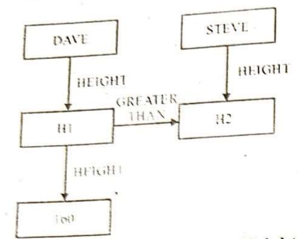


Fig.(3) : Comparison of Two Height

Section – C

Q.6.(a) Differantiate between Statistical reasoning and Symbolic reasoning.(10)

Ans. Symbolic Reasoning : The Symbolic methods basically represent uncertainty belief as being

(i) True.

(ii) False, or

(iii) Neither True nor False.

Some methods also had problems with

- Incomplete Knowledge
- Contradictions in the knowledge.

Statistical Reasoning : Statistical methods provide a method for representing beliefs that are not certain (or uncertain) but for which there may be some supporting (or contradictory) evidence.

Statistical methods offer advantages in two broad scenarios:

Genuine Randomness : Card games are a good example. We may not be able to predict any outcomes with certainty but we have knowledge about the likelihood of certain items (*e.g.* like being dealt an ace) and we can exploit this.

Exceptions : Symbolic methods can represent this. However if the number of exceptions is large such system tend to break down. Statistical techniques can summarise large exceptions without resorting enumeration.

Q.6.(b) What do you mean by Planning ? Describe planning in situational calculus. (10)

Ans. Planning : Planning is finding a sequence of actions that will achieve a goal. The planning refers to the process of computing several steps of the problem solving procedure before executing any of them.

The *planning* problem in Artificial Intelligence is about the decision making performed by intelligent creatures like robots, humans, or computer programs when trying to achieve some goal. It involves choosing a sequence of actions that will (with a high likelihood) transform the state of the world, step by step, so that it will satisfy the goal. The world is typically viewed to consist of atomic *facts*, and actions make some facts true and some facts false. In the following we discuss a number of ways how to formalize planning, and how the planning problem can be solved automatically.

Planning in situational calculus : Planning is required because the world is :

1. **Dynamic :** As the world is not static, it is dynamic, therefore planning is required.

2. **Stochastic :** Stochastic means random. A stochastic process is one whose behavior is non-deterministic, in that a system's subsequent state is determined both by the process's predictable action and by a random element. In artificial intelligence, stochastic programs work by using probabilistic methods to solve problems, as in simulated annealing, stochastic neural networks, and genetic algorithms. A problem itself may be stochastic as well as in planning under uncertainty.

3. **Partially observable :** As the thing in the world are not completely observable, therefore planning is required.

Planning is also required because actions:

(i) take time to reach in final state

(ii) have continuous effects.

Situation Calculus :

→ Situation calculus means the formalization of actions in first -order logic (McCarthy & Hayes, 1969). Search is performed by logical inference in AI.

→ In situation calculus, Situations denotes the logical terms denoting states of the world.

→ In situation calculus Actions and facts are represented as logical terms called fluents.e.g. puton (A, B): represented the action. The action of putting block A on block B.

→ On (A,B) : represents the proposition i.e. The proposition that block A is on block B.

→ in situation calculus Propositional fluents are asserted to be true in a particular state by using the predicate : holds

holds (on(A, B), s) : A is on B in situation s

→ in situation calculus Situations resulting from performing an action in another situation are represented using the function : result

result (puton (A, B), s) : The situation resulting from putting A on B in situation s.

→ in situation calculus Axioms are used to represent preconditions and effects of actions.

$\forall s \, \forall x \, \forall y$ [holds (clear(x),s) $\wedge$holds (clear(y),s)

→ holds (on(x, y), result(puton(x, y),s))]

However, in situation calculus, state that does not change must be explicity stated when an action is performed.

"$\forall s \, \forall x \, \forall y \, \forall c$[holds (color (x,c),s) →

holds (color (x, c), result (puton(x, y), s))]

→ in situation calculus other logic such as temporal and modal logics have also been developed for reasoning.

Q.7.(a) Define fuzzy reasoning. What are the various operations on fuzzy sets? (10)

Ans. Fuzzy reasoning is the single rule with single antecedent.

Rule: if x is A then y is B

Fact: x is A'

Conclusion: y is B'

Fuzzy Set Operations :

Union : The membership function of the Union of two fuzzy sets A and B with membership function μ_A and μ_B respectively is defined as the maximum of the two individual membership functions. This is called the maximum criterion.

$$\mu_A \cup E = \max(\mu_A, \mu_B)$$

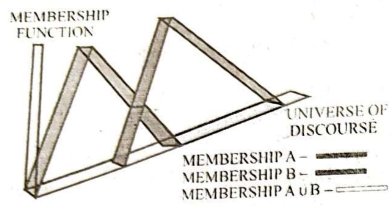


Fig.(1) : Membership function for Union.

The Union operation in Fuzzy set theory is the equivalent of the OR operation in Boolean algebra.

Intersection : The membership function of the Intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

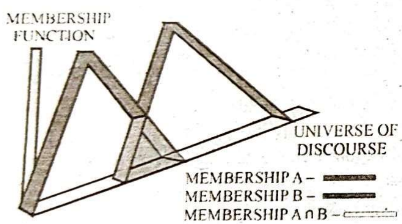


Fig.(2) : Membership function for Intersection.

The Intersection operation in Fuzzy set throry is the equivalent of the AND operation in Boolean algebra.

Complement : The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as the negation of the specified membership function. This is caleed the negation criterion.

$$\mu_{\bar{A}} = 1 - \mu_A$$

The Complement operation in Fuzzy set theory is the equivalent of the NOT operation in Boolean algebra.

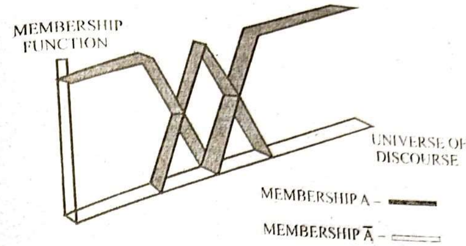


Fig.(3) : Membership function for Complement.

The following rules which are common in classical set theory also apply to Fuzzy set theory.

De Morgans Law : $\overline{(A \cap B)} = \bar{A} \cap \bar{B}, \overline{(A \cap B)} = \bar{A} \cap \bar{B}$

Associativity : $(A \cap B) \cap C = A \cap (B \cap C)$

$(A \cup B) \cup C = A \cup (B \cup C)$

Commutativity : $A \cap B = B \cap A, A \cup B = B \cup A$

Distributivity : $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Q.7.(b) Explain Temporal reasoning with detail. **(10)**

Ans. Temporal reasoning : The central component of any knowledge representation that supports Natural Language is the treatment of verbs and time.

Classification:

Types of Time

(i) Instantaneous point assignment with some transition in the world

- e.g. light turning on, someone finding a pen

(ii) Extended stretch over which some event occurs

- e.g. "John drove his car to work at 5pm."

(iii) All intervals have durations

- e.g. five minutes long
- Points cannot have durations

Parsing Text for TemporalExpressions

(i) Markers for Time

Noun/Noun Phrase/Proper Noun: "day", "Friday night", "Wednesday"

(ii) Prepositional Phrase: "in a week"

(iii) Adjective: "current", "future"

(iv) Adverb: "recently", "hourly"

(v) Adjective/Adverb Phrase: "two weeks ago", "nearly half an hour ago"

(vi) Number: 3 (as in "He arrived at 3.")

(vii) Subordinate Clauses: "…when the market stabilized"

Examples of Current Methods

- Logics
- Interval-based Temporal Logic
- Tense
- TIMEX2
- TimeML
- DAML Ontology of time

Interval-based Temporal Logic

- Only based on intervals.
- 13 basic binary relations between time intervals: before, after, overlaps, overlapped by, starts, started by, finishes, finished by, during, contains, meets, met by, equal to
- Incomplete temporal information common innatural-language is captured by a disjunction of several of these relations.

Section – D

Q.8. What is an expert system ? Describe the architecture of expert system with various components. **(20)**

Ans. Expert System : An expert system is software that attempts to provide an answer to a problem, or clarify uncertainties where normally one or more human experts would need to be consulted.

An expert system compared with traditional computer.

Inference engine + Knowledge = Expert system

(Algorithm + Data structures = Program in traditional computer)

MYCIN and DENDRAL are two expert system in history.

Some of the important advantages of expert systems are as follows :

(a) Ability to capture and preserve irreplaceable human experience;

(b) Ability to develop a system more consistent than human experts;

(c) Minimize human experts needed at a number of locations at the same time (especially in a hostile environment that is dangerous to human health);

(d) Solutions can be developed faster than human experts.

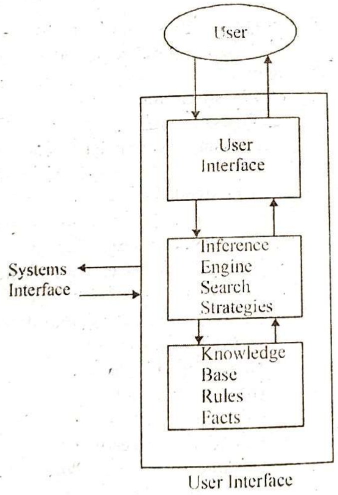


Fig.(a)

Architecture of an Expert System :

(1) *Knowledge Base :* The knowledge base stores all relevant information, data, rules, cases, and

relationships used by the expert system, i.e. the knowledge base of expert systems contains both factual and heuristic knowledge.

Factual knowledge is that knowledge of the task domain that is wideloy shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less accurate more experiential more judgemental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is knowledge of good practice, good judgement, and believable reasoning in the field. It is the knowledge that under lies the "art of good guessining".

A knowledge base can combine the knowledge of multiple human experts. A rule is a conditional statement that links given conditions to actions or outcomes.

A frame is another approach used to capture and store knowledge in a knowledge base. It relates an object or item to various facts or values. A frame-based representation is ideally suited for object-oriented programming techniques. Expert systems making use of frames to store knowledge are also called frame-based expert systems.

(2) *Inference Engine :* The purpose of the inference engine is to seek information and relationship from the knowledge base and to provide answers, predictions, and suggestions in the way a human expert would. The inference engine must find the right facts, interpretations, and rules and assemble them correctly. Two types of inference methods are commonly used-Backward chaining and forward chaining.

(a) *Forward chaining(Data-Driven) :* Forward chaining starts with the facts and works forward to the conclusions.

(b) *Backward Chaining(Goal-Driven) :* Backward chaining is the process of starting with conclusions and working backward to the supporting facts.

Q.9. Briefly explain : **(20)**

(a) Neural Networks

(b) Natural language processing

(c) Genetic algorithms

Ans. (a) Neural Network : Neural network is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.

Neural Networks are identified by different names such as

(i) Artificial Neural Network

(ii) Connectionists Network.

(iii) Parallel distributed processing systems.

Types : Artificial neural network types vary from those with only one or two layers of single direction logic, to complicated multi–input many directional feedback loops and layers. On the whole, these systems use algorithms in their programming to determine control and organization of their functions. Some may be as simple as a one-neuron layer with an input and an output, and others can mimic complex systems such as dANN, which can mimic chromosomal DNA through sizes at the cellular level, into artificial organisms and simulate reproduction, mutation and population sizes. Most systems use "weights" to change the parameters of the throughput and the varying connections to the neurons. Artificial neural networks can be autonomous and learn by input from outside "teachers" or even self-teaching from written-in rules.

Advantages :

(1) *Adaptive learning* : An ability to learn how to do tasks based on the data given for training or initial experience.

(2) *Self-Organization* : An ANN can create its own organization or representation of the information it receives during learning time.

(3) *Real Time Operation* : ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

(4) *Fault Tolerance via Redundant Information Coding* : Paratial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

(5) *Flexibility* : Artificial neural networks have the ability to generalize and learn. They acquire knowledge from their surroundings by adapting to internal and external parameters. The network learns from examples and adapts to situations based on its findings. It generalizes knowledge to produce adequate responses to unknown situations. Artificial neural networks solve complex problems that are difficult to manage by approximation.

(6) *Non linearity* : A computational neuron can produce a linear or a non-linear answer. A non-linear artificial network is made by the interconnection of non-linear neurons. Non-linear systems have inputs that are not proportional to the outputs. This function allows the network to efficiently acquire knowledge through learning. This is a distinct advantage over a traditionally linear network that is inadequate when it comes to modeling non-linear data.

Ans.(b)Natural language processing : Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.

The analysis of Natural Language is broken into various broad levels such as morphological, lexical, syntactic, semantic, pragmatic and discourse analysis.

(1) *Morphological* : Morphology deals with the componential nature of word, which are composed of morphemes the smallest units of meaning.

For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root registra. and the suffix tion.

Since the meaning of each morpheme remains the same across words, unknown word can be broken down into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix - ed to a verb, conveys that the action of the verb took place in the past.

(2) *Lexical analysis* : The aim is to divide the text into paragraphs, sentences and words. The lexical analysis cannot be performed in isolation from morphological and syntactic analysis.

In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur.

A single lexical unit is decomposed into its more basic properties. Given that there is a set of semantic primitives used across all words, these simplified lexical representations make it possible to unify meaning across words and to produce complex interpretations much the same as humans do.

The lexical level may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon.

(3) *Syntactic Analysis* : This level focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence. The output of this level of processing is the representation of the sentence that reveals the structural dependency relationships between the words.

For example consider two sentences : 'The dog chased the cat.' and 'The cat chased the dog.' These 2 sentences differ only in terms of syntax, yet convey different meanings.

Some word sequences may be rejected if they violet the rules of the language for how words may be combined. For example, the cat the dog chased.

(4) *Semantic Analysis* : Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between the syntactic structures and the objects in the task domain. The structures for which no such mapping is possible are rejected.

For example, the sentence "colorless green ideas...." would be rejected because colorless and green make no sense although the sentence is correct. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence.

(5) *Discourse Integration* : The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.

Example : The word "it" in the sentence, "you wanted it" depends in the prior discourse context.

(6) *Pragmatic* : This is high-level-knowledge which relates the use of sentences in different contexts and how the contexts affects the meaning of the sentence.

The approaches taken in developing language understanding programs generally follow the above levels or stages. When a string of words has been detected, sentences are parsed or analyzed to determine their structure and grammatical correctness. Semantics of the structure are determined and appropriate representation structures created for inferencing the programs. The whole process is a series or transformations from the basic speech sounds to a complete set of initial representation structures.

Ans.(c)Genetic algorithm : Genetic algorithms (GAs) are the main paradigm of evolutionary computing. GAs are inspired by Darwin's theory about evolution – the 'survival of the fittest'. In nature cometition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

Genetic operators are analogous to those which occur in the natural world : Reproduction (or selection)

Crossover (or recombination); and Mutation.

The cycle of a Genetic Algorithm is presented in the figure :

Each cycle in Genetic Algorithms produces a new generation of possible solutions for a given problem. In the first phase, an initial population, describing representatives of the potential solution, is created to initiate the search process. The elements of the population are encoded into bit-string, called chromosomes. The performance of the strings, often called fitness, is then evaluated with the help of some functions, representing the constrains of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. It should be noted that the selection process is mainly responsible for assuring survival of the best-fit individuals. After selection of the population strings is over, the genetic manipulation process is carried out. In the first step, the crossover operation that recombines the bits (genes) of each two selected strings (chromosomes) is executed.

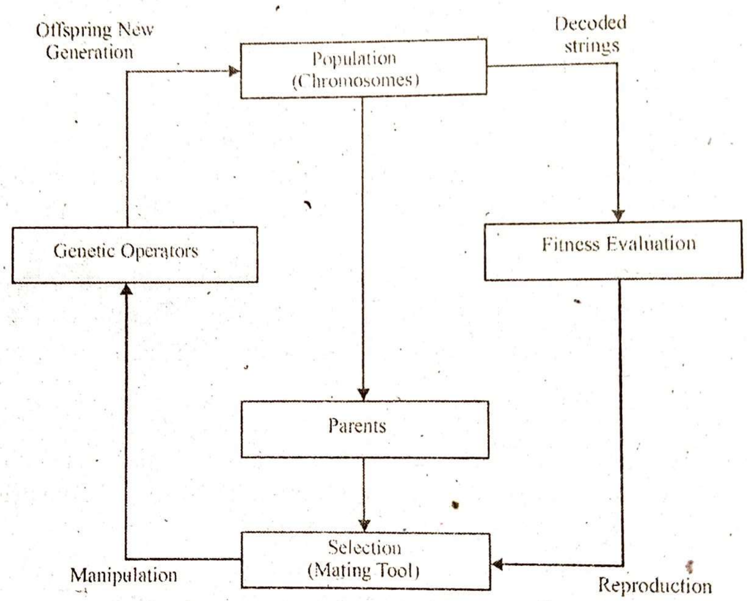


Fig.

INTELLIGENT SYSTEMS

May - 2019

Paper Code:-CSE-304-F

Note : ***Attempt five questions in all, selecting one question from each Section. Question No. 1 is compulsory. All questions carry equal marks.***

Q.1. Explain the following : **(20)**

(a) Describe various AI applications to robotics.

(b) What is knowledge base ? How it is different from Database ?

(c) Explain the concept of Knowledge Acquisition.

(d) Learning by Analogy.

Ans.(a) AI Applications to Robotics are as follows :

(1) *The Intelligent Home* : Automated systems can now monitor home security, environmental conditions and energy usage. Door and windows can be opened automatically and appliances such as lighting and air conditioning can be pre programmed to activate. This assists occupants irrespective of their state of mobility. Cye and the Carebot will vacuum the house, even if you're not at home!

(2) *Exploration* : Robots can visit environments that are harmful to humans. An example is monitoring the environment inside a volcano or exploring our deepest oceans. NASA has used robotic probes for planetary exploration since the early sixties.

(3) *Military Robots* : Airborne robot drones are used for surveillance in today's modern army. In the future automated aircraft and vehicles could be used to carry fuel and ammunition or clear minefields.

(4) *Farms* : Automated harvesters can cut and gather crops. Robotic dairies are available allowing operators to feed and milk their cows remotely.

(5) *The Car Industry* : Robotic arms that are able to perform multiple tasks are used in the car manufacturing process. They perform tasks such as welding, cutting, lifting, sorting and bending. Similar applications but on smaller scale are now being planned for the food processing industry in particular the trimming, cutting and processing of various meats such as fish, lamb, beef.

(6) *Hospitals* : Under development is a robotic suit that will enable nurses to lift patients without damaging their backs. Scientists in Japan have developed a power-assisted suit which will give nurses the extra muscle they need to lift their patients - and avoid back injuries.

(7) *Disaster Areas* : Surveillance robots fitted with advanced sensing and imaging equipment can operate in hazardous environments such as urban setting damaged by earthquakes by scanning walls, floors and ceilings for structural integrity.

(8) *Entertainment* : Interactive robots that exhibit behaviours and learning ability. SONY has one such robot which moves freely, plays with a ball and can respond to verbal instructions.

(9) *Education* : Education is integrating technologies in a creative format and robotics involves all key learning areas such as maths, arts (i.e. materials and design), English, sciences (i.e. chemistry, physics, mechanics, electronics) and social skills.

Ans.(b)Knowledge base : A knowledge base is a technology used to store complex structured and unstructured information used by a computer system.

Database are typically used for storing sets of related data such as accounting records, questions on yahoo answers, etc.

A Knowledge base is used in AI. When an AI algorithm tries to make a decision, it queries its knowledge base to determine how to act. Depending on the result it can then update the knowledge base. It is part of machine learning.

Difference between knowledge base and database are as follows :

Knowledge Base (KB)	Database
1. Dual knowledge base can be used to store cause and effect information and imprecise.	1. Database store data that is simple and flat (relational).
2. Knowledge may consists of fuzzy facts. There is a mechanism which manipulates uncertain information.	2. Database does not consist of fuzzy fact.
3. Knowledge base contains more sophisticated relationships between facts.	3. Database do not contains more sophisticated relationships between facts.
4. Knowledge bases emulate the decision making processess of humans.	4. Database do not emulate decision factual and processes.
5. Knowledge base of expert systems contain both factual and heuristic knowledge.	5. Database do not contain factual and heuristic knowledge.
6. Knowledge base capture and distribute knowledge.	6. Database systems do not capture and distribute knowledge.
7. Knowledge bases are dependable.	7. Database systems are not dependable.
8. Knowledge bases are accurate and consistent.	8. Database systems are sometimes not consistent.
9. Knowledge bases are profitable.	9. Database systems are sometimes not profitable.
10. An expert knowledge of the domain in needed for updating knowledge base.	10. A database expert in needed to update data basis.

Ans.(c)Knowledge Acquistion : Knowledge acquistion means one can learn by experience and by storing the experience in a knowledge base. One basic example of this type is rote learning.

The domain expertise that needs to be transferred to an expert system is a collection of definitions, relations, specialized facts, procedures and assumptions. The transfer of the knowledge from some knowledge source to a computer system is called knowledge acquistion. To acquire knowledge from human experts is known as knowledge engineering. And to extract the human expert's knowledge via interviews or tools is called knowledge elicitation. The three models of knowledge acquistions defined by Bunchanan and Shortliffe are :

(a) *Handcrafting* : Handcrafting means Code knowledge is converted into program directly.

(b) *Knowledge Engineering* : Knowledge engineering means working with an expert system to organize his/her knowledge in a suitable form an expert system to use.

(c) *Machine learning* : Machine learning means to Extract the knowledge from training examples.

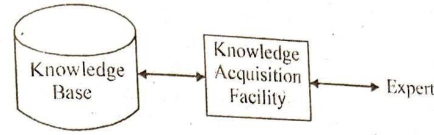


Fig. : Knowledge Acquisition Facility

Knowledge acquisition can be divided into following five stages :

(i) *Identification* : Identification means to Define an appropriate problem and determine the characteristics.

(ii) *Conceptualization* : Conceptualization means to Find the concepts (objects, relations, information etc.) to represent the knowledge.

(iii) *Formalization* : Formalization means to choose a knowledge representation method and an inference mechanism.

(iv) *Implementation* : Implementation means to formulate knowledge in the chosen formalism (rules, frames etc.)

(v) *Testing* : Testing means to verify the knowledge and validate the system.

Ans.(d)Learning By Analogy : Analogy is a powerful inference tool. It involves a complicated mapping between what might appear to be two dissimilar concepts.

Bill was built like a large outdoor brick lavatory.

He was like putty in her hands

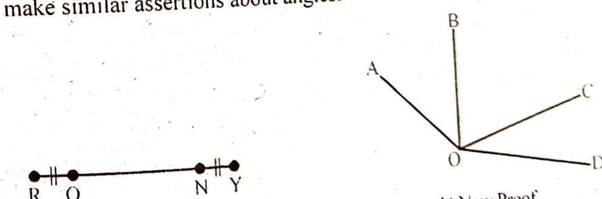
Human quickly recognise the abstractions involved and understand the meaning.

There are two methods of analogical problem method studied in AI.

Transformational Analogy : Transformational Analogy Looks for a similar solution and copy it to the new situation making suitable substitutions where appropriate.

For example, Geometry

If you know about lengths of line segments and a proof that certain lines are equal (Fig.) then we can make similar assertions about angles.



(a) Old Proof (b) New Proof

Fig. : Translational Analogy Example

- We know that lines RO = NY and angles AOB = COD
- We have seen that RO + ON = ON + NY – additive rule.
- So we can say that angles AOB + BOC = BOC + COD
- So by a transitive rule line RN = OY

- So similarly angle AOC = BOD

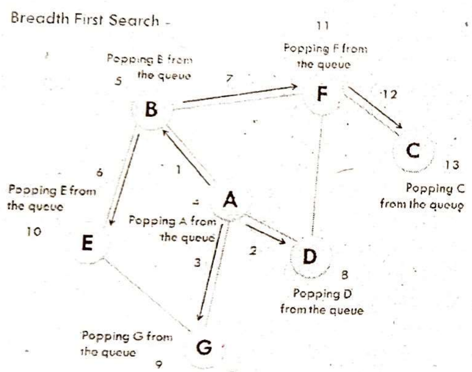
Carbonell (1983) describes a T-space method to transform old solutions into new ones.

- Whole solutions are viewed as states in a problem space – the T-space.
- T-operators prescribe methods of transforming existing solution states into new ones.
- Reasoning by analogy becomes a search in T-space : starting with an old solution, use means-end analysis or some method to find a solution to the current problem.

Section – A

Q.2.(a) What is the difference between Breath first search and Depth first search ? Explain with the help of example. (10)

Ans. Breadth First Search (BFS) is the traversing method used in graphs. It uses a queue for storing the visited vertices. In this method the emphasize is on the vertices of the graph, one vertex is selected at first then it is visited and marked. The vertices adjacent to the visited vertex are then visited and stored in the queue sequentially. Similarly, the stored vertices are then treated one by one, and their adjacent vertices are visited. A node is fully explored before visiting any other node in the graph, in other words, it traverses shallowest unexplored nodes first.



Output - A, B, D, G, E, F, C.

Example : We have a graph whose vertices are A, B, C, D, E, F, G. Considering A as starting point. The steps involved in the process are:

· Vertex A is expanded and stored in the queue.

· Vertices B, D and G successors of A, are expanded and stored in the queue meanwhile Vertex A removed.

· Now B at the front end of the queue is removed along with storing its successor vertices E and F.

· Vertex D is at the front end of the queue is removed, and its connected node F is already visited.

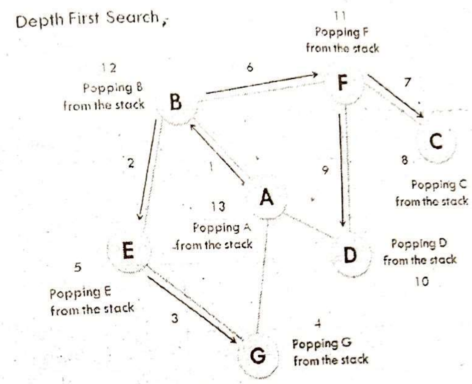
· Vertex G is removed from the queue, and it has successor E which is already visited.

· Now E and F are removed from the queue, and its successor vertex C is traversed and stored in the queue.

· At last C is also removed and the queue is empty which means we are done.

· The generated Output is – A, B, D, G, E, F, C.

Definition of DFS : Depth First Search (DFS) traversing method uses the stack for storing the visited vertices. DFS is the edge based method and works in the recursive fashion where the vertices are explored along a path (edge). The exploration of a node is suspended as soon as another unexplored node is found and the deepest unexplored nodes are traversed at foremost. DFS traverse/visit each vertex exactly once and each edge is inspected exactly twice.



Output - A, B, E, G, F, C, D.

Example : Similar to BFS lets take the same graph for performing DFS operations, and the involved steps are:

· Considering A as the starting vertex which is explored and stored in the stack.

· B successor vertex of A is stored in the stack.

· Vertex B have two successors E and F, among them alphabetically E is explored first and stored in the stack.

· The successor of vertex E, i.e., G is stored in the stack.

· Vertex G have two connected vertices, and both are already visited, so G is popped out from the stack.

· Similarly, E s also removed.

· Now, vertex B is at the top of the stack, its another node(vertex) F is explored and stored in the stack.

· Vertex F has two successor C and D, between them C is traversed first and stored in the stack.

· Vertex C only have one predecessor which is already visited, so it is removed from the stack.

· Now vertex D connected to F is visited and stored in the stack.

· As vertex D doesn't have any unvisited nodes, therefore D is removed.

· Similarly, F, B and A are also popped.

· The generated output is – A, B, E, G, F, C, D.

Key Differences Between BFS and DFS

1. BFS is vertex-based algorithm while DFS is an edge-based algorithm.
2. Queue data structure is used in BFS. On the other hand, DFS uses stack or recursion.
3. Memory space is efficiently utilized in DFS while space utilization in BFS is not effective.
4. BFS is optimal algorithm while DFS is not optimal.
5. DFS constructs narrow and long trees. As against, BFS constructs wide and short tree.

Comparison Chart :

Basis For Comparison	BFS	DFS
Basic	Vertex-based algorithm	Edge-based algorithm
Data structure used to store the nodes	Queue	Stack
Memory consumption	Inefficient	Efficient
Structure of the constructed tree	Wide and short	Narrow and long
Traversing fashion	Oldest unvisited vertices are explored at first.	Vertices along the edge are explored in the beginning.
Optimality	Optimal for finding the shortest distance, not in cost.	not optimal
Application	Examines bipartite graph, connected component and shortest path present in a graph.	Examines two-edge connected graph, strongly connected graph, acyclic graph and topological order.

Q.2.(b) Explain various problems and their solutions in hill climbing algorithm. **(20)**

Ans. Problems in Hill Climbing : Both basic & steepest ascent hill climbing may fail to find a solution. Either algorithm may terminate not by finding a goal state but by getting to a state from which no better states can be generated. This will happen if the program has reached either of the following states :

(i) Local Maximum
(ii) Plateau
(iii) Ridge

(i) Local Maximum : A state that is better than all its neighbours but not so when compared to states that are father away. Local maximum are particularly frustrating because they often occur almost within sight of a solution. In this case, they are called foot-hills.

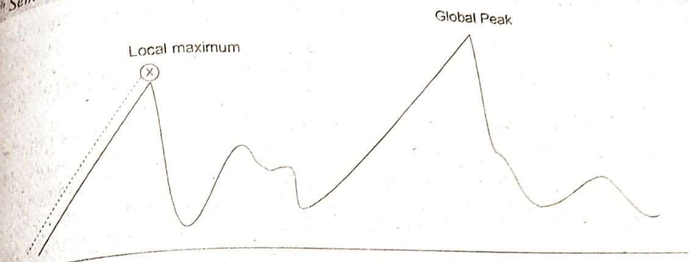


Fig. (a)

(ii) Plateau : A flat area of the search space in which all neighbours have the same value. On a plateau, it is not possible to determine the best direction in which to move by making loacal comparisons.

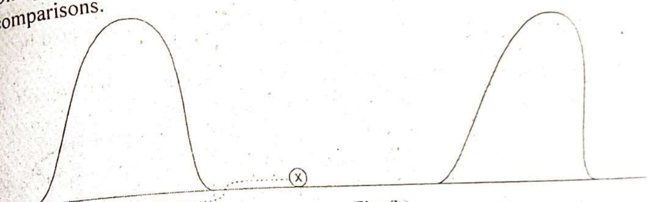


Fig. (b)

(iii) Ridge : A ridge is a special kind of local maximum. It is an area of search space that is higher than the surrounding areas & that iftself has a slope. But the orientation of the high region makes it impossible to transverse a ridge by single moves.

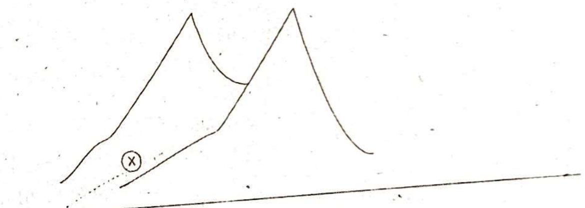


Fig. (c)

Solutions of Hill Climbing Problems :

(i) Back tracking to some some earlier node & try a different direction. This is a good way of dealing with local maximum.

(ii) Make a big jump in some direction to a new search. This is good way of dealing with plateaus.

(iii) Apply two or more rules before doing the test. This corresponds to moving in several directions at once. This is a particularly good strategy for dealing with ridges.

Hill climbing becomes inefficient in large problem spaces & when combinational explosion occurs. But it is very useful when combined with other methods.

Q.3.(a) What is best first search ? Explain A* algorithm with the help of example. (10)

Ans. Best First Search : Best first search is a little like hill climbing, in that it uses an evaluation function and always chooses the next node to be that with the best score. However, it is exhaustive, in that it should eventually try all possible paths. It uses an agenda as in breadth/depth first search, but instead of taking the first node off the agenda (and generating its successors) it will take the best node off. *i.e.* the node with the best score. The successors of the best node will be evaluated (*i.e.* have a score assigned to them) and added to the list. To implement the algorithm two types of lists.

Open : Nodes that have been generated, but have not examined. This is organized as a priority queue.

Closed : Nodes that have already been examined.

Whenever a new node is generated, check whether it has been generated before.

The A* Algorithm : In its simplest form best first search is useful, but doesn't take into account the cost of the paths when choosing which node to search from next. So, a solution may be find out but it may be not a very good solution.

There is a variant of best first search known as A* which attempts to find a solution which minimizes the total length or cost of the solution path. It combines advantanges of breadth first search, where the shortest path is found first, with advantages of best first search, where the node that is guesssed closest to the solution is explored next.

In the A* algorithm the score which is assigned to a node is a combination of the cost of the path and the estimated cost to solution. This is normally expressed as an evaluation function $f(n)$, which involves the sum of the values returned by two functions g(n) and h(n), i.e.

$$f(n) = g(n) + h(n) \quad \text{(n represents node)}$$

where h(n) = cost of the cheapest path from node n to a goal state.

g(n) = cost of the cheapest path from the initial state to node n.

Or

$$f^*(n) = g^*(n) + h^*(n)$$

h*(n) (heuristic factor) = estimate of h(n).

g*(n) (depth factor) = approximation of g(n) found by A* so far.

The A* algorithm then looks the same as the simple best first algorithm, slightly more complex evaluation function is used here.

Algorithm is as follows :

1. Start with OPEN holding the initial nodes.
2. Pick the BEST node on OPEN such that Or $f^*(n) = g^*(n) + h^*(n)$ is minimal.
3. If BEST is goal node quit and return the path from initial to BEST Otherwise.
4. Remove BEST from OPEN and all of BESTs children, labelling each with its path from initial node.

Yes A* algorithm is guaranteed to find an optimal goal path if one exits.

Q.3.(b) What is alpha and beta pruning ? Explain with example. (10)

Ans. Alpha-Beta Pruning : In MINIMAX search, number of game state increases exponentially. To reduce search, the pruning is done. Alpha-beta is one such pruning technique. It maintains two threshold values one is called alpha (or 'α') and other beta ('β').

These threshold values are defined as follows:

α = Lower bound on maximum value of utility function. It is the best acceptable value of utility function in maximizing ply.

β = Upper bound on minimum value of utility function. It is the best highest acceptable value of utility function in case minimizing ply.

In searching the game tree in mini max search, the part of tree having utility value less than α indicates that this particular move will not at all be useful. Hence part of tree having utility value less than alpha will be pruned. That means, in future all the nodes below it will never be explored. Similarly, if utility value comes out to be more than beta in case of minimizing ply, it will indicate that this move is at all useful and that sub free will be pruned on similar ground. Let us consider an example:

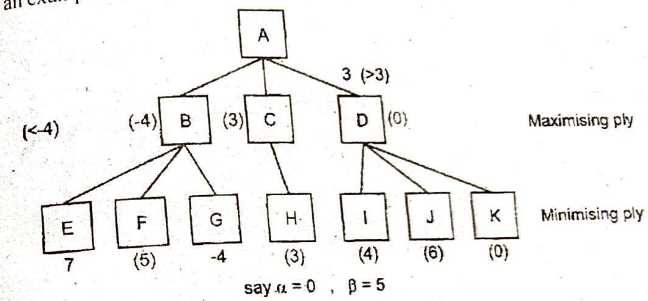


Fig. : α–β pruning

Here the possibilites are considered up to 2 ply. At min ply, the best value from three nodes is – 4, 3, 0. These will be back propagated towards root and a maximizing move 3 will be taken. Now node E having utility value 7 is far more, then accepted (as it is minimizing ply). So further node E will not be explored. In the situation when more plies are considered, whole sub tree below E will be pruned.

Similarly if alpha = 0 and beta = 5, all notes and related sub trees having value of unitly function less than 0 at maximizing ply and more than 5 at minimizing ply will be pruned. The procedure of minimax search is as follows.

Procedure MINIMAX *A*, *B* (Position)

{Procedure Minimax (Position, Depth, Player)

1. If reached last ply (Position, Depth)

then return the structure VALUE = STATIC (Position, Player)

2a. Path = nil

2b. Otherwise, generate one more ply of the tree and set SUCCESSOR to the list, it returns.

3. If SUCCESSOR, is empty, then return the same structure.
4. Otherwise, examine each element and find best one.
5. After examining all the nodes, return the structure
VALUE = BEST-SCORE
PATH = BEST-PATH}

The worthiness of α, β pruning depends upon the order in which paths are examined. If the worst successor is generated first then no cut offs will occur. Kunth and Moore have analyzed that for a search tree of branching factor b and depth, d the α - β search needs examining only $b^{d/2}$ nodes to pick up best move, instead of b^d for mini-max. That means, effective branching factor becomes √b instead of b. For a chess game, where the value of 'b' is 35, with alpha beta pruning the effective branching factor will become 6. This indicates a significant amount of saving in search.

Section – B

Q.4.(a) Explain various knowledge representation issues in detail. (10)

Ans. The fundamental goal of knowledge representation is to facilitate inferencing (conclusions) from knowledge. The issues that arises while using knowledge representation techniques are many. Some of them are :

Important Attributes : Important attributes means are there any attributes that occur in many different types of problem?

There are two instance *is* and *isa* and each is important because each supports property inheritance.

Relationships : It means the relationship between the attributes of an object, such as, inverses, existence, techniques for reasoning about values and single valued attributes. We can consider an example of an inverse in

band (John Zorn, Naked City)

This can be treated as John Zorn plays in the band Naked City or John Zorn's band is Naked City.

Another representation is band = Naked City

band-members = John Zorn, Bill Frissell, Fred Frith, Joey Barron, ...

Granularity : Granularity means what level should the knowledge be represented and what are the primitives. Choosing the Granularity of Representation Primitives are fundamental concepts such as holding, seeing, playing and as English is a very rich language with over half a million words it is clear we will find difficulty in deciding upon which words to choose as our primitives in a series of situations.

If *ram feeds a dog* then it could become:

feeds(ram, dog)

If *ram gives the dog a bone* like:

gives (ram, dog, bone) Are these the same?

In any sense does giving an object food constitute feeding?

If *give (x, food)* → *feed (x)* then we are making progress.

Q.4.(b) Write a short note on with example : (10)

(i) Semantic Nets

(ii) Frames

(iii) Inheritance

Ans.(i) Semantic Nets : It is another form of representing the knowledge. The major idea is that :

→ The meaning of a concept comes from its relationship with other concepts.

→ The information is stored by interconnecting nodes with labelled arcs.

Representation in a Semantic Net :

The physical attributes of a person can be represented as in Fig.(a).

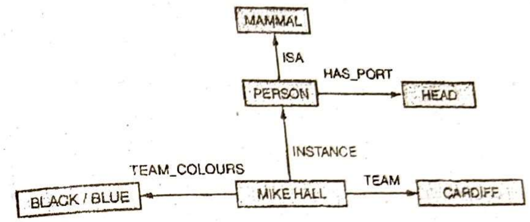


Fig (a) : A Semantic Network

These values can also be represented in logic as : is a (person, mammal), instance(Milk-Hall, person) team(Milk-Hall, Cardiff).

Features of Sementic Network :

1. Sementic nets were used was to find relationships among objects by spreading activation out from each of two nodes & seeing where the activation net. This process is called intersection search.

2. It is a natural way to represent relationships that would appear as ground instance of binary predicates in predicate logic.

3. It is particularly useful for representing the contents of a typical declarative sentence that describes several aspects of a particular events.

Sementic Networks for the facts. Every dog has bitten every mail carrier.

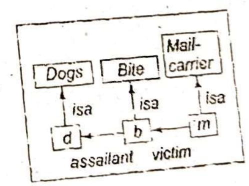


Fig (b) : Using Partitioned Semantic Nets

Ans.(ii) Frames : A frame is a collection of attributes or slots and associated values that describe some real world entity. Frames on their own are not particularly helpful but frame systems are a powerful way of encoding information to support reasoning. Set theory provides a good basis for understanding frame systems.

Each frame represents :

- a class (set), or
- an instance (an element of a class).

Flexibility in Frames : Slots in a frame can contain

- Information for choosing a frame in a situation
- Relatinoships between this and other frames
- Procedures to carry out after various slots filled
- Default information to use where input is missing
- Blank slots: left blank unless required for a task
- Other frames, which gives a hierarchy

Frames can also be expressed in first order logic.

Example : Frame Representation

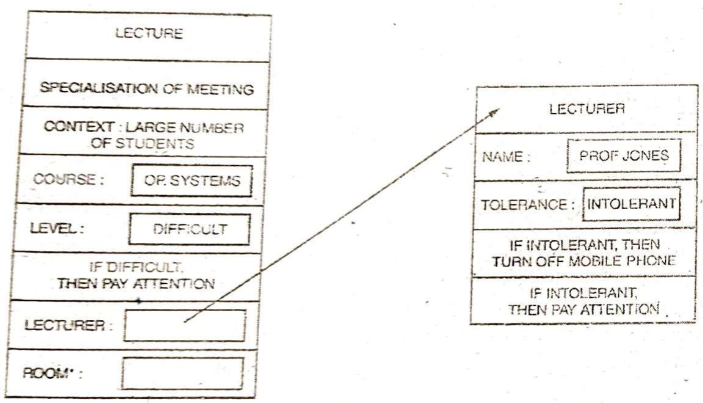


Fig.

Ans.(iii) Inheritance : The "*is a*" and "*instance*" representation provides a mechanism to implement this.

Inheritance also provides a means of dealing with *default reasoning*. For example, we could represent :

- Emus are birds..
- Typicaly birds fly and have wings.
- Emus run.

In the following Semantic net :

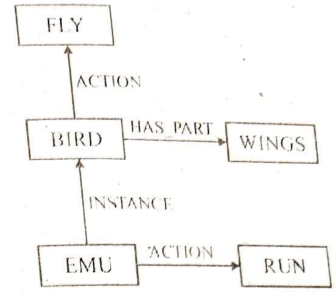


Fig.(1) : A Semantic Network for a Default Reasoning

In making certain inferences we will also need to distinguish between the link that defines a new entity and holds its value and the other kind of link that relates two existing entities. Consider the example shown where the height of two people is depicted and we also wish to compare them.

We need extra nodes for the concept as well as its value.

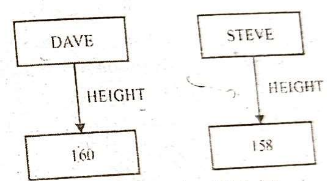


Fig.(2) : Two Height

Special procedures are needed to process these nodes, but without this distinction the analysis would be very limited.

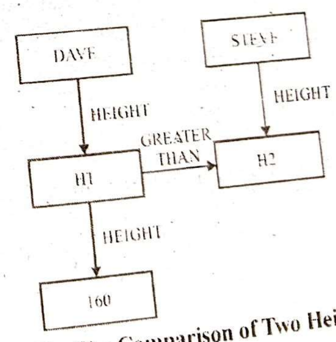


Fig.(3) : Comparison of Two Height

Q.5. What is reasoning under uncertainty ? How Baye's probabilistic interferences theory is helpful for solving probability based problems. (20)

Ans. Reasoning under uncertainty : Unfortunately the world is an uncertain place. Uncertainty can arise from a variety of sources. The infromation available about one thing may be incomplete or volatile or may be some important facts and details about the problem could be missing. Or the facts available may be vague or fuzzy. Despite of all these problems, human beings can deal with uncertainities on a daily basis and can arrive at some reasonable solutions.

Any AI system that seeks to model and reasoning in such a world must be able to deal with this.

In particular it must be able to deal with :

(i) Incompleteness — compensate for lack of knowledge.

(ii) Inconsistencies — resolve ambiguities and contradictions.

(iii) Change — it must be able to update its world knowledge base over time. Clearly in order to deal with this, some decisions that are made are more likely to be true (or false) than others and we must introduce methods that can cope with this uncertainty.

There are three basic methods that can deal with uncertainty are :

- Symbolic methods.
- Statistical methods.
- Fuzzy logic methods.

Bayesian Theorem : Bayes' theorem shows the relation between two conditional probabilities which are the reverse of each other. This theorem is named for Thomas Bayes and often called Bayes' law or Bayes' rule. Bayes' theorem expresses the conditional probability, or "posterior probability", of a hypothesis H (*i.e.* its probability after evidence E is observed) in terms of the "prior probability" of H, the prior probability of E, and the conditional probability of E given H. It implies that evidence has a stronger confirming effect if it is more unlikely before being observed.

This states: $$P(H_i \backslash E) = \frac{P(E \backslash H_i)P(H_i)}{\sum_{k=1}^{n} P(E \backslash H_k)P(H_k)}$$

This statement/equation can be read as that given some evidence E then probability that hypothesis H_i is true is equal to the ratio of the probability that E will be true given H_i times the a priori evidence on the probability of H_i and and the sum of the probability of E over the set of all hypotheses times the probability of these hypotheses.

Section – C

Q.6.(a) Define fuzzy reasoning. What are the various operations on fuzzy sets ? (10)

Ans. Fuzzy reasoning is the single rule with single antecedent.

Rule: if x is A then y is B

Fact: x is A'

Conclusion: y is B'

Fuzzy Set Operations :

Union : The membership function of the Union of two fuzzy sets A and B with membership function μ_A and μ_B respectively is defined as the maximum of the two individual membership functions. This is called the maximum criterion.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

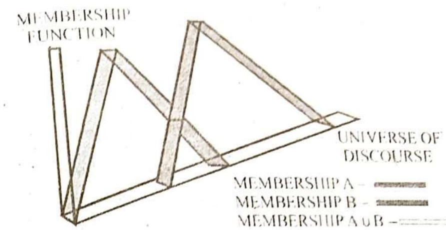


Fig.(1) : Membership function for Union.

The Union operation in Fuzzy set theory is the equivalent of the OR operation in Boolean algebra.

Intersection : The membership function of the Intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

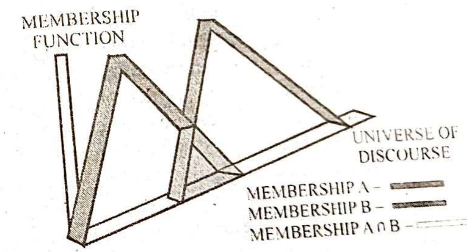


Fig.(2) : Membership function for Intersection.

The Intersection operation in Fuzzy set throry is the equivalent of the AND operation in Boolean algebra.

Complement : The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as the negation of the specified membership function. This is caleed the negation criterion.

$$\mu_{\bar{A}} = 1 - \mu_A$$

The Complement operation in Fuzzy set theory is the equivalent of the NOT operation in Boolean algebra.

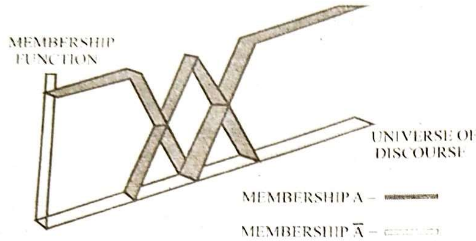


Fig.(3) : Membership function for Complement.

The following rules which are common in classical set theory also apply to Fuzzy set theory.

De Morgans Law : $\overline{(A\cap B)}=\overline{A}\cap\overline{B}.\overline{(A\cap B)}=\overline{A}\cap\overline{B}$

Associativity : $(A\cap B)\cap C=A\cap(B\cap C)$

$(A\cup B)\cup C=A\cup(B\cup C)$

Commutativity : $A\cap B=B\cap A.A\cup B=B\cup A$

Distributivity : $A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$

$A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$

Q.6.(b) Explain frame problem in planning. **(10)**

Ans. Frame Problem : In the confined world of a robot, surroundings are not static. Many varying forces or actions can cause changes or modifications to it. The problem of forcing a robot to adapt to these changes is the basis of the frame problem in artificial intelligence. Information in the knowledge base and the robot's conclusions combine to form the input for what the robot's subsequent action should be. A good selection from its facts can be made by discarding or ignoring irrelevant facts and ridding of results that could have negative side effects. (Dennett).

A robot must introduce facts that are relevant to a particular moment. That is, a robot will examine its current situation, and then look up the facts that will be beneficial to choosing its subsequent action. The robot should also search for any changeable facts. It then examines these facts to determine if any of them have been changed during a previous examination. There are two basic types of change :

- **Relevant Change :** inspect the changes made by an action
- **Irrelevant Change :** do not inspect facts that are not related to the task at hand.

Facts may be examined utilizing two levels :

- **Semantic Level :** This level interprets what kind of information is being examined. Solutions should become obvious by the assumptions of how an object should behave. There are believers in a purely semantic approach who believe that correct information can be reached via meaning. However, this hypothesis has yet to be proven.

- **Syntactic Level :** This level simply decides in which format the information should be inspected. That is, it forms solutions based on the surface and patterns of facts.

When inspecting the facts, various problems can occur :

- Sometimes an implication can be missed.
- Considering all facts and all their subsequent side effects is time-consuming.
- Some facts are unnecessarily examined when they are unneeded.
- **Problems Related to the Frame Problem :**

(1) The Qualification Problem : The Qualification problem was introduced by John McCarthy. It suggests that one is never completely positive if a specific rule will work. It also suggests that the robot does not necessarily know which rules to ignore in a given situation. Modifications in the environment can "confuse" the robot as certain rules will become obselete and new rules will be necessary before they exist.

For example,

If you turn the ignition key of your car the engine will start ...

unless the battery is dead ...

or it is out of gas ...

or there is a potato blocking the tailpipe ...

... etc.

(2) The Representational Problem : The Representational Problem is the difficulty of generating truths about the current environment. For example, how can be one program the notions of up and down? These are relative to each other, and can not be simply described buy direction. To partially rectify this problem, successor-state axioms are used. These axioms show all the true and false possibilities of a rule.

(3) The Inferential Problem : Difficulty with examining the methods by which the world is judged is the Inferential Problem. There are two kinds of purposes. The General Purpose is to inspect the entire world of things that are changeable. The Special Purpose is to only inspect actions that can modify over a small area of surroundings.

(4) The Ramification Problem : This problem describes how an action can cause deviations within its environment. For example, a robotic arm has been given the task of picking up a brick and placing it on its side in a different location. If the brick has been knocked over, what can the robot do to rectify the problem? Will it still know which side should be facing up without the ability of human sight? Should these deviations be examined individually each time an action has taken place?

(5) The Predictive Problem : The Predictive Problem deals with the benefits of predictions. That is, it is uncertain if a given prediction will cause a positive change in the environment. If the change will not be positive, "either the laws or description of the given situation must be imperfect."

Q.7.(a) Explain Non-monotonic reasoning with the help of example. **(10)**

Ans. Non-monotonic Reasoning : In monotonic logic, addition of new axioms and facts are consistent with earlier stored knowledge and facts so knowledge grows monotonically but in uncertain situations addition of new axioms may contradict with earlier and might be required to be removed from the knowledge base and the size of knowledge base grows non-monotonically. As described in the above example of shopping, new facts contradict the earlier ones and remove the same from the database. This type of reasoning is called non-monotonic reasoning. Consider following example :

Given the fact that X is an intelligent student and he scores good marks. Based on this information and the knowledge about an intelligent student which is generally true, you guess that X will be an obedient student. However, being intelligent and obedient are unrelated characteristics and obedience is a personal characteristics (i.e., an intelligent student may be in specific situation become arrogant). An intelligent student will be obedient is only a belief of listener, which may be wrong also. Hence if for specific instance of X, it is told that student is not obedient, this will contradict the general belief and to accommodate this fact in the knowledge base, the old knowledge will require modification.

Two general approaches used for nonmonotonic reasoning are :

• *Abduction :* It works on the principle of inferring some situation based on evidence is encountered, the previous evidence is removed and new one is considered, e.g. if a patient has symptoms of vomiting, as per general belief the disease is diagnosed as Diarrhoea, but after more investigation it is found that the disease actually is dehydration and hence, the earlier information would be replaced by the new one.

• *Property inheritance :* It is another form of non-monotonic reasoning. It works on the principle that a subclass will inherit the characteristics of parent class unless otherwise told. In the situations where the subclass does not inherit properties of parent class, it is specifically mentioned. For example, we know that birds can fly and Ostrich is a bird. Hence, we would infer that Ostrich can fly but it is wrong because evidence contradict that. Hence, flying, thought is inherited property of birds, special mention would be required about some birds that cannot fly.

Q.7(b) Explain the partial order planning algorithm by giving suitable example. **(10)**

Ans. Partial order planning algorithm :

```
PartialOrderPlanner(Gs)
  Inputs
        Gs: set of atomic propositions to achieve
  Output
        linear plan to achieve Gs
  Local
        Agenda: set of (P,A) pairs where P is atom and A an action
        Actions: set of actions in the current plan
        Constraints: set of temporal constraints on actions
        CausalLinks: set of (act_0,P,act_1) triples
  Agenda ←{(G,finish):G ∈ Gs}
  Actions ←{start,finish}
  Constraints ←{start<finish}
  CausalLinks ←{}
  repeat
        select and remove (G,act_1) from Agenda
        either
              choose act'_0 ∈ Actions such that act_0 achieves G
        or
              choose act_0 ∉ Actions such that act_0 achieves G
              Actions ←Actions ∪ {act_0}
              Constraints ←add_const(start<act_0,Constraints)
              for each CL ∈ CausalLinks do
                    Constraints ←protect(CL,act_0,Constraints)
              Agenda ←Agenda ∪ {(P,act_0): P is a precondition of act_0}
        Constraints ←add_const(act_0<act_1,Constraints)
        CausalLinks ∪ {(act_0,G,act_1)}
        for each A ∈ Actions do
              Constraints ←protect((act_0,G,act_1),A,Constraints)
  until Agenda={}
  return total ordering of Actions consistent with Constraints
```


Example : Consider the goal $\neg swc \wedge \neg mw$, where the initial state contains $RLoc=lab$, swc, $\neg rhc$, mw, $\neg rhm$.

Initially the agenda is ,

$$(\neg swc,finish),(\neg mw,finish).$$

Suppose $(\neg swc,finish)$ is selected and removed from the agenda. One action exists that can achieve $\neg swc$, namely deliver coffee, dc, with preconditions off and rhc. At the end of the **repeat** loop, *Agenda* contains

$$(off,dc),(rhc,dc),(\neg mw,finish).$$

Constraints is $\{start<finish,\ start < dc,\ dc <finish\}$. There is one causal link, $(dc,\neg swc,finish)$. This causal link means that no action that undoes $\neg swc$ is allowed to happen after dc and before $finish$.

Suppose $(\neg mw,finish)$ is selected from the agenda. One action exists that can achieve this, pum, with preconditions mw and $RLoc=mr$. The causal link $(pum,\neg mw,finish)$ is added to the set of causal links; (mw,pum) and (mr,pum) are added to the agenda.

Suppose (mw,pum) is selected from the agenda. The action $start$ achieves mw, because mw is true initially. The causal link $(start,mw,pum)$ is added to the set of causal links. Nothing is added to the agenda.

At this stage, there is no ordering imposed between dc and pum.

Suppose (off,dc) is removed from the agenda. There are two actions that can achieve off: mc_cs with preconditions cs, and mcc_lab with preconditions lab. The algorithm searches over these choices. Suppose it chooses mc_cs. Then the causal link (mc_cs,off,dc) is added. The first violation of a causal link occurs when a move action is used to achieve (mr,pum). This action violates the causal link (mc_cs,off,dc), and so must happen after dc (the robot goes to the mail room after delivering coffee) or before mc_cs.

Section – D

Q.8. What is an expert system ? Describe the various components of expert system in detail. **(20)**

Ans. Expert System : An expert system is software that attempts to provide an answer to a problem, or clarify uncertainties where normally one or more human experts would need to be consulted.

An expert system compared with traditional computer.

Inference engine + Knowledge = Expert system

(Algorithm + Data structures = Program in traditional computer)

MYCIN and DENDRAL are two expert system in history.

Various componets of an Expert System :

(1) *Knowledge Base :* The knowledge base stores all relevant information, data, rules, cases, and relationships used by the expert system, i.e. the knowledge base of expert systems contains both factual and heuristic knowledge.

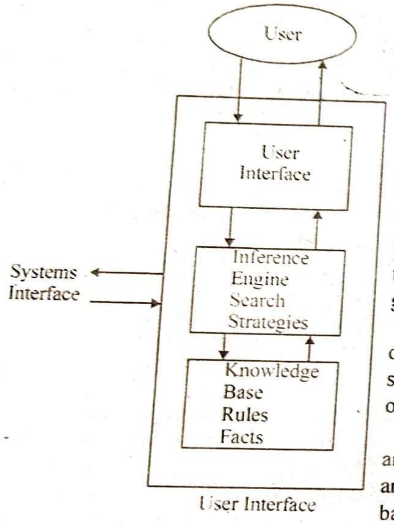


Fig.(a)

Factual knowledge is that knowledge of the task domain that is wideloy shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less accurate more experiential more judgemental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is knowledge of good practice, good judgement, and believable reasoning in the field. It is the knowledge that under lies the "art of good guessining".

A knowledge base can combine the knowledge of multiple human experts. A rule is a conditional statement that links given conditions to actions or outcomes.

A frame is another approach used to capture and store knowledge in a knowledge base. It relates an object or item to various facts or values. A frame-based representation is ideally suited for object-oriented programming techniques. Expert systems making use of frames to store knowledge are also called frame-based expert systems.

(2) *Inference Engine :* The purpose of the inference engine is to seek information and relationship from the knowledge base and to provide answers, predictions, and suggestions in the way a human expert would. The inference engine must find the right facts, interpretations, and rules and assemble them correctly. Two types of inference methods are commonly used-Backward chaining and forward chaining.

(a) *Forward chaining(Data-Driven) :* Forward chaining starts with the facts and works forward to the conclusions.

(b) *Backward Chaining(Goal-Driven) :* Backward chaining is the process of starting with conclusions and working backward to the supporting facts.

Q.9. Write short notes on the following : (20)

(a) Artificial neural networks

(b) Learning from examples

(c) Princples of Natural Language Processing (NLP)

Ans.(a) Artificial neural networks : An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve specific problems. The weights on the connections encode the knowledge of a network. Each neuron has a local memory and the output of each neuron depends upon only the input signals arriving at the neuron and value in neuron's memory. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well.

In many tasks such as recognizing human and understanding speech, current AI systems cannot do better than humans. It is conjectured that the structure of the brain is somehow suited to these tasks and not suited to these such as high speed arithmetic calculation. Rather than by being programmed an ANN can solve the problems simple by example mapping.

The intelligence of a neural network emerges from the collective behavior of neurons, each of which performs only very limited operation. Even though each individual neuron works slowly, they can still quickly find the solution by working in parallel. This fact can explain why humans can recognize a visual scene faster than a digital computer, while an individual brain cell responds much more slowly than a digital cell in VLSI circuit.

Neural Networks are identified by different names such as

(1) Artificial Neural Networks

(2) Connectionists Networks

(3) Parallel distributed processing systems.

Why Use Neural Network : Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information is has been given to analyze. Other advantages include :

(1) Adaptive learning : An ability to learn how to do tasks based on the data given for training or initial experience.

(2) Self-Organization : An ANN can create its own organization or representation of the information it receives during learning time.

(3) Real Time Operation : ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

(4) Fault Tolerance via Redundant Information Coding : Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Ans.(b) Learning From Examples :

(1) Inductive Learning : Inductive Learning involves the process of *learning by example*, where a system tries to induce a general rule from a set of observed instances.

This involves classification. Classification is the process of assigning, to a particular input, the name of a class to which it belongs. Classifications is important to many problem solving tasks.

A learning system has to be capable of evolving its own class description :

- Initial class definitions may not be adequate.
- The world may not be well understood or rapidly changing.

The task of constructing class definitions is called induction or concept learning.

The techniques used for constructing class definitions (or concept learning) are :

- Winston's learning program
- Version spaces
- Decision trees

(2) Winston's learning-Block World Learning Program : Winston describes a Blocks World Learning Program. This program operated in a simple blocks domain. The goal is to construct representation of the definition of concepts in the blocks domain.

Example : Concepts such a "house".

Start with input, a line drawing of a blocks world structure. It learned concepts House, Tent, Arch as :

Brick (rectangular block) with a wedge (riangular block) suitably placed on top of it, tent – as 2 wedges touching side by side, or an arch – as 2 non-touching bricks supporting a third wedge or brick.

The program for each concept is learned through near miss. A near miss is an object that is not an instance of the concept but a very similar to such instances.

The program uses procedures to analyze the drawing and construct a semantion net representation.

An example of such an structural for the house is shown below :

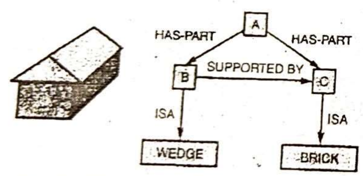


Fig.(1) : House object and semantic net.

Node A represents entire structure, which is composed of two parts :

Node B, a Wedge, and node C, a Brick.

Links in network include supported-by, has-part and isa.

There are three basic steps to the problem of concept formulation :

(1) Select one know instance of the concept. Call this the concept definition.

(2) Examine definitions of other known instance of the concept. Generalise the definition to include them.

(3) Examine descriptions of near misses. Restrict the definition to exclude these.

Both steps 2 and 3 rely on comparison and both similarities and differences need to be identified.

(3) Decision Trees : A second approach to concept learning is the induction of decision tree as exemplified by the Quinlan in his ID3 system introduced the idea of decision trees.

ID3 is a program that can build trees automatically from given positive and negative instance. Basically each leaf of a decision tree asserts a positive or negative concept. To classify a particular input we start at the top and follow assertions down until we reach as answer.

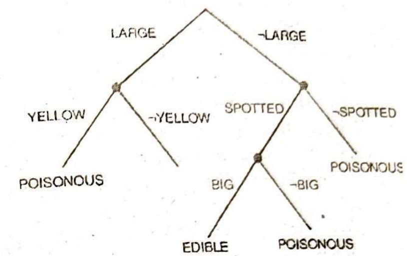


Fig.(2) : Edible Mushroom decision tree.

Ans.(c) Natural language processing : Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications.

The analysis of Natural Language is broken into various broad levels such as morphological, lexical, syntactic, semantic, pragmatic and discourse analysis.

(1) Morphological : Morphology deals with the componential nature of word, which are composed of morphemes the smallest units of meaning.

For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root registra. and the suffix tion.

Since the meaning of each morpheme remains the same across words, unknown word can be broken down into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix - ed to a verb, conveys that the action of the verb took place in the past.

(2) Lexical analysis : The aim is to divide the text into paragraphs, sentences and words. The lexical analysis cannot be performed in isolation from morphological and syntactic analysis.

In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur.

A single lexical unit is decomposed into its more basic properties. Given that there is a set of semantic primitives used across all words, these simplified lexical representations make it possible to unify meaning across words and to produce complex interpretations much the same as humans do.

The lexical level may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon.

(3) Syntactic Analysis : This level focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence. The output of this level of processing is the representation of the sentence that reveals the structural dependency relationships between the words.

For example consider two sentences : 'The dog chased the cat.' and 'The cat chased the dog.' These 2 sentences differ only in terms of syntax, yet convey different meanings.

Some word sequences may be rejected if they violet the rules of the language for how words nay be combined. For example, the cat the dog chased.

(4) Semantic Analysis : Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. The structures created by the syntactic analyzer are assigned meaning. Thus, a mapping is made between the syntactic structures and the objects in the task domain. The structures for which no such mapping is possible are rejected.

For example, the sentence "colorless green ideas...." would be rejected because colorless and green make no sense although the sentence is correct. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence.

For example, amongst other meanings, 'file' as a noun can mean either a folder for storing papers, or a tool to shape one's fingernails, or a line of individuals in a queue. If information from the rest of the sentence were required for the disambiguation, the semantic, not the lexical level, would do the disambiguation. A wide range of methods can be implemented to accomplish the disambiguation.

(5) Discourse Integration : The meaning of an individual sentence may depend on the sentences that precede it and may influence the meaning of the sentences that follow it.

Example : The word "it" in the sentence, "you wanted it" depends in the prior discourse context.

(6) Pragmatic : This is high-level-knowledge which relates the use of sentences in different contexts and how the contexts affects the meaning of the sentence.

ARTIFICIAL INTELLIGENCE

July - 2021

Paper Code:PCC-CSE-304-G

Note : ***Attempt five questions in all, selecting one question from each Section. Question No. 1 is compulsory. All questions carry equal marks.***

Q.1.(a) Differentiate between informed and uninformed search. (3)

Ans.

Informed Search	Uninformed Search
It uses knowledge for the searching process.	It doesn't knowledge for searching process.
It finds solution more quickly.	It finds solution slow as compared to informed search.
It may or may not be complete.	It is always complete.
Cost is low.	Cost is high.
It consumes less time.	It consumes moderate time.
It provides the direction regarding the solution.	No suggestion is given regarding the solution in it.
It is less lengthy while implementation.	It is more lengthy while implementation.
Ex. : Greedy Search, A* Search	Ex. : Depth first search, Breatdh first search

RC PUBLICATION

Q.1.(b) Write a short note on Semantic Network. (3)

Ans. Semantic Nets :

It is another form of representing the knowledge.

The major idea is that :

→ The meaning of a concept comes from its relationship with other concepts.

→ The information is stored by interconnecting nodes with labelled arcs.

Representation in a Semantic Net **:**

The physical attributes of a person can be represented as in Fig.(a).

These values can also be represented in logic as : is a (person, mammal), instance(Milk-Hall, person) team(Milk-Hall, Cardiff).

Features of Sementic Network **:**

1. Sementic nets were used was to find relationships among objects by spreading activation out from each of two nodes & seeing where the activation net. This process is called intersection search.

2. It is a natural way to represent relationships that would appear as ground instance of binary predicates in predicate logic.

3. It is particularly useful for representing the contents of a typical declarative sentence that describes several aspects of a particular events.

Sementic Networks for the facts. Every dog has bitten every mail carrier.

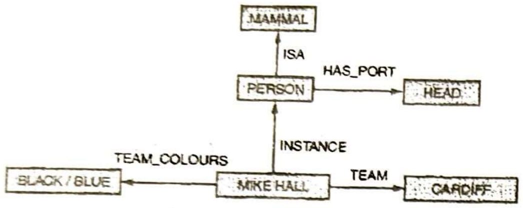


Fig (a) : A Semantic Network

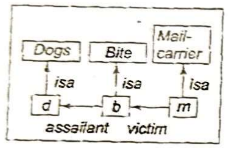


Fig (b) : Using Partitioned Semantic Nets

Q.1.(c) What do you mean by term Heuristics ? **(3)**

Ans. *Heuristic* is a rule of thumb that probably leads to a solution. Heuristics play a major role in search strategies because of exponential nature of the most problems. Heuristics help to reduce the number of alternatives from an exponential number to a polynomial number. In Artificial Intelligence, heuristic search has a general meaning, and a more specialized technical meaning. In a general sense, the term heuristic is used for any advice that is often effective, but is not guaranteed to work in every case. Within the heuristic search architecture, however, the term heuristic usually refers to the special case of a heuristic evaluation function.

The following algorithms make use of heuristic evaluation functions.

(1) Hill Climbing (2) Constraint Satisfaction
(3) Best-First Search (4) A* Algorithm
(5) AO* Algorithm (6) Beam search

Q.1.(d) Explain the term Non-monotonic reasoning. **(3)**

Ans. Nonmonotonic Reasoning : The conclusions derived from the monotonic logics are valid conclusions. If we go on adding new axioms(fact and axioms) in the knowledge base, it only increase the amount of knowledge contained in the knowledge base rather than decreasing.

Nonmonotonic is another form of reasoning. In this, new facts which are contradicting and invalidating the old knowledge came into picture. As new facts came into picture, old knowledge in the knowledge base become invalid thereby requiring further retractions. This retraction led to nonmonotonic growth in the knowledge at times.

In this real world, it is not reasonable to expect that the knowledge acquired by us at the first time is complete or valid. The initial knowledge may be incomplete, vague and can contain

redundancies, inconsistencies etc. therefore to attempt to model real world, commonsense reasoning is required.

A non-monotonic logic is a formal logic whose consequence relation is not monotonic. A logic is non-monotonic if the truth of a proposition may change when new information (axioms) are added.

→ Allows a statement to be retracted.

→ Used to formalize plausible (believable) reasoning.

The Non-monotonic reasoning are of the type :

→ Defaulat reasoning

→ Circumscription

→ Truth maintenance systems.

Q.1.(e) Write a short note on genetic algorithms. **(3)**

Ans. Genetic Algorithm : Genetic algorithms (GAs) are the main paradigm of evolutionary computing. GAs are inspired by Darwin's theory about evolution – the 'survival of the fittest'. In nature cometition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

Genetic operators are analogous to those which occur in the natural world : Reproduction (or selection)

Crossover (or recombination); and Mutation.

The cycle of a Genetic Algorithm is presented below :

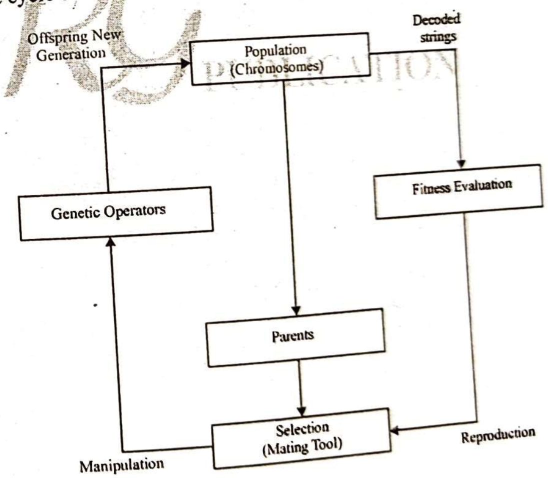


Fig.

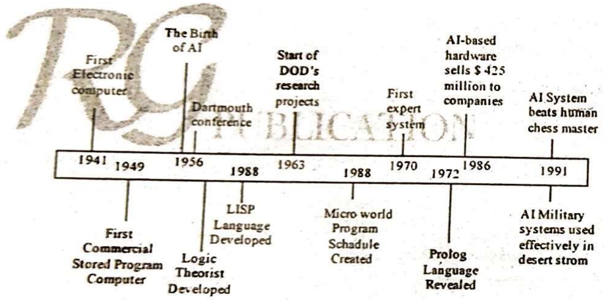
Each cycle in Genetic Algorithms produces a new generation of possible solutions for a given problem. In the first phase, an initial population, describing representatives of the potential solution, is created to initiate the search process. The elements of the population are encoded into bit-string, called chromosomes. The performance of the strings, often called fitness, is then evaluated with the help of some functions, representing the constrains of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. It should be noted that the selection process is mainly responsible for assuring survival of the best-fit individuals. After selection of the population strings is over, the genetic manipulation process is carried out. In the first step, the crossover operation that recombines the bits (genes) of each two selected strings (chromosomes) is executed.

Unit – I

Q.2.(a) Define term Artificial Intelligence. What are applications of AI in the various fields ? **(7)**

Ans. Artificial Intelligence(AI) is the intelligence of machines and the branch of computer science that aims to create it.

Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occurs in people, many animals and some machines. And according to John McCarthy, who coined the term Artificial Intelligence in 1956, defines it as "the science and engineering of making intelligent machines", especially intelligent computer programs.



The History or origin of Artificial Intelligence : Nobert wiener was one of the first Americans to make observations on the principle of feedback theory. In late 1955, Newell and Simon developed The Logic Therorist, considered by many to be the first AI program. The program, representing each problem as a tree model, would attempt to solve it by selecting the branch that would most likely result in the correct conclusion. The impact that the logic theorist made on both the public and the field of AI has made it a crucial stepping stone in developing the AI field.

In 1956 John McCarthy regrded as the father of AI, organized a conference to draw the talent and experitse of others interested in machine intelligence for a month of brainstorming. After 7 years AI began to pick up momentum. During the 1980's AI was moving at a faster pace, and further into the corporate sector. The 1980's introduced to its place in the corporate marketplace, and showed the technology had real life uses, ensuring it would be a key in the 21st

century. With the popularity of the AI computer growing the interest of the public has also grown. Applications for the Apple Macintosh and IBM compatible computer, such as voice and character recognition have become available. Also AI technology has made steadying camcorders simple using fuzzy logic. With a greater demand for AI-related technology, new advancement are becoming available. Inevitably Artificial Intelligence has, and will continue to affecting our lives.

Application of AI : Application of AI are as follows :

(1) Finance : Banks use artifical intelligence system to organize operations, invest in stocks, and manage properties.

(ii) Hospitals : A medical clinic can use artificial intelligence systems to organize bed schedules make a staff rotation, and provide medical information.

(iii) Heavy industry : Robots have become common in many industries. They are often given jobs that are considered dangerous to humans. Robots have proven effective in jobs that are very respectivies which may lead to mistakes or accidents due to a lapse in concentration and other jobs which humans may find degrading.

(iv) Music : The evolution of music has always been affected by technology. With AI, scientists are trying to make the computer emulate the activities of the skillful musician. Composition, performance, music theory, sound processing are some of the major areas on which research in Music and Artificial Intelligence are focusing.

(v) Game playing : Games are made by creating human level artificial intelligent entities e.g. enemies, partners, and support character that act just like humans.

Q.2.(b) Explain Hill Climbing strategy with example. What are the problems faced while applying this strategy ? **(8)**

Ans. Hill climbing is one of the Heuristic Search techniques. Hill Climbing strategies expand the current state in the search and evaluate its children. The best child is selected for further expansion and neither its siblings nor its parent are retained. Search halts when it reaches a state that is better than any of its children. Hill climbing is named for the strategy that might be used by an eager, but blind mountain climber, go uphill along the steepest possible path until you can go no further.

Problems in Hill Climbing : Both basic & steepest ascent hill climbing may fail to find a solution. Either algorithm may terminate not by finding a goal state but by getting to a state from which no better states can be generated. This will happen if the program has reached either of the following states :

(i) Local Maximum

(ii) Plateau

(iii) Ridge

(i) Local Maximum : A state that is better than all its neighbours but not so when compared to states that are father away. Local maximum are particularly frustrating because they often occur almost within sight of a solution. In this case, they are called foot-hills.

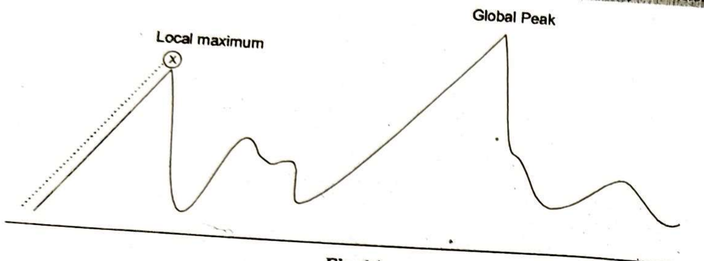


Fig. (a)

(ii) Plateau : A flat area of the search space in which all neighbours have the same value. On a plateau, it is not possible to determine the best direction in which to move by making loacal comparisons.

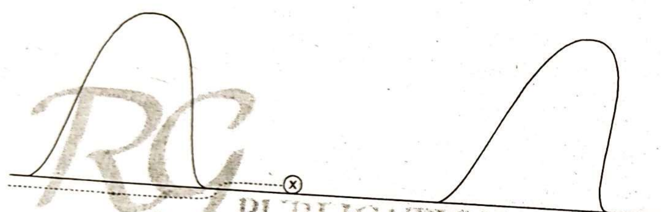


Fig. (b)

RG PUBLICATION

(iii) Ridge : A ridge is a special kind of local maximum. It is an area of search space that is higher than the surrounding areas & that iftself has a slope. But the orientation of the high region makes it impossible to transverse a ridge by single moves.

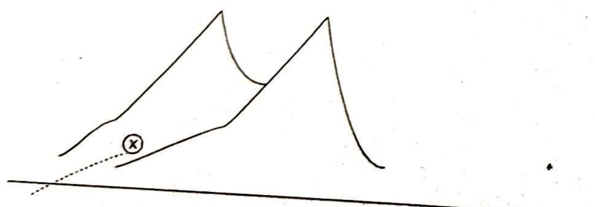


Fig. (c)

Solutions of Hill Climbing Problems :

(i) Back tracking to some some earlier node & try a different direction. This is a good way of dealing with local maximum.

(ii) Make a big jump in some direction to a new search. This is good way of dealing with plateaus.

(iii) Apply two or more rules before doing the test. This corresponds to moving in several directions at once. This is a particularly good strategy for dealing with ridges.

Hill climbing becomes inefficient in large problem spaces & when combinational explosion occurs. But it is very useful when combined with other methods.

Q.3.(a) What do you mean by Game playing in AI ? (7)

Ans. Game Playing is an important domain of artificial intelligence. Games don't require much knowledge; the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game.

Both players try to win the game. So, both of them try to make the best move possible at each turn. Searching techniques like BFS(Breadth First Search) are not accurate for this as the branching factor is very high, so searching will take a lot of time. So, we need another search procedures that improve –

- **Generate procedure** so that only good moves are generated.
- **Test procedure** so that the best move can be explored first.

The most common search technique in game playing is Minimax search procedure. It is depth-first depth-limited search procedure. It is used for games like chess and tic-tac-toe.

RG PUBLICATION

Q.3.(b) Explain Alpha-Beta pruning with example. What is the need of pruning a game tree ? (8)

Ans. Alpha-Beta Pruning : In MINIMAX search, number of game state increases exponentially. To reduce search, the pruning is done. Alpha-beta is one such pruning technique. It maintains two threshold values one is called alpha (or 'α') and other beta ('β').

These threshold values are defined as follows:

α = Lower bound on maximum value of utility function. It is the best acceptable value of utility function in maximizing ply.

β = Upper bound on minimum value of utility function. It is the best highest acceptable value of utility function in case minimizing ply.

In searching the game tree in mini max search, the part of tree having utility value less than α indicates that this particular move will not at all be useful. Hence part of tree having utility value less than alpha will be pruned. That means, in future all the nodes below it will never be explored. Similarly, if utility value comes out to be more than beta in case of minimizing ply, it will indicate that this move is at all useful and that sub free will be pruned on similar ground. Let us consider an example:

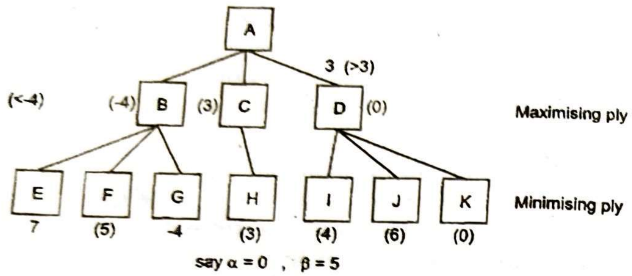


Fig. : α–β pruning

Here the possibilites are considered up to 2 ply. At min ply, the best value from three nodes is – 4, 3, 0. These will be back propagated towards root and a maximizing move 3 will be taken. Now node E having utility value 7 is far more, then accepted (as it is minimizing ply). So further node E will not be explored. In the situation when more plies are considered,whole sub tree below E will be pruned.

Similarly if alpha = 0 and beta = 5, all notes and related sub trees having value of unitly function less than 0 at maximizing ply and more than 5 at minimizing ply will be pruned. The procedure of minimax search is as follows.

Procedure MINIMAX *A, B* (Position)

{Procedure Minimax (Position, Depth, Player)

1. If reached last ply (Position, Depth)

then return the structure VALUE = STATIC (Position, Player)

2a. Path = nil

2b. Otherwise, generate one more ply of the tree and set SUCCESSOR to the list, it returns.

3. If SUCCESSOR, is empty, then return the same structure.

4. Otherwise, examine each element and find best one.

5. After examining all the nodes, return the structure

VALUE = BEST-SCORE

PATH = BEST-PATH}

The worthiness of α, β pruning depends upon the order in which paths are examined. If the worst successor is generated first then no cut offs will occur. Kunth and Moore have analyzed that for a search tree of branching factor b and depth, d the α - β search needs examining only $b^{d/2}$ nodes to pick up best move, instead of b^d for mini-max. That means, effective branching factor becomes vb instead of b. For a chess game, where the value of 'b' is 35, with alpha beta pruning the effective branching factor will become 6. This indicates a significant amount of saving in search.

Unit – II

Q.4.(a) Differentiate between propositional logic and predicate logic. (8)

Ans.

Propositional Logic	Predicate Logic
1. Propositional logic is the logic that deals with a collection of declarative statements which have a truth value, true or false.	Predicate logic is an expression consisting of variables with a specified domain. It consists of objects, relations and functions between the objects.
2. It is the basic and most widely used logic. Also knows as Boolean logic.	It is an extension of propositional logic covering predicates and quantification.
3. A proposition has a specific truth value, either true or false.	A predicate's truth value depends on the variables value.
4. Scope analysis is not done in propositional logic.	Predicate logic helps and analyze the scope of the subject over the predicate. There are three quantifiers : Universal Quantifier ($\forall$) depicts for all, Existential Quantifier ($\exists$) depicting there exists some and Uniqueness Quantifier ($\exists!$) depicting exactly one.
5. Propositions are combined with Logical Operators or Logical Connectives like Negation ($\neg$), Disjunction($\vee$), Conjunction($\wedge$), Exclusive OR($\oplus$), Implication($\Rightarrow$), Bi-Conditional or Double Implication($\Leftrightarrow$).	Predicate Logic adds by introducing quantifiers to the existing proposition.
6. It is a more generalized representation.	It is a more specialized representation.
7. It cannot deal with sets of entities.	It can deal with set of entities with the help of quantifiers.

Q.4.(b) What do you mean by Skolemization ? (7)

Ans. Skolemisation : In predicate logic, the existential and universal quantifiers are used. Skolemisation (after the name of logician Thoraf Skolem) is the process of managing these quantifiers. In the process of skolemisation, for universal quantifiers nothing changes and such variables are always turned into pattern matching variables. The universal quantification becomes implicit in the fact that such a variable will match anything.

The removal of existential variable is trickier. Every existential variable, whose arguments are the universal variable, must be turned into a function term from quantifirs, whose scope include that of existential quantifier. This term than replaces every occurrence of variable.

The Skolemisation is summarized as :

(*i*) Determine which variables are existential and which are universal.

(*ii*) Replace each existentially quantified variable by a function. The arguments of that function are universally quantified variables, which include the existential in their scope.

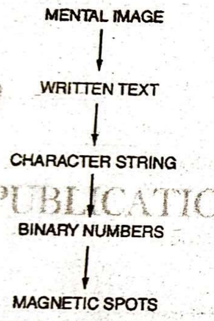
(*iii*) If two different universally quantified variables have same name, rename one of them.

(*iv*) Replace each universally quantified variable 'v' by simply v, i.e., $\forall x : v(x)$ is replaced by v.

Q.5. Explain the various ways of knowledge representation in AI. **(15)**

Ans. Knowledge representation is the method used to encode knowledge in an intelligent system's knowledge base. The object of knowledge representation is to express knowledge in computer-tractable form, such that it can be used to help intelligent system perform well.

Representation of Knowledge : As knowledge consists of facts, concepts, rules and so on. It can be represented as in different forms as mental image, as spoken or written words in some language, as graphical or other pictures and as character strings or collection of magnetic spots stored in the computer.



RG PUBLICATION

Fig. : Different Levels of Knowledge

Knowledge Representation Schemes :

(1) Relational Knowledge : This knowledge associates elements of one domain with another domain.

Relational knowledge is made up of objects consists of attributes and their corresponding associated values. The results of this knowledge type is a mapping of elements among different domains.

The table below shows a simple way to store facts.

The facts about a set of objects are put systematically in columns.

This representation provides little opportunity for inference.

Player	Height	Width	Bats-Throws
Ram	6 – 0	160	Right-left
Shyam	5 – 5	180	Right-right

Given the facts it is not possible to answer simple question such as :

"Who is the heaviest player?"

But if a procedure for finding heaviest player is provided, then these facts will enable that procedure to compute an answer. We can ask things like who "bats-left" and "throws-right".

(2) Inheritable Knowledge : Relational knowledge is made up of objects consisting of

(i) attributes

(ii) and their corresponding associated values

But in this the knowledge elements inherit the attributes from their parents and the data must be organised into a hierarchy of classes. Within the hierarchy, elements inherit attributes from their parents, but in many cases not all attributes of the parent elements be prescribed to the child elements.

The hierarchical structure is also called semantic network or collection of frames or slot-and-filler structure. The most useful froms of inference is property inheritance.

Property inheritance : The objects or elements of specific classes inherit attributes and values from more general classes. The classes are organized in a generalized hierarchy.

In this hirerarchy :

Boxed nodes : Boxed nodes represents objects and values of attributes of objects. Values can be objects with attributes and so on.

Arrows : Arrows point from object to its value.

This structure is known as a slot and filler structure, semantic network or a collection of frames.

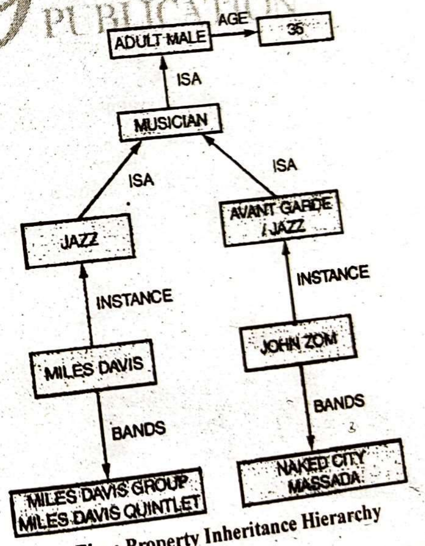


Fig. : Property Inheritance Hierarchy

(3) Inferential Knowledge : This knowledge generates new information from the given information. This new information does not require further data gathering form source, but does require analysis of the given information to generate new knowledge.

Example :

(i) given a set of relations and values, one may infer other values or relations.

(ii) a predicate logic (a mathematical deduction) is used to infer from a set of attributes.

(iii) inference through predicate logic uses a set of logical operations to relate individual data.

(iv) The symbols used for the logic operations are :

"$\rightarrow$" (implication), "$\neg$" (not), "$\vee$" (or), "$\wedge$" (and),

"$\forall$" (for all, "$\exists$" (there exists).

(4) Procedural Knowledge : Here, the knowledge is a mapping process between domains that specify "what to do when" and the representation is of "how to make" rather than "what it is". The procedural knowledge :

(i) may have inferential efficiency, but no inferential adequacy and acquisitional efficiency.

(ii) are represented as small programs that know how to do specific things, how to proceed.

Example : A parser in a natural language has the knowledge that a noun phrase may contain articles, adjectives and nouns. It thus accordingly call routines that know how to process articles, adjectives and nouns.

Unit – III

Q.6.(a) Explain how uncertainity is managed in AI. **(5)**

Ans. Uncertainty can arise from a variety of sources. The information available about one thing may be incomplete or volatile or may be some important facts and details about the problem are missing. Or the facts available may be vague or fuzzy.

Any AI system that seeks to model and reasoning in such a world must be able to deal with this. In particular it must be able to deal with :

→ Incompleteness -compensate for lack of knowledge.

→ Inconsistencies – resolve ambiguities and contradictions.

→ Change – it must be able to update its world knowledge base over time.

There are three basic methods that can deal with uncertainty are :

(1) Symbolic methods.

(2) Statistical methods.

(3) Fuzzy logic methods.

(1) Symbolic methods : The (Symbolic) methods basically represent uncertainty belief as being

→ True,

→ False, or

→ Neither True nor False.

Some methods also had problems with

→ Incomplete Knowledge

→ Contradictions in the knowledge.

(2) Statistical Methods : Statistical methods provide a method for representing beliefs that are not certain (or uncertain) but for which there may be some supporting (or contradictory) evidence is present.

Advantages of Statistical methods in two broad scenarious are :

Genuine Randomness : Playing Card games are a good example. We may not be able to predict any outcomes with certainty but we have knowledge about the likelihood of certain items (e.g. like being dealt an ace) and we can exploit this.

Exceptions : This advantage can be represented by symbolic methods. However if the number of exceptions is large, such system tend to break down. For example many common sense and expert reasoning tasks. Statistical techniques can summarise large exceptions without resorting enumeration.

(3) Fuzzy Logic methods : Fuzzy logic also deals with uncertainty problem. Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic theory, where binary sets have two-valued logic: true of false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degree may be managed by specific functions.

Q.6.(b) Explain Dempster shafer theory with the help of example. **(10)**

Ans. Dempster Shafer Theory : There is some information that probability cannot describe. For example, ignorance. Consider the following example: If we have absolutely no information about the coin, in probability theory, we will assume that it would be 50% head and 50% tail. However, in another scenario, we know the coin is fair, so we know for a fact that it would be 50% head and 50% tail. Therefore, in the two different scenarios, we arrive at the same conclusion. How we present total ignorance in probability theory becomes a problem.

Dempster-Shafer theory can effectively solve this problem: In Dempster-Shafter Theory, for the ignorance scenario, the belief of Head and the belief of Tail would be 0. For the fair coin scenario, the belief of Head would be 0.5, the belief fo Tail would also be 0.5.

An alternative parameterization can say that the probability of heads is p and tails is $1 - p$. Upon ignorance, we say that p is uniformly distributed on the $[0, 1]$ interval. Upon a fair coin assumption, we say that $p = 0.5$ with probability 1 and has other values with probability zero. Another kind of ignorance might be that p has some beta distribution on the interval $[0, 1]$.

The basic idea in representing uncertainty in this model is:

(i) Set up a confidence interval : An interval of probabilities within which the true probability lies with a certain confidence - based on the Belief B and plausibility PL provided by some evidence E for a proposition P.

(ii) The belief brings together all the evidence that would lead us to believe in P with some certainty.

(iii) The plausibility brings together the evidence that is compatible with P and is not inconsistent with it.

This method allows for further additions to the set of knowledge and does not assume disjoint outcomes.

If Ω is the set of possible outcomes, then a mass probability, M, is defined for each member of the set 2Ω and takes values in the range [0, 1].

The Null set, θ is also a memebr of 2Ω.

M is probability density function defined not just for Ω but for **em all** subsets.

So Ω is the set (Flu (F), Cold (C), Pneumonia (P)) then 2Ω is the set {θ, (F), (C), (P), (F, C), (F, P), (C, P), (F, C, P)}.

The confidence interval is then defined as [$B(E)$, $PL(E)$] where

$$B(E) = \sum_{A} M$$

where $A \subseteq E$ i.e. all the evidence that makes us believe in the correctness of P, and

$$PL(E) = 1 - B(\neg E)$$
$$= 1 - \sum_{A} M$$

where $\neg L \subseteq \neg E$ *i.e.* all the evidence that contradicts P.

Q.7.(a) Discuss partial-order plan with example. **(7)**

Ans. A partial-order plan or partial plan is a plan which specifies all actions that need to be taken, but only specifies the order between actions when necessary. It is the result of a partial-order planner. A partial-order plan consists of four components:

- A set of actions (also known as operators).
- A partial order for the actions. It specifies the conditions about the order of some actions.
- A set of causal links. It specifies which actions meet which preconditions of other actions. Alternatively, a set of bindings between the variables in actions.
- A set of open preconditions. It specifies which preconditions are not fulfilled by any action in the partial-order plan.

In order to keep the possible orders of the actions as open as possible, the set of order conditions and causal links must be as small as possible.

A plan is a solution if the set of open preconditions is empty.

A linearization of a partial order plan is a total order plan derived from the particular partial order plan; in other words, both order plans consist of the same actions, with the order in the linearization being a linear extension of the partial order in the original partial order plan.

Example : For example, a plan for baking a cake might start:

- go to the store
- get eggs; get flour; get milk
- pay for all goods
- go to the kitchen

This is a partial plan because the order for finding eggs, flour and milk is not specified, the agent can wander around the store reactively accumulating all the items on its shopping list until the list is complete.

Q.7.(b) How do we represent states, goals and actions in planning ? Explain with example. **(8)**

Ans. Representing States, Actions, and Goals : To reason about what to do, an agent must have goals, some model of the world, and a model of the consequences of its actions. A deterministic **action** is a partial function from states to states. It is partial because not every action can be carried out in every state. For example, a robot cannot carry out the action to pick up a particular object if it is nowhere near the object. The **precondition** of an action specifies when the action can be carried out. The **effect** of an action specifies the resulting state.

Example : Consider a delivery robot world with mail and coffee to deliver. Assume a simplified domain with four locations as shown in Figure.

Features to describe states

RLoc	- Rob's location
RHC	- Rob has coffee
SWC	- Sam wants coffee
MW	- Mail is waiting
RHM	- Rob has mail

Actions

mc	- move clockwise
mcc	- move counterclockwise
puc	- pickup coffee
dc	- deliver coffee

pum

- pickup mail

dm

- deliver mail

Figure : The delivery robot domain

The robot, called Rob, can buy coffee at the coffee shop, pick up mail in the mail room, move, and deliver coffee and/or mail. Delivering the coffee to Sam's office will stop Sam from wanting coffee. There can be mail waiting at the mail room to be delivered to Sam's office. This domain is quite simple, yet it is rich enough to demonstrate many of the problems in representing actions and in planning.

The state can be described in terms of the following features :

- the robot's location (*RLoc*), which is one of the coffee shop (*cs*), Sam's office (*off*), the mail room (*mr*), or the laboratory (*lab*).

- whether the robot has coffee (*RHC*). Let *rhc* mean Rob has coffee and $\neg rhc$ mean Rob does not have coffee.

- whether Sam wants coffee (*SWC*). Let *swc* mean Sam wants coffee and $\neg swc$ mean Sam does not want coffee.

- whether mail is waiting at the mail room (*MW*). Let *mw* mean there is mail waiting and $\neg mw$ mean there is no mail waiting.

- whether the robot is carrying the mail (*RHM*). Let *rhm* mean Rob has mail, and $\neg rhm$ mean Rob does not have mail.

Suppose Rob has six actions :

- Rob can move clockwise (*mc*).

- Rob can move counterclockwise (*mcc*).

- Rob can pick up coffee if Rob is at the coffee shop. Let *puc* mean that Rob picks up coffee. The precondition of *puc* is $\neg rhc \wedge RLoc{=}cs$; that is, Rob can pick up coffee in any state where its location is *cs*, and it is not already holding coffee. The effect of this action is to make *RHC* true. It does not affect the other features.

- Rob can deliver coffee if Rob is carrying coffee and is at Sam's office. Let *dc* mean that Rob delivers coffee. The precondition of *dc* is $rhc \wedge RLoc{=}off$. The effect of this action is to make *RHC* false and make *SWC* false.

- Rob can pick up mail if Rob is at the mail room and there is mail waiting there. Let *pum* mean Rob picks up the mail.

- Rob can deliver mail if Rob is carrying mail and at Sam's office. Let *dm* mean Rob delivers mail.

Assume that it is only possible for Rob to do one action at a time. We assume that a lower-level controller can implement these actions.

Unit – IV

Q.8. What is an Expert System ? Explain its architecture in detail. Also, write its applications in the various domains. (15)

Ans. Expert System : An expert system is software that attempts to provide an answer to a problem, or clarify uncertainties where normally one or more human experts would need to be consulted.

An expert system compared with traditional computer.

Inference engine + Knowledge = Expert system

(Algorithm + Data structures = Program in traditional computer)

MYCIN and DENDRAL are two expert system in history.

Various componets of an Expert System :

(1) *Knowledge Base :* The knowledge base stores all relevant information, data, rules, cases, and relationships used by the expert system, i.e. the knowledge base of expert systems contains both factual and heuristic knowledge.

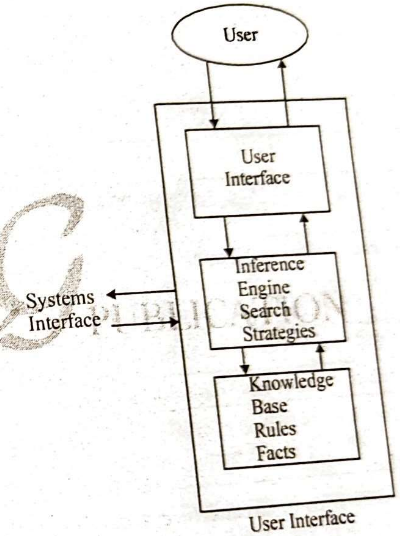


Fig.(a)

Factual knowledge is that knowledge of the task domain that is wideloy shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less accurate more experiential more judgemental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is knowledge of good practice, good judgement, and believable reasoning in the field. It is the knowledge that under lies the "art of good guessining".

A knowledge base can combine the knowledge of multiple human experts. A rule is a conditional statement that links given conditions to actions or outcomes.

A frame is another approach used to capture and store knowledge in a knowledge base. It relates an object or item to various facts or values. A frame-based representation is ideally suited for object-oriented programming techniques. Expert systems making use of frames to store knowledge are also called frame-based expert systems.

(2) *Inference Engine* : The purpose of the inference engine is to seek information and relationship from the knowledge base and to provide answers, predictions, and suggestions in the way a human expert would. The inference engine must find the right facts, interpretations, and rules and assemble them correctly. Two types of inference methods are commonly used-Backward chaining and forward chaining.

(a) *Forward chaining(Data-Driven)* : Forward chaining starts with the facts and works forward to the conclusions.

(b) *Backward Chaining(Goal-Driven)* : Backward chaining is the process of starting with conclusions and working backward to the supporting facts.

Applications of Expert System : The following table shows wherer Expert System can be applied :

Application	Description.
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis systems to deduce cause of disease from observed data, conduction medical operations on humans.
Monitoring Systems	Comparing data continuously with observed system or with prescribed behavior such as leakage monitoring in long petroleum pipeline.
Process control systems	Controlling a physical process based on monitoring.
Knowledage Domain	Finding out faults in vehicles, computers.
Finance/commerce	Detection of possible fraud, suspicious transaction, strock market trading, Airline scheduling, cargo scheduling.

Q.9.(a) Explain ANN with its architecture. How artificial Neural networks are similar to biological neural networks ? **(8)**

Ans. An artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements(neurons) working in parallel to solve specific problems. The weights on the connections encode the knowledge of a network. Each neuron has a local memory and the output of each neuron depends upon only the input signal arriving at the neuron and value in neuron's memory. ANNs, like people, learn by example.

ANN architecture is based on the structure and function of the biological neural network. Similar to neurons in the brain, ANN also consists of neurons which are arranged in various layers. Feed forward neural network is a popular neural network which consists of an input layer to receive the external data to perform pattern recognition, an output layer which gives the

problem solution, and a hidden layer is an intermediate layer which separates the other layers. The adjacent neurons from the input layer to output layer are connected through acyclic arcs. The ANN uses a training algorithm to learn the datasets which modifies the neuron weights depending on the error rate between target and actual output. In general, ANN uses the back propagation algorithm as a training algorithm to learn the datasets. The general structure of ANN is shown in Fig.

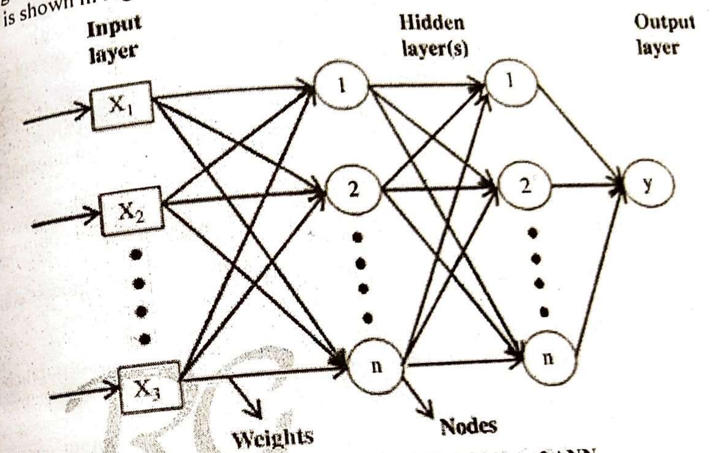


Fig. : General structure of ANN.

Similarities between artificial Neural network and biological neural networks :

(1) Biological neural networks process information in parallel; this is also true of artificial neural networks.

(2) Learning in biological neural networks is through past experiences which improve their performance level; this is also true of artificial neural networks.

(3) Learning in biological neural networks involves adjustment of the synaptic connections; learning in artificial neural networks is also by adjustment of weights. Weight in artificial neural networks is similar to synapse in biological neural networks.

(4) Information transmission in biological neural networks involves using electrical signals. In artificial neural networks, electrical signals are also used in information transmission.

(5) Information storage in biological neural networks is at the synapses, in artificial neural networks information is also stored in weights matrix.

Q.9.(b) Explain ANN applications in the various fields. **(7)**

Ans. Applications of artificial neural network : Neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.

(1) Business Applications : The 1988 DARPA Neural Network Study [DARP88] lists various neural network applications, beginning in about 1984 with the adaptive channel equalizer. This device, which is an outstanding commercial success, is a single-neuron network used in long-distance telephone systems to stabilize voice signals. The DARPA report goes on to list other commercial applications, including a small word recognizer, a process monitor, a sonar classifier, and a risk analysis system. Neural networks have been applied in many other fields since the DARPA report was written. A list of some applications mentioned in the literature follows.

(2) Aerospace : High performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, aircraft component fault detection.

(3) Automotive : Automobile automatic guidance system, warranty activity analysis.

(4) Banking : Check and other document reading, credit application evaluation.

(5) Credit Card Activity Checking : Neural networks are used to spot unusual credit card activity that might possibly be associated with loss of a credit card.

(6) Defense : Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, features extraction and noise suppression, signal/image identification.

(7) Electronics : Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modeling.

(8) Entertainment : Animation, special effects, market forecasting.

(9) Financial : Real estate appraisal, loan advisor, mortgage screening, corporate bond rating credit-line use analysis, portfolio trading program, corporate financial analysis, currency price prediction.

(10) Industrial : Neural networks are being trained to predict the output gasses of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past.

RG PUBLICATION

(11) Insurance : Policy application evaluation, product optimization.

(12) Manufacturing : Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, beer testing, welding quality analysis, paper quality prediction, computer-chip quality analysis, analysis of grinding operations, chemical product design analysis, machine maintenance analysis, project bidding, planning and management, dynamic modeling of chemical process system.

(13) Medical : Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, emergency-room test advisement.

(14) Oil and Gas : Exploration.

(15) Robotics : Trajectory control, forklift, robot, manipulator controllers, vision systems.

(16) Speech : Speech recognition, speech compression, vowel classification, text-to-speech synthesis.

(17) Securities : Market analysis, automatic bond rating, stock trading advisory systems.

(18) Telecommunications : Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems.

(19) Transportation : Truck brake diagnosis systems, vehicle scheduling , routing systems.

ARTIFICIAL INTELLIGENCE

July – 2022

Paper Code : PCC-CSE-304-G

Note : *Attempt five questions in all, selecting one question from each Section. Question No. 1 is compulsory. All questions carry equal marks.*

Q.1. Explain the following questions :

(a) History of AI (2.5)

(b) Ant Colony Optimization (2.5)

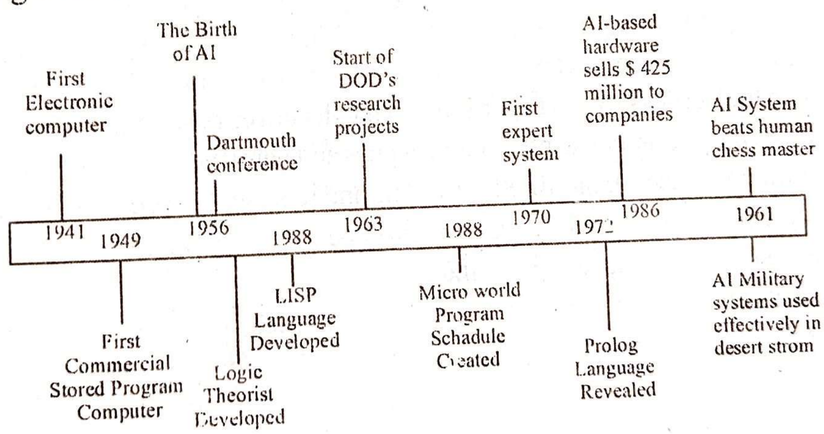
(c) Bayesian Reasoning (2.5)

(d) Transformational Analogy (2.5)

(e) Neural Networks (2.5)

(f) Informed Search (2.5)

Ans.(a) The History or origin of Artificial Intelligence : Nobert wiener was one of the first Americans to make observations on the principle of feedback theory. In late 1955, Newell and Simon developed The Logic Therorist, considered by many to be the first AI program. The program, representing each problem as a tree model, would attempt to solve it by selecting the branch that would most likely result in the correct conclusion. The impact that the logic theorist made on both the public and the field of AI has made it a crucial stepping stone in developing the AI field.



In 1956 John McCarthy regrded as the father of AI, organized a conference to draw the talent and experitse of others interested in machine intelligence for a month of brainstorming. After 7 years AI began to pick up momentum. During the 1980's AI was moving at a faster pace, and further into the corporate sector. The 1980's introduced to its place in the corporate marketplace, and showed the technology had real life uses, ensuring it would be a key in the 21st century. With the popularity of the AI computer growing the interest of the public has also grown. Applications for the Apple Macintosh and IBM compatible computer, such as voice and character recognition have become available. Also AI technology has made steadying camcorders simple using fuzzy logic. With a greater demand for AI-related technology, new advancement are becoming available. Inevitably Artificial Intelligence has, and will continue to affecting our lives.

Ans.(b) Ant colony optimization (ACO) is a population-based metaheuristic that can be used to find approximate solutions to difficult optimization problems.

In ACO, a set of software agents called *artificial ants* search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants (hereafter ants) incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a *pheromone model*, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants.

The easiest way to understand how ant colony optimization works is by means of an example. We consider its application to the traveling salesman problem (TSP). In the TSP a set of locations (e.g. cities) and the distances between them are given. The problem consists of finding a closed tour of minimal length that visits each city once and only once.

To apply ACO to the TSP, we consider the graph defined by associating the set of cities with the set of vertices of the graph. This graph is called *construction graph*. Since in the TSP it is possible to move from any given city to any other city, the construction graph is fully connected and the number of vertices is equal to the number of cities. We set the lengths of the edges between the vertices to be proportional to the distances between the cities represented by these vertices and we associate pheromone values and heuristic values with the edges of the graph. Pheromone values are modified at runtime and represent the cumulated experience of the ant colony, while heuristic values are problem dependent values that, in the case of the TSP, are set to be the inverse of the lengths of the edges.

The ants construct the solutions as follows. Each ant starts from a randomly selected city (vertex of the construction graph). Then, at each construction step it moves along the edges of the graph. Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited. An ant has constructed a solution once it has visited all the vertices of the graph. At each construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will

choose that particular edge. Once all the ants have completed their tour, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).

This procedure is repeatedly applied until a termination criterion is satisfied.

Ans.(c) Bayesian Reasoning : The Bayes' theorem calculates the conditional probability P(A/B) of an event A to happen when the event B has already happened, with the help of the inverse in time conditional probability P(B/A), the prior probability P(A) and the posterior probability P(B). Since the changes of the value of P(A) result to different values of P(A/B), **Bayesian Reasoning** defines a multi-valued logic treating the existing due to the imprecision of the values of P(A) uncertainty in a way analogous to fuzzy logic. Therefore, **Bayesian Reasoning** could be considered as an interface between bivalent and fuzzy logic.

Ans.(d)Transformational Analogy : Transformational Analogy Looks for a similar solution and copy it to the new situation making suitable substitutions where appropriate.

For example, Geometry.

If you know about lengths of line segments and a proof that certain lines are equal (Fig.) then we can make similar assertions about angles.

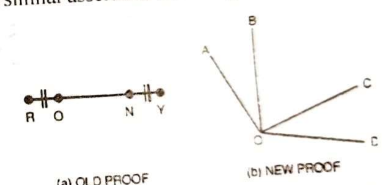


Fig. : Transformational Analogy Example.

- We know that lines $RO = NY$ and angles $AOB = COD$
- We have seen that $RO + ON = ON + NY$ – additive rule.
- So we can say that angles $AOB + BOC = BOC + COD$
- So by a transitive rule line $RN = OY$
- So similarly angle $AOC = BOD$

Carbonell (1983) describes a T–space method to transform old solutions into new ones.

- Whole solutions are viewed as states in a problem space – the T–space.
- T–operators prescribe methods of transforming existing solution states into new ones.

– Reasoning by analogy becomes a search in T–space : starting with an old solution, use means–end analysis or some other method to find a solution to the current problem.

Ans.(e) Neural Network : Neural network is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.

Neural Networks are identified by different names such as

(i) Artificial Neural Network

(ii) Connectionists Network.

(iii) Parallel distributed processing systems.

Ans.(f) Informed Search Algorithms: Here, the algorithms have information on the goal state, which helps in more efficient searching. This information is obtained by something called a *heuristic*.

Heuristic is a function which is used in Informed Search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. Heuristic function estimates how close a state is to the goal. It is represented by $h(n)$, and it calculates the cost of an optimal path between the pair of states. The value of the heuristic function is always positive.

Informed search algorithms are :

1. Greedy Search
2. A* Tree Search
3. A* Graph Search

Section – A

Q.2. What do you mean by Game Playing : Min.–Max. Algorithm and Alpha–Beta Pruning. Explain in detail. **(15)**

Ans. Game Playing is an important domain of artificial intelligence. Games don't require much knowledge; the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game.

Both players try to win the game. So, both of them try to make the best move possible at each turn. Searching techniques like BFS(Breadth First Search) are not accurate for this as the branching factor is very high, so searching will take a lot of time. So, we need another search procedures that improve –

- **Generate procedure** so that only good moves are generated.

- **Test procedure** so that the best move can be explored first.

The most common search technique in game playing is Minimax search procedure. It is depth-first depth-limited search procedure. It is used for games like chess and tic-tac-toe.

Mini–Max Algorithm : (a) Searchng game tree using the Mini–Max algorithm.

(b) Steps used in picking the next move :

– Since its may turn to move, the start node is MAX node with current board configuration.

– Expand nodes down (play) to some depth of look–ahead in the game.

– Apply evaluation function at each of the leaf nodes.

– "Backup" values for each non–leaf nodes until computed for the root node.

(c) At MIN nodes, the backed up value is maximum of the values associated with its children.

(d) At MAX node, the backed up value is the maximum of the values associated with its children.

Note : The process of "backing up" values gives the optimal strategy that is, both players assuming that your opponent is using the same static evaluation function as you are :

Example :

(1) The MAX player considers all three possible moves.

(2) The opponent MIN player also considers all possible moves.

(3) The evaluation function is applied to leaf level only.

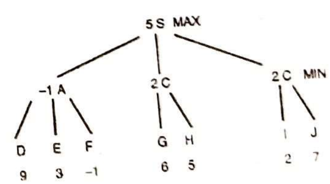


Fig.

Apply Evaluation Function :

(*i*) Apply static evaluation function at leaf nodes and begin backing up.

(*ii*) First compute backed up values at the parents of the leaves.

– Node A is a MIN node, i.e. it is the opponent's turn to move.

- A's backed–up value is –1, i.e. min of (9, 3, –1), meaning if opponent every reaches this node, then it will pick the move associated with the arc from *A* to *F*.
- Similarly, B's backed–up value is 5 and C's backed up value is 2.

(*iii*) Next, backup values to next higher level,

- Node *S* is a MAX node, i.e. its' our turn to move.
- Look best on backed–up values at each of S's children.
- The best child is *B* since value is 5, i.e. MAX of (–1, 5, 2).
- So the minimax value for the root node S is 5, and the move selected is associated with the arc from *S* to *B*.

The problem with Mini–Max algorithm is that the number of game states it has to examine is exponential in the number of moves.

Alpha–Beta Pruning : In MINIMAX search, number of game state increases exponentially. To reduce search, the pruning is done. Alpha-beta is one such pruning technique. It maintains two threshold values one is called alpha (or 'α') and other beta ('β').

These threshold values are defined as follows:

α = Lower bound on maximum value of utility function. It is the best acceptable value of utility function in maximizing ply.

β = Upper bound on minimum value of utility function. It is the best highest acceptable value of utility function in case minimizing ply.

In searching the game tree in mini max search, the part of tree having utility value less than α indicates that this particular move will not at all be useful. Hence part of tree having utility value less than alpha will be pruned. That means, in future all the nodes below it will never be explored. Similarly, if utility value comes out to be more than beta in case of minimizing ply, it will indicate that this move is at all useful and that sub free will be pruned on similar ground. Let us consider an example:

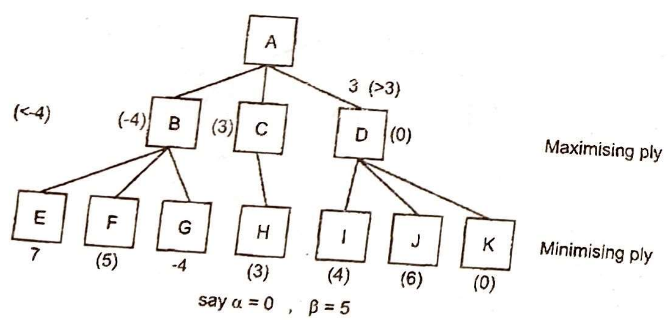


Fig. : α–β pruning

Here the possibilites are considered up to 2 ply. At min ply, the best value from three nodes is –4, 3, 0. These will be back propagated towards root and a maximizing move 3 will be taken. Now node E having utility value 7 is far more, then accepted (as it is minimizing ply). So further node E will not be explored. In the situation when more plies are considered, whole sub tree below E will be pruned.

Similarly if alpha = 0 and beta = 5, all notes and related sub trees having value of unitly function less than 0 at maximizing ply and more than 5 at minimizing ply will be pruned. The procedure of minimax search is as follows.

Procedure MINIMAX *A*, *B* (Position)

{Procedure Minimax (Position, Depth, Player)

1. If reached last ply (Position, Depth)

then return the structure VALUE = STATIC (Position, Player)

2a. Path = nil

2b. Otherwise, generate one more ply of the tree and set SUCCESSOR to the list, it returns.

3. If SUCCESSOR, is empty, then return the same structure.

4. Otherwise, examine each element and find best one.

5. After examining all the nodes, return the structure

VALUE = BEST-SCORE

PATH = BEST-PATH}

The worthiness of α, β pruning depends upon the order in which paths are examined. If the worst successor is generated first then no cut offs will occur. Kunth and Moore have analyzed that for a search tree of branching factor b and depth, d the α - β search needs examining only $b^{d/2}$ nodes to pick up best move, instead of b^d for mini-max. That means, effective branching factor becomes vb instead of b. For a chess game, where the value of 'b' is 35, with alpha beta pruning the effective branching factor will become 6. This indicates a significant amount of saving in search.

Q.3.(a) Explain A* algorithm. (8)

Ans. The A* Algorithm : In its simplest form best first search is useful, but doesn't take into account the cost of the paths when choosing which node to search from next. So, a solution may be find out but it may be not a very good solution.

There is a variant of best first search known as A* which attempts to find a solution which minimizes the total length or cost of the solution path. It combines advantanges of breadth first search, where the shortest path is found first, with advantages of best first search, where the node that is guesssed closest to the solution is explored next.

In the A* algorithm the score which is assigned to a node is a combination of the cost of the path and the estimated cost to solution. This is normally expressed as an evaluation function $f(n)$, which involves the sum of the values returned by two functions $g(n)$ and $h(n)$, i.e.

$$f(n) = g(n) + h(n) \qquad \text{(n represents node)}$$

where $h(n)$ = cost of the cheapest path from node n to a goal state.

$g(n)$ = cost of the cheapest path from the initial state to node n.

Or $$f^*(n) = g^*(n) + h^*(n)$$

$h^*(n)$ (heuristic factor) = estimate of $h(n)$.

$g^*(n)$ (depth factor) = approximation of $g(n)$ found by A* so far.

The A* algorithm then looks the same as the simple best first algorithm, slightly more complex evaluation function is used here.

Algorithm is as follows :

1. Start with OPEN holding the initial nodes.
2. Pick the BEST node on OPEN such that Or $f^*(n) = g^*(n) + h^*(n)$ is minimal.
3. If BEST is goal node quit and return the path from initial to BEST Otherwise.
4. Remove BEST from OPEN and all of BESTs children, labelling each with its path from initial node.

Yes A* algorithm is guaranteed to find an optimal goal path if one exits.

Q.3.(b) What is Hill Climbing ? Explain simple Hill Climbing. (7)

Ans. Hill Climbing : Hill climbing is a varient of generat & test in which feedback from the test procedure is used to help the generator decide which direction to move in the search space. In a pure generate-and-test procedure, the test fuction responds with only yes or no. But in hill climbing the test fuction is provided with heuristic function which provides an estimate of how close given state is to goal state.

Hill climbing is often used when a good neuristic function is avbailable for evaluating states but when no other useful information is available.

Simple Hill Climbing : The simplest way to implement hill climbing is as follows :

Algorithm :

1. Evaluate the initial state. If it is also a goal state, then return it & quit . Otherewise, continue with the initial state as current state.

2. Loop until a solution is found or until there are no new operators left to be applied in the current state :

(a) Select an operator that has not yet been applied to the current state & apply it to produce a new state.

(b) Evaluate the new state.

(i) If it is a goal state, then return it & quit.

(ii) If it is not a goal state but it is better then the current state, then make it the current state.

(iii) If it is not better than current state, then continue in the loop.

Section – B

Q.4. Describe different Approaches to Knowledge Representation. (15)

Ans. Knowledge representation is the method used to encode knowledge in an intelligent system's knowledge base. The object of knowledge representation is to express knowledge in computer-tractable form, such that it can be used to help intelligent system perform well.

Representation of Knowledge : As knowledge consists of facts, concepts, rules and so on. It can be represented as in different forms as mental image, as spoken or written words in some language, as graphical or other pictures and as character strings or collection of magnetic spots stored in the computer.

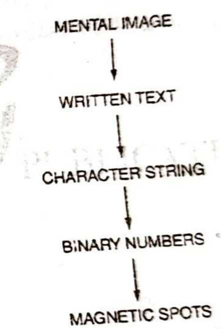


Fig. : Different Levels of Knowledge

Knowledge Representation Schemes :

(1) Relational Knowledge : This knowledge associates elements of one domain with another domain.

Relational knowledge is made up of objects consists of attributes and their corresponding associated values. The results of this knowledge type is a mapping of elements among different domains.

The table below shows a simple way to store facts.

The facts about a set of objects are put systematically in columns.

This representation provides little opportunity for inference.

Player	Height	Width	Bats-Throws
Ram	6 – 0	160	Right-left
Shyam	5 – 5	180	Right-right

Given the facts it is not possible to answer simple question such as :

"Who is the heaviest player?"

But if a procedure for finding heaviest player is provided, then these facts will enable that procedure to compute an answer. We can ask things like who "bats-left" and "throws-right".

(2) Inheritable Knowledge : Relational knowledge is made up of objects consisting of

(i) attributes

(ii) and their corresponding associated values

But in this the knowledge elements inherit the attributes from their parents and the data must be organised into a hierarchy of classes. Within the hierarchy, elements inherit attributes from their parents, but in many cases not all attributes of the parent elements be prescribed to the child elements.

The hierarchical structure is also called semantic network or collection of frames or slot-and-filler structure. The most useful froms of inference is property inheritance.

Property inheritance : The objects or elements of specific classes inherit attributes and values from more general classes. The classes are organized in a generalized hierarchy.

In this hirerarchy :

Boxed nodes : Boxed nodes represents objects and values of attributes of objects. Values can be objects with attributes and so on.

Arrows : Arrows point from object to its value.

This structure is known as a slot and filler structure, semantic network or a collection of frames.

(3) Inferential Knowledge : This knowledge generates new information from the given information. This new information does not require further data gathering form source, but does require analysis of the given information to generate new knowledge.

Example :

(i) given a set of relations and values, one may infer other values or relations.

(ii) a predicate logic (a mathematical deduction) is used to infer from a set of attributes.

(iii) inference through predicate logic uses a set of logical operations to relate individual data.

(iv) The symbols used for the logic operations are : "$\rightarrow$" (implication), "$\neg$" (not), "$\vee$" (or), "$\wedge$" (and), "$\forall$" (for all, "$\exists$" (there exists).

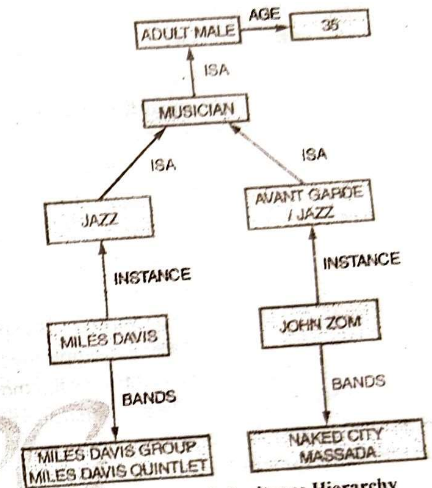


Fig. : Property Inheritance Hierarchy

(4) Procedural Knowledge : Here, the knowledge is a mapping process between domains that specify "what to do when" and the representation is of "how to make" rather than "what it is". The procedural knowledge :

(i) may have inferential efficiency, but no inferential adequacy and acquisitional efficiency.

(ii) are represented as small programs that know how to do specific things, how to proceed.

Example : A parser in a natural language has the knowledge that a noun phrase may contain articles, adjectives and nouns. It thus accordingly call routines that know how to process articles, adjectives and nouns.

Q.5. Explain Rule–Base–System with example. (15)

Ans. Rule based system : Rule-based systems (also known as production systems or expert systems) are the simplest form of artificial intelligence. A rule based system uses rules as the knowledge representation for knowledge coded into the system. The definitions of rule-based system depend almost entirely on expert systems, which are system that mimic the reasoning of human expert in solving a knowledge intensive problem. Instead of representing knowledge in a declarative static way as a set of things which are true, rule-based system represent knowledge in terms of a set of rules that tells what to do or what to conclude in different situations.

A rule-based system is a way of encoding a human expert's knowledge in a fairly narrow area into an automated system. A rule-based system can be simply created by using a set of assertions and a set of rules that specify how to act on the assertion set. Rules are expressed as a set of if-then statements (called IF-THEN rules or productioin rules) :

IF P THEN Q

which is also equivalent to :

$P \Rightarrow Q$.

A rule-based system consists of a set of IF-THEN rules, a set of facts and some interpreter controlling the application of the rules, given the facts. The idea of an expert system is to use the knowledge from an expert system and to encode it into a set of rules. When exposed to the same data, the expert system will perform (or is expected to perform) in a similar manner to the expert. Rule-based systems are very simple models and can be adapted and applied for a large kind of problems. The requirement is that the knowledge on the problem area can be expressed in the form of if-then rules. The area shold also not be that large because a high number of rules can make the problem solver (the expert system) inefficient.

Elements of a rule-based system : Any rule-based system consists of a few basic and simple elements as follows :

1. A set of *facts*. These facts are actually the assertions and should be any thing relevant to the beginning state of the system.
2. A set of *rules*. This contains all actions that should be taken within the scope of a problem specify how to act on the THEN part. The system should contain only relevant rules and avoid the irrelevant ones because the number of rules in the system will affect its performance.
3. A termination criterion. This is a condition that determines that a solution has been found or that none exists. This is necessary to terminate some rule-based systems that find themselves in infinite loops otherwise.

Facts can be seen as a collection of data and conditions. Data associated the value of characteristics with a thing and conditions perform tests of the values of characteristics to determine if something is of interest, perhaps the correct classification of something or whether an event has taken place.

For instance, if we have the fact :

temperature <)

then temperature is the data and the condition is < 0.

Rules do not interact directly with data, but only with conditions either singly or multiple (joined by logical operators as shown below).

Example : Trouble shooting of water pumps

(1) If pump failure then the pressure is low

(2) If pump failure then check oil level

(3) If power failure then pump failure

- Rule based system consists of a library of such rules.
- Rules reflect essential relationship within the domain.
- Rules reflect ways to reason about the domain.
- Rules draw conclusions and points to actions, when specific information about the domain comes in. This is called inference.
- The inference is a kind of chain reaction like : If there is a power failure then (see rules 1, 2, 3 mentioned above).

How to deal uncertainties' in rule base systems ?

A problem with rule–based systems is that often the connections reflected by the rules are not absolutely certain (i.e., deterministic), and the gathered information is often subject to uncertainty.

In such cases, a certainty measure is added to the premises as well as the conclusions in the rules of the system

A rule then provided a function that describes: how much a change in the certainty of the premise will change the certaity of the conclusion.

In its simplest form, this looks like :

If A (with certainty x) then B (with certainty f (x))

This is a new rule, say rule 4, added to earlier three rules.

There are many schemes for treating uncertainty in rule based systems.

The most common are :

- Adding certainty factors.
- Adoptions of Dempster–Shafter belief functions.
- Inclusion of fuzzy logic.

Applications : Two major applications of probability theory in everyday life are in

- Risk Assessment
- Commodity Markets.

Goverment typically apply probabilistic methods in environmental regulation where it is called "pathway analysis", often measuring well–being using methods that are stochastic in nature, and choosing projects to undertake based on statistical analyses of their probable effect on the population as a whole.

Another significant application of probability theory in everyday life is reliability. Many consumer products, such as automobiles and consumer electronics, utilize reliability theory in the design of the product in order to reduce the probability of failure. The probability of failure may be closely associated with the product's warranty.

Section – C

Q.6. Explain in detail Dempster–Shafer Theory. (15)

Ans. Dempster Shafer Theory : There is some information that probability cannot describe. For example, ignorance. Consider the following example: If we have absolutely no information about the coin, in probability theory, we will assume that it would be 50% head and 50% tail. However, in another scenario, we know the coin is fair, so we know for a fact that it would be 50% head and 50% tail. Therefore, in the two different scenarios, we arrive at the same conclusion. How we present total ignorance in probability theory becomes a problem.

Dempster-Shafer theory can effectively solve this problem: In Dempster-Shafter Theory, for the ignorance scenario, the belief of Head and the belief of Tail would be 0. For the fair coin scenario, the belief of Head would be 0.5, the belief fo Tail would also be 0.5.

An alternative parameterization can say that the probability of heads is p and tails is 1 $-p$. Upon ignorance, we say that p is uniformly distributed on the [0, 1] interval. Upon a fair coin assumption, we say that $p = 0.5$ with probability 1 and has other values with probability zero. Another kind of ignorance might be that p has some beta distribution on the interval [0, 1].

The basic idea in representing uncertainty in this model is:

(i) Set up a confidence interval : An interval of probabilities within which the true probability lies with a certain confidence - based on the Belief B and plausibility PL provided by some evidence E for a proposition P.

(ii) The belief brings together all the evidence that would lead us to believe in P with some certainty.

(iii) The plausibility brings together the evidence that is compatible with P and is not inconsistent with it.

This method allows for further additions to the set of knowledge and does not assume disjoint outcomes.

If Ω is the set of possible outcomes, then a mass probability, M, is defined for each member of the set 2Ω and takes values in the range [0, 1].

The Null set, θ is also a memebr of 2Ω.

M is probability density function defined not just for Ω but for **em all** subsets.

So Ω is the set (Flu (F), Cold (C), Pneumonia (P)) then 2Ω is the set $\{\theta, (F), (C), (P), (F, C), (F, P), (C, P), (F, C, P)\}$.

The confidence interval is then defined as $[B(E), PL(E)]$ where

$$B(E) = \sum_{A} M$$

where $A \subseteq E$ i.e. all the evidence that makes us believe in the correctness of P, and

$$PL(E) = 1 - B(\neg E)$$

$$= 1 - \sum_{A} M$$

where $\neg L \subseteq \neg E$ i.e. all the evidence that contradicts P.

Q.7. What is Probability Theory in terms of Reasoning under Uncertainty ? Elaborate. (15)

Ans. Bayesian Theorem : Bayes's theorem shows the relation between two conditional probabilities which are the reverse of each other. This theorem is named for Thomas Bayes and often called Bayes' law or Bayes' rule. Bayes' theorem expresses the conditional probability, or "posterior probability", of a hypothesis H (i.e. its probability after evidence E is observed) in terms of the "prior probability" of H, the prior probability of E, and the conditional probability of E given H. It implies that evidence has a stronger confirming effect if it was more unlikely before being observed.

This sates :

$$P(H_i / E) = \frac{P(E \setminus H_i) P(H_i)}{\sum_{k=1}^{n} P(E \setminus H_k) P(H_k)}$$

This statement/equation can be read as that given some evidence E then probability that hypothesis H_i is true is equal to the ratio of the probability that E will be true given H_i times the a priori evidence on the probability of H_i and and the sum of the probability of E over the set of all hypotheses times the probability of these hypotheses.

– The set of all hypothese must be mutually exclusive and exhaustive.

– Thus to find if we examine medical evidence to diagnose an illness. We must know all the prior probabilities of find symptom and also the probability of having an illness based on certain symptoms being observed.

Simple Statement of Theorem : Thomas Bayes addressed both the case of discrete probability distributions of data and the more complicated case of continuous probability distributions. In the discrete case, Bayes' theorem relates the conditional and marginal probabilities of events A and B, provided that the probability of B does not equal zero :

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)}$$

In Bayes' theorem, each probability has a conventional name :

– $P(A)$ is the prior probability (or "unconditional" or "marginal" probability) of A. It is "prior" in the sense that it does not take into account any information about B; however, the event B need not occur after event A.

– $P(A | B)$ is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.

– $P(B | A)$ is the conditional probability of B given A. It is also called the likelihood.

– $P(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

Bayes' theorem in this form gives a mathematical representation of how the conditional probability of event A given B is related to the converse conditional probability of B given A.

Section – D

Q.8. Explain in detail Expert systems. **(15)**

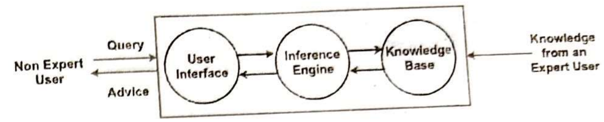
Ans. Artificial Intelligence is a piece of software that simulates the behaviour and judgement of a human or an organization that has experts in a particular domain is known as an expert system. It does this by acquiring relevant knowledge from its knowledge base and interpreting it according to the user's problem. The data in the knowledge base is added by humans that are expert in a particular domain and this software is used by a non-expert user to acquire some information. It is widely used in many areas such as medical diagnosis, accounting, coding, games etc.

An expert system is AI software that uses knowledge stored in a knowledge base to solve problems that would usually require a human expert thus preserving a human expert's knowledge in its knowledge base. They can advise users as well as provide explanations to them about how they reached a particular conclusion or advice. *Knowledge Engineering* is the term used to define the process of building an Expert System and its practitioners are called *Knowledge Engineers*. The primary role of a knowledge engineer is to make sure that the computer possesses all the knowledge required to solve a problem. The knowledge engineer must choose one or more forms in which to represent the required knowledge as a symbolic pattern in the memory of the computer.

Example : There are many examples of an expert system. Some of them are given below –

- MYCIN – One of the earliest expert systems based on backward chaining. It can identify various bacteria that can cause severe infections and can also recommend drugs based on the person's weight.
- DENDRAL – It was an artificial intelligence-based expert system used for chemical analysis. It used a substance's spectrographic data to predict its molecular structure.
- R1/XCON – It could select specific software to generate a computer system wished by the user.
- PXDES – It could easily determine the type and the degree of lung cancer in a patient based on the data.
- CaDet – It is a clinical support system that could identify cancer in its early stages in patients.
- DXplain – It was also a clinical support system that could suggest a variety of diseases based on the findings of the doctor.

Components of an Expert System :



Architecture of an Expert System

- *Knowledge Base* – The knowledge base represents facts and rules. It consists of knowledge in a particular domain as well as rules to solve a problem, procedures and intrinsic data relevant to the domain.
- *Inference Engine* – The function of the inference engine is to fetch the relevant knowledge from the knowledge base, interpret it and to find a solution relevant to the user's problem. The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts. Inference engines can also include an explanation and debugging abilities.
- *Knowledge Acquisition and Learning Module* – The function of this component is to allow the expert system to acquire more and more knowledge from various sources and store it in the knowledge base.
- *User Interface* – This module makes it possible for a non-expert user to interact with the expert system and find a solution to the problem.
- *Explanation Module* – This module helps the expert system to give the user an explanation about how the expert system reached a particular conclusion.

The Inference Engine generally uses two strategies for acquiring knowledge from the Knowledge Base, namely –

- Forward Chaining
- Backward Chaining

Forward Chaining – Forward Chaining is a strategic process used by the Expert System to answer the questions – What will happen next. This strategy is mostly used for managing tasks like creating a conclusion, result or effect. Example – prediction or share market movement status.

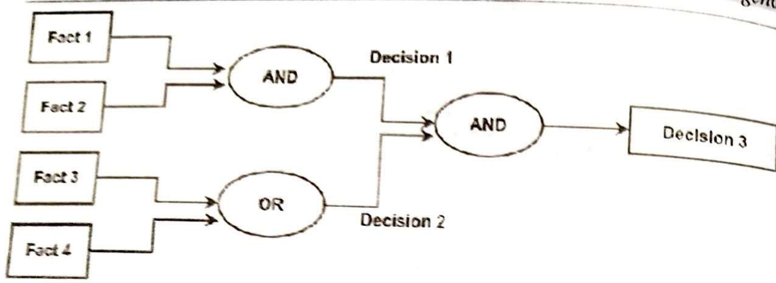


Fig. : Forward Chaining

Backward Chaining – Backward Chaining is a storage used by the Expert System to answer the questions – Why this has happened. This strategy is mostly used to find out the root cause or reason behind it, considering what has already happened. Example – diagnosis of stomach pain, blood cancer or dengue, etc.

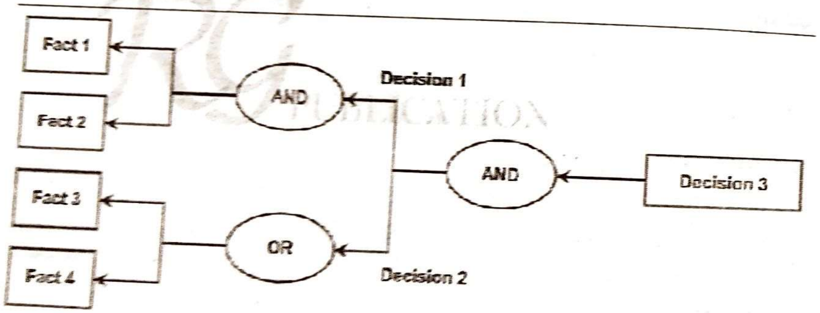


Fig. : Backward Chaining

Characteristics of an Expert System:

- Human experts are perishable, but an expert system is permanent.
- It helps to distribute the expertise of a human.
- One expert system may contain knowledge from more than one human experts thus making the solutions more efficient.
- It decreases the cost of consulting an expert for various domains such as medical diagnosis.
- They use a knowledge base and inference engine.

- Expert systems can solve complex problems by deducing new facts through existing facts of knowledge, represented mostly as if-then rules rather than through conventional procedural code.
- Expert systems were among the first truly successful forms of artificial intelligence (AI) software.

Limitations:

- Do not have human-like decision-making power.
- Cannot possess human capabilities.
- Cannot produce correct result from less amount of knowledge.
- Requires excessive training.

Advantages:

- Low accessibility cost.
- Fast response.
- Not affected by emotions, unlike humans.
- Low error rate.
- Capable of explaining how they reached a solution.

Disadvantages:

- The expert system has no emotions.
- Common sense is the main issue of the expert system.
- It is developed for a specific domain.
- It needs to be updated manually. It does not learn itself.
- Not capable to explain the logic behind the decision.

Applications: The application of an expert system can be found in almost all areas of business or government. They include areas such as –

- Different types of medical diagnosis like internal medicine, blood diseases and show on.
- Diagnosis of the complex electronic and electromechanical system.
- Diagnosis of a software development project.
- Planning experiment in biology, chemistry and molecular genetics.
- Forecasting crop damage.
- Diagnosis of the diesel-electric locomotive system.
- Identification of chemical compound structure.
- Scheduling of customer order, computer resources and various manufacturing task.
- Assessment of geologic structure from dip meter logs.
- Assessment of space structure through satellite and robot.
- The design of VLSI system.
- Teaching students specialize task.
- Assessment of log including civil case evaluation, product liability etc.

Expert systems have evolved so much that they have started various debates about the fate of humanity in the face of such intelligence, with authors such as Nick Bostrom (Professor of Philosophy at Oxford University), pondering if computing power has transcended our ability to control it.

Q.9. Demonstrate a discussion of AI, its current trends, limitations and applications. **(15)**

Ans. Artificial Intelligence : Artificial Intelligence (AI) is the machine-displayed intelligence that simulates human behavior or thinking and can be trained to solve specific problems. AI is a combination of Machine Learning techniques and Deep Learning. Types of artificial intelligence models are trained using vast volumes of data have the ability to make intelligent decisions.

AI Trends : Artificial intelligence adoption in different enterprises has grown due to the pandemic. In 2019, IDC predicted that spending on AI technologies would increase to $97.9 billion by 2023. Since the COVID-19 pandemic hit the world, the potential value of artificial intelligence has only grown. A survey published by the McKinsey State of AI in November 2020 suggested that at least half of the organisations have adopted AI functions in their organization. This article on Artificial intelligence trends will help you understand more about upcoming Artificial Intelligence trends in 2022.

AI becomes increasingly important as organisations continue to automate day-to-day operations and understand COVID-affected datasets. Businesses are more digitally connected than ever since the lockdown and work from home were implemented.

The focus of AI adoption is restricted to improving the efficiency of operations or the effectiveness of operations. It can be leveraged to improve the stakeholder experience as well. Let us take a look at the top trends expected in 2022

Greater Cloud and AI collaboration : Rico Burnett, the director of client innovation at legal services provider Exigent, says that Artificial Intelligence will play a significant role in the broad adoption of Cloud Solutions in 2021. Through the deployment of artificial intelligence, it will be possible to monitor and manage cloud resources and the vast amount of available data.

AI solutions for IT : The number of AI solutions that are being developed for IT will increase in 2021. Capgemini's Simion predicts that AI solutions that can detect common IT problems on their own and self-correct any small malfunctions or issues will see an increase in the upcoming years. This will reduce downtime and allow the teams in an organization to work on high-complexity projects and focus elsewhere.

AIOps become more popular : Over the last few years, the complexity of IT systems has increased. Forrester recently said that vendors would want platform solutions that combine more than one monitoring discipline, such as application, infrastructure, and networking. IT

operations and other teams can improve their key processes, decision-making, and tasks with AIOps solutions and improved analysis of the volumes of data coming its way. Forrester advised the IT leaders to find AIOps providers who will empower the cross-team collaboration through end-to-end digital experiences, data correlation, and integration of the IT operations management toolchain.

AI will help in structuring data : In the future, we will see more unstructured data structured with natural language processing and machine learning processes. Organisations will leverage these technologies and create data that RPA or robotic process automation technology can use when they want to automate transactional activity in an organization. RPA is one of the fastest-growing areas in the software industry. The only limitation that it faces is that it can only use structured data. With the help of AI, unstructured data can easily be converted into structured data, which can provide a defined output. This is one of the most important AI trends.

Artificial intelligence talent will remain tight : The supply of talent is expected to be an issue in adopting artificial intelligence in 2021. There has been a persistent gap in AI talent, and organisations have finally realized this potential. It is essential to address this gap and ensure that a wider group of people learn artificial intelligence. Ensuring that a broader set of users have access to artificial intelligence to focus on technology, learning strategies, and supporting a change in the working environment is essential in 2021. This is one of the most important AI trends.

Large-scale adoption of AI in the IT industry : We have seen continuous growth in the adoption of AI within the IT industry. However, Simion predicts that organisations will use AI in production and start using them on a large scale. With the help of artificial intelligence, an organization can get ROI in real time. This means that organisations will see their efforts being paid off. This is one of the most important AI trends.

AI Ethics is the focus : Natalie Cartwright, co-founder, and COO of Finn AI, an AI banking platform, predicts that in 2021, organisations will deliver expertise on how to leverage artificial intelligence against major global problems, stimulate innovation and economic growth, and ensure inclusion and diversity. As AI ethics become more important to organisations, transparency of data and algorithm fairness are two of the issues that are in the spotlight.

Augmented Processes have become increasingly popular : Artificial intelligence and data science will be a part of the bigger picture regarding innovation and automation in 2021. Data ecosystems are scalable and lean and also provide data on time to heterogeneous sources. However, providing a foundation to adapt and foster innovation is necessary. According to Ana Maloberti, a big data engineer at Globant, companies will go further to optimize their augmented business and development processes. Using Artificial Intelligence, software development processes can be optimized, and we can look for a wider collective intelligence and improved collaboration. We must foster a data-driven culture and grow out of the experimental stages to move into a sustainable delivery model. This is one of the most important AI trends.

Artificial Intelligence will become more explainable : The senior director of product at customer data hub Tealium, Dave Lucas, says that there will be a bigger focus on explainability. As more data regulations come into play, trust in AI will be pivotal. To clearly understand and articulate how each characteristic will contribute to the machine learning model's end prediction or the result.

Voice and Language Driven Intelligence : Particularly in customer care centers, the increase in remote working has driven a great opportunity to adopt NLP or ASR (automated speech recognition) capabilities. Less than 5% of all customer contacts are routinely checked for quality feedback, according to ISG's Butterfield. Due to the lack of one-on-one coaching, organisations can use artificial intelligence to complete routine quality checks on customer understanding and intent to ensure continued compliance.

Limitations of Artificial Intelligence :

1. *High Costs* : The ability to create a machine that can simulate human intelligence is no small feat. It requires plenty of time and resources and can cost a huge deal of money. AI also needs to operate on the latest hardware and software to stay updated and meet the latest requirements, thus making it quite costly.

2. *No creativity* : A big disadvantage of AI is that it cannot learn to think outside the box. AI is capable of learning over time with pre-fed data and past experiences, but cannot be creative in its approach. A classic example is the bot Quill who can write Forbes earning reports. These reports only contain data and facts already provided to the bot. Although it is impressive that a bot can write an article on its own, it lacks the human touch present in other Forbes articles.

3. *Unemployment* : One application of artificial intelligence is a robot, which is displacing occupations and increasing unemployment (in a few cases). Therefore, some claim that there is always a chance of unemployment as a result of chatbots and robots replacing humans.

For instance, robots are frequently utilized to replace human resources in manufacturing businesses in some more technologically advanced nations like Japan. This is not always the case, though, as it creates additional opportunities for humans to work while also replacing humans in order to increase efficiency.

4. *Make Humans Lazy* : AI applications automate the majority of tedious and repetitive tasks. Since we do not have to memorize things or solve puzzles to get the job done, we tend to use our brains less and less. This addiction to AI can cause problems to future generations.

5. *No Ethics* : Ethics and morality are important human features that can be difficult to incorporate into an AI. The rapid progress of AI has raised a number of concerns that one day, AI will grow uncontrollably, and eventually wipe out humanity. This moment is referred to as the AI singularity.

6. *Emotionless* : Since early childhood, we have been taught that neither computers nor other machines have feelings. Humans function as a team, and team management is essential for achieving goals. However, there is no denying that robots are superior to humans when functioning effectively, but it is also true that human connections, which form the basis of teams, cannot be replaced by computers.

7. *No Improvement* : Humans cannot develop artificial intelligence because it is a technology based on pre-loaded facts and experience. AI is proficient at repeatedly carrying out the same task, but if we want any adjustments or improvements, we must manually alter the codes. AI cannot be accessed and utilized akin to human intelligence, but it can store infinite data.

Machines can only complete tasks they have been developed or programmed for; if they are asked to complete anything else, they frequently fail or provide useless results, which can have significant negative effects. Thus, we are unable to make anything conventional.

Application of AI : Application of AI are as follows :

(1) Finance : Banks use artifical intelligence system to organize operations, invest in stocks, and manage properties.

(ii) Hospitals : A medical clinic can use artificial intelligence systems to organize bed schedules make a staff rotation, and provide medical information.

(iii) Heavy industry : Robots have become common in many industries. They are often given jobs that are considered dangerous to humans. Robots have proven effective in jobs that are very respectivies which may lead to mistakes or accidents due to a lapse in concentration and other jobs which humans may find degrading.

(iv) Music : The evolution of music has always been affected by technology. With AI, scientists are trying to make the computer emulate the activities of the skillful musician.

Composition, performance, music theory, sound processing are some of the major areas on which research in Music and Artificial Intelligence are focusing.

(v) Game playing : Games are made by creating human level artificial intelligent entities e.g. enemies, partners, and support character that act just like humans.

RG PUBLICATION