IGENT SYSTEMS

s Work	: 50 Marks
Exam	: 100 Marks
Total	: 150 Marks
on of Exam. :	3 Hrs.

ry of AI, Ai problems and
PROLOG- problem spaces
istic search techniques Hill
-algorithms, game playing-

ing, semantic nets- frames
rules, rules based deduction
probabilistic interferences

oning, Temporal reasoning,
epresentation for planning,

xplanation based learning,
essing, rule based systems
pplication to robotics, and

wo questions from each
. This Q.1 is compulsary
20marks). Students have
h section.

INTELLIGENT SYSTEMS

May 2015

Paper Code: CSE-304-F

Note: *Attempt five questions in all, selecting one question from each section. Q. No. 1 is compulsory.*

Q.1.(a) What is heuristic search ? Give an example. (5)

Ans. Heuristic search is an AI search technique that employs heuristic for its moves. 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. InArtificial 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.

Following is a list of heuristic search techniques.

(1) Pure Heuristic Search
(2) A* algorithm
(3) Iterative-Deepening A*
(4) Depth-First Branch-And-Bound
(5) Heuristic Path Algorithm
(6) Recursive Best-First Search

Q.1.(b) Explain various knowledge representation issues. (5)

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.1.(c) What do you mean by reasoning under uncertainty ? (5)

Ans. 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.

Q.1.(d) Mention current trends in AI. (5)

Ans. Artificial Intelligence is all around us. Fuzzy logic, for example, is widely used in washing machines, cars, and elevator control mechanisms. (Note that no one would claim that as a result those machines were intelligent, or anything like it! They are simply using techniques that enable them to behave in a more intelligent way than a simples control mechanism would allow.)

Intelligent agents are widely used. For example, there are agents that help us to solve problems while using our computers and agents that traverse the Internet, helping us to find documents that might be of interest. The physical embodiment of agents, robots, are also becoming more widely used. Robots are used to explore the oceans and other worlds, being able to travel in environments inhospitable to humans. It is still not the case, as was once predicted, that robots are widely used by households, for example, to carry shopping items or to play with children, although the AIBO robatic dog produced by Sony and other similar toys are a step in this directions.

Expert systems are used by doctors to help with symptoms that are hard to diagnose or to prescribe treatments in case where even human experts have difficulty.

Artificial Intelligence system are used in a wide range of industries, from helping travel agents select suitable holidays to enabling factories to schedule machines.

Artificial Intelligence is particularly useful in situations where traditional methods would be too slow. Combinatorial problems, such as scheduling teachers and pupils to classrooms, are not well solved by traditional computer science techniques. In such cases, the heuristics and techniques provided by Artificial Intelligence can provide excellent solutions.

Many computer games have been designed based on Artificial Intelligence. In order to provide more realistic play, the computer game Republic: The Revolution, launched in 2003, contained a million individual Artificial Intelligence, each capable of interacting with the world and with the player of the game, as well as capable of being manipulated by the player.

It is likely that Artificial intelligence will become more prevalent in our society. And whether or not we eventually create an Artificial Intelligence that is truly intelligent, we are likely to find computers, machines, and other objects appearing to become more intelligent – at least in terms of the way they behave.

Section - A

Q.2.(a) Explain intelligence and artifical intelligence. Also explain major components of AI. (10)

Ans. Intelligence : Intelligence is the computational part of the ability to achieve goals in the world. But there is a variation in kinds and degrees of intelligence that occur in people, many animals and some machines.

Artifical Intelligence : It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computer to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Major components of AI : The major componens of AI are :

(1) The user interface : The user interface is the means of communication between a user and the expert systems problem-solving processes. A good expert system is not very useful unless it has an effective interface. It has to be able to accept the queries or instructions in a form that the user enters and translate them into working instructions for the rest of the system. It also has to be able to translate the answers, produced by the system, into a form that the user can understand. Careful attention should be given to the screen design in order to make the expert system appear 'user friendly'.

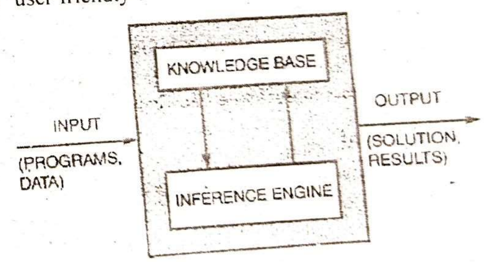

Fig. : Components of AI.

(2) The knowledge base : The knowledge base stores all the facts and rules about a particular problem domain. It makes these available to the inference engine in a form that it can use. The facts may be in the form of background information built into the system or facts that are input by the user during a consultation. The rules include both the production rules that apply to the domain of the expert system and the heuristics or rules-of-thumb that are provided by the domain expert in order to make the system find solutions more efficiently by taking short cuts.

(3) The shall or inference engine : The inference engine is the program that locates the appropriate knowledge in the knowledge base, and infers new knowledge by applying logical processing and problem-solving strategies.

Q.2.(b) Explain hill climbing algorithm. (10)

Ans. Hill Climbing : Hill climbing is a varient of general & 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. Otherwise, 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.

Q.3. Explain A* algorithm. Is A* algorithm guaranteed to find an optimal goal path if one exits. (10)

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.

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

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.

Section -B

Q.4.(a) Describe the properties of knowledge representation system. (10)

Ans. Properties of Knowledge Representation Systems : A knowledge representation system should possess the following properties:

(1) Representation Adequacy : Knowledge Representation Systems must have the ability to represent the required knowledge adequately (correctly).

(2) Inferential Adequacy : It is the ability to manipulate the knowledge represented to product new knowledge corresponding to that inferred from the original.

(3) Inferential Efficiency : It is the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides.

(4) Acquisitional Efficiency : It is the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

To date no single system optimises all of the above.

Q.4.(b) What do you mean by knowledge aquisition ? Write different methods used for knowledge acquisition. (10)

Ans. 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.

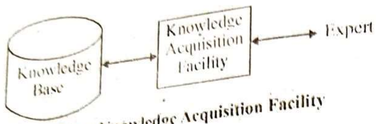

Fig. : Knowledge Acquisition Facility

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.

Many techniques have been developed to help elicit knowledge from an expert. These are referred to as knowledge elicitation or knowledge acquistion (KA) techniques. The term "KA techniques" is commonly used.

The following list gives a brief introduction to the types of techniques used for acquiring, analysing and modelling knowledge :

Protocol-generation Techniques : Protocol-generation techniques include various types of interviews (unstructured, semi-structured and structured), reporting techniques (such as self-report and shadowing) and observational techniques.

Protocol analysis Techniques : Protocol analysis techniques are used with transcripts of interviews or other text-based information to identify various types of knowledge, such as goals, decisions, relationships and attributes. This acts as a bridge between the use of protocol-based techniques and knowledge modeling techniques.

Hierarchy-generation Techniques : Hierarchy-generation techniques such as laddering, are used to build taxonomies or other hierarchical structures such as goal trees and decision networks.

Matrix-based Techniques : Matrix-based techniques involve the construction of grids indicating such things as problems encountered against prossible solutions. Important types include the use of frames for representing the properties of concepts and the repertory grid technique used to elicit, rate, analyse and categorise the properties of concepts.

Sorting techniques : Sorting techniques are used for capturing the way people compare and order concepts, and can lead to the revelation of knowledge about classes, properties and priorities.

Limited-information and constrained-processing tasks : Limited-information and constrained-processing tasks are techniques that either limits the time and/or information available to the expert when performing tasks. For instance, the twenty-question technique provides an efficient way of accessing the key information in a domain in a prioritised order.

Diagram-based techniques : Diagram-based techniques include the generation and use of concept maps, state transition networks, event diagrams and process maps. The use of these is particularly important in capturing the "what, how, when, who and why" of tasks and events.

Q.5. Write notes on : (10 + 10 = 20)

(a) Semantic nets.

(b) Baye's theorem.

Ans.(a) 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).

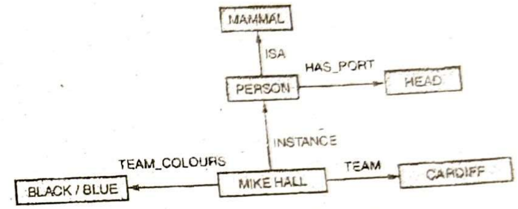

Fig (a) : A Semantic Network

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. Semantic 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.

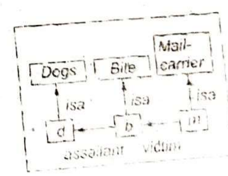

Fig (b) : Using Partitioned Semantic Nets

Ans. (b) Baye's theorem : Bayesian view of probability is related to degree of belief. It is a measure of the plausibility of an event given incomplete knowledge.

Bayes' theorem is also knwon as Bayes' rule or Bayes' law; or called bayesian reasoning.

The probability of an event A conditional on another even B i.e. P(A|B) is generally different from probability of B conditional on A is P(B|A).

There is a definite relationship between the two, P(A|B) and P(B|A), and Bayes' theorem is the statement of that relationship.

Bayes' theorem is a way to calculate P(A|B) from a knowledge of P(B|A).

Bayes' theorem is a result that allows new information to be used to update the conditional probability of an event.

Bayes' Theorem : Let S be a sample space.

Let $A_1, A_2, ..., A_n$ be a set of mutually exclusive events from S.

Then Bayes' Theorem descibes following two probabilities :

$$P(Ak|B) = \frac{P(Ak \cap B)}{P(A_1 \cap B) + P(A_2 \cap B) + ... + (A_n \cap B)}$$

and by invoking the fact $p(Ak \cap B) = P(Ak).P(B|Ak)$ the probability

$$P(Ak|B) = \frac{P(Ak).P(B|Ak)}{P(A_1).P(B|A_1) + P(A_2).P(B|A_2) + ... + P(A_n).P(B|A_n)}$$

Bayes' theorem is applied while following conditions exist ;

- The sampe space S is partitioned into a set of mutually execlusive events $\{A_1, A_2, ...A_n\}$
- within S there exists an event B, for which P(B) > 0.
- the goal is to compute a conditional probability of the form : P(Ak|B).
- you know at least one of the two sets of probabilities escibed below

(i) $P(A_k \cap B)$ for each A_k

(ii) $P(A_k)$ and $P(B | A_k)$ for each A_k.

Section - C

Q.6.(a) What do you mean by planning ? Explain various steps in planning process. **(10)**

Ans. 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 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 method 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 things 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.

Basic planning problem:

(i) *Given* : start state, goal conditions, actions

(ii) *Find* : sequence of actions leading from start to goal

(iii) *Typically* : states correspond to possible worlds, actions and goals specified using a logical formalism. (e.g., STRIPS, situation calculus)

Various steps in planning process : The process of planning includes the determination of objectives and outlining the future actions that are needed to achieve these objectives. Various steps that are followed in the process of planning are:

(i) Identifying the problem: It involves the identification of the aim for the fulfillment of which the plan is being formulated. If a new plan is require or the modification of an existing plan could help in achieving these aims.

(ii) Gathering information about the activities involve: An effective plan needs complete knowledge of the activities involved and their effect on other external and internal activities.

(iii) Analysis of information: This information is then analysed minutely and the information, related with similar subjects is classified so that similar type of data can be kept together.

(iv) Determining alternate plans: There are alternate plans available for the achievement of the objectives and ingenuity and creativeness are required as some plans are also developed at this stage.

(v) Selecting the plan: At this stage the plan which is acceptable to the operating personnel is proposed. The adaptability and the cost of the plan are also taken into consideration.

(vi) Detailed seauence and timing: Detailed like who will perform which activity under the plan and the time within which the plan should be carried out is determining in this step.

(vii) Progress check of the plan: The provisions are made for the follow up of the plan as the success of any plan can be measured by the results only.

Q.6.(b) Explain non monotonic reasoning. **(10)**

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:

(1) Defaulat reasoning

(2) Circumscription

(3) Truth maintenance systems.

(1) Default Reasoning : This is a very common from of non-monotonic resoning. The conclusions are drawn based on what is most likely to be true. There are two approaches, both are logic type, to Default reasoning; One is Non-monotonic logic and the other is Default logic.

Nonmonotonic logic : It has already been defined. It says, "the truth of a proposition may change when new information (axioms) are added and a logic may be build to allows the statement to be retracted."

Example :

$\forall$ x : plays_instrument(x) $\wedge$ M manage (x) - > jazz_musician(x)

States that for all x, the x plays an instrment and if the fact that x can manage is consistent with all other knowledge then we can conclude that x is a jazz musician.

Default logic : Default logic initiates a new inference rule : $\frac{A:B}{C}$ where

A is known as the prerequisite,

B as the justification, and

C as the consequent

Read the above inference rule as :

"If A, and if it is consistent with the rest of what is known to assume that B, then conclude that C".

The rule says that given the prerequisite, the consequent can be inferred, provided it is consistent with the rest of the data.

Example : Rule that "birds typically fly" would be represented as

$$\frac{bird(x) : flies(x)}{flies(x)}$$ which says

"If x is a bird and the claim that x flies is consistent with what we know, then infer that x flies".

(2) Circumscription : Circumscription is a non-monotonic logic to formalize the common sense assumtion. Circumscription is a formalized rule of conjecture (gusess) that can be used along with the rules of inference of first order logic.

Circumscription involves formulating rules of thumb with "abnormality" predicates and then restricting the extension of these predicates, circumscribing them, so that they apply to only those things to which they are currently known.

Circumscription can also cope with default reasoning.

Suppose we know : bird(tweety)

$\forall x$: penguin (x) → bird (x)

$\forall x$: penguin (x) → ¬flies (x)

and we wish to add the fact that typically, birds fly.

In circumscription this phrase would be stated as :

A bird will fly if it not abnormal

and can thus be represented by :

$\forall x$: bird (x) $\wedge$ ¬abnormal (x) → flies (x)

However, this is not sufficient

We cannot conclude

flies (tweety)

since we cannot prove

¬abnormal (tweety).

This is where we apply circumscription and, in this case, we will assume that those things that are shown to be abnormal are the only things to be abnormal. Thus we can rewrite our default rule as :

$\forall x$: bird (x) $\wedge$ ¬flies (x) → abnormal (x)

and add the following :

$\forall x$: ¬abnormal (x)

since there is nothing that cannot be shown to be abnormal.

If we now add the fact :

penguin (tweety)

Clearly we can prove

abnormal (tweety).

If we circumscribe abnormal now we would add the sentence, a penguin (tweety) is the abnormal thing :

$\forall x$: abnormal (x) → penguin (x)

(3) Truth Maintenance Systems : Truth maintenance systems have been implemented to permit a form of nonmonotonic reasoning by permitting the addition of changing the statements to a knowledge base.

A truth maintenance system, or TMS, is a knowledge representation method for representing both beliefs and their dependencies. The name truth maintenance is due the ability of these systems to restore consistency.

(10 + 10 = 20)

Q.7.(a) Write notes on :

(a) Fuzzy reasoning

(b) Partial order planning.

Ans.(a)Fuzzy reasoning : 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.

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

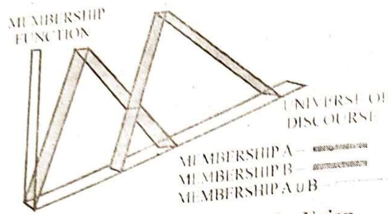

Fig.(1) : Membership function for Union.

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.

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

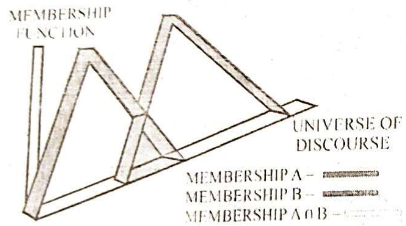

Fig.(2) : Membership function for Intersection.

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.

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

De Morgans Law : $(A \cap B) = A \cap B; (A \cap B) = A \cap B$

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

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

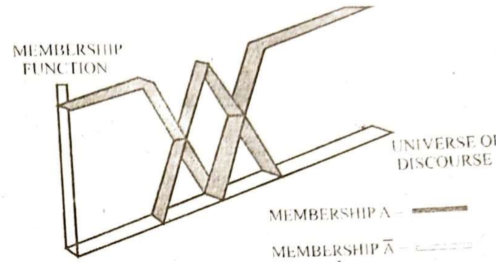

Fig.(3) : Membership function for Complement.

Ans. (b)Partial order planning : In formal AI planning, a partial plan is a plan which specifies all actions that need to be taken, but doesnt' specify an exact order for the actions as the order doesn't matter.

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 & milk is not specified, the agent can wonder around the store reactively accumulating all the items on its shopping list until the list is complete.

Partial order planning enables us to "take advantage of problem decomposition". The algorithm works on several subgoals independently, solve them with several subplans, and then combines the subplans. In addition, "such an approach also has the advantage of flexibility in the order in which it constructs the plan". That is, the planner can work on 'obvious' or 'important' decision first, rather than being forced to work on steps in chronological order.

Section –D

Q.8. What is an expert system ? Describe various componets of an expert system. Mention some advantages and disadvantages of expert systems. (10)

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 components 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.

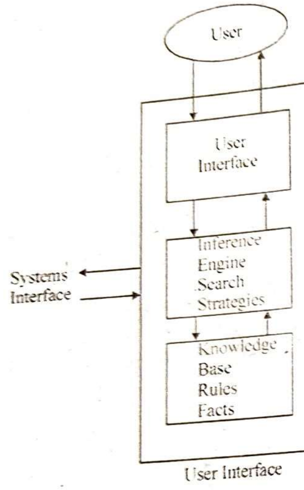

Fig.(a)

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.

Advantages of Expert Systems :

(1) Permanence : Expert systems do not forget, but human experts may.

(2) Reproducibility : Many copies of an expert system can be made, but training new human experts is time-consuming and expensive.

(3) If there is a maze of rules (e.g. tax and auditing), then the expert system can "unravel" the maze.

(4) Efficiency : can increase throughput and decrease personnel costs. Although expert systems are expensive to build and maintain, they are inexpensive to operate . Development and maintenance costs can be spread over many users. The overall cost can be quite reasonable when compared to expensive and scarce human experts. Cost savings: Wages - (elimination of a room full of clerks) Other costs - (minimize loan loss).

(5) Consistency : With expert systems similar transactions handled in the same way. The system will make comparable recommendations for like situations.

(6) Humans are influenced by recency effects (most recent information having a disproportionate impact on judgment) primacy effects (early information dominates the judgment).

(7) Documentation : An expert system can provide permanent documentation of the decision process.

(8) Completeness : An expert system can review all the transactions, a human expert can only review a sample

(9) Timeliness : Fraud and/or errors can be prevented. Information is available sooner for decision making.

(10) Breadth : The knowledge of multiple human experts can be combined to give a system more breadth that a single person is likely to achieve.

Disadvantages of Expert Systems :

(1) Common sense : In addition to a great deal of technical knowledge, human experts have common sense. It is not yet known how to give expert systems common sense.

(2) Creativity : Human experts can respond creatively to unusual situations, expert systems cannot.

(3) Learning : Human experts automatically adapt to changing environments; expert systems must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.

(4) Sensory Experience : Human experts have available to them a wide range of sensory experience; expert systems are currently dependent on symbolic input.

(5) Degradation : Expert systems are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

(10 + 10 = 20)

Q.9. Write notes on :

(a) Rule based system.

(b) Natural language processing.

Ans. (a) 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).

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 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.

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.

INTELLIGENT SYSTEMS

May 2016
Paper Code: CSE-304-F

Note: *Attempt five questions in all, selecting one question from each section. Q. No. 1 is compulsory.*

Q.1. Explain the following 4 × 5 = 20

(a) Effect of over-estimation and under-estimation of h'(n) on A* algorithms.

(b) Various knowledge representation issues.

(c) Reasoning under uncertainty.

(d) AI applications to robotics.

Ans.(a) Effect of over estimation and under estimation of h'(n) on A algorithms:

Underestimating h'(n) : Referring to fig.(1), let us assume that A* applied and we are at the intermediate stage that is mentioned. Here, it has been removed from the open list and the stages (where we are having the nodes/successor expanded) are represented. Now we can see that here Q node's evaluation function $f(Q) = h(Q) + g(Q)$ is equal to 6. Since this is the lowest cost, this node is selected. Next, S is the only successor of Q. $f(S)$ is same as $f(P)$. Here we continue with our expansion, so we reach to T. Now, $f(T) = 9$, whereas $f(P) = 7$; so we go back and expand P. Thus, due to underestimation of $h'(P)$, efforts an memory are wasted.

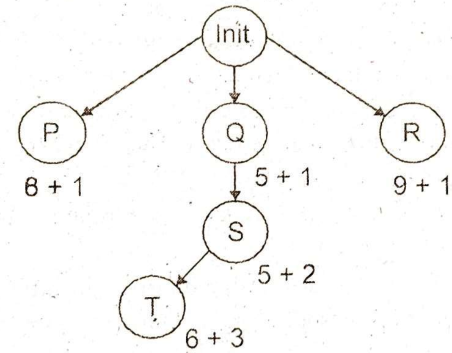

Fig.(1) : Underestimating h.

Overestimating ***h'(n)*** **:** Referring to fig.(2), now consider, where the heuristic function is overestimated. In the process of reaching to the goal state, we can see that from node Q, we move to S. $f(S) = 7$, and hence, being the lowest, it is selected. A head in the process, T is selected as $f(T) = 8$ and then we reach to U(say our goal state). Suppose there is direct path from P to goal state giving path length less than 3, we will not be able to find it. So we can always say that if h' over estimates h, then, we have no assurance of getting the cheapest path.

Then the question is raised that if this overstimation does not occur, is the algorithm of A^* admissible ? We cannot guarantee whether h' will underestimate or overestimate A^*. The only way is to set it to zero, but then this wil lead to BFS.

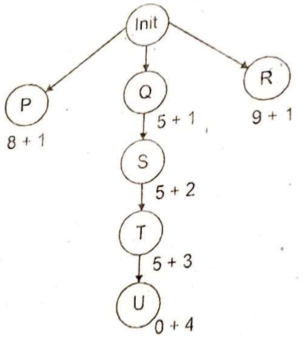

Fig.(2) : Overestimating h.

Here, heuristic $h(n)$ is admissible, if for every node the heuristic cost $h(n)$, is always less than or equal to the actual cost to reach goal state. The heuristic that never over estimates the cost is an admissible heuristic.

Ans.(b) Various knowledge representation issues : 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.

Ans.(c) Reasoning under uncertainty : Unfortunately the world is an uncertain place. 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 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.

Ans.(d) 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.

Section - A

Q.2.(a) Write a LISP expression that reverse a list at all levels. For example reversall1 [((a b c) ((d e) f) g)] to return (g (f (e d)) (c b a))]. (10)

Ans.
```
(dfun revall1 (1)
    (cond
        ( (nu ll 1) 1)
        ( (atom (car 1)) (append (revall1 (cdr 1) )(cons (car 1) NIL)))
        ( T (append (reva ll 2 (cdr 1) )(cons (revall 1 (car 1) ) NIL) ) )
    )
)
(defun reva ll 2 (1)
    (cond
        ( (nu ll 1) 1)
        ( (atom 1) 1)
        (T (append (revall 2 (cdr 1) )
                (cons (revall 2 (car 1)), NIL)))
    )
)
(defun revall 3 (1)
    (cond
        ((atom 1) 1)
        (T (append (revall 3 (cdr 1))
                (cons (revall 3 (car 1)) NIL)))
    )
)
```

Q.2.(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:

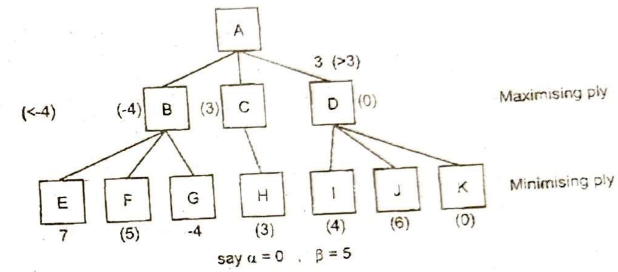

Fig. : α–β pruning

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 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 various problems and their solutions in hill climbing algorithm. **(10)**

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.

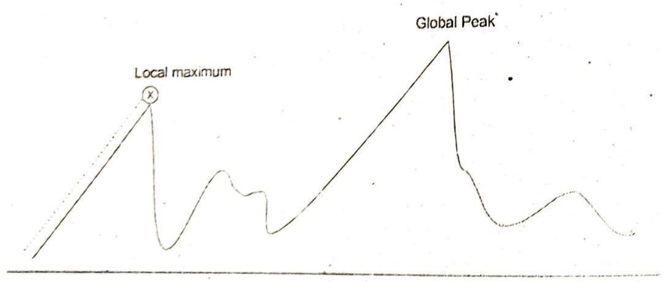

Fig. (a)

(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.

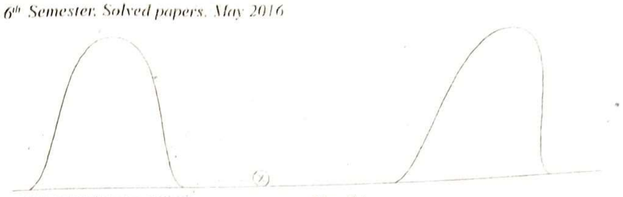
Fig. (b)

(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.

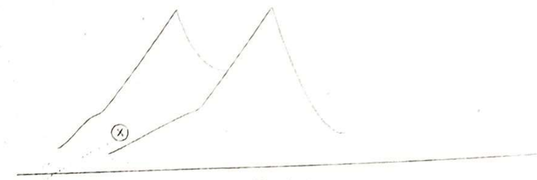
Fig. (c)

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.(b) Discuss the relative merits of BFS and DFS. Suggest some applications to which each is best suited ? **(10)**

Ans. Advantages of Breadth First Search (BFS) : The breadth first search is not caught in a blind alley. This means that, it will not follow a single unfruitful path for very long time or forever, before the path actually terminates in a state that has no successors. In the situations where solution exists, the breadth first search is guaranted to find it. Besides this, in the situations where there are multiple solutions, the BFS finds the minimal solution. The minimal solution is one that requires the minimum number of steps. This is because of the fact that is breadth first search, the longer paths are never explored until all shorter ones have already been examined. Thus, if the goal state is found during the search of shorter paths, longer paths would not be required to be searched, saving time and efforts.

Travelling sales person problem can be solved using Breadth-First Search technique. It will simply explore all the paths possible in the tree and will ultimately come out with the shortest path desired. However, this strategy works well only if the number of cities is less. If we have large number of cities in the list, It fails miserably because number of paths hence the time takento perform the search become too big to be controlled by this method efficiently.

Advantages of depth first search : Depth First search strategy has following advantages:

(i) Requires less space and hence, less memory, since nodes on the current path we stored.

(ii) It may find a solution without examining much of search space, because we may get the desired solution in the very first go. Hence for the problems which have only one solution or one considered sufficient, this technique is advantages.

Out of the two most elementary search techniques described, the depth-first search (DFS) has some problems as compared to breadth-first search (BFS). The DFS, unlike BFS may follow a single unfruiful path for a very long time. Theoretically, in the situation when there are no successors, only then it will stop searching. In the problems, where production rules from a loop, it is probable that DFS may be stuck up in the loop.

Also, the DFS does not guarantee to find an optimal solution because as soon as a solution is found, it will stop the search and this solution may not be the optimal one. It may find the answer in more number of steps by unnecessary exploring the wrong paths.

On the other hand, BFS technique takes lot of time and becomes unmangeable if number of paths is large. It is more suitable if the problem has more than one solution and we have to find optimal solution. However, for the problems having single solution, it may take unnecessary time in exploring all the paths in spite of getting a solution early.

Hence, a better approach would be device the search strategies that combine the advantages of both DFS and BFS.

Section - B

Q.4.(a) What do you mean by knowledge acquisition ? Write different methods used for knowledge acquisition. (10)

Ans. 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.

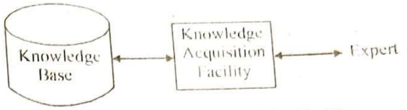

Fig. : Knowledge Acquisition Facility

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.

Many techniques have been developed to help elicit knowledge from an expert. These are referred to as knowledge elicitation or knowledge acquistion (KA) techniques. The term "KA techniques" is commonly used.

The following list gives a brief introduction to the types of techniques used for acquiring, analysing and modelling knowledge :

Protocol-generation Techniques : Protocol-generation techniques include various types of interviews (unstructured, semi-structured and structured), reporting techniques (such as self-report and shadowing) and observational techniques.

Protocol analysis Techniques : Protocol analysis techniques are used with transcripts of interviews or other text-based information to identify various types of knowledge, such as goals, decisions, relationships and attributes. This acts as a bridge between the use of protocol-based techniques and knowledge modeling techniques.

Hierarchy-generation Techniques : Hierarchy-generation techniques such as laddering, are used to build taxonomies or other hierarchical structures such as goal trees and decision networks.

Matrix-based Techniques : Matrix-based techniques involve the construction of grids indicating such things as problems encountered against possible solutions. Important types include the use of frames for representing the properties of concepts and the repertory grid technique used to elicit, rate, analyse and categorise the properties of concepts.

Sorting techniques : Sorting techniques are used for capturing the way people compare and order concepts, and can lead to the revelation of knowledge about classes, properties and priorities.

Limited-information and constrained-processing tasks : Limited-information and constrained-processing tasks are techniques that either limits the time and/or information available to the expert when performing tasks. For instance, the twenty-question technique provides an efficient way of accessing the key information in a domain in a prioritised order.

Diagram-based techniques : Diagram-based techniques include the generation and use of concept maps, state transition networks, event diagrams and process maps. The use of these is particularly important in capturing the "what, how, when, who and why" of tasks and events.

Q.4.(b) Write down about the procedural as well as declarative knowledge in brief. **(10)**

Ans. Declarative Knowledge : It is the knowledge, which gives the simple facts about any organization or phenomenon. Declarative knowledge means representation of facts or assertions. This tells 'what' about a situation, e.g., the facts about the college like its building, its courses, location, organizational setup consist of declarative knowledge. The facts may be static facts, or dynamic facts. The static facts do not change with time whereas the dynamic facts changes with time, e.g. in the collage, it may be possible that the location of college is permanent, then it will become static fact, howeve, the new courses may be added in curriculum pattern of collage, then it will become dynamic fact. The frame representation method can be used to represent static knowledge, i.e. it gives the information about existing things.

Procedural knowledge : The declarative knowledge not tell anything regarding functioning of the concerned object. In context of collage, it does not tell anything about running of the college. It does not tell how a student is examined, how syllabus is framed, how fees deposit is made etc The procedural knowledge represents the functioning of organization. It describes dynamic attributes using production rules, e.g., following rules may give information regarding operation of a college :

If : Student has deposited fees and
Student has opted a course and
Student has attended 90% classes and
Student has passes the examination
Then : declare the student pass.

Or

If: Student has scored distinction in all subjects and
student has good communication skills and
Student has participated in extra-curricular activities
Then : Student is the best student

Similarly, procedures regarding other operations of a college may be declared. The procedural known in AI program is represented as production rules. These rules can be easily coded in lisp structures.

Q.5. Write notes on :
(a) Statistical reasoning **(10)**
(b) Dempster shafer theory. **(10)**

Ans. (a) Statistical reasoning : Refer Q.7(a) of paper May 2017.

(b) Dempster Shafer Theory : Refer Q.4(b) of paper May 2017.

Section -C

Q.6.(a) What do you mean by planning ? Explain various steps in planning process. **(10)**

Ans. 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 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 method 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 things 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.

Basic planning problem:

(i) *Given* : start state, goal conditions, actions

(ii) *Find* : sequence of actions leading from start to goal

(iii) *Typically* : states correspond to possible worlds, actions and goals specified using a logical formalism. (e.g., STRIPS, situation calculus)

Various steps in planning process : The process of planning includes the determination of objectives and outlining the future actions that are needed to achieve these objectives. Various steps that are followed in the process of planning are:

(i) Identifying the problem: It involves the identification of the aim for the fulfillment of which the plan is being formulated. If a new plan is require or the modification of an existing plan could help in achieving these aims.

(ii) Gathering information about the activities involve: An effective plan needs complete knowledge of the activities involved and their effect on other external and internal activities.

(iii) Analysis of information: This information is then analysed minutely and the information related with similar subjects is classified so that similar type of data can be kept together.

(iv) Determining alternate plans: There are alternate plans available for the achievement of the objectives and ingenuity and creativeness are required as some plans are also developed at this stage.

(v) Selecting the plan: At this stage the plan which is acceptable to the operating personnel is proposed. The adaptability and the cost of the plan are also taken into consideration.

(vi) Detailed sequence and timing: Detailed like who will perform which activity under the plan and the time within which the plan should be carried out is determining in this step.

(vii) Progress check of the plan: The provisions are made for the follow up of the plan as the success of any plan can be measured by the results only.

Q.6.(b) Describe planning in situational calculus. (10)

Ans. 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. Write note on : (20)

(a) Symbolic reasoning

(b) Fuzzy reasoning

Ans. (a) 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.

Ans.(b)Fuzzy reasoning : 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.

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

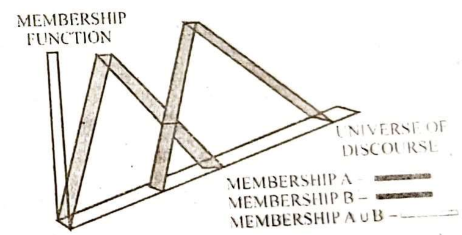

Fig.(1) : Membership function for Union.

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.

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

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.

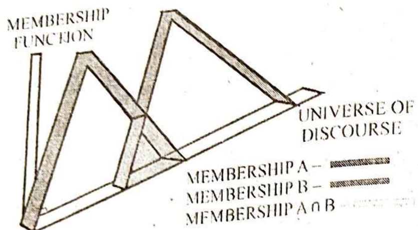

Fig.(2) : Membership function for Intersection.

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

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

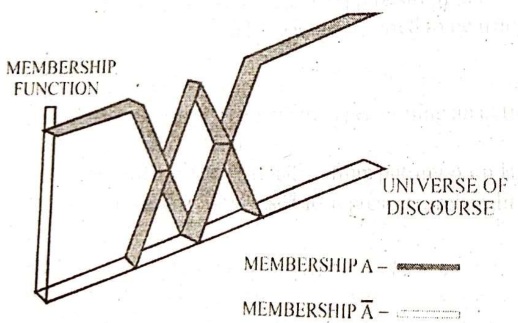

Fig.(3) : Membership function for Complement.

Section - D

Q.8. What is an expert system ? Describe various components of an expert system ? Compare a human expert with an artificial expert system. (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 components 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 widely 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 guessing".

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.

Human Express V/S Expert Systems : Human experts have distinct advantages over expert system. On the other hand, expert system have advantages over human experts also in more than one aspect.

Advantages of Human Expert Over Expert System : Although modern technology has facilitated developement of expert systems that compete well with the expertise of human experts, it would be a mistake to overestimate the ability of expert systems while comparing those with human experts. Human experts still have advantages over the machines in some specific areas, as mentioned below :

(i) Depth of knowledge base : Human experts not only possesses deep knowledge of the problem domain, but also are able to apply heuristic knowledge and common sense as and when deep required. On the other hand expert system find difficulty in capturing deep knowledge. Their functioning depends upon the knowledge base they posses and that limitation of knowledge affects the output the provide of a particular problem. They also lack in application of common sense knowledge, e.g., MYCIN does not posses any knowledge about human physiology. It does not know anything about the function of spinal chord. A famous legendary saying is that once MYCIN asked whether the patient was pregnant during selection of drugs for treatment of meningitis, even though it was told that the patient was a male. This might be a joke in its true sense, such incidents indicates towards the narrowness of knowledge of expert systems. A human expert will hardly commit such type of mistake.

(ii) Ability to provide explanations: Human experts are not only able to provide best possible solution of a problem in a hand, but also are able to explain why is the solution most appropriate and about the approach applied to find the solution of a particular problem. Expert systems are not able to explain the above facts because of the lack of deep knowledge they posses.

(iii) Flexibility and robustness : If humans are encounted with a problem that is difficult for them to solve immediatly, they can wait and find some strategy to solve the problem. Expert systems cannot do so. Whatever result they will produce today, the same they will provide ever after.

(iv) Learning from experience : Human experts keep on adding to their knowledge by experience they acquire over the years without and extra effort and they are able to use that enhanced knowledge with the problems that may come before them late on. Whereas expert systems once built will not learn anything from experience and their performance will remain same unless modifications are done to their program.

In spite of the limitations of expert systems mentioned above, they have proved their worth in a number of applications. Because of this fact, development of expert systems is a burning area of research and their use in almost all the areas is increasing day-by-day.

Advantages of Expert Systems Over Human Experts : Although it is hard to eliminate human experts, expert systems have taken over them in many areas because of the ease, efficiency and economy in using expert systems for solving intricate problems. Some of the advantages expert systems posses over the humans are mentioned below :

(i) Easy availability : Experts system once developed is easily available to any computer. One expert system can be used at different places at the same time. Thus, expert systems are tools of mass production. Human experts are not easily available and they have limitations of working hours. In addition, one expert can deal with one problem at a time a single place.

(ii) Economy : Use of expert systems is much more economical compared to humans.

(iii) Permanence : The expertise of expert systems is forever. Human experts may quit or die, but expert systems will last indefinitely.

(iv) Multiple expertise : Multiple expert systems can be used to work simultaneously on one problem and the combined expertise may exceed that of a single human expert. It is difficult to use multiple humans to work on a single problem at a time.

(v) Fast response : Expert systems respond faster than humans depending upon the hardware and software used in developing them. It is beneficial to use expert systems where fast response is required in some emergency situations.

(vi) Steady and unemotional : Expert systems do not have emotions, hence their results are unaffected by circumstances. On the other hand, results produced by human experts are sometimes affected by emotions and may not be ideal solutions of the problem. Also, the efficiency of the expert remains same irrespective to the duration of their use, while human experts get fatigued and their working efficiency tends to decrease with time. Expert systems are steady and will provide same reults of the same problem every time, whereas human experts may come out with different resutls of the same problem some other time.

(vii) Intelligent tutor : Expert systems can act as tutor to someone who is interested in learning the reasoning and approach use in solving a particular problem by using the system for various examples. This facility is not available with human expert.

(viii) Intelligent database : Expert systems can be used to access a database and the same can be used for some other purpose, whereas it is impossible to read the heart and mind of a human expert.

(ix) Refined knowledge base : The knowledge base of expert systems is built by acquiring the knowledge from humans. The implicit knowledge present in human's mind is converted into explicit knowledge before entering into knowledge base of an expert systems and this knowledge can be checked and refined before doing so. Hence, quality of knowledge possessed by expert systems is improved. Knowledge possessed by human experts on the other hand, cannot be exploited by others and hence cannot be refined or corrected if need be.

Q.9. Write notes on : (20)

(a) Knowledge acquisition concepts

(b) Natural language processing

(c) Genetic algorithm.

Ans. (a) 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 therr 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. (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 level's 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. [illegible] 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 NLU 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.

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.

(ii) 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.

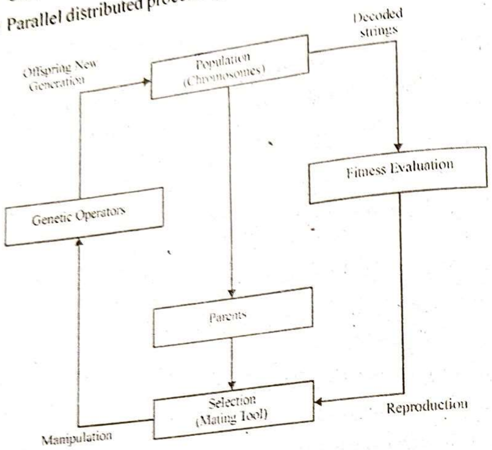

Fig.

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.

INTELLIGENT SYSTEM

May - 2017
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 : (4 × 5 = 20)

(a) Blind searh and heuristic search technique.
(b) Various knowledge representation issues
(c) Partial order planning.
(d) Genetic algorithm

Ans. (a) Breadth-first search : This is also a brute-force technique. Here searching progress level by level unlike *DFS* which goes deep into the tree. An operator is employed to generate all possible children of a node.

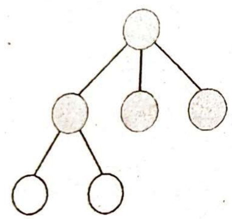

Fig. : Breadth-first search: Expand all the nodes of one level first.

The search action consists of :

(1) Selecting a node as a current node.

(2) Testing the current node for meeting the goal test criteria.

(3) If the goal state is not reached, select the next node in a breadth-wise order.

(4) The search ends either when all the nodes have been searched or if the goal has been found.

Heuristic Searching : Heuristic search is an AI search technique that employs heuristic for its moves. *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

Ans.(b) Various knowledge representation issues : 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.

Ans.(c) Partial order planning : In formal AI planning, a partial plan is a plan which specifies all actions that need to be taken, but doesnt' specify an exact order for the actions as the order doesn't matter.

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

Þ Go to the store,

Þ Get eggs; get flour, get milk

Þ Pay for all goods.

Þ Go to the kitchen

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

Partial order planning enables us to "take advantage of problem decomposition". The algorithm works on several subgoals independently, solve them with several subplans, and then combines the subplans. In addition, "such an approach also has the advantage of flexibility in the order in which it constructs the plan". That is, the planner can work on 'obvious' or 'important' decision first, rather than being forced to work on steps in chronological order.

Ans.(d) 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 :

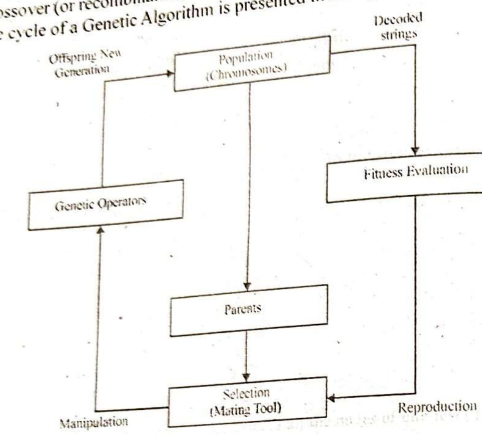

Fig.

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.

Section - A

Q.2.(a) Explain intelligent and artificial intelligence. How do you distinguish between the two ? Give different definitions based on various form of intelligence. (10)

Ans. Intelligence : Intelligence is the computational part of the ability to achieve goals in the world. But there is a variation in kinds and degrees of intelligence that occur in people, many animals and some machines.

Artifical Intelligence : It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computer to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

Distinguish between the intelligent and artificial intelligence : Intelligence is the quality of being intelligent. It is the quality of a being to be well-informed and smart. Artificial intelligence, on the other hand, are computers that can be well trained to think like humans do.

Different definitions based on various form of intelligence :

"Intelligence is the ability to learn, to deal with different situations to a acquire, understand and apply knowledge and to analyze and reason."

"Intelligence is a state grasping the truth, involving reasons concerned with action about what is good or bad for human being."

"AI is the branch of engineering employed for the creation of computers that possess some form of intelligence and can be used to solve real world problems and function within a limited."

"AI is the automation of activities that we associate with human thinking activities such as decision making, problem solving learing ..."

Bellman, 1978

" AI is concerned with designing intelligent computer systems which exhibit the characteristic we associate with intelligence in human behiviour."

-Barr and Feignenbaum, 1981

"AI is the exciting new effort to make computes think ..... machines with minds, in the full and literal sense."

-Haugeland, 1985

"AI is the study of mental faculties through the use of computational models."

-Charniak and McDermott, 1985

"AI is the art of creating machines that perform functions that require intelligence when performed by people."

-Kurzwell, 1990

"AI is the study of how to make computers do things at which, at the moment, people are better."

-Rich and Knight, 1991

"AI is the study of the combinations that make it possible to perceive, reason and act.

-Winston, 1992

Q.2.(b) Write a program in PROLOG to solve a tower a Hanoi Problem. (10)

Ans. The tower of Hanoi problem requires moving of disks from one tower to another in such a way that

- One and only one disk is to be moved at one time
- A large disk may neverr be placed on a smaller disk.

The prolog code is as follows :

```
domains
loc = a ; b ; c ;
predicates
    tower_of_hanoi (integer)
    move (integer, loc, loc, loc)
    output (loc, loc)
clauses
    tower_of_hanoi (N) :- move (N, a, b, c).
    move (1, A, B, ) :- output (A, B)
    move (N, A, B, C)
    N > = 2,
    N_1 = N – 1,
    move (N_1, A, C, B).
    output (A, B)
    move (N_1, C, B, A).
output (X, Y) :- write ("\n move on disk from ",X, "to " Y).
```

running this code will be as follows.

Goal : tower_of_hanoi (4)

Move on disk from *a* to *c*
Move on disk from *a* to *b*
Move on disk from *c* to *b*
Move on disk from *a* to *c*
Move on disk from *b* to *a*
Move on disk from *b* to *c*
Move on disk from *a* to *c*
Move on disk from *a* to *b*
Move on disk from *c* to *b*
Move on disk from *c* to *a*
Move on disk from *b* to *a*
Move on disk from *c* to *b*
Move on disk from *a* to *c*
Move on disk from *a* to *b*
Move on disk from *c* to *b*

Q.3. Explain AO* algorithm. Is AO* algorithm guaranteed to terminate even on graphs that have cycles. (20)

Ans. AO* Algorithm : In this algorithm a single structure GRAPH is used. Each node in the graph will point both down to its immediate successor and upto its immediate predecessor. Each node in the graph will have a associated value *h* and value of *g* is not stored in this algorithm. FUTILITY IS THE TERM USED in this algorithm and it is the threshold to stop the algorithm i.e. if the value of the estimated cost of the solution becomes greather than *FUTILTY*. THEN SEARCH STOPS.

Algorithm :

(1) Initialise the graph to start node.

(2) Traverse the graph following the current path accumulating nodes that have not yet been expanded or solved.

(3) Pick any of these nodes and expand it and if it has no successors call this values *FULTILITY* otherwise calculate only f' for each of the successors.

(4) If f' is 0 then mark the node as SOLVED.

(5) Change the value of f' for the newly created node to reflect its sucessors by back propagation.

(6) Wherever possible use the most promising routes and if a node is marked as *SOLVED* then mark the parent node as *SOLVED*.

(7) If starting node is *SOLVED* or value greater than *FUTILITY*, stop, else repeat from 2.

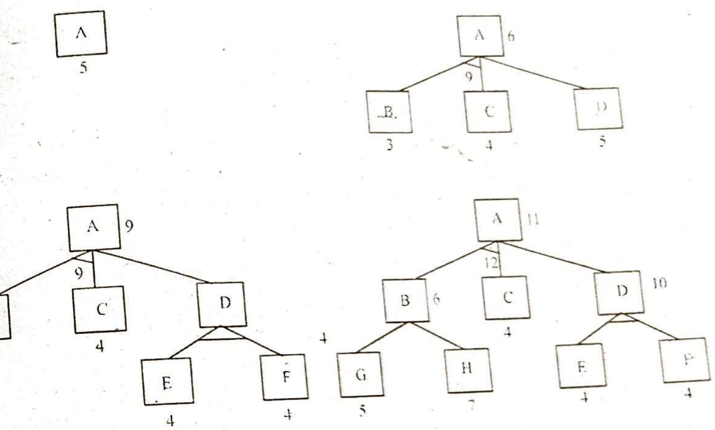

Fig. : AO* Algorithm

The process is explained as follows with the help of example.

Step 1 : *A* is only node.

Step 2 : *A* is expanded thus giving three nodes *B*, *C*, *D*. it is assumed that every operation has a uniform cost, so each are with a single successor has cost of 1 and each AND are with multiple successors has a cost of 1 for each component. Now as value at *B* is 3. but there is an are between *B* and *C*, therefore total cost at the arc becomes 3 + 4 + 1 +1 i.e. *C* + *B* + 2 and if we use path through *D* then total cost becomes *D* + 1 i.e. 5 + 1 = 6. As 6 < 9, therefore node *D* is chosen for expansion.

Step 3 : This process produces new AND are *E* and *F* with combined cost of 10 (*E* + *F* +2 i.e. 4 + 4 + 2). Therefore the cost of *D* becomes 10 (updated). AS the AND are between *B* and *C* is better than the arc between *E* and *F* (because 9 < 10) so it is lageled as current best path. In this step A is traversed and found that B and C are not expanded yet. We have to expand B and C eventually. Any node can be taken B or C. Lets choose B first.

Step 4 : When B is expanded, two new nodes are G and H. Now B is updated and its value becomes 6. This also requires updating the cost of AND are BC to 12.

After this step again D is the better path from A, so it is recoded as the best path and either node E or F will be chosen for expansion in this step 4. This process continues until a solution is found or all paths have led to dead ends.

Yes. AO is guaranteed to terminate even on graphs that have cycles.*

Section - B

Q.4.(a) Describe the properties of knowledge representation system. (10)

Ans. Properties of Knowledge Representation Systems : A knowledge representation system should possess the following properties:

(1) Representation Adequacy : Knowledge Representation Systems must have the ability to represent the required knowledge adequately (correctly).

(2) Inferential Adequacy : It is the ability to manipulate the knowledge represented to product new knowledge corresponding to that inferred from the original;

(3) Inferential Efficiency : It is the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides.

(4) Acquisitional Efficiency : It is the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

To date no single system optimises all of the above.

Q.4.(b) Describe Dempster Shafer theory. (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 $\{\theta, (F), (C), (P), (F, C), (F, P), (C, P), (F, C, P)\}$.

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.5. Write notes on : (20)

(a) Rule based deduction systems

(b) Baye's theorem

Ans. (a) Rule based deduction systems : 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 charactẽristics 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).

Ans.(b) Baye's theorem : Bayesian view of probability is related to degree of belief. It is a measure of the plausibility of an event given incomplete knowledge.

Bayes' theorem is also knwon as Bayes' rule or Bayes' law, or called bayesian reasoning.

The probability of an event A conditional on another even B i.e. P(A|B) is generally different from probability of B conditional on A is P(B|A).

There is a definite relationship between the two, P(A|B) and P(B|A), and Bayes' theorem is the statement of that relationship.

Bayes' theorem is a way to calculate P(A|B) from a knowledge of P.(B|A).

Bayes' theorem is a result that allows new information to be used to update the conditional probability of an event.

Bayes' Theorem : Let S be a sample space.

Let $A_1, A_2, ..., A_n$ be a set of mutually exclusive events from S.

Then Bayes' Theorem descibes following two probabilities :

$$P(Ak|B) = \frac{P(Ak \cap B)}{P(A_1 \cap B) + P(A_2 \cap B) + ... + (A_n \cap B)}$$

and by invoking the fact $p(Ak \cap B) = P(Ak).P(B|Ak)$ the probability

$$P(Ak|B) = \frac{P(Ak).P(B \mid Ak)}{P(A_1).P(B \mid A_1) + P(A_2).P(B \mid A_2) + ... + P(A_n).P(B \mid A_n)}$$

Bayes' theorem is applied while following conditions exist :

- The sampe space S is partitioned into a set of mutually execlusive events $\{A_1, A_2, ...A_n\}$

- within S there exists an event B, for which P(B) > 0.
- the goal is to compute a conditional probability of the form : P(Ak|B).
- you know at least one of the two sets of probabilities escibed below
  (i) $P(A_k \cap B)$ for each A_k
  (ii) $P(A_k)$ and $P(B \mid A_k)$ for each A_k.

Section - C

Q.6.(a) Explain non-monotonic reasoning. (10)

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:

(1) Defaulat reasoning

(2) Circumscription

(3) Truth maintenance systems.

(1) Default Reasoning : This is a very common from of non-monotonic resoning. The conclusions are drawn based on what is most likely to be true. There are two approaches, both are logic type, to Default reasoning; One is Non-monotonic logic and the other is Default logic.

Nonmonotonic logic : It has already been defined. It says, "the truth of a proposition may change when new information (axioms) are added and a logic may be build to allows the statement to be retracted."

Example :

$\forall$x : plays_instrument(x) $\wedge$ M manage (x) → jazz_musician(x)

States that for all x, the x plays an instrment and if the fact that x can manage is consistent with all other knowledge then we can conclude that x is a jazz musician.

Default logic : Default logic initiates a new inference rule : $\frac{A : B}{C}$ where

A is known as the prerequisite,

B as the justification, and

C as the consequent

Read the above inference rule as :

"If A, and if it is consistent with the rest of what is known to assume that B, then conclude that C".

The rule says that given the prerequisite, the consequent can be inferred, provided it is consistent with the rest of the data.

Example : Rule that "birds typically fly" would be represented as

$$\frac{bird(x) : flies(x)}{flies(x)}$$ which says

"If x is a bird and the claim that x flies is consistent with what we know, then infer that x flies".

(2) Circumscription : Circumscription is a non-monotonic logic to formalize the common sense assumtion. Circumscription is a formalized rule of conjecture (gusess) that can be used along with the rules of inference of first order logic.

Circumscription involves formulating rules of thumb with "abnormality" predicates and then restricting the extension of these predicates, circumscribing them, so that they apply to only those things to which they are currently known.

Circumscription can also cope with default reasoning.

Suppose we know : bird(tweety)

$\forall x$: penguin $(x) \rightarrow$ bird (x)

$\forall x$: penguin $(x) \rightarrow \neg$flies (x)

and we wish to add the fact that typically, birds fly.

In circumscription this phrase would be stated as :

A bird will fly if it not abnormal

and can thus be represented by :

$\forall x$: bird $(x) \wedge \neg$ abnormal $(x) \rightarrow$ flies (x)

However, this is not sufficient

We cannot conclude

flies (tweety)

since we cannot prove

$\neg$abnormal (tweety).

This is where we apply circumscription and, in this case, we will assume that those things that are shown to be abnormal are the only things to be abnormal. Thus we can rewrite our default rule as :

$\forall x$: bird $(x) \wedge \neg$ flies $(x) \rightarrow$ abnormal (x)

and add the following :

$\forall x$: $\neg$ abnormal (x)

since there is nothing that cannot be shown to be abnormal.

If we now add the fact :

penguin (tweety)

Clearly we can prove

abnormal (tweety).

If we circumscribe abnormal now we would add the sentence, a penguin (tweety) is the abnormal thing :

$\forall x$: abnormal $(x) \rightarrow$ penguin (x)

(3) Truth Maintenance Systems : Truth maintenance systems have been implemented to permit a form of nonmonotonic reasoning by permitting the addition of changing the statements to a knowledge base.

A truth maintenance system, or TMS, is a knowledge representation method for representing both beliefs and their dependencies. The name truth maintenance is due the ability of these systems to restore consistency.

Q.6.(b) What do you mean by planning? Explain various steps in planning process. **(10)**

Ans. 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 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 method 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 things 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.

Basic planning problem:

(i) *Given* : start state, goal conditions, actions

(ii) *Find* : sequence of actions leading from start to goal

(iii) *Typically* : states correspond to possible worlds, actions and goals specified using a logical formalism. (e.g., STRIPS, situation calculus)

Various steps in planning process : The process of planning includes the determination of objectives and outlining the future actions that are needed to achieve these objectives. Various steps that are followed in the process of planning are:

(i) Identifying the problem: It involves the identification of the ai[illegible] or the fulfillment of which the plan is being formulated. If a new plan is require or the modification of an existing plan could help in achieving these aims.

(ii) Gathering information about the activities involve: An effective plan needs complete knowledge of the activities involved and their effect on other external and internal activities.

(iii) Analysis of information: This information is then analysed minutely and the information related with similar subjects is classified so that similar type of data can be kept together.

(iv) Determining alternate plans: There are alternate plans available for the achievement of the objectives and ingenuity and creativeness are required as some plans are also developed at this stage.

(v) Selecting the plan: At this stage the plan which is acceptable to the operating personnel is proposed. The adaptability and the cost of the plan are also taken into consideration.

(vi) Detailed sequence and timing: Detailed like who will perform which activity under the plan and the time within which the plan should be carried out is determining in this step.

(vii) Progress check of the plan: The provisions are made for the follow up of the plan as the success of any plan can be measured by the results only.

Q.7. Write notes on : **(20)**
(a) Statistical reasoning
(b) Temporal reasoning.

Ans. (a) Statistical reasoning : An important goal for many problem solving systems is to collect evidence as the system goes along and to modify its behavior on the basis of evidence. To model this behavior, we need statistical theory of evidence and Bayesian statistics is such theory.

The fundamental notation of Bayesian statistics is that of condition probability. 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 summaries large exceptions without resorting enumeration.

Ans.(b) 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 various components and applications of an expert system. Mention some advantages and disadvantages of expert system.(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.

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.

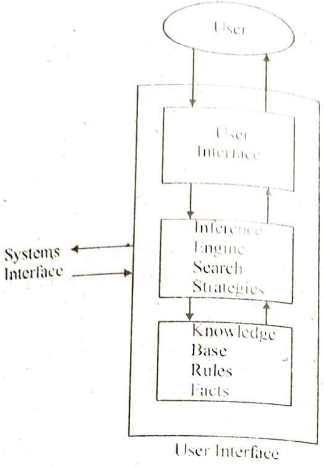

Fig.(a)

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.

Advantages of Expert Systems :

(1) *Permanence* : Expert systems do not forget, but human experts may.

(2) *Reproducibility* : Many copies of an expert system can be made, but training new human experts is time-consuming and expensive.

(3) If there is a maze of rules (e.g. tax and auditing), then the expert system can "unravel" the maze.

(4) *Efficiency* : can increase throughput and decrease personnel costs. Although expert systems are expensive to build and maintain, they are inexpensive to operate . Development and maintenance costs can be spread over many users. The overall cost can be quite reasonable when compared to expensive and scarce human experts. Cost savings: Wages - (elimination of a room full of clerks) Other costs - (minimize loan loss).

(5) *Consistency* : With expert systems similar transactions handled in the same way. The system will make comparable recommendations for like situations.

(6) Humans are influenced by recency effects (most recent information having a disproportionate impact on judgment) primacy effects (early information dominates the judgment).

(7) *Documentation* : An expert system can provide permanent documentation of the decision process.

(8) *Completeness* : An expert system can review all the transactions, a human expert can only review a sample

(9) *Timeliness* : Fraud and/or errors can be prevented. Information is available sooner for decision making.

(10) *Breadth* : The knowledge of multiple human experts can be combined to give a system more breadth that a single person is likely to achieve.

Disadvantages of Expert Systems :

(1) *Common sense* : In addition to a great deal of technical knowledge, human experts have common sense. It is not yet known how to give expert systems common sense.

(2) *Creativity* : Human experts can respond creatively to unusual situations, expert systems cannot.

(3) *Learning* : Human experts automatically adapt to changing environments; expert systems must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.

(4) *Sensory Experience* : Human experts have available to them a wide range of sensory experience; expert systems are currently dependent on symbolic input.

(5) *Degradation* : Expert systems are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

Q.9. What do you mean by natural language processing. Explain various stages of natural language processing. Also mention need and advantages of natural language processing. (20)

Ans. 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.

Various stages of natural language processing : The analysis of Natural Language is broken into various broad stages 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.

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.

Need of natural language processing : There are a number of different NLP tasks incorporated into software programs today, including:

(i) Sentence segmentation, part-of-speech tagging, and parsing : Natural language processing can be used to analyze parts of a sentence to better understand the grammatical construction of the sentence.

(ii) Deep analytics: Deep analytics involves the application of advanced data processing techniques in order to extract specific information from large or multi-source data sets. Deep analytics is particularly useful when dealing with precisely targeted or highly complex queries with unstructured and semi-structured data. Deep analytics is often used in the financial sector, the scientific community, the pharmaceutical sector, and biomedical industries. Increasingly, however, deep analysis is also being used by organizations and companies interested in mining data of business value from expansive sets of consumer data.

(iii) Machine translation: Natural language processing is increasingly being used for machine translation programs, in which one human language is automatically translated into another human language.

(iv) Named entity extraction: In data mining, a named entity definition is a phrase or word that clearly identifies one item from a set of other items that have similar attributes. Examples include first and last names, age, geographic locations, addresses, phone numbers, email addresses, company names, etc. Named entity extraction, sometimes also called named entity recognition, makes it easier to mine data.

(v) Co-reference resolution: In a chunk of text, co-reference resolution can be used to determine which words are used to refer to the same objects.

(vi) Automatic summarization: Natural language processing can be used to produce a readable summary from a large chunk of text. For example, one might us automatic summarization to produce a short summary of a dense academic article.

Advantages of natural language processing : The benefits of natural language processing are innumerable. Natural language processing can be leveraged by companies to improve the efficiency of documentation processes, improve the accuracy of documentation, and identify the most pertinent information from large databases. For example, a hospital might use natural language processing to pull a specific diagnosis from a physician's unstructured notes and assign a billing code.

INTELLIGENT SYSTEMS

May - 2018

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.(a) Differentiate between Depth first search and Breath first search with example. (5)

Ans. Difference between Depth first search and Breadth first search :

(1) Depth first search selects any one of the adjacent vertices as the start vertex. Then any one of its adjacent vertices is visited. When no further movement is possible, backtracking takes place to visit the unvisited adjacent vertices. In breadth first search, we first visit all the adjacent vertices of the start vertex and then visit all the unvisited vertices adjacent to these, and so on.

(2) Depth first search wents its way through the graph, storing on the stack the points where other paths branch off. Breadth first search 'sweeps through' the graph, using a queue to remember the frontier of the visited places.

(3) Depth first search 'explores' the graph by looking for a new vertex for away from the start point, taking closer vertices only when dead ends are encountered. Breadth first search completely covers the area close to the starting point, moving further away only when everything close has been looked at.

Q.1.(b) What is knowledge base ? How it is different from Database ? (5)

Ans. 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.

Q.1.(c) What are the various uses of intelligent systems ? Briefly explain. (5)

Ans. Various uses of intelligent systems are as follows :

Game playing: You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation—looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

Speech recognition : In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

Understanding natural language : Just getting a sequence of words into a computer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

Computer vision : The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

Expert systems : A "knowledge engineer" interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the

knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expert systems depends on their users having common sense.

Heuristic classification : One of the most feasible kinds of expert system given the present knowledge of AI is to put some information in one of a fixed set of categories using several sources of information. An example is advising whether to accept a proposed credit card purchase. Information is available about the owner of the credit card, his record of payment and also about the item he is buying and about the establishment from which he is buying it (e.g., about whether there have been previous credit card frauds at this establishment).

Q.1.(d) Differentiate between Monotonic and Non-monotonic reasoning. Explain with examples. (5)

Ans. Monotonic reasoning : In monotonic reasoning if we enlarge at set of axioms we cannot retract any existing assertions or axioms. Axioms are sentence that are true with the system.

Most formal logics have a monotonic consequence relation, meaning that adding a formula to a theory never produces a reduction of its set of consequences. In other words, a logic is monotonic if the truth of a proposition does not change when new information (axioms) are added. The traditional logic is monotonic.

But human reasoning is non-monotonic in nature this means that we reach to the conclusion from certain premises that we would not reach if certain other sentences are included in our premises.

The non-monotonic human reasoning is caused by the fact that our knowledge about the world is always incomplete and therefore we are forced to reason in the absence of complete information. Therefore, we often revise our conslusions, when new information becomes available.

Thus, the need for non-monotonic reasoning in AI was recognized, and several formalizations of non-monotonic reasoning.

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.

Section – A

Q.2.(a) Define brute force search and heuristic searching techniques with the help of suitable examples. **(10)**

Ans. Brute Force : The most general search algorithms are brute-force searches since they do not require any domain specific knowledge. All that is required for a brute-force search is a state description, a set of legal operators, an initial state, and a descriptions of the goal state. So brute-force search is also called *uninformed search* and *blind search*.

Brute-force search should proceed in a systematic way by exploring nodes in some predetermined order or simply by selecting nodes at random. Search programs either return only a solution value when a goal is found or record and return the solution path.

Various types of Brute Force searching techniques are :

(1) Depth-first search (DFS)
(2) Breadth-first search

(1) Depth-first search : This is a very simple type of brute-force technique. The search begins by expanding the initial node, i.e. by using an operator, generate all successors of initial node and the test them. One of the most important features for a depth-first search is the ability to keep track of the nodes that have already been visited.

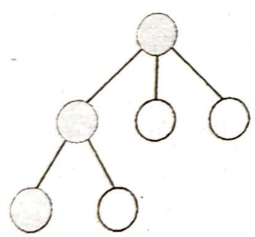

Fig. : Depth-first search :Expand one of the nodes at the deepest level

Algorithm :

Step 1 : Put the initial node on a list START.
Step 2 : If (START is empty) or START = GOAL terminate search.
Step 3 : Remove the first node from START. Call this node as *A*.
Step 4 : If (*A* = GOAL) terminate search with success.
Step 5 : Else if node *A* has Successors, generate all of them and add them at the begining of START.
Step 6 : Goto step 2.

(2) Breadth-first search : This is also a brute-force technique. Here searching progress level by level unlike *DFS* which goes deep into the tree. An operator is employed to generate all possible children of a node.

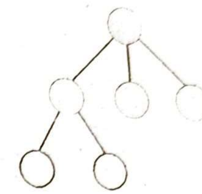

Fig. : Breadth-first search: Expand all the nodes of one level first.

The search action consists of :

(1) Selecting a node as a current node.
(2) Testing the current node for meeting the goal test criteria.
(3) If the goal state is not reached, select the next node in a breadth-wise order.
(4) The search ends either when all the nodes have been searched or if the goal has been found.

Algorithm :

Step 1 : Put the initial node on a list START.
Step 2 : If (START is empty) or START = GOAL terminate search.
Step 3 : Remove the first node from START. Call this node as *A*.
Step 4 : If (*A*=*GOAL*) terminate search with success.
Step 5 : Else if node *A* has successors, generate all of them and add them at the tail of START.
Step 6 : Goto step 2.

Heuristic Searching : Heuristic search is an AI search technique that employs heuristic for its moves. *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

Hill Climbing : Hill climbing algorithm, also called discrete optimization algorithm, uses a simple heuristic function viz., amount of distance the node is from the goal. The ordering of choices is a heuristic measure of the remaining distance one has to traverse to reach the goal node.

In fact, there is practically no difference between hill-climbing and depth-first search except that the children of the node that has been expanded are sorted by the remaining distance.

Algorithm :

Step 1 : Put the initial node on a list START

Step 2 : If (START is empty) or (START = GOAL) terminate search

Step 3 : Remove the first node from START. Call this node a

Step 4 : If (a = GOAL) terminate search success

Step 5 : Else if node a has successors, generate all of them. Find out how far they are from the goal node. Sort them by the remaining distance from the goal and add them to the beginning of START.

Step 6 : Goto Step 2.

Best-First Search : There is only a minor variation between hill-climbing and best-first search. In the former, we sorted the children of the first node being generated. Here, we have to sort the entire list to identify the next node to be expanded.

Algorithm :

Step 1 : Put the initial node on a list START

Step 2 : If (START is empty) or (START = GOAL) terminate search

Step 3 : Remove the first node from START. Call this node a

Step 4 : If (a = GOAL) terminate search with success

Step 5 : Else if node a has successors, generate all of them. Find out how far they are from the goal node. Sort all the children generated so far by the remaining distance from the goal.

Step 6 : Name this list as START 1

Step 7 : Replace START with START 1

Step 8 : Goto Step 2.

A* Algorithm : If it is possible for one to obtain the evaluation function values and the cost function values, then A* algorithm can be used. The basic principle is that sum the cost and evaluation function values for a state to get its "goodness" worth and use this as a yardstick instead of the evaluation function value in best-first search.

Algorithm :

Step 1 : Put the initial node on a list START

Step 2 : If (START is empty) or (START = GOAL) terminate search

Step 3 : Remove the first node from START. Call this node a

Step 4 : If (a = GOAL) terminate search with success

Step 5 : Else if node a has successors, generate all of them. Estimate the fitness number of the successors by totaling the evaluation function value and the cost-function value. Sort the list by fitness number.

Step 6 : Name the new list as START 1

Step 7 : Replace START with START 1

Step 8 : Goto Step 2.

Beam Search : This is an attractive heuristic search tehcnique because it permits searching to be done on a multi-processor machine, thereby reducing computations.

The searching process is similar to breadth-first search wherein searching proceeds level by level. At each level, heuristic functions are applied to reduce the number of paths to be explored.

Algorithm :

Step 1 : Let width of beam = w

Step 2 : Put the initial node on a list START

Step 3 : If (START is empty) or (START = GOAL) terminate search

Step 4 : Remove the first node from START. Call this node a

Step 5 : If (a = GOAL) terminate search with success

Step 6 : Else if node a has successors, generate all of them and add them at the tail of START

Step 7 : Use a heuristic function to rank and sort all the elements of START

Step 8 : Determine the nodes to be expanded. The number of nodes should not be greater than w. Name these as START 1

Step 9 : Replace START with START 1

Step 10 : Goto Step 2.

Q.2.(b) Explain AO* algorithm with the help of example. (10)

Ans. AO* Algorithm : In this algorithm a single structure GRAPH is used. Each node in the graph will point both down to its immediate successor and upto its immediate predecessor. Each node in the graph will have a associated value h and value of g is not stored in this algorithm. FUTILITY IS THE TERM USED in this algorithm and it is the threshold to stop the algorithm i.e. if the value of the estimated cost of the solution becomes greather than *FUTILTY*, THEN SEARCH STOPS.

Algorithm : (1)Initialise the graph to start node.

(2) Traverse the graph following the current path accumulating nodes that have not yet been expanded or solved.

(3) Pick any of these nodes and expand it and if it has no successors call this values *FULTILITY* otherwise calculate only f for each of the successors.

(4) If f is 0 then mark the node as SOLVED.

(5) Change the value of f for the newly created node to reflect its sucessors by back propagation.

(6) Wherever possible use the most promising routes and if a node is marked as *SOLVED* then mark the parent node as *SOLVED*.

(7) If starting node is *SOLVED* or value greater than *FUTILITY*, stop, else repeat from 2.

The process is explained as follows with the help of example.

Step 1 : A is only node.

Step 2 : A is expanded thus giving three nodes B, C, D. it is assumed that every operation has a uniform cost, so each are with a single successor has cost of 1 and each AND are with multiple successors has a cost of 1 for each component. Now as value at B is 3, but there is an are between B and C, therefore total cost at the arc becomes $3 + 4 + 1 + 1$ i.e. $C + B + 2$ and if we use path through D then total cost becomes $D + 1$ i.e. $5 + 1 = 6$. As $6 < 9$, therefore node D is chosen for expansion.

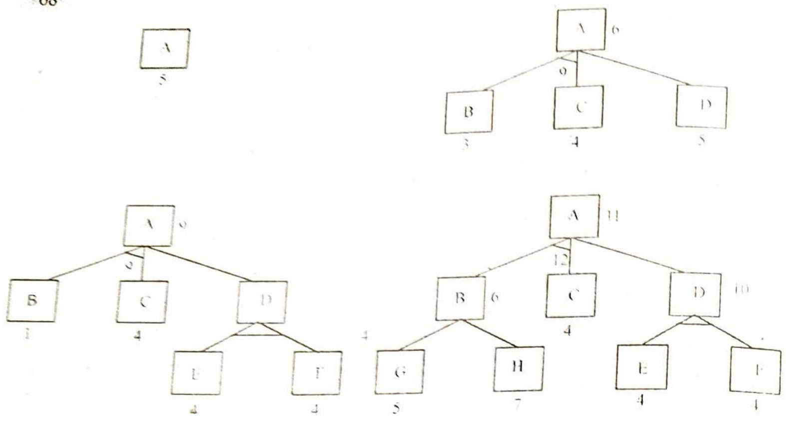

Fig. : AO* Algorithm

Step 3 : This process produces new AND are E and F with combined cost of 10 ($E + F + 2$ i.e. $4 + 4 + 2$). Therefore the cost of D becomes 10 (updated). AS the AND are between B and C is better than the arc between E and F (because $9 < 10$) so it is lageled as current best path. In this step A is traversed and found that B and C are not expanded yet. We have to expand B and C eventually. Any node can be taken B or C. Lets choose B first.

Step 4 : When B is expanded, two new nodes are G and H. Now B is updated and its value becomes 6. This also requires updating the cost of AND are BC to 12.

After this step again D is the better path from A, so it is recoded as the best path and either node E or F will be chosen for expansion in this step 4. This process continues until a solution is found or all paths have led to dead ends.

Q.3.(a) Discuss different features of LISP and Prolog. (10)

Ans. Prolog vs Lisp : Prolog and Lisp are two of the most popular AI (Artificial Intelligence) computer programming languages today. They are built with two different programming paradigms. Prolog is a declarative language, while Lisp is a functional language. Both are used for various AI problems but Prolog is used most for logic and reasoning problems, while Lisp is used for problems with rapid prototyping needs.

Prolog : Prolog is an AI programming language. It belongs to the family of logic programming languages. Prolog is a declarative language, in which computations are carried over by running queries over the relations (which represent program logic), which are defined as rules and fact. Developed in 1970, prolog is one of the oldest logic programming languages and one of the most popular AI programming languages today (along with Lisp). It is a free language, but many commercial variants are available. It was first used for natural language processing, but now it is been used for various tasks such as expert systems, automated answering systems, games and advanced control systems. Prolog has only one data type called the term. A Term can

be an atom, number, variable or a compound term. Numbers can be float or integers. Prolog supports lists and string as collection of items. Prolog defines relations using clauses. Clauses can be either rules or facts. Prolog allows iteration thorough its recursive predicates.

Lisp : Lisp is a family of computer programming languages. And the most famous Lisp dialects used for general purpose programming today are Common Lisp and Scheme. The name LISP comes from "LISt Processing" and as it hints, Lisp's major data structure is the linked list. In fact the whole source is written using lists (using prefix notation), or more correctly parenthesized lists (called s-expressions). For example, a function call is written as (f a1 a2 a3), which means function f is called using a1, a2 and a3 as input arguments for the function. Therefore it is called an expression oriented language, where all data and code are written as expressions (there is no distinction between expressions and statements in Lisp). This nice feature is very special to Lisp, where it could be used to extend the language to the problem at hand by writing helpful macros. Although tail-recursion is used by programmers to express loops, all frequently seen Lisp dialects do include control structures like loop. Furthermore, Common Lisp and scheme have mapcar and map that are examples of functions, which provide looping functionality by applying the function successively to all its elements and then collects the results in to a list.

What is the difference between Prolog and Lisp?

Although, Prolog and Lisp are two of the most popular AI programming languages, they have various differences. Lisp is a functional language, while Prolog is a logic programming and declarative languages. Lisp is very flexible due to its fast prototyping and macro features, so it actually allows extending the language to suit the problem at hand. In the areas of AI, graphics and user interfaces, Lisp has been used extensively because of this rapid prototyping ability. However, due to its inbuilt logic programming abilities, Prolog is ideal for AI problems with symbolic reasoning, database and language parsing applications. Choice of one over the other completely depends on the type of AI problem that need to be solved.

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:

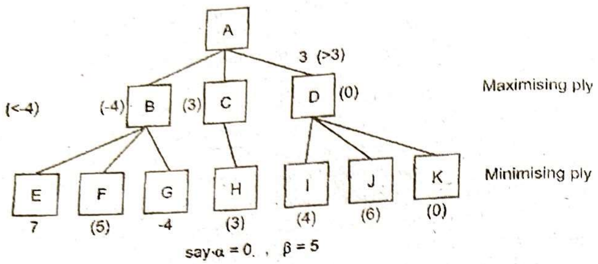

Fig. : α–β pruning

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 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.

Section – B

Q.4. Explain Dempster Shafer Theory. How does it remove the disadvantages of Bayes Probability Inference. Using Dempster Shafer Approach, find the uncertainty of the following prediction.

"There are 80% chances of rain today. However there is uncertainty regarding the type of cloud cover. Some experts tell he is confident that there are 90% chances of these types of clouds bringing rains." **(20)**

Ans. Dempster Shafer Theory : DST is mathematical theory of evidence based on belief functions and plausible reasoning. It is used to combine separate pieces of information (evidence) to calculate the probability of an event.

The idea is to allocate a number between 0 and 1 to indicate degree of belief on a proposal as in the probability framework.

However, it is not considered a probability but a belief mass the distribution of masses is called basic belief assignment.

Need of Dempster-Shaft 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 to Tail would also be 0.5

An alternative parametrization 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 with in 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 member 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.

Dempster-Shafter Theory considers set of propositions and assigns to each of them on interval

[Belief, plausibility]

In which the degree of belief must lie Belief[usually denote Bel] measures the strength of the evidence in favour of a set of propositions. It ranges from 0 (indicating no evidence) 0 to 1(denoting certainty).

Plausibility (P1) is denned to be

$$P1(s) = 1 - Bil(-s)$$

It also ranges from 0 to 1 and measures the extent to which evidence in favour of –s leaves room for belief in *s*. In particular, if we have certain evidence in favour of –*s*, then Bell (–*s*) will be 1 and: 1(*s*) will be 0. This tells us that the only possible value of Bel (*s*) is also 0.

The belief-plausibility interval we have just defined measure not only our level of belief in some propositions, but also the amount of information we have suppose that we are currently considering three comperting hypothesis : *A*, *B* and *C*. If we have no information, we represent that by saying, for each of them, that the true likelihood is in the range [0,1]. As evidence is accumulated, this interval can be expected to shrink, representing increased confidence that we know how likely each hypothesis is. Note that this contrasts with a pure Bay's probability inference, in which we would probably begin by distributing the prior probability equally among the hypothesis and thus assort for each that $P(h) = 0.33$. The interval approach makes it clear that we have no information when we start. The Bay's probability inference does not, since we could end up with same probability values if we collected volumes of evidence, which taken together suggest that the three value, occur equally often.

- Assign probabilities to all combinations of values for all the random variables, based on observations.
- The sum must be 1.
- Given the joint, uncondítonal and conditional probabilities can be compared.

	Sunny	Rainy	Cloudy	Snowing
Cold	0.01	0.10	0.04	0.20
Hold	0.50	0.05	0.10	0.00

- P(hot) = 0.50
- P(hot) = 0.50 + 0.05 + 0.10 + 0.00 = 0.65
- P(hot & sunny) = 0.01 + 0.50 + 0.10 + 0.00 = 0.65
- P(hot & sunny) = P(hot) + P(sunny) – P(hot & sunny) = 0.65+0.51–0.50 = 0.66
- P(hot/sunny) = P(hot & sunny)/P(sunny) = 0.50/0.51 = 0.98

Given probability of rain, P(R) = 0.8,
hence P(notR) = 0.2.
The confidence associated with the evidence, i.e. clouds is 0.9,
hence
Belief for rain, bel(R) = 0.8 × 0.9 = 0.72, and
bel(notR) = 0.2 × 0.9 = 0.18.
Thus, plausibility of rain, pl(R) = 1 – bel(notR) = 1 – 0.18 = 0.82
The belief interval is [0.82, 0.72]
Hence, uncertainty associated with the prediction = pl(R) – bel(R)
= 0.82 – 0.72
= 0.10

Hence, uncertainty associated with the prediction = 10%

Q.5. Write a short note on with example : **(20)**
(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).

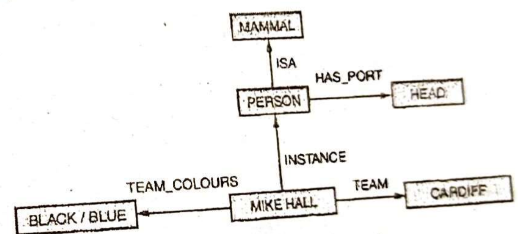

Fig (a) : A Semantic Network

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.

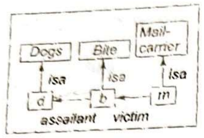

Fig (b) : Using Partitioned Semantic Nets

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

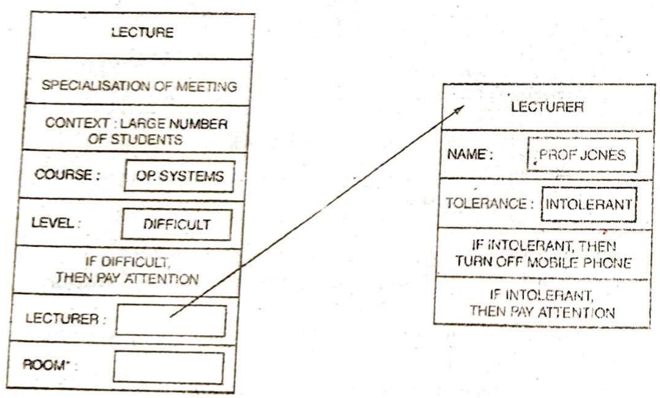

Fig.

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 :

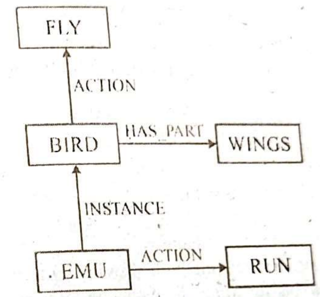

Fig.(1) : A Semantic Network for a Default Reasoning

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.

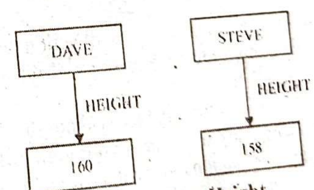

Fig.(2) : Two Height

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