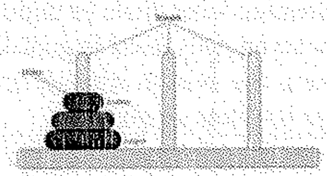
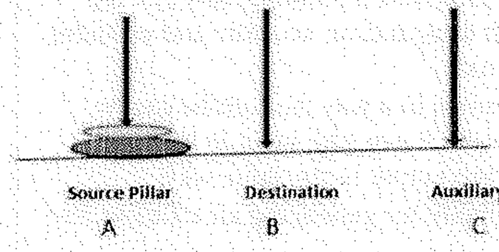
| SCHEME OF VALUATION | | | | |
|---|---|---|---|---|
| (Scoring Indicators) | | | | |
| **Revision: 2021** | | | | |
| **Course Name: Artificial Intelligence** | | | | |
| **Course Code: 3344** | | **QID: 2109230386** | | |
| **Qs. No** | **Scoring Indicator** | **Split up Score** | **Sub Total** | **Total** |
| **I** | **PART A** | | | 9 |
| 1) | Artificial Intelligence is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions. | 1 | 1 | |
| 2) | Turing Test. | 1 | 1 | |
| 3) | The Knowledge base is a type of storage that stores Knowledge acquires from the different experts of the particular domain. | 1 | 1 | |
| 4) | Tautology. | 1 | 1 | |
| 5) | Uninformed Search. | 1 | 1 | |
| 6) | Greedy Search. | 1 | 1 | |
| 7) | In AI, the Divide and Conquer approach refers to a problem solving strategy that involves breaking down a complex problem into smaller, more ,manageable sub problems. | 1 | 1 | |
| 8) | 1.High Level Expertise 2.Understandable 3.Reliable 4.Highly Responsive | 1 | 1 | |
| 9) | MYCIN and EMYCIN. | 1 | 1 | |
| **II** | **PART B** | | | 24 |
| 1) | Speech recognition in artificial intelligence is also known as automatic speech recognition (ASR). It converts spoken language into written text to understand and respond. Automatic Speech recognition works by analysing audio input and applying complex algorithms and cutting-edge technologies like machine learning (ML) and neural networks to recognize and interpret spoken words. | 3 | 3 | |
| 2) | Alan Turing introduced a test to check whether a machine can think like a human or not, this test is known as the Turing Test. In this test, Turing proposed that the computer can be said to be an intelligent if it can mimic human response under specific conditions.<br>Explain the turing method in short. | 3 | 3 | |
| 3) | An Agent is anything that takes actions according to the information it gains from the environment. A human agent has sensory organs to | 1.5 | 3 | |

| | | | | |
|---|---|---|---|---|
| | sense the environment and the body parts to act, while a robot agent has sensors to perceive the environment<br><br>An environment is everything in the world which surrounds the agent, but it is not a part of an agent itself. An environment can be described as a situation in which an agent is present.The environment is where agent lives, operate and provide the agent with something to sense and act upon it. | 1.5 | | |
| 4) | An ontology is a basic term of knowledge as a collection of ideas within an area and their connections. Classes, individuals, characteristics, and relations, as well as rules, limitations, and axioms, must all be explicitly specified for such a description to be possible. As a result, ontologies not only provide a reusable and sharable knowledge representation, but they may also contribute new domain information. The ontology data model may be applied to a set of individual facts to form a knowledge graph, which is a collection of things whose kinds and connections are represented by nodes and edges connecting them. | 3 | 3 | |
| 5) | Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.<br><br>• It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.<br>• It is also a way which describes how we can represent knowledge in artificial intelligence.<br>Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human. | 3 | 3 | |
| 6) | Uninformed search in AI refers to a type of search algorithm that does not use additional information to guide the search process. Instead, these algorithms explore the search space in a systematic, but blind, manner without considering the cost of reaching the goal or the likelihood of finding a solution. Examples of uninformed search algorithms include Breadth-First search (BFS), Depth-First search (DFS), and Depth-Limited search. | 3 | 3 | |
| 7) | Rule based systems in AI are a type of knowledge representation and reasoning system. They operate on a set of explicitly defined rules or logical statements to make decisions or draw conclusions. These systems consists of two main components:<br>a) Inference Engine b) Knowledge base<br><br>Write in short about inference engine and knowledge base. | 3 | 3 | |
| 8) | Branching is the process of generating sub problems. Bounding refers to ignoring partial solutions that cannot be better than the current best solution. It is a search procedure to find the optimal | 3 | 3 | |

| | | | | |
|---|---|---|---|---|
| | solution. It eliminates those parts of a search space which doesnot contain the better solution. | | | |
| 9) | Forward Search: In forward search, the system starts from known facts or initial information and uses rules or inference mechanisms to reach a conclusion. It works by applying rules or knowledge in a step-by-step manner, moving from premises to conclusions.This approach is often used in rule-based expert systems, where the system follows a chain of rules to make inferences and reach a decision.<br><br>Backward search: The system begins with a goal or a desired outcome and works backward to determine what facts or conditions must be true to achieve that goal. It is commonly used in goal-driven reasoning or diagnostic systems, where the system tries to find the causes or explanations for a particular problem or goal. | 1.5<br><br>1.5 | 3 | |
| 10) | • High-Level Expertise : The most helpful feature of the expert system is the highlevel expertise which it gives while solving any problem. It offers best thinking, just like top experts and end up with imaginative, accurate and efficient solutions to the problems.<br>• Understandable: It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.<br>• Reliable: It is much reliable for generating an efficient and accurate output.<br>• Highly responsive: ES provides the result for any complex query within a very short period of time. | 3 | 3 | |
| | **PART C** | | | 42 |
| III) | Applications of AI<br>1. Language Models<br>2. Information Extraction<br>3. Information Retrieval<br>4. Speech Recognition<br>5. Machine Translation<br>6. Natural Language Processing<br><br>Write any 4 application and its description. | 7 | 7 | |
| | OR | | | |

| | | | | |
|---|---|---|---|---|
| IV) | Tower of hanoi is mathematical game puzzle where we have three pile(pillars) and n numbers of disk.<br>This game has some rules (Rules of game)<br>1. Only one disk will move at a time.<br>2. The larger disk should always be on the bottom and the smaller disk on top of it.(Even during intermediate move)<br>3. Move only the uppermost disk.<br>4. All disk move to destination pile from source pile.<br>So, here we are trying to solve that how many moves are required to solve a problem (It depends on number of disk).<br><br>When we have two disk and 3 pillars (pile, A, B, C)<br><br><br>In the above diagram, following the rule of the game our target is move the disks<br>from source pile (pillar) to the destination pillar. (lets take a look how many steps/moves are required to make this happen).<br>Step1: Move small disk to the auxiliary pillar (A).<br>Step2: Move large disk to the Destination pillar (B).<br>Step3: Move small disk to the Destination pillar(B).4<br>So, basically when we have 2 disks we required 3 move to reach the destination .<br>Tower of hanoi problem is an example of recursion and backtracking. | 4<br>+<br>3 | 7 | |

<table>
<tr><td></td><td></td><td>3</td><td></td><td></td></tr>
<tr><td>V)</td><td>Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. But how machines do all these things comes under knowledge representation and reasoning.<br><br>Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.<br><br>It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.<br><br>It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.</td><td></td><td>7</td><td></td></tr>
<tr><td></td><td>Objects represents all the information related to the objects present in the world. For example, buses need drivers, cars have wheels etc.<br><br>Events represents occurrence that takes place in a system. These occurrences could be instantaneous or have a duration.<br><br>Actions represents processes or behaviours that cause changes in a system. They are the driving force behind events.<br><br>Relations describe how objects are connected or interact with each other. These relationships can be binary or hierarchical.</td><td>4</td><td></td><td></td></tr>
</table>

OR

| | | | | |
|---|---|---|---|---|
| VI) | Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions. A proposition is a declarative statement which is either true or false. It is a technique of knowledge representation in logical and mathematical form.<br>Example:<br>1. a) It is Sunday.<br>2. b) The Sun rises from West (False proposition)<br>3. c) 3+3= 7(False proposition)<br>4. d) 5 is a prime number.<br><br>Basic facts about propositional logic:<br>• Propositional logic is also called Boolean logic as it works on 0 and 1.<br>• In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q,R, etc.<br>• Propositions can be either true or false, but it cannot be both.<br>• Propositional logic consists of an object, relations or function, and logical connectives.<br>• These connectives are also called logical operators.<br>• The propositions and connectives are the basic elements of the propositional logic.<br>• Connectives can be said as a logical operator which connects two sentences.<br>• A proposition formula which is always true is called tautology, and it is also called a valid sentence.<br>• A proposition formula which is always false is called Contradiction.<br><br>• Statements which are questions, commands, or opinions are not propositions such as "Where is Rohini", "How are you", "What is your name", are not propositions.<br>Syntax of propositional logic:<br>The syntax of propositional logic defines the allowable sentences for the knowledge representation. There are two types of Propositions:<br>a. Atomic Propositions<br>b. Compound propositions<br>o Atomic Proposition: Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must<br>be either true or false.<br>Example:<br>1. a) 2+2 is 4, it is an atomic proposition as it is a true fact. | 7 | 7 | |

2. b) "The Sun is cold" is also a proposition as it is a false fact.

o Compound proposition: Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.

Example:

1. a) "It is raining today, and street is wet."
2. b) "Ankit is a doctor, and his clinic is in Mumbai."

Logical Connectives:

Logical connectives are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives. There are mainly five connectives, which are given as follows:

1. Negation: A sentence such as ¬ P is called negation of P. A literal can be either Positive literal or negative literal.
2. Conjunction: A sentence which has ∧ connective such as, P ∧ Q is called a conjunction.
Example: Rohan is intelligent and hardworking. It can be written as,
P=Rohan is intelligent,
Q= Rohan is hardworking. → P∧ Q.
3. Disjunction: A sentence which has ∨ connective, such as P ∨ Q. is called disjunction, where P and Q are the propositions.
Example: "Ritika is a doctor or Engineer",
Here P= Ritika is Doctor. Q= Ritika is Doctor, so we can write it as P ∨ Q.
4. Implication: A sentence such as P → Q, is called an implication. Implications are also known as if-then rules. It can be represented as
If it is raining, then the street is wet.
Let P= It is raining, and Q= Street is wet, so it is represented as P → Q
5. Biconditional: A sentence such as P⇔ Q is a Biconditional sentence, example If I am breathing, then I am alive
P= I am breathing, Q= I am alive, it can be represented as P ⇔ Q.

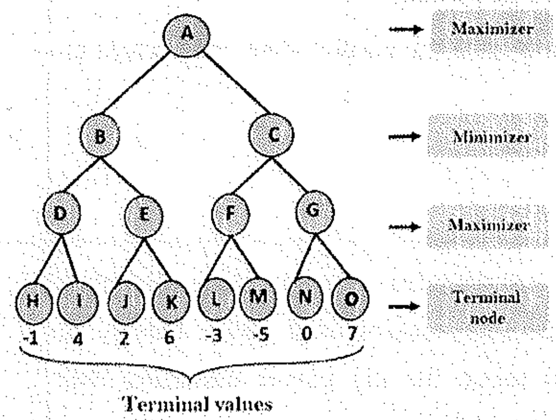
*Include Truth table also

| | | | | |
|---|---|---|---|---|
| VII) | Breadth First Search is an uninformed search technique(blind search/brute force method).<br><br>BFS algorithms will have:<br>• A problem **graph,** containing the start node S and the goal node G.<br>• A **strategy,** describing the manner in which the graph will be traversed to get to G.<br>• A **fringe,** which is a data structure used to store all the possible states (nodes) that you can go from the current states.<br>• A **tree,** that results while traversing to the goal node.<br>• A solution **plan,** which the sequence of nodes from S to G.<br><br>Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures.<br><br>It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level.<br><br>It is implemented using a queue.<br><br>*Note: After the definition and description , write the working of Breadth first search. | 7 | 7 | |
| | OR | | | |
| VIII) | Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbour has a higher value.<br><br>It is also called greedy local search as it only looks to its good immediate neighbour state and not beyond that.<br><br>**No backtracking:** It does not backtrack the search space, as it does not remember the previous states. | 3 | 7 | |

| | | | | |
|---|---|---|---|---|
| | **Algorithm for Simple Hill Climbing:**<br>o **Step 1:** Evaluate the initial state, if it is goal state then return success and Stop.<br>o **Step 2:** Loop Until a solution is found or there is no new operator left to apply.<br>o **Step 3:** Select and apply an operator to the current state.<br>o **Step 4:** Check new state:<br>a. If it is goal state, then return success and quit.<br>b. Else if it is better than the current state then assign new state as a current state.<br>c. Else if not better than the current state, then return to step2.<br>o **Step 5:** Exit. | 4 | | |
| IX) | o Mini-max algorithm is a recursive or backtracking algorithm which is used in decision-making and game theory. It provides an optimal move for the player assuming that opponent is also playing optimally.<br>o Mini-Max algorithm uses recursion to search through the game-tree.<br>o Min-Max algorithm is mostly used for game playing in AI. Such as Chess, Checkers, tic-tac-toe, go, and various tow-players game. This Algorithm computes the minimax decision for the current state.<br>o In this algorithm two players play the game, one is called MAX and other is called MIN.<br>o Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.<br>o Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value.<br>o The minimax algorithm performs a depth-first search algorithm for the exploration of the complete game tree. | 3<br><br>4 | 7 | |

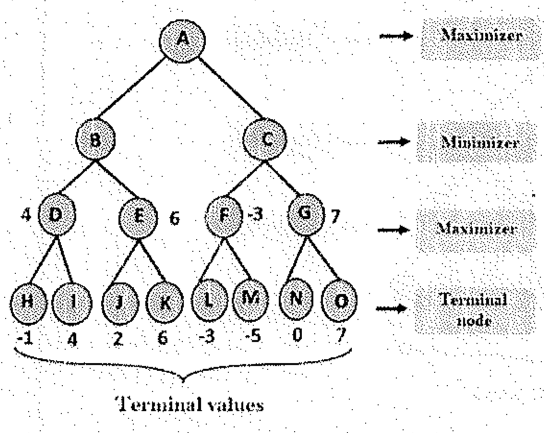
- The minimax algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

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



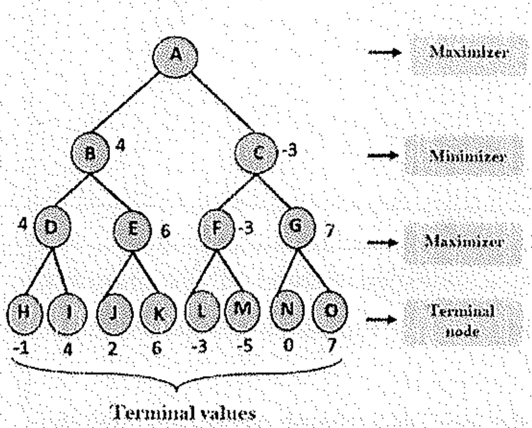

**Step 2:** Now, first we find the utilities value for the Maximizer, its initial value is -∞, so we will compare each value in terminal state with initial value of Maximizer and determines the higher nodes values. It will find the maximum among the all.

- For node D max(-1,- -∞) => max(-1,4)= 4
- For Node E max(2, -∞) => max(2, 6)= 6
- For Node F max(-3, -∞) => max(-3,-5) = -3
- For node G max(0, -∞) = max(0, 7) = 7



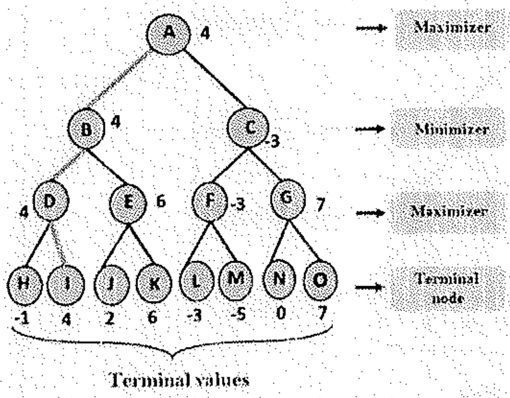

**Step 3:** In the next step, it's a turn for minimizer, so it will compare all nodes value with $+\infty$, and will find the 3rd layer node values.

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




**Step 4:** Now it's a turn for Maximizer, and it will again choose the maximum of all nodes value and find the maximum value for the root node. In this game tree, there are only 4 layers, hence we reach immediately to the root node, but in real games, there will be more than 4 layers.

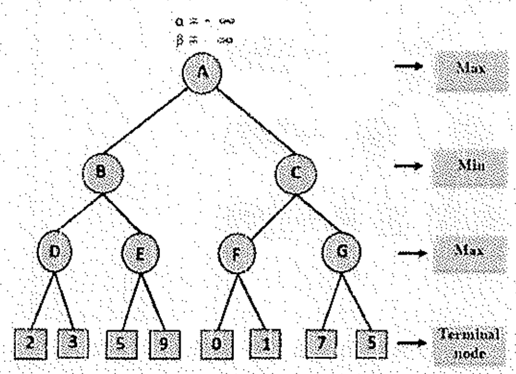
- For node A max(4, -3)= 4

<table>
<tr><td></td><td>
</td><td></td><td></td><td></td></tr>
<tr><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td>X)</td><td>
<ul>
<li>Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.</li>
<li>As we have seen in the minimax search algorithm that the number of game 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 game tree we can compute the correct minimax decision, and this technique is called pruning. This involves two threshold parameter Alpha and beta for future expansion, so it is called alpha-beta pruning. It is also called as Alpha-Beta Algorithm.</li>
<li>Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leaves but also entire sub-tree.</li>
<li>The two-parameter can be defined as:</li>
</ul>
</td><td>3</td><td>7</td><td></td></tr>
</table>

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

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 +∞.

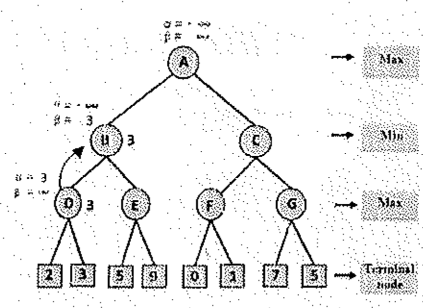
**Step 1:** At the first step the, Max player will start first move from node A where α= -∞ and β= +∞, these value of alpha and beta passed down to node B where again α= -∞ and β= +∞, and Node B passes the same value to its child D.




4

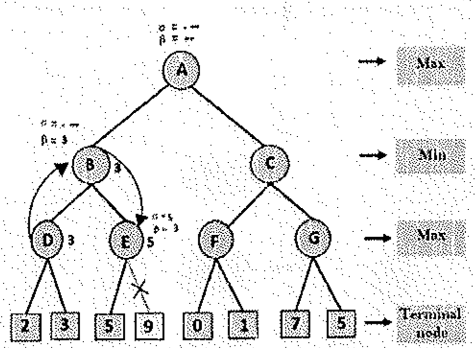
**Step 2:** At Node D, the value of α will be calculated as its turn for Max. The value of α is compared with firstly 2 and then 3, and the max (2, 3) = 3 will be the value of α at node D and node value will also 3.

**Step 3:** Now algorithm backtrack to node B, where the value of β will change as this is a turn of Min, Now β= +∞, will compare with the available subsequent nodes value, i.e. min (∞, 3) = 3, hence at node B now α= -∞, and β= 3.



In the next step, algorithm traverse the next successor of Node B which is node E, and the values of α= -∞, and β= 3 will also be passed.

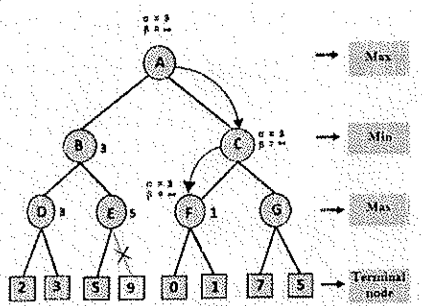
**Step 4:** At node E, Max will take its turn, and the value of alpha will change. The current value of alpha will be compared with 5, so max (-∞, 5) = 5, hence at node E α= 5 and β= 3, where α>=β, so the right successor of E will be pruned, and algorithm will not traverse it, and the value at node E will be 5.



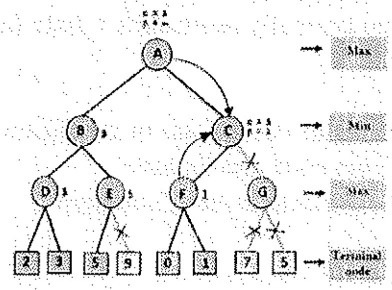

**Step 5:** At next step, algorithm again backtrack the tree, from node B to node A. At node A, the value of alpha will be changed the maximum available value is 3 as max (-∞, 3)= 3, and β= +∞, these two values now passes to right successor of A which is Node C.

At node C, α=3 and β= +∞, and the same values will be passed on to node F.

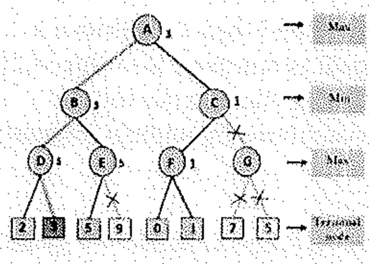
**Step 6:** At node F, again the value of α will be compared with left child which is 0, and max(3,0)= 3, and then compared with right child which is 1, and max(3,1)= 3 still α remains 3, but the node value of F will become 1.



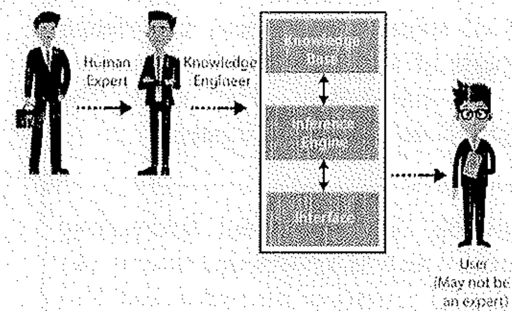
**Step 7:** Node F returns the node value 1 to node C, at C α= 3 and β= +∞, here the value of beta will be changed, it will compare with 1 so min (∞, 1) = 1. Now at C, α=3 and β= 1, and again it satisfies the condition α>=β, so the next child of C which is G will be pruned, and the algorithm will not compute the entire sub-tree G.




**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.



| | | | | |
|---|---|---|---|---|
| XI) | Goal stack planning is a technique often associated with classical AI planning systems and was first introduced in the STRIPS (Stanford Research Institute Problem Solver) planning system in the early 1970s.<br><br>The steps involved in goal stack planning:<br><br>1. Representation of Goals: Goals are represented as logical expressions or conditions. Each goal typically represents a state or condition that you want to achieve.<br><br>2. Initial State: The planner starts with an initial state or condition representing the current situation. This state can include information about the environment, the objects in it, and their current properties or states.<br><br>3. Goal Stack: The goals to be achieved are maintained in a stack. Initially, the stack contains the top-level goals that need to be accomplished. These goals may have subgoals, which are pushed onto the stack as well.<br><br>4. Search for Actions: The planner searches for actions or operators that can be applied to the current state to achieve the top goal on the stack. These actions are selected based on their preconditions (the conditions that must be true for the action to be applicable) and their effects (the changes they make to the state).<br><br>5. Achieving Subgoals: If an action is found that can achieve a subgoal, it is executed, and the subgoal is removed from the stack. If an action has unsatisfied preconditions, the planner may recursively try to achieve those preconditions by adding subgoals to the stack.<br><br>6. Backtracking: If an action cannot be found to achieve a goal, the planner backtracks by undoing previous actions and trying alternative actions or strategies. This is where search algorithms and heuristics play a role in finding a valid plan.<br><br>7. Termination: The planner continues this process until all goals in the stack have been achieved, or it determines that no valid plan exists.<br><br>8. Plan Execution: Once a plan is found, it can be executed in the real world or simulated environment to achieve the desired goals.<br><br>Goal stack planning is particularly useful for complex planning problems where goals can be decomposed into a hierarchy of subgoals. It allows for a structured approach to problem-solving and is widely used in various domains, including robotics, automated manufacturing, and AI-based game playing. | 7 | 7 | |

| | | | | |
|---|---|---|---|---|
| | | | | |
| OR | | | | |
| XII) | An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert.<br>➢ It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.<br>➢ The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance.<br>➢ One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.<br><br>**Architecture**<br><br>An expert system mainly consists of three components:<br>• o User Interface<br>• o Inference Engine<br>• o Knowledge Base<br><br>1. User Interface<br>With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, it is an interface that helps a non-expert user to communicate with the expert system to find a solution. | 3<br><br>4 | 7 | |

| | | | | |
|---|---|---|---|---|
| | 2. Inference Engine(Rules of Engine)<br>• o The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.<br>• o With the help of an inference engine, the system extracts the knowledge from the knowledge base.<br>• o There are two types of inference engine:<br>• o Deterministic Inference engine: The conclusions drawn from this type of inference engine are assumed to be true. It is based on facts and rules.<br>• o Probabilistic Inference engine: This type of inference engine contains uncertainty in conclusions, and based on the probability.<br><br>Inference engine uses the below modes to derive the solutions:<br>• o Forward Chaining: It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.<br>• o Backward Chaining: It is a backward reasoning method that starts from the goal and works backward to prove the known facts.<br><br>3. Knowledge Base<br>The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledgebase, the more precise will be the Expert System. | | | |
| XIII) | Humans are best at understanding, reasoning, and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. **But how machines do all these things comes under knowledge representation and reasoning.**<br>Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.<br>• o It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.<br>• o It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent | 4<br><br>3 | 7 | |

| | | | | |
|---|---|---|---|---|
| | machine to learn from that knowledge and experiences so that it can behave intelligently like a human.<br><br>What to Represent:<br>Following are the kind of knowledge which needs to be represented in AI systems:<br>• o **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.<br>• o **Events:** Events are the actions which occur in our world.<br>• o **Performance:** It describe behavior which involves knowledge about how to do things.<br>• o **Meta-knowledge:** It is knowledge about what we know.<br>• o **Facts:** Facts are the truths about the real world and what we represent.<br>• o **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).<br><br>**Knowledge:** Knowledge is awareness or familiarity gained by experiences of facts, data, and situations. | | | |
| | OR | | | |
| XIV) | MYCIN was a ground breaking expert system in the field of artificial intelligence, and here are some more details about its key features and contributions:<br><br>1.Development and Purpose: MYCIN was developed at Stanford University in the 1970s by Edward Shortliffe. Its primary purpose was to assist physicians in diagnosing bacterial infections and recommending appropriate antibiotics. It focused on the domain of infectious diseases, where there is often a complex interplay of symptoms and medical knowledge.<br><br>2. Knowledge Representation: MYCIN used a rule-based knowledge representation system. It had an extensive knowledge base comprising a large number of rules. These rules were written in the form of "IF-THEN" statements, where the system would match patient data to these rules to make diagnostic and treatment recommendations.<br><br>3. Inference Engine: The inference engine in MYCIN was responsible for executing the rules and drawing conclusions based on the information provided by the user. It used forward chaining, which means it started with available data and worked forward to | 7 | 7 | |

| | | | | |
|---|---|---|---|---|
| | reach a diagnosis or treatment recommendation.<br><br>4. Uncertainty Handling: MYCIN was notable for its ability to handle uncertainty. It could deal with incomplete or uncertain information, which is crucial in medical diagnosis. It used certainty factors (values between 0 and 1) to represent the degree of confidence in its conclusions.<br><br>5. User Interaction: MYCIN interacted with users through a question-and-answer dialogue. It would ask the physician a series of questions about the patient's symptoms, medical history, and test results to gather information for diagnosis.<br><br>6. Contributions: MYCIN demonstrated the potential of expert systems in the medical domain. It showcased that computers could assist healthcare professionals by providing accurate and consistent diagnostic recommendations. It also paved the way for subsequent expert systems in various domains.<br><br>MYCIN was a significant milestone in the history of AI and continues to be studied and referenced in the field of artificial intelligence and healthcare informatics as an early example of knowledge-based expert systems. | | | |