No.:
Date:
YOUVA

3) to develop the prototype. (If-then rule).
4) Test and refine the prototype.
5) Develop and complete the expert system.
6) To maintain the expert system.

→ BENEFITS OF EXPERT SYSTEM:-

- It can be used in risky ~~basics~~ places.
- High speed responce to A queso query
- The performace of system performs steady.
- Error possibilities are less.
- The systems are reproduceable.

→ LIMITATION -

- Response of expert system may get wrong if knowledge base has wrong information.
- Maintainace and development cost is high.
- Knowledge accession is a very difficult task.
- For each domain we need a specific expert system.
-

30/8/22

→ ~~Disp~~ Discourse Integration:

- In this analysis our main focus is on the properties of a text as a whole that convey meaning by making connections between different component of the sentences. ~~of the~~.
- It means sense of content.
- This is the fifth and last phase of NLP.

* fragmented analysis:→

- In this analysis, we explain how extra meaning is read into text without actualy ~~bie~~ being encoding in there.

Regular expression → used to find pattern.

M T W T F S S
Page No.:
Date:
YOUVA

- This analysis require ~~post work~~
  - → world knowledge
  - → Understanding the intension, plan and goal.

Ex- Close the door → request or order depends on tone.

→ Applications of NLP:
- question answering → alexa, google assistant.
- Chatbots
- text summarization
- Infomation retrieval.
- Speach recognition
- spam detection
- Sentiment analysis.

8/9/22

→ Regular Expressions: Disjunctions

- letter inside square brackets [ ];

Pattern	Matches
[wW]oodchuck	Woodchuck, wood chuck
[1234567890]	Any digit.

- Ranges [A-Z]

Pattern	Matches	
[A-Z]	An upper case letter	Drenched Blossom
[a-z]	A lower case letter	my beans were impatient
[0-9]	A single digit	Chapter 1: Down the Rabbit Hole

→ Regular Expression : Negation in Disjunction

- Negations [^Ss]
- Carot means negation only when first in [].

Pattern	Matches	
[^A-Z]	Not an upper case letter	Oyfn pripetchik
[^Ss]	Neither 'S' nor 's'	I have no exquisite reason
[^e^]	Neither e nor ^	look here
a^b	The pattern a Carat b	look up a^b now

→ Regular Expression : More disjunction

- Woodchucks is another name for groundhog!
- The pipe | for disjunction.

Pattern	Matches.
groundhog | woodchuck	
yours | mine	yours mine
a|b|c	= [abc]
[gG]roundhog | [Ww]oodchuck	

→ Regular Expression : ? * + .

Pattern	Matches	
colou?r	optical prev char	Colour Color
oo*h!	0 or more of prev char	oh! ooh! oooh! ooooh!
o*h!		oh! ooh! oooh! ooooh!
baa+		baa baaa baaaa baaaaa
beg.n		begin begun began beg3n

→ Regular Expressions : Anchors ^ $

Pattern	Matches
^[A-Z]	Palo Alto
^[^A-Za-z]	1 "Hello"
\.$	The end.
.$	The end? The end!

→

Pattern	Matches.
{n}	n occurances
\d	[0-9] digits
\D	[^0-9] not digit
\w	[a-zA-Z0-9] alphanumeric.

15/9/22

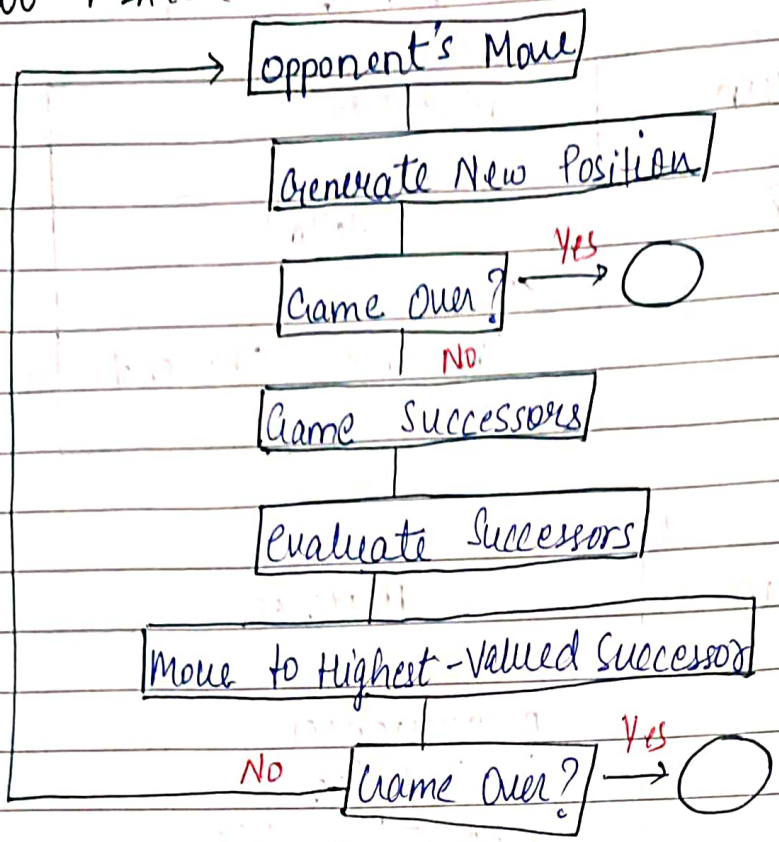
→ Game playing :-

Why its imp to play?

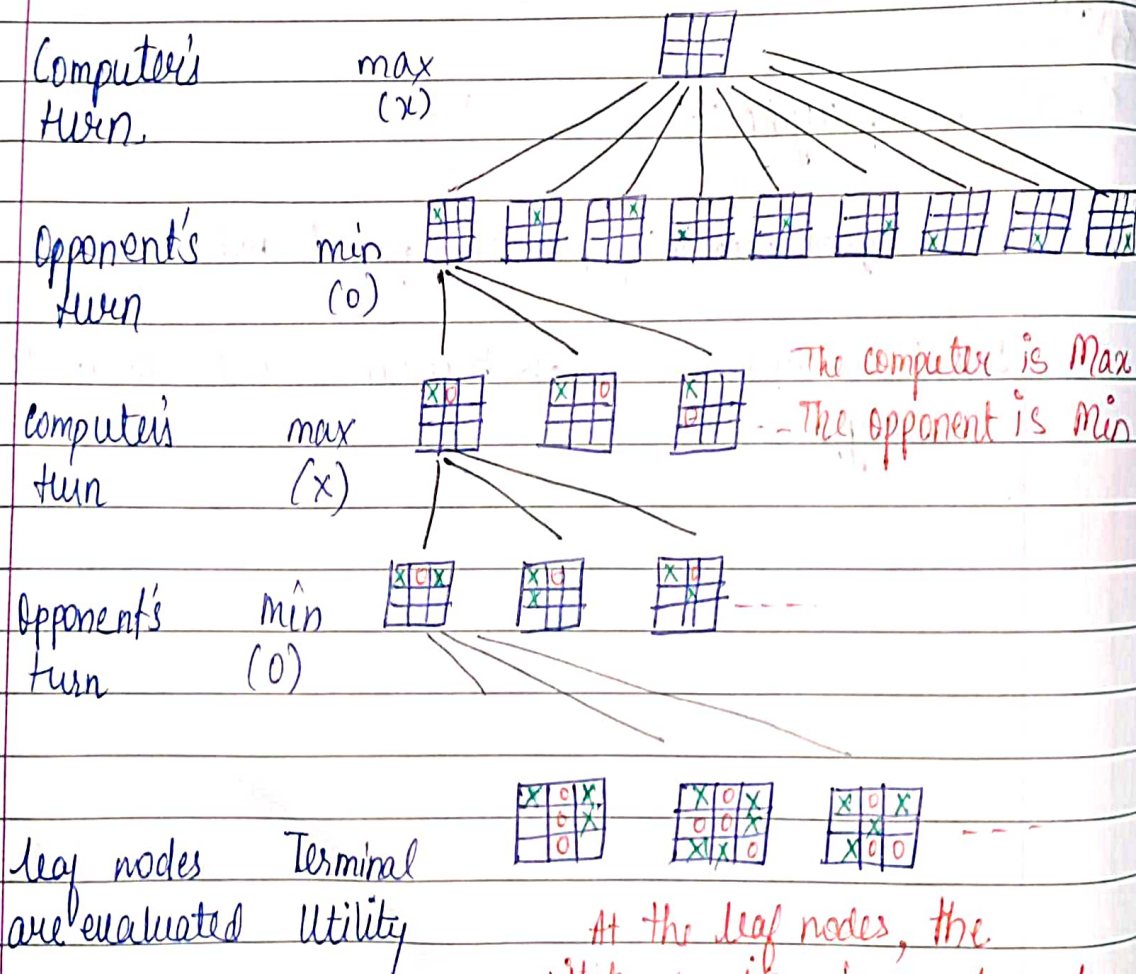
1) It's a good reasoning problem, formal, ~~and non~~

2) Direct comparison with humans and computers is easy.

→ TWO-PLAYER GAME :-



→ Game Tree (2-player Deterministic, Turns)

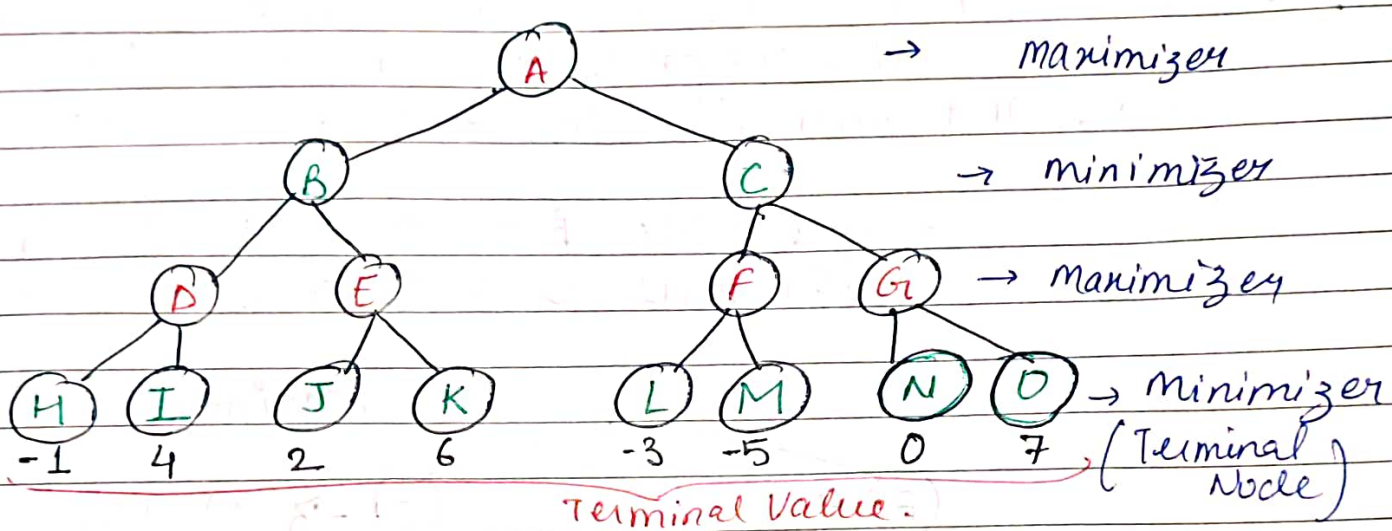


Mon 2

M T W T F S S
Page No.:
Date:
YOUVA

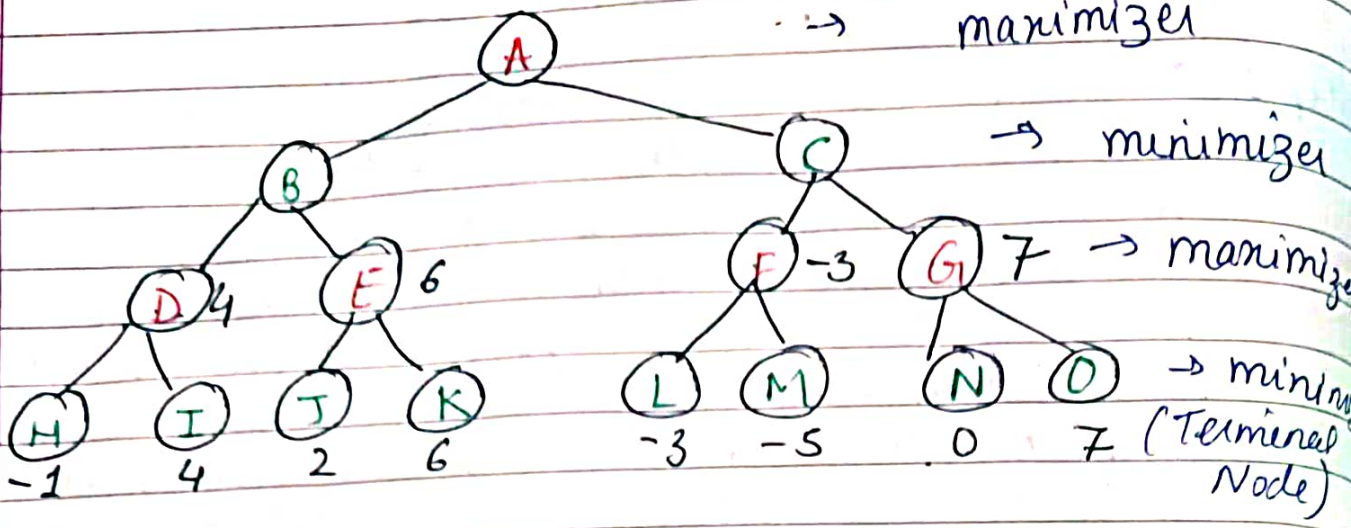
→ Min - Max Algorithem :-

Step-1. In the first step, the algorithm generates the entire game-tree and apply the ~~utit~~ utility function to get the utility values for the terminal states. In below tree diagram, let's take A is the initial state of the tree. Suppose maximizer takes first turn which has worse-case initial value = -infinity and minimizer will take next turn which has worst-case initial value = -infinity.



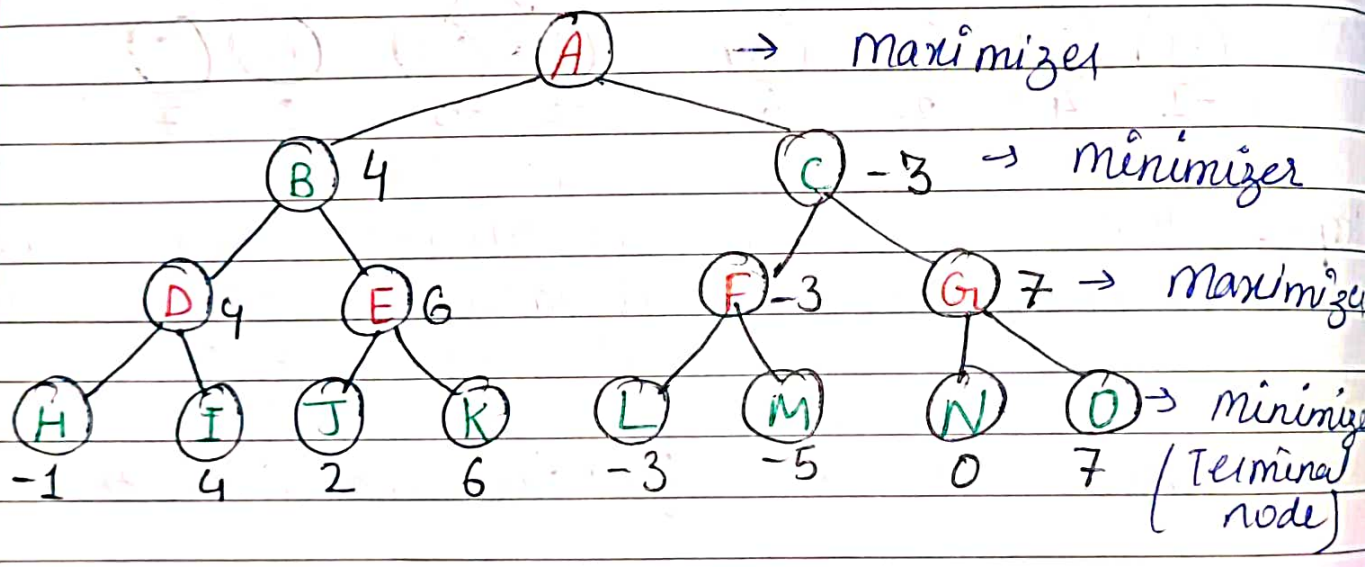
Step-2 Now first we find the utilities value for the maximizer, its initial value is $-\infty$, so we will compare each terminal state with initial value of maximizer and determine the higher node's values. It will find the maximum ~~at~~ all.

- for node D: max $(-1, -\infty) \Rightarrow$ max $(-1, 4) = 4$
- for node E: max $(2, -\infty) \Rightarrow$ max $(2, 6) = 6$
- for node F: max $(-3, -\infty) \Rightarrow$ max $(-3, -5) = -3$
- for node G: max $(0, -\infty) =$ max $(0, 7) = 7$.



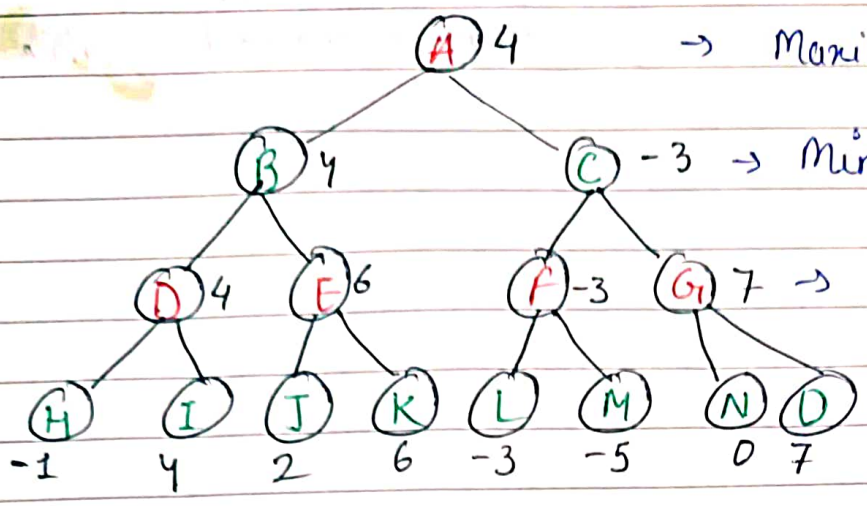
Step-3 In the next step, it is a turn for minimizer, so it will compare all nodes value with $+\infty$, and will find the 3rd layer node values.

- for node B = min (4, 6) = 4
- for node C = min (-3, +7) = -3



Step-4 Now its a turn for maximizer and it will again choose the maximum of all nodes value and find the minimum value for the root node. In this game tree, there are only 4 layers, hence we reached immediately to the root node, but in real games, there will be more than 4 layers.

· For node A max (4, -3) = 4



→ Properties of Min-Max Algorithm:-

1) Complete - Min-max algorithm is complete. It will definetely find a solution (if exist) in the finite search tree.

2) Optimal - Min-max algorithm is optimal if both opponent are playing optimally.

3) Time Complexity - As it performs DFS for the game-tree so the time complexity of min-max algorithm is $O(b^m)$, where b is branching factor of the game-tree, and m is the maximum depth of the tree.

4) Space Complexity - Space complexity of min-max algorithm is also similiar to DFS which is $O(bm)$.

→ Limitation of the maximum Algorithm:-
The main drawback of the minimum algorithm is that gets really slow for complex game such as chess, go etc. This type of game has a huge branching factor and the player has lot of choices

to decide. The limitation of the ~~maxi~~ minimax algorithm can be improved from alpha-beta pruning which we have discussed in the next topic.

to decide. The limitation of the ~~maxi~~ minimax algorithm can be improved from alpha-beta pruning which we have discussed in the next topic.

28/9/22

→ Alpha-Beta Pruning :-

- Alpha-beta pruning is modified version of the minimax algorithm.
- It is an optimization technique for the minmax algorithm.
- As we have seen in the minmax search algorithm that the number of games states it has to examine are exponential in depth of the tree.
- Since we cannot eliminate the exponent, but we can cut it to half.
- Hence there is a technique by which without checking each node of the same tree, we can compute the correct minimax decision, and this technique is called ~~prunir~~ pruning
- This involves two threshold parameters Alpha and beta for future expansion, so it is called Alpha-beta pruning.
- It is also called as Alpha-beta Algorithm.
- Alpha-beta ~~algorithm~~ pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leave but also entire sub-tree.
- ~~The~~ The two-parameter can be defined as :-

a) Alpha :
The best (highest-value) choice we have found so far at any point along the ~~point~~ path of Maximizer. The initial value of

alpha is $-\infty$

(b) Beta:
The best (lowest value) choice we have found so far at any point along the path of minimizer. The initial value of beta is $+\infty$

- Alpha-beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does, but it removes all the node which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm fast.

→ Key-Points about alpha-beta pruning:-

- The Max player will only update the value of alpha.
- The Mix player will only update the value of beta.
- While back-tracking the tree, the node value will be passed to upper nodes instead of value of alpha and beta.
- We will only pass the alpha, beta values to the child nodes.

→ Pseudo-code for Alpha-beta Pruning:-

- Function minimax(node, depth, alpha, beta, maximizingPlayer) is --
- if depth == 0 or node is a terminal node then,
- return static evaluation for node.

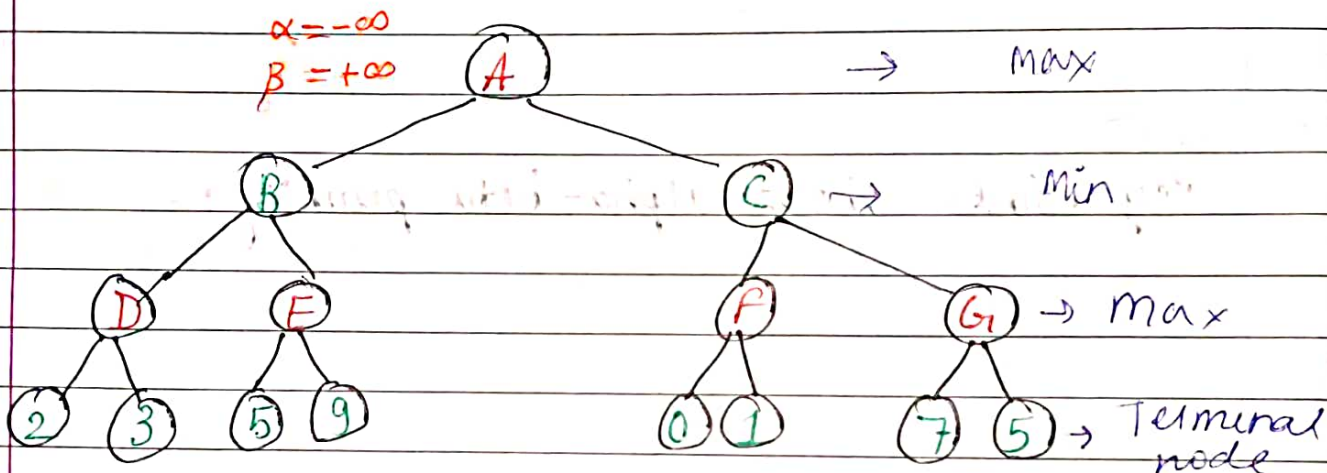
29/9/22

→ Working of Alpha-Beta pruning:

Let's take an example of two players search tree to understand the working of alpha-beta pruning.

Step-1

At the first step, Max players will start first move from node A where $\alpha = -\infty$ and $\beta = +\infty$, these values of alpha and beta passed down to node B where again, $\alpha = -\infty$, and $\beta = +\infty$, and node B passes the same value to its child D.



Step 2

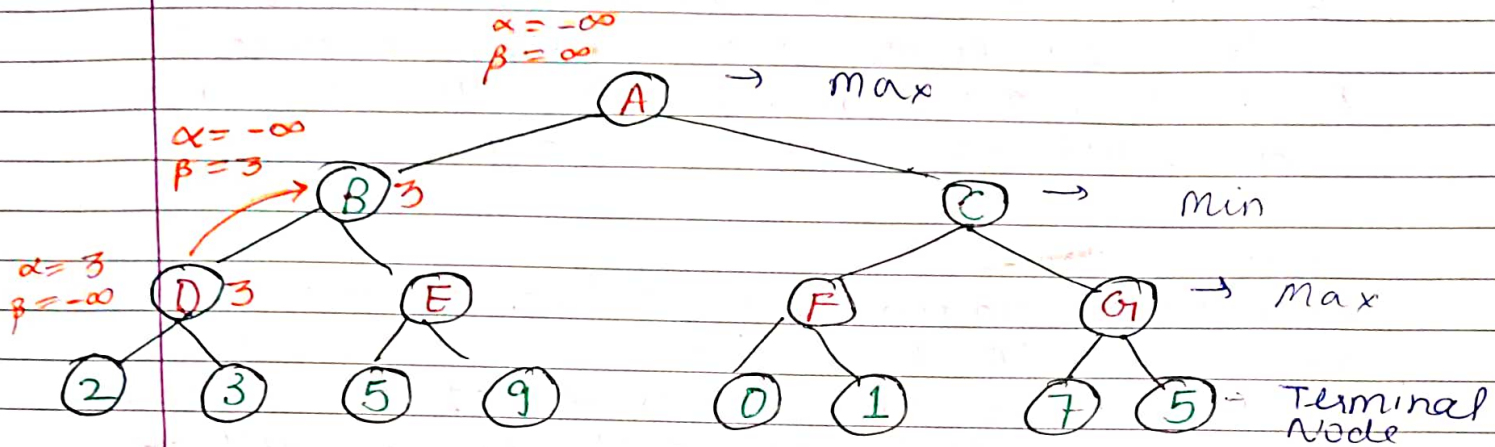
At node D, the value of α will be calculated at its turn for max. The value of α is compared with firstly 2 and then 3, and the max (2,3) = 3 will be the value of α at node D, and node value will also be 3.

Step 3

Now algorithem backtrack to node B, where the value of β will change as this is a turn of Min, Now $\beta = +\infty$, will compare with the available subsequent nodes value, i e min

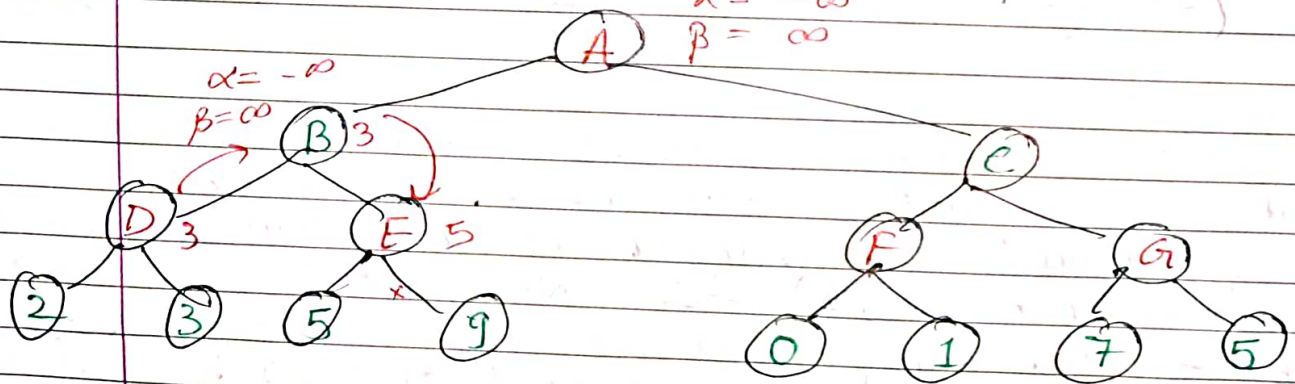
$(\infty, 3) = 3$, hence at node B now $\alpha = -\infty$ and $\beta = 3$

In the next step, algorithm traverse the next successor of node B, which is node E, and the value of $\alpha = -\infty$, $\beta = 3$ will also be passed.



Step 4:

At node E, Max will take its turn, and the value of alpha will change. The current value of alpha will be compared with 5, so max $(-\infty, 5) = 5$, hence at node E $\alpha = 5$ and $\beta = 3$, we have $\alpha >= \beta$, so the right successor of E will be pruned, and algorithm will not traverse it, and the value at node E will be 5.

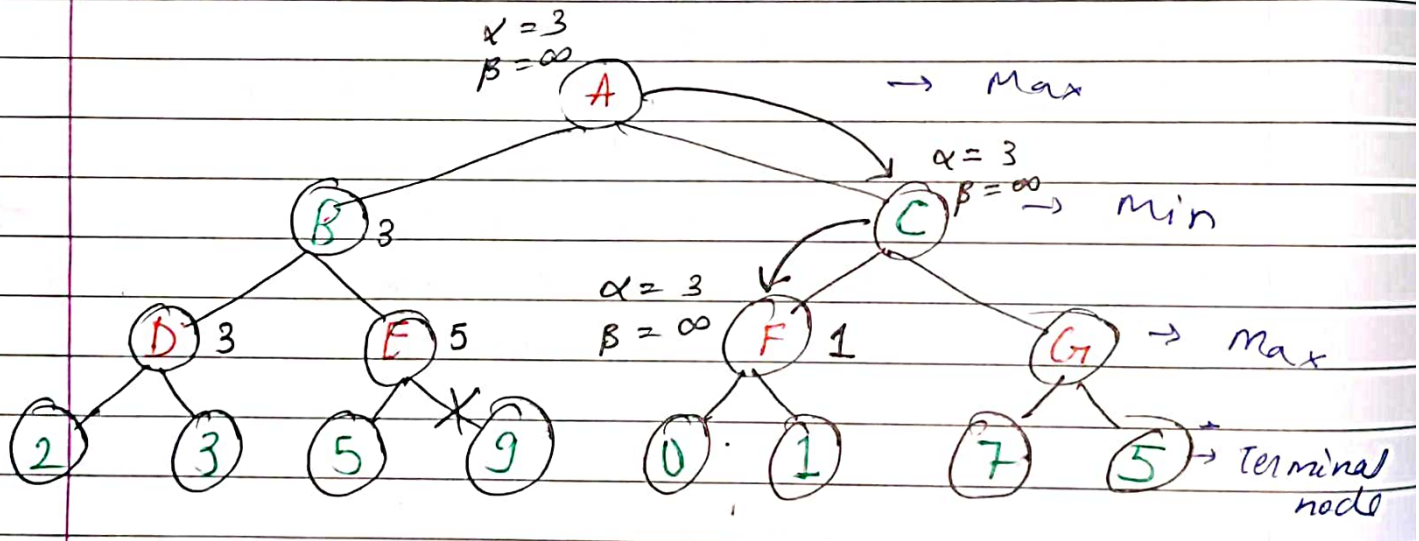


Step 5:

At step (next), algorithm again backtrack the tree, from node B to node A. At node A, the value of alpha will be changed the maximum available value is 3 as max $(-\infty, 3) = 3$ and $\beta = +\infty$, these two values now passes to right successor of A which is node C. At node C, $\alpha = 3$ and $\beta = +\infty$, and the same values will be passed on to F.

Step 6:

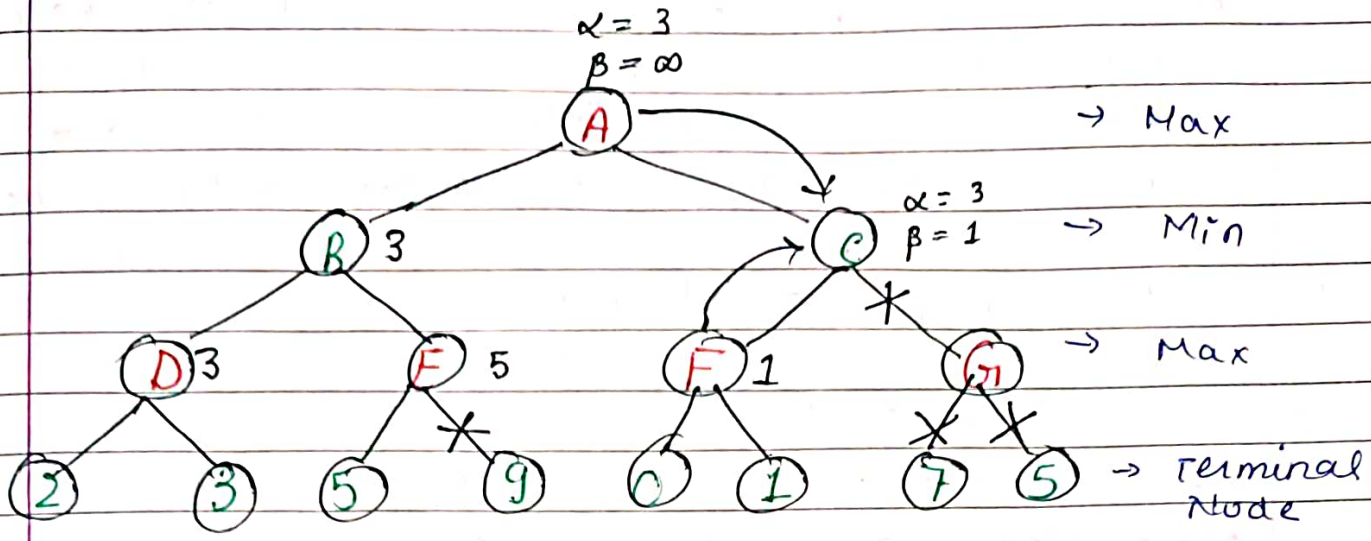
At node F, again the value of α will be compared with left child which is 0, and max $(3, 0) = 3$ and then compared with right child which is 1, and $(3, 1)$ max $= 3$, still α remains 3, but node value of f will become 1.



Step 7:

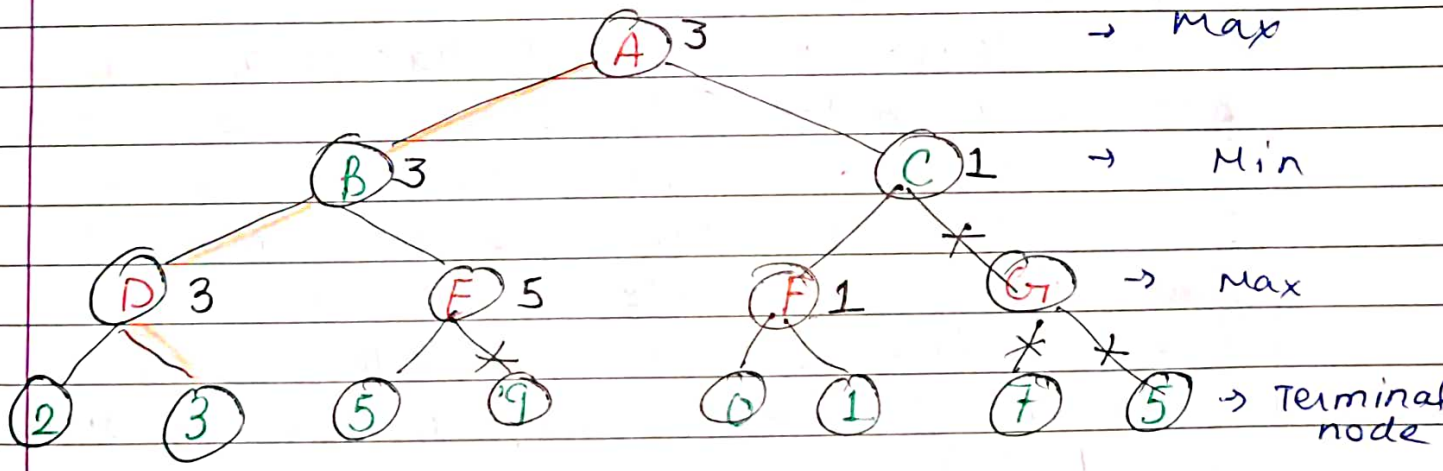
Node F return the node value 1 to node C at C $\alpha = 3$ and $\beta = +\infty$, here the value of beta will be changed, it will compare with 1 so min $(\infty, 1) = 1$. Now at C, $\alpha = 3$ and $\beta = 1$, and again satisfies the condition ~~$\alpha < \beta$~~ $\alpha >= \beta$, so the next child of C which is G

will be pruned, and algorithm will not compute the entire sub-tree G1.



Step 8:

C now returns the value of 1 to A here the best value for A is max (3, 1) = 3. Following is the final game tree which is the showing the nodes which are computed and nodes which has never computed. Hence the optimal value for the maximizer is 3 for this example.



M T W T F S S

Page No.: YOUVA

Date:

MOVE ORDERING IN ALPHA-BETA PRUNING:-

The effectiveness of alpha-beta pruning is highly dependent on the order in which each node is examined. Move order is an important aspect of alpha-beta pruning

- It can be of two types:

- **Worst Ordering:**
In some cases, alpha-beta pruning algorithm does not prune any of the leaves of the tree, and works exactly as minimax algorithm. In this case, it also consumes more time because of alpha-beta factors, such a move of pruning is called worst ordering. In this case, the best move occurs on the right side of the tree. The time complexity for such an order is $O(b^m)$.

- **Ideal Ordering:**
The ideal ordering for alpha-beta pruning occurs when lot of pruning happens in the tree, and best moves occurs at the left side of the tree. We apply DFS hence it first search left of the tree and go deep twice as minimax algorithm in the same amount of time. Complexity in ideal ordering is $O(b^{m/2})$

uses of array, stack, linked list → in real life.
↳ free.

M T W T F S S
Page No.:
Date:
YOUVA

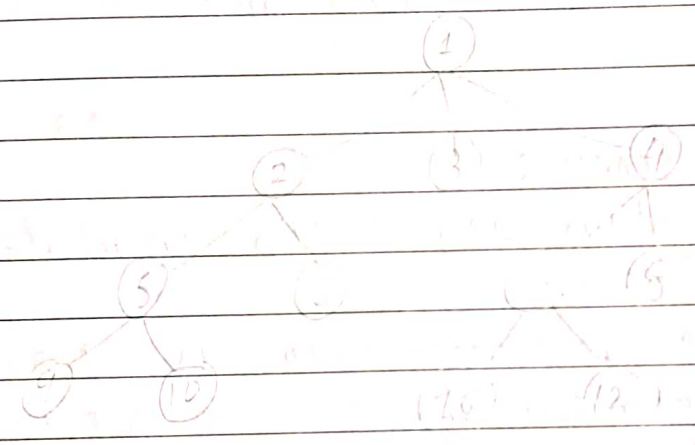
→ Rules to find good ordering :-

·→ Occurs the best move from the shallowest node.

·→ Order the nodes in the tree such that the best node are cracked first.

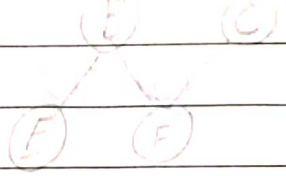
·→ Use domain knowledge while finding the best move. Ex: For chess, try order: capture first, then threats, then forward moves, backward moves.

·→ We can bookkeep the states, as there is a possibility that states may repeat.



BFS = 1 2 3 4 5 6 7

DFS = 1 2 5 6 3 4 7



BFS = A B C D E F

DFS = A B E F C D

→ Search algorithms in AI.

- Search techniques are universal problem solving methods
- Problem solving agents in AI mostly use these search techniques or algorithms to solve a specific problem.
- Problem solving agents are the goal-based agents.
- Based on search problem, we can classify the search algorithm into uninformed and informed search.

1) UN-INFORMED SEARCH:

- It doesnot contain any domain knowledge such as location of goals.
- It operates in a brutforce way as it only includes information about how to ~~traves~~ traverss the tree and how to identify leaf and root nodes.
- Un-informed search applies a way in which search tree is search without any information about search space.
- It is also called blind search.
- It examines each nodes of tree until it achieves the goal node.
- It can be ~~devided into~~ divided into 5 main types.

(1) Breadth first Search.
(2) Depth first search.
(3) Depth limited search.

BFS = Queue → First in First out
DFS = Stack → First in last out
last in first out

M T W T F S S
Page No :
Date:
YOUVA

(2) INFORMED SEARCH:-

- Informed search algorithm use domain knowledge
- In an informed search, problem information is available which can guide the search.
- Informed search strategy can find a solution more efficiently than an uninformed search strategy.
- Informed search is also called a Heuristic search.
- A heuristic search is a way which might not always be guaranteed for best solution but guarenteed to find a good solution in reasonable time.

(1) Breadth first Search:-

- Breadth-first search is the most common search strategy for traversing a tree or graph.
- This algorithm searches breadth wise in a tree or graph, so it is called breadth-first search.
- BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- The BFS algorithm is an example of general-graph search algorithm
- BFS implemented using FIFO queue data structure.

- Advantages:

→ BFS will provide a solution if any solution exist.

→ If there are more than one solution for a given problem, then BFS will provide the

minimal solution which requires the least number of steps.

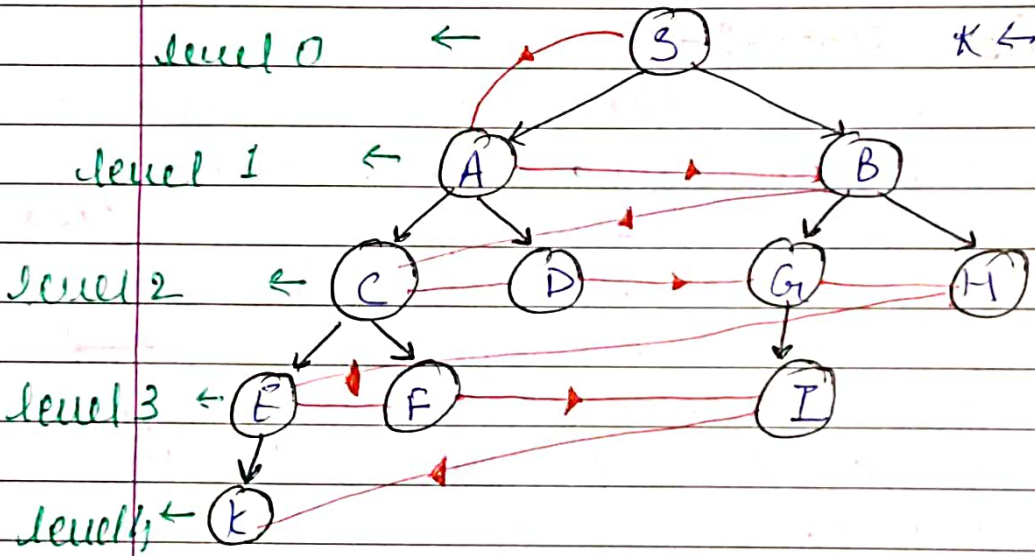
- **Disadvantages:**

→ It requires a lot of memory since each level of the tree must be saved into memory to expand the next level.

→ BFS needs lot of time if the solution is far away from the root node.

- **Example:**

$S \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow G \rightarrow H \rightarrow E \leftarrow F \leftarrow I \leftarrow K$



→ Time complexity:

- Time complexity of BFS algorithm can be obtained by the ~~no~~ number of nodes traversed in BFS until the shallowest Node. where the d= depth of shallowest solution and b is a node at every state.

$$T(b) = 1 + b^2 + b^3 + \dots + b^d = O(b^d)$$

→ Space Complexity :
Space complexity of BFS algorithm is given by the memory size of frol frontier which is $O(b^d)$

(2) Depth-first search:-
- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation
- The process of DFS algorithm is similar to the BFS algorithm.
- Note - Backtracking is an algorithm technique for finding all possible solutions using recursion.

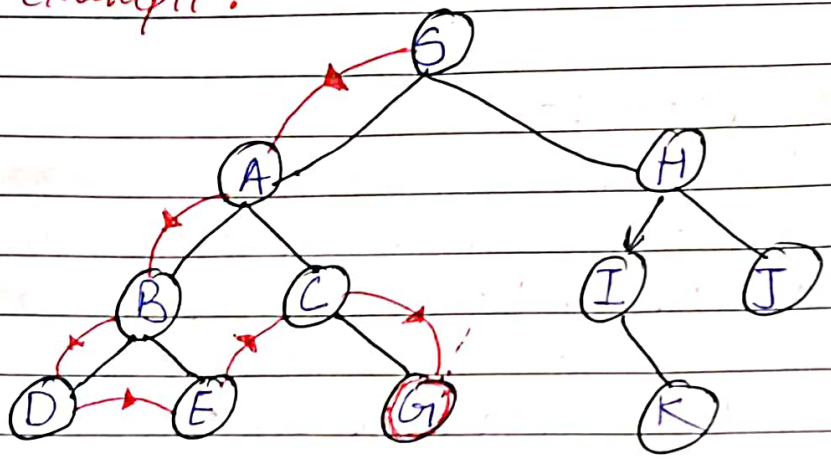
- Advantages:
→ DFS requires very less memory as it only need to store a stack of the node on the path from root node to the current node.
→ It takes less time to reach to the goal node than the BFS algorithm (if it traverses in right path).

- Disadvantages:

→ There is ~~possibly~~ possibility that many states keep re-occurring and there is no guarantee of finding the solution.

- DFS algorithm goes for # deep-down searching and sometimes it may go to the infinite loop.

- Example :-

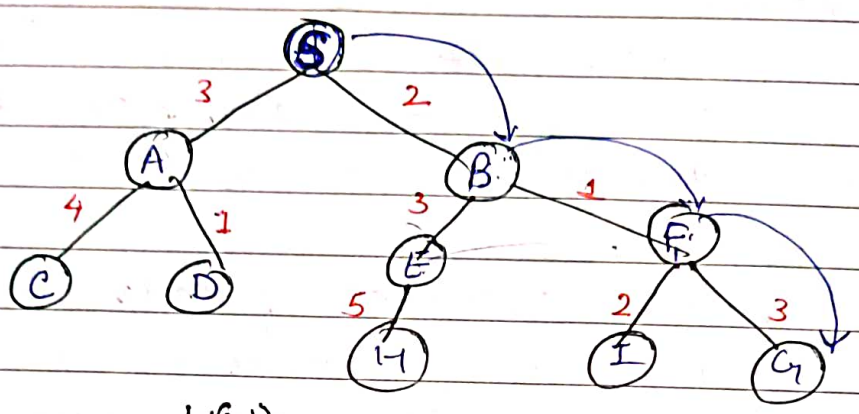


DFS => S → A → B → D → E → C → G → H → I → K → J.

1/11/22

HEURIBIIC FUNCTION :-

Heuristic is a function which is used in informed search and finds most promising path.



node -	H(N)
A	12
B	4
C	7
D	3
E	8
F	2
G	0
H	4
I	9
S	13

(1) Best first search.
(Greedy Search).

OPEN - we have not visited the node.

close - we have visited.

Initial

OPEN -	A, B	E, F, A	AEIG	I
CLOSE -	S	B	F	G.

Shortest path

- we have to check the value of open nodes with their Heuristic value. (H,N)

∴ Path ⇒ S → B → F → G.

Goal Node → end of tree.

(b) Best first algorithm :-

OPEN

Step 1. Place start node into open list.

Step 2. If open list is empty, stop and return failure.

3. Remove the node 'N' from open list which has lowest value of (HN) and place it to CLOSED list.

4. Expand node 'N' and generate successor of node 'N'.

5. Check each successor of node 'N' and find whether any node is goal node or not, if any successor node is goal node, return success and terminate search. else and return to 6 (step)

6. For each successor node algorithm checks for evaluation function f(n) ~~for~~ ec and check if node has been in either CLOSE or OPEN list. If node has not ~~to~~ been in both list then add it to OPEN list.

7. Return to ~~set~~ step 2. and continue till we reach goal node.

---

int → initial
OPEN [A,B]
CLOSE [S]

inter 1 →
OPEN [A]
CLOSE [S, B]

inter 2 →
OPEN [E, A]
CLOSE [S, B, F]

iter 3 →
OPEN [I, G, E, A]
CLOSE [S, B, F]

iter 4 →
OPEN [I, E, A]
CLOSE [S, B, F, G]
end node.

AO* Search (AND-OR) GRAPH :-

The depth first search and Breadth first search given earlier for OR trees or graphs can be easily adopted by AND-OR graph. The main difference lies in the way termination condition are determined, since all goals following an AND nodes must be realised; where as a single goal node following an OR node will do. So for this purpose we are using AO* Algorithm.

→ Algorithm :-

Step 1: Place the starting node into OPEN.

Step 2: Compute the most promissing solution tree say TO.

Step 3: Select a node n that is both on OPEN and member of TO. Remove it from OPEN and place it in CLOSE

Step 4: If n is the terminal goal then leveled n as solved and levelled all the ancestors of n as solved. If the starting node is marked as solved then success and exit.

Step 5: If n is not a solvable node, then mark n as unsolvable. If starting node is marked as unsolvable, then return failure and exit.

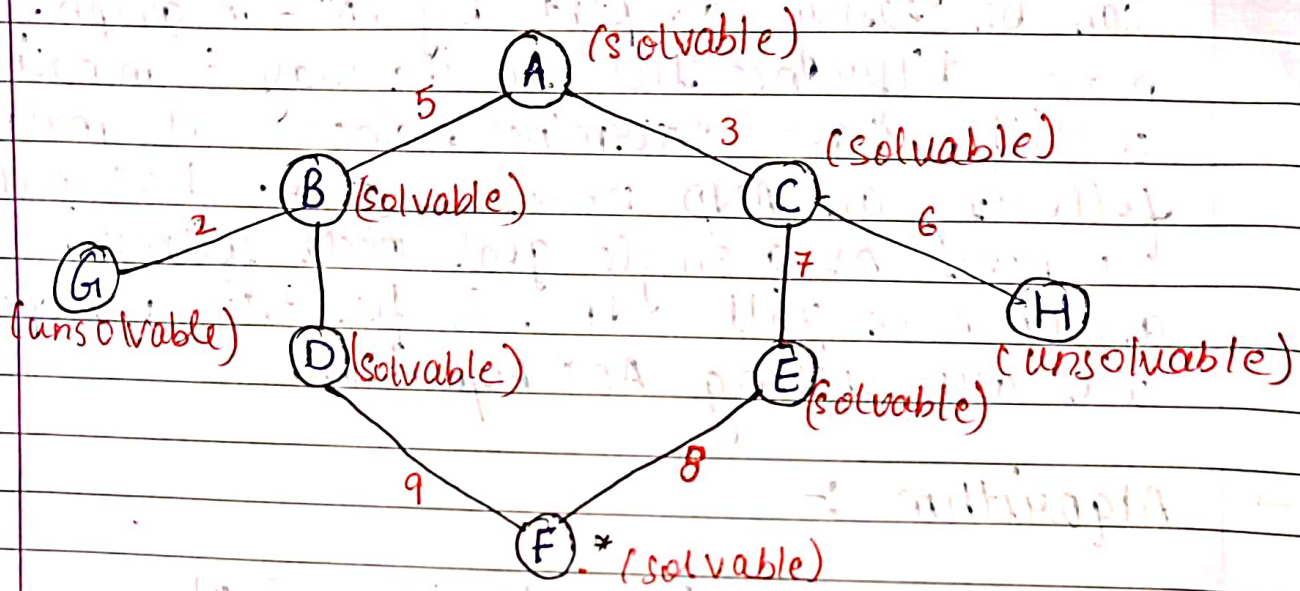
Step 6: Expand n. Find all its successors and find their h (n) value, push them into OPEN.

Step 7: Return to step 2.

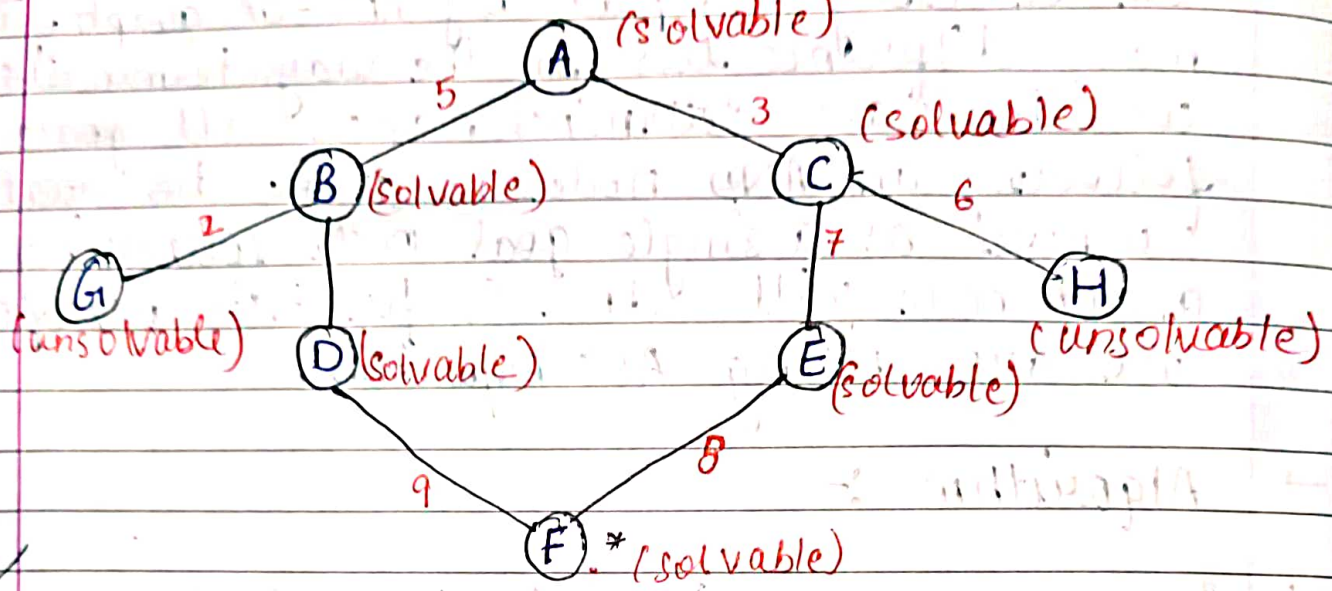
Step 8: Exit.

→ IMPLEMENTATION:

let us take the following example to implement the Ao* algorithm.



→ IMPLEMENTATION:
Let us take the following example to implement the AO* algorithm.



22 | 11 | 22

→ Knowledge Represent (KR, KRR):-

* Declarative knowledge -
- Includes facts and figures.
- Also called descriptive knowledge.

* Procedural knowledge -
- Responsible for knowing how to do something.
- Rules, strategy, procedures, agenda.
- Procedural knowledge is based on strategy and rules.

* Meta Knowledge -
- Knowledge about other type of knowledge.

* Heuristic Knowledge:-
  - Knowledge of experts in particular subject.
  - Heuristic knowledge is thumbrule based on previous knowledge, experience and approaches.

* Structural Knowledge:-
  - Basic knowledge to problem solving.
  - It describe relationship b/w various concepts and logic.

→ Preposition logic:-
works on 0 and 1 so its called boolian logic

- Constructed by combining 2 or more propositio using parantheses.

Truth Table:-

→ Negation:-

P	¬P
True	False
False	True

→ Disjunction:- or ∨

P	Q	P ∨ Q
T	True	True
F	True	T
T	False	T
F	F	F

→ Conjunction → and.

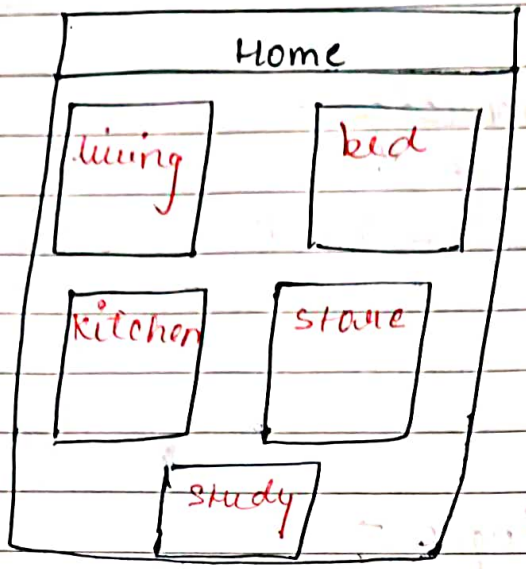
P	Q	P ∧ Q
True	T	T
False	F	F
T	T	F
F	F	F

→ Implication → TAVB

P	Q	P → Q
T	T	T
T	F	F
F	T	T
F	F	T

→ SCRIPT - ROBBERY TRACK: successfull snatch.

(1) → file - 1 frames.

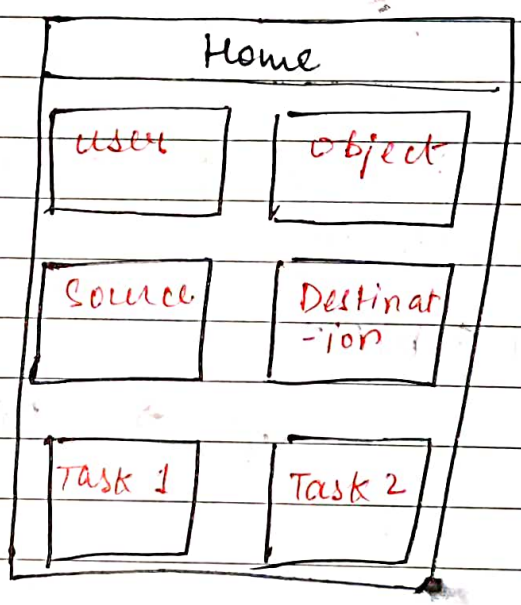


← slot

→ Declarative information

← slot values

file - 2



→ procedural information

→ Script discuss of procedure, process & situation knowledges.

(13500)

→ Situation :-

(1) Getting Gun.
(2) Hold up bank.
(3) Escape with money.

→ Props :-

→ We use symbols for props.
Gun - G.
Bag - B
Loot - L
Getaway car - C

→ Roles :-

- Robber → R
- Casier → M
- Bank Manage → O
- Policemen → P.

→ Result :-

- R has more money.
- O is angry.
- M is shock.
- P is shot.

→ SYMBOLS :-

- ATRANS → Abstract or transfering something.
  → transfer of abstract relationship.
- P TRANS → Transfer of physical location.
- MTRANS → Transfer of Mental information
- PROPEL → Propelling an object by physical force

→ GRASP → Grasping of an object by an actor.
→ INGEST → Ingesting
→ EXPEL → Expelling of something from a body of animal or Human
→ MBUILD → Mentally Building
→ SPEAK → Speaking by producing sounds.

• Abstract Transfer	ATRANS	give, take
• Physical Transfer	PTRANS	go, come
• Mental Transfer	MTRANS	ask, tell.
• Propelling	PROPEL	push, pull.
• Grasp	GRASP	Clutching, catching
• Ingesting	INGEST	eat, drink, taking in
• Expelling	EXPEL	sweat, cry
• Mentally Building	MBUILD	Decide, inferring
• Speaking by producing sound	SPEAK	talking
• Attending to stimulus by focusing a sense organ toward it.	ATTEND	see, hear.
• Movement of actor	MOVE	KICK, punch

Scripts:-

Scene 1:- Getting a gun.

R: PTRANS R into Gunshop
R: MBUILD R choice of Gn
R MTRANS choice
R: ATRANS buys Gn

Scene 2 Holding up Bank.

Bank of Baroda

R PTRANS R into Bank. ([redacted])
R ATTENDS eyes M, O and P
R MOVE R to M position.
R GRASP G
R Move G to point to M
R MTRANS " Give me money or else " to M.
P MTRANS " Hands up " to R.
R PROPEL shots G
P INGEST BULLET from G.
M ATRANS L to R
M ATRANS L puts in B
M PTRANS EXITS
O ATRANS RAISES Alarm.

SCENE-3:- The Gateway

- The goal of this theory are to help us drawing inferences from instances
- To be independent of the words used in original input.
- for any 2 or more sentences that are identical in meaning, there should be only one representation of that meaning, it has been used in many programs that pretends to understand english → MARGIE, SAM, PAM
- Margie → meaning → analysis & response generation and inference on english.

SAM - Script applier mechanism → scripts to understand stories.
PAM - Plan applier mechanism

→ Advantage (~~Disadvantage~~)

- CD provides a structure into which notes representing information can be placed.
- It provides a specific set of primitives
- Ex. of primitive acts are - ATRANS, PTRANS, PROPEL, MTRANS, MBUILD, MOVE, ~~SE~~ ATTEND, INGEST, EXPEL, SPEAK, GRASP,
- There are 6 primitive conceptual categories. which are set of allowable dependency in a sentence:

PP
ACT
PA
AA
T

LOC

PP - real world object → pen, Human etc.
ACT - Real world actions → dance, fight
PA - Attributes of objects. → intelligent, fool.
AA - Attribute of action
T - Times.
LOC - Locations,

① John gives mary a book
John ⇔ ATRAS (BOOK) ← R [from mary, to John]

Double ~~arrow~~ arrow indicates 2 way links b/w actor PP and action AC

② John took the book.

PP[John] ⇒ ACT (TOOK) ←° [PP[BOOK]

→ In preposition logic we represent statement that are facts but unfortunately PL is not sufficient to represent complex sentence or natural language statements, because i.e PL has very less express powers.

→ Some human are intelligent.

→ Sachin likes cricket.

FOL → Predicate logic → first order logic.
FOPL → first order predicate logic.

First order logic is powerful logic language that represent information about object and also represent relationship b/w object.
• It is extension of propositional logic.

→ TERMS :-

→ OBJECT - people, numbers.

→ Relation → unary relation ⇒ Red, round
Single operation
↳ Binary relation → double operator.
→ sister of, brother of

2 main parts

① Syntax
Syntax of FOL express collection of basic elements of FOL.
Constants - 1, 2, A, John
Variable - X, Y, A, B
Predicate - Relation

(1) Atomic Sentence
(2) Compound sentence
(Predicate) → Term, Term, ... n Terms)
Complex sentences are made by ∘

Predicate → Relation which binds object in sentence
Subject → main part of sentence.

e.g. $\underset{\text{Subject}}{x}$ is an integer.
predicate.

- Quotifier is language ~~which~~ element which generates quantification (quantity of specimen)
- These are symbols that permit to determine the range and scope of variable in a logical expression.

(1) → Universal Quantifier :-
This is a symbol of logical representation which specifies that statement within its range is true for everything.

'∀' symbol

Note → universal quantifier is always used with implication

If x is also available, then → ∀x, ∈
for each x.
for every x.

(1) And. (2)

M T W T F S S
Page No.:
Date:
YOUVA

① All men drinks coffee.

men(x) ∀

~~∀(x)me~~

∀men(x) ⟶ coffee(x)

② Existential Quantifier :-
Which express that the statement within ~~the~~ is true for atleast 1 instance ~~for~~ of something

Note → Existential Quatifier we always use AND or conjuction and (∧) (^)

- ∃x. (there exist a X)
- for some x.

ex → Some boys are intelligent.
∃x boys(x) ∧ intelligent.

ex → Some girls are beautiful and intelligent.
∃x girls (x) ∧ (beautiful (x) ∧ intelligent(x)

ex → Every man respect his parent

∀x ∃man(x) → respect (x, parent)

~~Some boys~~

→ (not) all students like both maths & scien