Artificial intelligence

- Saroj Kaushik

Scanned with CamScanner

ARTIFICIAL INTELLIGENCE

Assignment - I

1) Briefly Explain about the history of Artificial intelligence?

The history of AI had cycles of success and failures but kept on introducing new creative approaches and systematically refining the best ones. There was no relation between human intelligence and machine till Early 1950 Even though philosophers in 400 BC. had concieved the ideas that the mind operates on knowledge encoded in some internal language in the same ways as machine does.

John McCarthy organized a conference on machine intelligence in 1956 and since then the field was known as Artificial intelligence. In 1957, the first version of a new program named as General problem solver (GPS) was developed and tested. This program was also developed by Newell and Simon. The GPS was capable of solving to some Extent the problems requiring common sense. Since then, many programs were developed. and McCarthy announced his new development called LIst processing language (LISP) in 1958. LISP was adopted as the language of choice bymost AI developers.

Scanned with CamScanner

Marvin Minsky of MIT demostrated that computer programs could solve spatial and logic programs when confined to a specific domain. Another program, named STUDENT, was developed during late 1960 which could solve algebra story problems fuzzy set and logic was developed by L. Zadeh in 1960 that had the unique ability to make decisions under uncertain conditions. Also neural networks were considered as possible ways of achieving artificial intelligence. During the same time, the system named SHRDLU was developed by Terry Winograd at the MIT, AI Laboratory, as a part of the micro worlds project, which consisted of research and programming in small worlds. SHRDLU was a program that carried out a simple dialogue with a user in English. Frame theory was one of the new methods developed by Minsky during 1970 in the develop-ment of AI for storing structured knowledge to be used by AI programs. Another development during the same time was the PROLOG language that was initially proposed by R Kowalski of Imperial college, London.

Advent of the Expert System appeared in 1970, when Expert systems were designed and developed to predict the probability of a solution under set conditions.

Scanned with CamScanner

Research Organizations and the coorporate sector during 1980 started developing AI systems at faster pace. Companies such as Digital Electronics were using XCON, an Expert system designed to the large VAX computers. Work in AI has started from agent point of view. So agents can be used in solving any problem which requires intelligence. Understanding of a problem and situations is embedded in agents and the problem is solved.

2) Write about the intelligent system, its categorization and its components?

Intelligent system:

AI is a combination of computer science, physiology, and philosophy. AI is a broad topic, consisting of different fields, from machine vision to Expert systems. John McCarthy was one of the founders of AI field who stated that AI is the science and Engineering of making intelligent machines, especially intelligent computer programs. further he defines intelligence as the computational part of the ability to achieve goals in the world. Different people think of AI differently and there is no unique definition. various authors defined AI differently as given below.

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

Scanned with CamScanner

- The art of creating machines that perform functions which require intelligence when performed by people. (Kurzweil, 1990)
- AI is a field of study that seeks to Explain and emulate intelligent behaviour interms of computational processes (Schalkoff, 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 computations that make it possible to percieve, reason, and act (Winston, 1992).
- AI is the branch of computer science that is concerned with the automation of intelligent behaviour (Luger and Stubblefield, 1993).

⇒ Categorization:-

In order to design intelligent systems it is important to categorize these systems. There are four possible categories of such system

- System that thinks like human.
- System that act like human.
- System that thinks rationally.
- System that acts rationally.

System that thinks like human requires cognitive modelling approaches. Most of the time it is the black box where we are not clear about our thought process. One has to know

Scanned with CamScanner

the functioning of the brain and its mechanism for processing information. It is an area of cognitive science. The stimuli are converted into mental representation and cognitive processes manipu-late this representation to build new representation that is used to generate actions. Neural network is a computing model for processing information similar to brain.

System that acts like human requires that the overall behaviour of the system should be human like which could be achieved by observation Turing test is an example.

System which thinks rationally relies on logic rather than human to measure correctness. For thinking rationally or logically. logical formulae and theories are used for synthesizing outcomes

System that acts rationally is the final category of intelligent system where by rational behaviour we mean doing the right thing. Even the method is illogical.

Components:-

AI techniques must be independent of the problem domain as far as possible. Any AI program should have knowledge base, and navigational capability which contains control strategy and inference mechanism.

Knowledge base:- AI programs should be learning in nature and update its knowledge accordingly. Knowledge base generally consists of facts and

Scanned with CamScanner

rules and has the following characteristics.

- It is voluminous in nature and requires proper structuring.
- It may be incomplete and imprecise.
- It may be dynamic and keep on changing.

Control strategy:> It determines which rule to be applied. To known this rule, some heuristics or thumb rules based on problem domain may be used.

Inference mechanism:> It requires search through knowledge base and derives new knowledge using the existing knowledge with the help of inference rules.

3) Write notes on ELIZA?

ELIZA was a program that conversed with user in English. people were amazed to see this program. The program was able to converse about any subject, because it stored subject information in data banks. Another feature of ELIZA was its ability to pick up speech patterns from users questions and provide responses using those patterns. In this mode, ELIZA mostly rephrased the user's statements as questions and posed those to the user.

<u>passage of ELIZA conversation.</u>

> Hello, I am ELIZA

* can you help me?

Scanned with CamScanner

> Don't you believe that I can help you?
* yes
> I see
* I feel depressed
> How long have you been feeling depressed?
* Since last month.

Since then lot many versions of ELIZA have been developed and are available on Internet. The main characteristics of ELIZA are briefly mentioned here.

Simulation of Intelligence:-

ELIZA programs are not intelligent at all in real sense. They donot understand the meaning of utterance Instead these programs simulate intelligent behaviour quite effectively by recognizing key words and phrases. By using a table lookup, one of a few ways of responding question is chosen.

Quality of Response:-

It is limited by the sophistication of the ways in which they can process the input text at a syntactic level.

Coherence:-

The earlier version of the system imposed no structure on the conversation. Each statement was based entirely on the current input and no context information was used. More complex versions of ELIZA can do a little better. Any sense of intelligence depends strongly on the

Scanned with CamScanner

coherence of the conversation as judged by the user.

Semantics:)

Such systems have no semantic representation of the content of either the user's input or the reply. That is why we say that it does not have intelligence of understanding of what we are saying. But it looks that it immitates the human conversation style. Because of this, it passed Turing test.

4) What are the foundations of AI?

Commonly used AI techniques and theories are rule based, fuzzy logic, neural networks, decision theory, statistics probability theory, genetic algorithms etc, since AI is inter-disciplinary in nature, foundations of AI are in various fields such as:

- Mathematics
- Neuroscience
- Control theory
- Linguistics.

Mathematics:)

AI systems use formal logical methods and Boolean logic (Boole, 1847), analysis of limits to what can be computed, probability theory, uncertainty that forms the basis for most modern approaches to AI, fuzzy logic etc.

Scanned with CamScanner

Neuroscience:-

This science of medicine helps in studying the functioning of brains. In early studies, injured and abnormal people were used to understand what parts of brain works. Now recent studies use accurate sensors to correlate brain activity to human thought. By monitoring individual neurons monkeys can now control a computer mouse using thought alone. Moore's law states that the computer will have as many gates as humans have neurons in the year 2020. Researchers are working to know as to how to have a mechanical brain. Such systems will require parallel computation, remapping and interconnections to a large extent.

Control therapy:-

Machines can modify their behaviour in response to the environment. Steam engine governer, thermostat and water-flow regulator are few examples of control theory. In 1950, control theory could only describe linear systems. AI largely rose as a response to this shortcoming. This theory of stable feedback systems helps in building systems that transition from initial state to goal state with minimum energy.

Linguistics:-

Speech demonstrates so much of human intelligence. Analysis of human language reveals thought taking place in ways not understood in other

Scanned with CamScanner

settings. children can create sentences they have never heard before. Languages and thoughts are believed to be tightly intertwined.

5) Mention the subareas of AI?

AI is an interdisciplinary area having numerous diverse subfields. Each one of these fields is an area of research in AI itself. Some of these are listed below.

- Knowledge representation models.
- Theorem proving mechanisms
- Game playing methodologies
- Common sense reasoning dealing with uncertainty and decision making.
- Learning models, inference techniques, pattern recognition, search and matching. Etc.
- Logic (fuzzy, temporal, modal)
- Planning and scheduling.
- Natural language understanding, speech recognition and understanding spoken utterances.
- Computer vision.
- Models for intelligent tutoring systems.
- Robotics.
- Data mining.
- Expert problem solving.
- Neural networks, AI tools, Etc.
- Web agents.

Scanned with CamScanner

6) Give the applications of AI?

AI finds applications in almost all areas of real life applications. Broadly speaking, business engineering, medicine, Education and manufacturing are the main areas.

- Business : financial strategies, give advice.
- Engineering: check design, offer suggestions to create new product, expert systems for all engineering applications.
- Manufacturing: assembly, inspection and maintenance
- Medicine: monitoring, diagnosing, and prescribing.
- Education: in teaching.
- fraud detection.
- object identification.
- space shuttle scheduling
- Information retrieval.

7) Write notes on production system and production rule?

production system is one of the formalisms that helps AI programs to do search process more conviniently in state space problems. This system comprises of state state and goal state of the problem along with one or more databases consisting of suitable and necessary information for the particular task. Generally, Knowledge representation schemes are used to structure information in these databases. PS consists of number of production rules in which Each

Scanned with CamScanner

production rule has left side that determines the applicability of the rule and a right side that describes the action to be performed if the rule is applied. Left side of the rule is current state, whereas the right side describes the new state that is obtained from applying the rule. These production rules operate on the databases that change as these rules are applied.

production system also consists of control strategies that specify the sequence in which the rules are applied when several rules match at once. The advantages of formalism in AI:

1. It is good way to model the strong state driven nature of intelligent action.
2. New rules can be easily added to account for new situations without disturbing the rest of the system.
3. It is quite important in real-time Environment and applications where new input to the database changes the behaviour of the system.

8) State water jug problem and describe its solution?

Problem statement:-

We have two jugs, a 5 gallon (5-g) and the

Scanned with CamScanner

other 3 gallon (3-g) with no measuring marker on them. There is Endless supply of water through tap. our task is to get 4 gallon of water in the 5-g jug.

solution:-

State space for this problem can be described as the set of ordered pairs of integers (x, y) such that x represents the number of gallons of water in 5-g jug and y for 3-g jug.

1. Start state is (0,0).
2. Goal state is (4, N) for any value of ($N \leq 3$)

The possible operations that can be used in this problem are listed as follows:

- Fill 5g jug from the tap and Empty the 5-g jug by throwing water down the drain.
- Fill 3-g jug from the tap and Empty the 3-g jug by throwing water down the drain.
- pour some or 3-g water from 5-g jug into the 3-g jug to make it full.
- pour some or full 3-g water jug water into the 5-g jug

These operations can formally be defined as production rules

P.T.O

Scanned with CamScanner

Production Rules for Water Jug problem.

Rule NO	Left of Rule (C.S)	Right of rule (G.S)	Description
1.	$(x, y / x<5)$	$(5, y)$	Fill 5-g jug
2.	$(x, y / x>0)$	$(0, y)$	empty 5-g jug
3.	$(x, y) / y<3)$	$(x, 3)$	Fill 3-g jug
4.	$(x, y) / y>0)$	$(x, 0)$	empty 3-g jug
5.	$(x, y / x+y \leq 5 \wedge y>0)$	$(x+y, 0)$	empty 3-g into 5-g jug
6.	$(x, y / x+y \leq 3 \wedge x>0)$	$(0; x+y)$	empty 5-g into 3-g jug.
7.	$(x, y / x+y \geq 5 \wedge y>0)$	$(5, y-(5-x))$ until 5-g jug is full	pour water from 3-g jug into 5-g jug.
8.	$(x, y / x+y \geq 3 \wedge x>0)$	$(x-(3-y), 3)$	pour water from 5-g jug into 3-g jug until 3-g jug is full.

- It should to be noted that there may be more than one solutions for a given problem. we have shown two possible solution paths. We have to choose appropriate control strategy.

Scanned with CamScanner

Solution path I

Rule applied	5-g Jug	3-g Jug	Step NO
Start state	0	0	
1	5	0	1
8	2	3	2
4	2	0	3
6	0	2	4
1	5	2	5
8	4	3	6
Goal state	4	-	

Solution path II

Rule applied	5-g Jug	3-g Jug	Step no
Start state	0	0	
3	0	3	1
5	3	0	2
3	3	3	3
7	5	1	4
2	0	1	5
5	1	0	6
3	1	3	7
5	4	0	8
Goal state	4	-	

Scanned with CamScanner

9) Give the production rule and solution path to the missionaries and cannibals problem?

<u>problem statement:</u>

Three missionaries and three cannibals want to cross a river. There is a boat on their side of the river that can be used by either one or two persons. How should they use this boat to cross the river in such a way that cannibals never outnumber missionaries on either side of river? If the cannibals ever outnumber the missionaries then the missionaries will be eaten How can they all cross over without anyone being eaten?

<u>Solution:</u>

State space for this problem can be described as the set of ordered pairs of left and right banks of the river as (L, R) where each bank is represented as a list [nM, mC, B]. Here n is the number of missionaries M, m is the number of cannibals C, and B represents the boat.

1. Start state: ([3M, 3C, 1B], [0M, 0C, 0B], 1B means that boat is present and 0B means it is absent.
2. Any state: ([n_1M, m_1C, _], [n_2M, m_2C, _]), with constraints / conditions at any state as $n_1 (\neq 0) \geq m_1$; $n_2 (\neq 0) \geq m_2$; $n_1 + n_2 = 3$, $m_1 + m_2 = 3$; boat can be either side.

Scanned with CamScanner

3. Goal State: ([0M, 0C, 0B], [3M, 3C, 1B])

It should be noted that by no means, this representation is unique. In fact, one may have number of representations for same problem.

The possible operations that can be used in this problem are listed as follows.

⊙ production rules of Missionaries and cannibals problem.

RN	Left side of rule	→	Right side of rule
Rules for boat going from left bank to right bank			
1.1	([n_1M, m_1C, 1B], [n_2M, m_2C, 0B])	→	([(n_1-2)M, m_1C, 0B], [(n_2+2)M, m_2C, 1B])
1.2	([n_1M, m_1C, 1B], [n_2M, m_2C, 0B])	→	([(n_1-1)M, (m_1-1)C, 0B], [(n_2+1)M, (m_2+1)C, 1B])
1.3	([n_1M, m_1C, 1B], [n_2M, m_2C, 0B])	→	([n_1M, (m_1-2)C, 0B], [n_2M, (m_2+2)C, 1B])
1.4	([n_1M, m_1C, 1B], [n_2M, m_2C, 0B])	→	[(n_1-1)M, m_1C, 0B], [(n_2+1)M, m_2C, 1B])
1.5	([n_1M, m_1C, 1B], [n_2M, m_2C, 0B])	→	([n_1M, (m_1-1)C, 0B], [n_2M, (m_2+1)C 1B])
Rules for boat coming from right bank to left bank			
R1	([n_1M, m_1C, 0B], [n_2M, m_2C, 1B])	→	([(n_1+2)M, m_1C, 1B], [(n_2-2)M, m_2C, 0B])
R2	([n_1M, m_1C, 0B], [n_2M, m_2C, 1B])	→	([(n_1+1)M, (m_1+1)C, 1B], [(n_2-1)M, (m_2-1)C, 0B])
R3	([n_1M, m_1C, 0B], [n_2M, m_2C, 1B])	→	([n_1M, (m_1+2)C, 1B], [n_2M, (m_2-2)C, 0B])
R4	([n_1M, m_1C, 0B], [n_2M, m_2C, 1B])	→	([(n_1+1)M, m_1C, 1B], [(n_2-1)M, m_2C, 0B])
R5	([n_1M, m_1C, 0B], [n_2M, m_2C, 1B])	→	([n_1M, (m_1+1)C, 1B], [n_2M, (m_2-1)C 0B]

Scanned with CamScanner

Solution path.

Rule Number	([3M, 3C, 1B], [0M, 0C, 0B]) ← Start state
L2:	([2M, 2C, 0B], [1M, 1C, 1B])
R4:	([3M, 2C, 1B], [0M, 1C, 0B])
L3:	([3M, 0C, 0B], [0M, 3C, 1B])
R5:	([3M, 1C, 1B], [0M, 2C, 0B])
L1:	([1M, 1C, 0B], [2M, 2C, 1B])
R2:	([2M, 2C, 1B], [1M, 1C, 0B])
L1:	([0M, 2C, 0B], [3M, 1C, 1B])
R5:	([0M, 3C, 1B], [3M, 0C, 0B])
L3:	([0M, 1C, 0B], [3M, 2C, 1B])
R5:	([0M, 2C, 1B], [3M, 1C, 0B])
L3:	([0M, 0C, 0B], [3M, 3C, 1B]) → Goal state.

10) Elaborate the concept of state space tree?

Similar to production system, state space is another method of problem representation that facilitates easy search. Using this method one can also find a path from start state to goal state while solving a problem. A state space basically consists of four components.

1. A set S containing start states of problem.
2. A set G containing goal states of problem.

Scanned with CamScanner

3. set of nodes in the graph/tree. Each node represen-ts the state in problem solving process.
4. set of arcs connecting nodes. Each arc corresponds to operator that is a step in a problem solving process.

A solution path is a path through the graph from a node in S to a node in G. The main objective of search algorithm is to determine a solution path in the graph. There may be more than one solution paths, as ~~they~~ there may be more than one ways of solving the problem. One would exercise a choice between various solution paths based on some criteria of goodness or on some heuristic function commonly used approach is to apply appropriate operator to transfer one state of problem to another. It is similar to production system search method where we use production rules instead of operators. Let us consider again the problem of "Missionaries and cannibals".

The possible operators that are applied in this problem are {2M0C, 1M1C, 0M2C, 1M0C, 0M1C}, Here M is missionary and C is cannibal. Digit before these characters indicates number of miss-ionaries and cannibals possible at any point in time. These operators can be used in both the situations i.e., if boat is on the left bank then, we write 'Operator →' and if the boat is on the right bank of the river then we write "operator ←".

Scanned with CamScanner

for the sake of simplicity, Let us represent state (L:R), where $L = n_1 M m_1 C1B$ and $R = n_2 M m_2 C0B$ Here B represents boat with 1 or 0 indicating presence or absence of the boat.

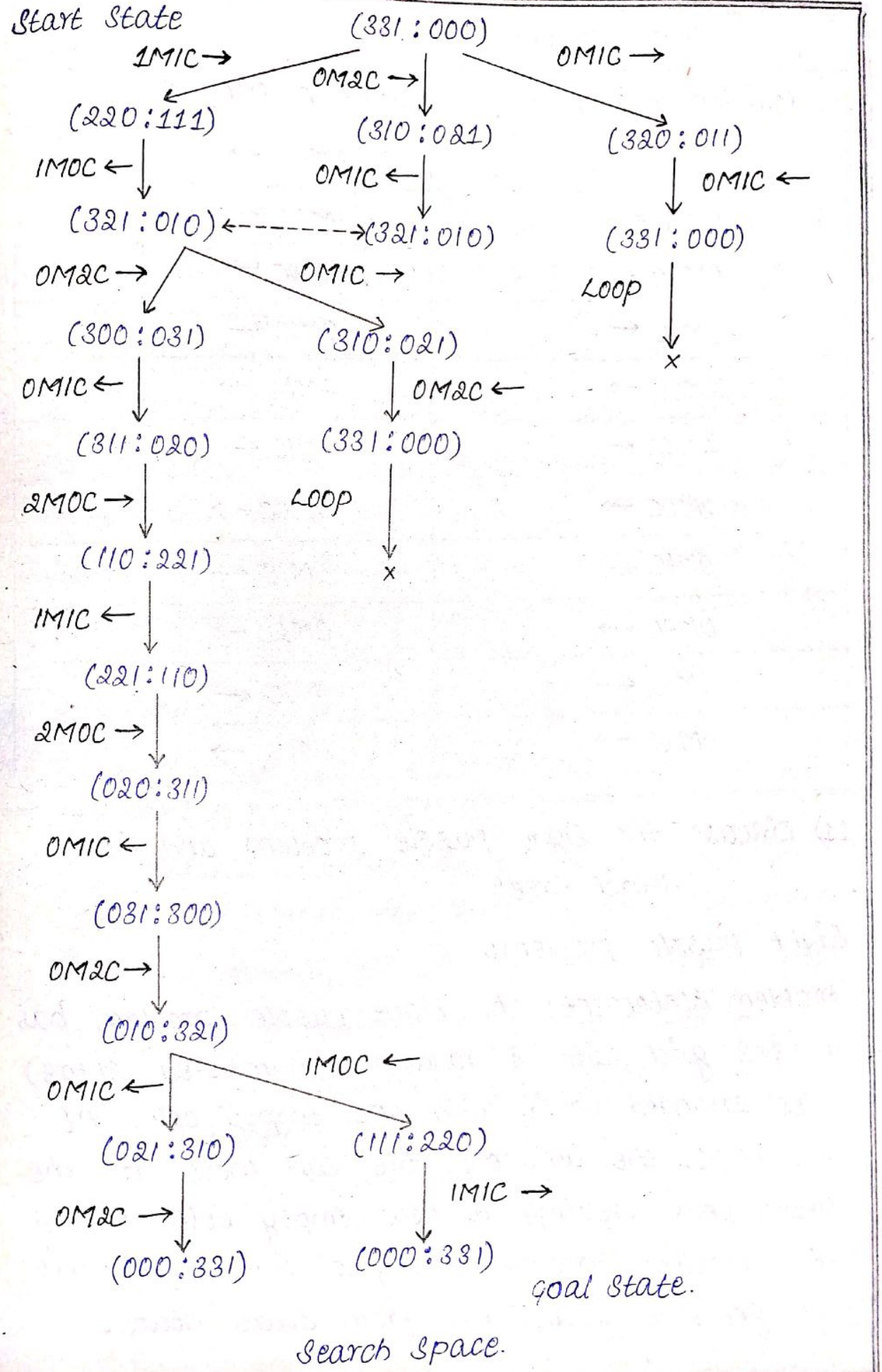
1. Start state : (3M3C1B : 0M0C0B) or simply (331:000)

2. Goal state: (0M0C0B: 3M3C1B) or (000:331)

To illustrates the progress of search we are required to develop a tree of nodes, with each node in the tree representing a state. The root node of the tree may be used to represent the start state. The arcs of the tree inside the application of one of the operators. The nodes for which no operators are applied are called leaf nodes, which have no arcs leading from them. To simplify, we have not generated entire search space and avoided illegal and looping states. Depth first or breadth first strategy or some intelli-gent heuristic searches is used to generate the search space. The search space generated using valid operators. The search of operators applied to this problem is given in sequence in search space tree.

P.T.O.

Scanned with CamScanner



Search Space.

Scanned with CamScanner

Two solution paths

solution path 1	solution path 2
1M1C →	1M1C →
1M0C ←	1M0C ←
0M2C →	0M2C →
0M1C ←	0M1C ←
2M0C →	2M0C →
1M1C ←	1M1C ←
2M0C →	2M0C →
0M1C ←	0M1C ←
0M2C →	0M2C →
0M1C ←	1M0C ←
0M2C →	1M1C →

11) Discuss the Eight puzzle problem and its search tree?

<u>Eight puzzle problem</u>:-

<u>problem statement</u>:- The Eight puzzle problem has a 3x3 grid with 8 randomly numbered (1 to 8) tiles arranged on it with one empty cell. At any point, the adjacent tile can move to the empty cells creating a new empty cell. solving this problem involves arranging tiles such that we get the goal state from start state.

Scanned with CamScanner

Start state

3	7	6
5	1	2
4	□	8

Goal state

5	3	6
7	□	2
4	1	8

Eight puzzle problem

- A state for this problem should keep track of the position of all tiles on the game board, with representing the blank position on the board. The start and goal states may be represented as follows with each list representing corresponding now:

1. Start state: [(3, 7, 6), (5, 1, 2), (4, 0, 8)]
2. The goal state could be represented as:

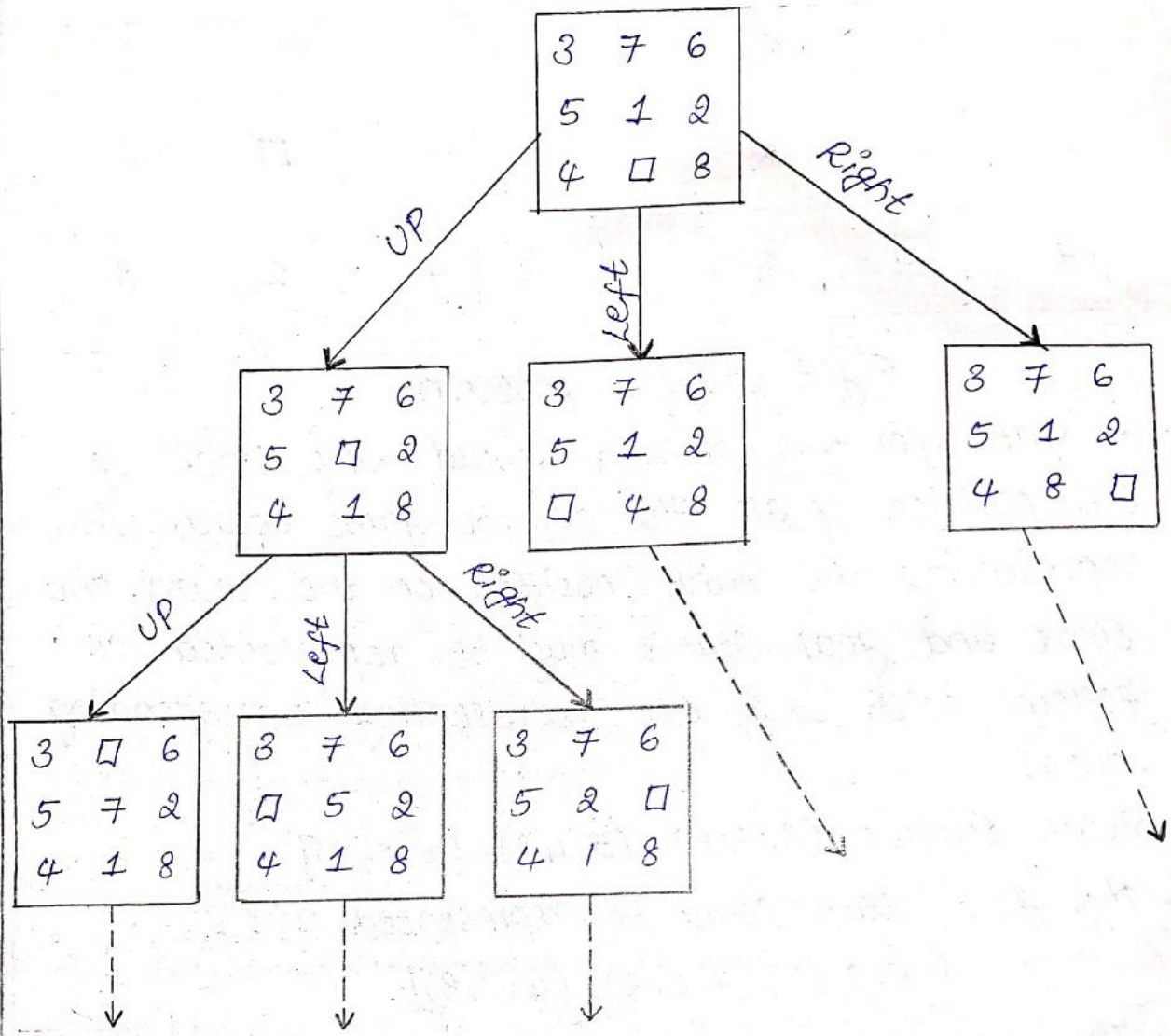
   [(5, 3, 6), (7, 0, 2), (4, 1, 8)]
3. The operators can be thought of moving {Up, Down, Left, Right}, the direction in which blank space effectively moves.

- To simplify, a search tree up to level 2 is drawn to illustrate the use of operators to generate next state.

P.T.O

Scanned with CamScanner

Initial state



Continue searching like this till we reach the goal state. The Exhaustive search can proceed using depth first and breadth first strategies Explained later. Some intelligent searches can also be made to find solution faster.

Scanned with CamScanner

12) Describe the concept of control strategies?

Control strategy is one of the most important components of problem solving that describes the order of application of the rules to the current state. control strategy should be such that it causes motion towards a solution. for example, if we select water jug problem, if we apply a simple control strategy of starting each time from the top of rule list and select the first applicable one, then we will never move towards solution. The second requirement of control strategy is that it should explore the solution space in a systematic manner. For example, if we select a control strategy where we select a rule randomly from the applicable rules, then definitely it causes motion and eventually will lead to a solution. But there is every possibility that we arrive to same state several times. This is because control strategy is not systematic. Depth first and breadth first are systematic control strategies but these are blind searches. In depth first strategy, we follow a single branch of the tree until it yields a solution or some pre specified depth has reached and then go back to immediate previous node and explore other branches using depth first strategy. In breadth first search, a search space tree is generated level wise until we find a solution or some specified depth is reached. These strategies are exhaustive, uninformed, and blind searches in

Scanned with CamScanner

nature. if the problem is simple, then any control strategy that causes motion and is systematic will lead to a solution. However, to solve some real-world problems, Effective control strategy must be read.

As mentioned earlier that the problem can be solved by searching for a solution. The main work in the area of search strategies is to find the correct search strategy for a given problem. There are two directions in which such a search could proceed.

- Data driven search, called forward chaining, from the start state.
- Goal driven search, called backward chaining, from the goal state.

<u>Forward chaining:-</u>

The process of forward chaining begins with known facts and works towards a conclusion. for eg in Eight puzzle problem, we start from the start state and work forward to the conclusion i.e., the goal state, In this case, we begin building a tree of move sequences with the root of the tree as start state. The states of next level of the tree are generated by finding all rules whose left sides match with root and use their right side to create the new state. This process is continued until a configuration that matches the goal

Scanned with CamScanner

state is generated. Language OPS5 uses forward reasoning rules. Rules are Expressed in the form of if then rules.

Backward chaining:-

It is a goal directed strategy that begins with the goal state and continues working backward, generating more sub-goals that must also be satisfied to satisfy main goal until we reach to start state. prolog language uses this strategy.

We can use both data driven and goal directed strategies for problem solving, depending on the nature of the problem.

13) Explain the characteristics of a problem?

Characteristics of problem:-

Before starting modelling the search and trying to find solution for the problem, one must analyze it along several key characteristics initially. Some of these are mentioned below.

Type of problem:-

There are three types of problems in real life, viz, ~~Igon~~ Ignorable, Recoverable, and Irrecoverable.

- Ignorable:

These are the problems when we can ignore the solution steps. for eg in proving a theorem if some lemma is proved to prove a theorem and later on we realize that it is not useful, then we can ignore this solution step and

Scanned with CamScanner

and prove another lemma. Such problems can be solved using simple control strategy.

⊙ Recoverable:-

These are the problems where solution steps can be undone. for Eg in water jug problem, if we have filled up the jug, we can empty it also. Any state can be reached again by undoing the steps. These problems are generally puzzles played by a single player. Such problems can be solved by backtracking, so control strategy can be implemented using a push down stack.

⊙ Irrecoverable:-

The problems where solution steps cannot be undone for Eg, any two player game such as chess, playing cards, snake and ladder, Etc., are examples of this category. Such problems can be solved by planning process.

Decomposability of a problem:-

Divide the problem into a set of independent smaller sub-problems, solve them and combine the solutions to get the final solution. The process of dividing subproblems continues till we get the set of the smallest subproblems for which a small collection of specific rules are used. Divide and conquer technique is the commonly used method for solving such problems It is an important and useful characteristic

Scanned with CamScanner

as each subproblem is simpler to solve and can be handed over to a different processor. Thus, such problems can be solved in parallel processing environment.

Role of Knowledge:

Knowledge plays an important role in solving any problem. Knowledge could be in the form of rules and facts which help generating search space for finding the solution.

Consistency of Knowledge Base used in solving problem:

Make sure that knowledge base used to solve problem is consistent. Inconsistent knowledge base will lead to wrong solutions. for eg, if we have knowledge in the form of rules and facts as follows:

- If it is humid, it will rain. If it is sunny, then it is daytime. It is sunny day. It is nighttime.
- This knowledge is not consistent as there is a contradiction because "it is a daytime" can be deduced from the knowledge ~~in the form of~~ and thus 'it is night time' and it is daytime are not possible at the same time. If knowledge base has such inconsistency, then some methods may be used to avoid such conflicts.

Scanned with CamScanner

Requirement of solution:-

We should analyze the problem whether solution required is absolute or relative. We shall call solution to be absolute if we have to find exact solution, whereas it is relative if we have reasonably good and approximate solution for eg, in water jug problem. if there are more than one ways to solve a problem, then we follow one path successfully. There is no need to go back and find a better solution In this case, the solution is absolute. In travelling salesman problem, our goal is to find the shortest route. Unless all routes are known, it is difficult to know the shortest route. This is a best path problem, whereas water jug is any path problem. Any path problem is generally solved in reasonable amount of time by using heuristics that suggest good paths to explore. Best path problems are computationally harder compared with any-path problems.

14) Illustrate depth first search tree generation with an example?

In the depth-first search (DFS), we go as far down as possible into search tree / graph before backing up and trying alternatives. It works by always generating a descendent of the most recently expanded node until some depth cut off is reached and then backtracks to next most recently expanded node and generates one of its

Scanned with CamScanner

descendants. DFS is memory efficient, as it only stores a single path from the root to leaf node along with the remaining unexpanded sibblings for each node on the path.

We can implement DFS by using two lists called OPEN and CLOSED. The OPEN list contains those states that are to be expanded, and CLOSED list keeps track of states already expanded Here OPEN and CLOSED. lists are maintained as stacks. If we discover that first element of OPEN is the goal state then search terminates successfully. We can get track of the path through state space as we traversed. but in those situations where many nodes after expansion are in the closed list, we fail to keep track of our path. This information can be obtained by modifying CLOSED. List by putting pointer back to its parent in the search tree.

<u>Algorithm</u>

Input: START and GOAL states.

Local variables: OPEN, CLOSED, ~~STATE-X~~ RECORD-X, SUCCS, FOUND;

Output: Yes or NO

Method:

- initialize OPEN list with START, nil and CLOSED = ϕ;
- FOUND = false;
- While (OPEN $\neq \phi$ and FOUND = false) do

  {

  - remove the first ~~state~~ record (initially (START, nil)) from OPEN and call it ~~STATE-X~~ RECORD-X;

- put ~~STATE-X~~ RECORD-X in the front of CLOSED list {maintai-
  -ned as stack};
- If STATE-X of RECORD-X = GOAL then FOUND = true else
  {
  - perform EXPAND operation on STATE-X, producing a list of ~~succs~~; records called SUCCS;
  - remove from successors those states, if any, that are in the CLOSED Lists:
  - ~~append~~ insert SUCCS ~~at~~ in the ~~End~~ front of the OPEN list:

  }

}/* End While */

- if FOUND = true then return ~~Yes~~ the path by tracing ~~else return NO~~ through the pointers to the parents on the CLOSED list else return NO.
- Stop.

Let us see the search tree generation from start state of the water jug problem using DFS algorithm. ~~At each state, we apply first applicable rule. if it generates previously generated state then cross it and try another rule in the sequence to avoid the looping. If new state is generated then cross it and try another rule in sequence to avoid the looping. If new state is generated then expand this state in breadth first problem/ fashion.~~

~~The rules applied and enclosed for water jug problem are:~~

Scanned with CamScanner

Let us see the search tree generation from start state of the water jug problem using DFS algorithm.

Water jug problem		
Search tree generation using DFS	OPEN list	CLOSED list
Start State (0,0)	[((0,0), nil)]	
↓ {1} (5,0)	[((5,0), (0,0))]	[((0,0), nil)]
{2} ↙ X ; {3} ↓ (5,3)	[((5,3), (5,0))]	[((5,0),(0,0)), ((0,0), nil)]
↓ {2} (0,3)	[((0,3), (5,3))]	[((5,3),(5,0)), ((5,0),(0,0)), ((0,0), nil)]
{1,4} ↙ X ; {5} ↓ (3,0)	[((3,0), (0,3))]	[((0,3),(5,3)), ((5,3),(5,0)), ((5,0),(0,0)), ((0,0), nil)]
↓ {3} (3,3)	[((3,3), (3,0))]	[((3,3),(3,0)), ((0,3),(5,3)), ((5,3),(5,0)), ((5,0),(0,0)), ((0,0), nil)]
{1,2,4} ↙ X ; {7} ↓ (5,1)	⋮	⋮
↓ {2} (0,1)	⋮	⋮
{1,2,4} ↙ X ; ↓ (1,0)	⋮	⋮
{1,2} ↙ X ; {3} ↓ (1,3)	⋮	⋮
{1,2,4} ↙ X ; {5} ↓ (4,0) goal state	[(4,0), (1,3)]	[((4,0),(1,3)), ((1,3),(1,0)), ((1,0),(0,1)), ((0,1),(5,1)), ((5,1),(3,3)), ((3,3),(3,0)), ((3,0),(0,3)), ((0,3),(5,3)), ((5,3),(5,0)), ((5,0),(0,0)), ((0,0), nil)]

Scanned with CamScanner

The path is obtained from the list stored in CLOSED. The solution path is

$(0,0) \rightarrow (5,0) \rightarrow (5,3) \rightarrow (0,3) \rightarrow (3,0) \rightarrow (3,3) \rightarrow (5,1) \rightarrow$
$(0,1) \rightarrow (1,0) \rightarrow (1,3) \rightarrow (4,0)$

- BFS is effective when the search tree has a low branching factor.
- BFS can work even in trees that are infinitely deep.
- BFS requires a lot of memory as number of nodes in level of the tree increases exponentially.
- BFS is superior when the GOAL exists in the upper right portion of a search tree.
- BFS gives optimal solution.
- DFS is effective when there are few sub trees in the search tree that have only one connection point to the rest of states.
- DFS is best when the GOAL exists in the lower left portion of search tree.
- DFS can be dangerous when the path closer to the START and farther from the GOAL has been choosen.
- DFS is memory efficient as the path from start to current node is stored. Each node should contain state and its parent.
- DFS may not give optimal solution.

Scanned with CamScanner

15) Illustrate breadth first search tree generation with an Example?

Breadth first search:

The breadth first search Expands all the states one step away from the start state and then Expands all states two steps from start, then three steps, Etc., until a goal state is reached. All successor states are Examined at the same depth before going deeper. The BFS always gives an optimal path or solution.

This search is implemented using two lists called OPEN and CLOSED. The OPEN list contains those states that are to be Expanded and CLOSED list keeps track of states already Expanded. Here OPEN list is maintained as a queue and CLOSED list as a stack. for the sake of simplicity, we are writing BFS algorithm for checking whether a goal node Exists or not. further, this algorithm can be modified to get a path from start to goal nodes by maintaining CLOSED list with pointer back to its parent in the search tree.

Algorithm:

Input: START and GOAL states

Local variables: OPEN, CLOSED, STATE-X, SUCCS, FOUND;

Output: yes or NO

Method:

- initialize OPEN list with START and CLOSED = ∅
- FOUND = false;

Scanned with CamScanner

- While (OPEN ≠ ∅ and FOUND = false) do

  {

  - ~~perform EXPAND operation~~ remove the first state from OPEN and call it STATE-X;
  - put STATE-X in the front of CLOSED list;
  - if STATE-X = GOAL then FOUND = true else

    {

    - perform EXPAND operation on STATE-X, producing a list of SUCCS;
    - remove from sucessors those states, if any, that are in the CLOSED list;
    - append SUCCS at the end of OPEN list;

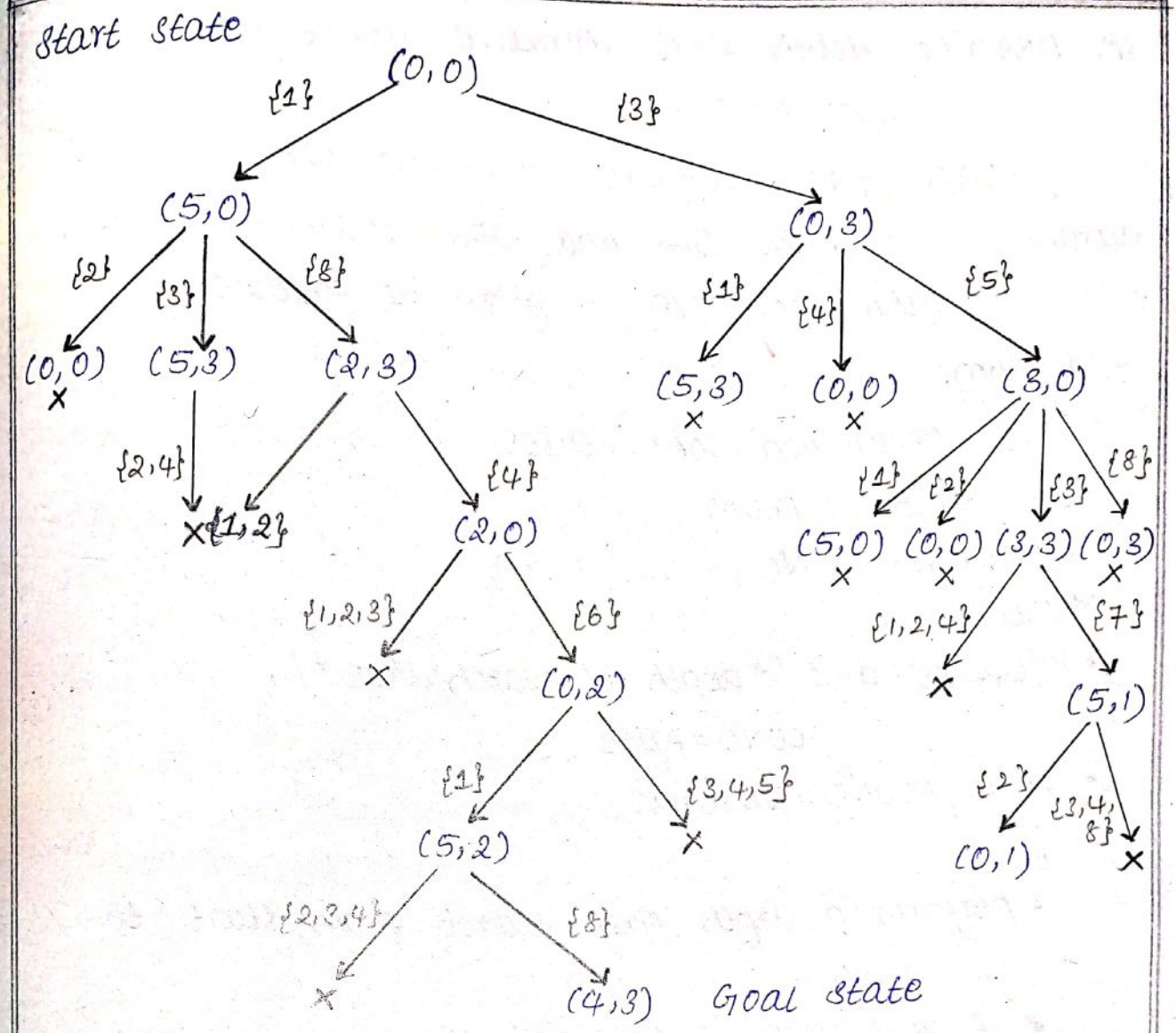
    }

  }/* End while */

- if FOUND = true then return yes else return NO
- stop

Let us see the search tree generation from start state of the water jug problem using BFS algorithm. At each state, we apply first applicable rule. If it generates previously generated state then cross it and try another rule in the sequence to avoid the looping. If new state is generated then expand this state in breadth first fashion.

Scanned with CamScanner



Search tree generation using BPS

Search tree is developed level wise. This is not memory efficient as partially developed tree is to be kept in the memory but it finds optimal solution or path. We can easily see the path from start to goal by tracing the tree from goal state to start state through parent link. This path is optimal and we cannot get a path shorter than is.

Solution path : $(0,0) \rightarrow (5,0) \rightarrow (2,3) \rightarrow (2,0) \rightarrow (0,2) \rightarrow (5,2) \rightarrow (4,3)$

Scanned with CamScanner

16) Describe depth first iterative depeening algorithm?

Depth first iterative deepening takes advantages of both BFS and DFS searches on trees.

⊙ The algorithm for DFID is given as follows:

<u>Algorithm:</u>

Input: START and GOAL states

Local variables: FOUND;

Output: yes or NO

Method

- initialize d = 1 /* depth of search tree */,
  FOUND = false
- While (FOUND = false) do
  {
  - perform a depth first search from start to depth d.
  - if goal state is obtained then FOUND = true
    else discard the nodes generated in the search of depth d
  - d = d + 1

  } /* end while */
- if FOUND = true then return yes otherwise return NO
- stop

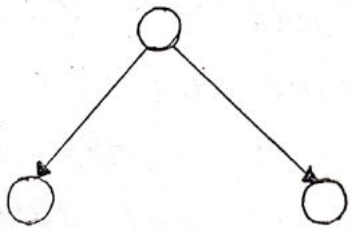
Since DFID expands all nodes at a given depth before explaining any nodes at greater depth it is guaranteed to find a shortest path or optimal solution from start to goal state. The

Scanned with CamScanner

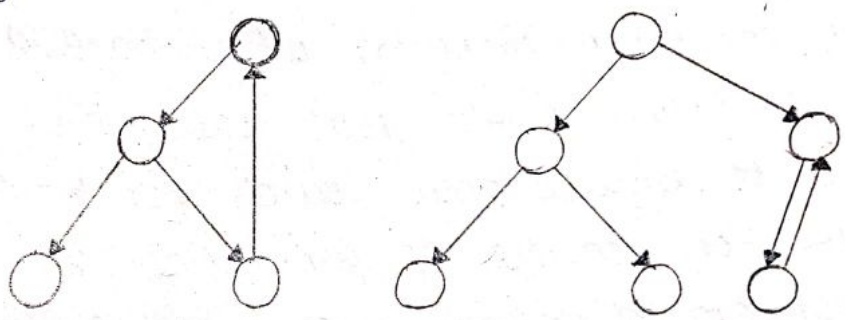
Working of DFID algorithm is given below.

Initial state

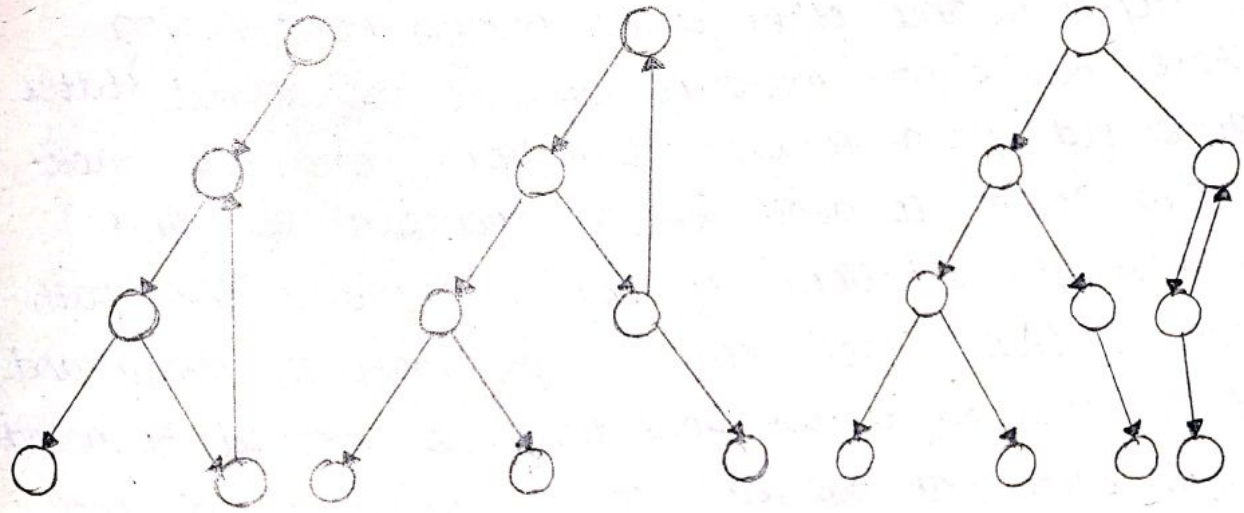
1st iteration



2nd iteration



3rd iteration



Continue this way

At any given time, it is performing a DFS and never searches deeper than depth 'd' thus, the space it uses is O(d). Disadvantage of DFIDS is that it performs wasted computation before reaching the goal depth.

Scanned with CamScanner

17) Explain bidirectional search graph with an Example?

Bidirectional search graph:-

Bidirectional search is a graph search algorithm that runs two simultaneous searches. one search moves forward from the start state and other moves backward from the goal and stops when the two meet in the middle. It is useful for those problems which have a single start state and single goal state. The DFID can be applied to bidirectional search for $k = 1, 2, \ldots$ The kth iteration consists of generating all states in the forward direction from start state up to depth k using BFS and goal state using DFS one to depth k and other to depth k+1 not storing states but simple matching against the stored states generated from forward direction. Here the backward search to depth k+1 is necessary to find odd-length solutions. If match is found, then path can be traced from start to the matched state and from matched to the goal state. It should be noted that each node has link to its successors as well as to its parent. These links will help generating complete path from start to goal state.

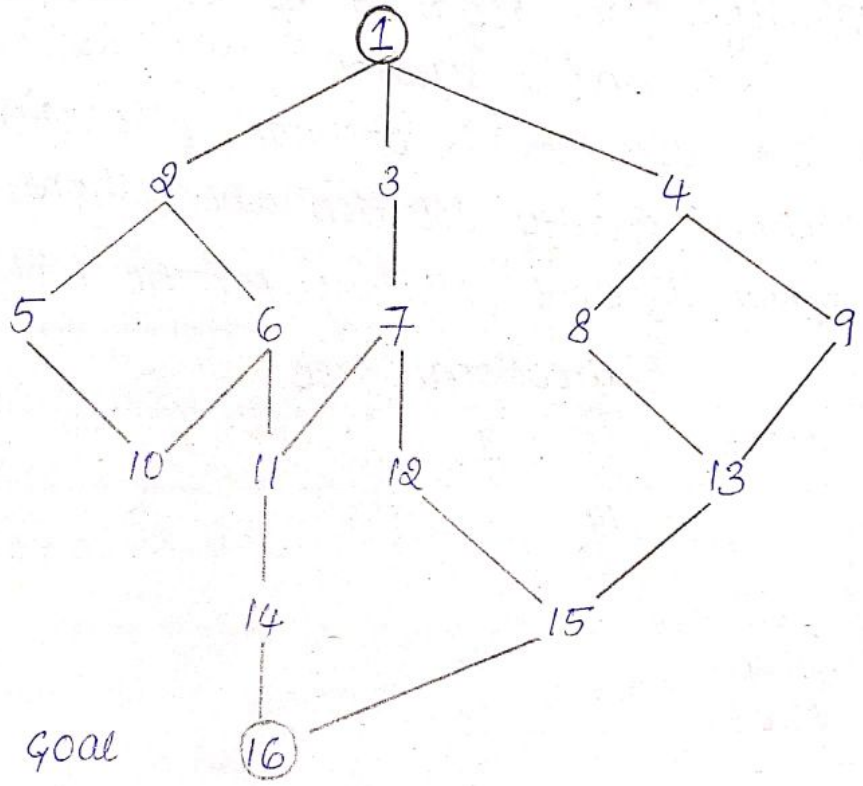
The reason for this approach is that each of the two searches has time complexity $O(b^{d/2})$, and $O(b^{d/2} + b^{d/2})$ is much less than the running time of one search from the beginning to goal

Scanned with CamScanner

which would be $O(b^d)$. This search can be made in already existing graph/tree or search graph tree can be generated as a part of search. Let us illustrate the working of this method using existing graph. Consider the following graph

⊙ find a path from node 1 to node 16.

Graph



Graph to be searched using Bidirectional Search

⊙ The trace of finding path from node 1 to 16 using bidirectional search is, We clearly see that the path obtained is:

1, 2, 6, 11, 14, 16.

Scanned with CamScanner

18) State the analysis of search methods?

Effectiveness of any search strategy in problem solving is measured in terms of:

- Completeness: completeness means that an algorithm guarantees a solution if it exists
- Time complexity: Time required by an algorithm to find a solution.
- Space complexity: Space required by an algorithm to find a solution.
- optimality: the algorithm is optimal if it finds the highest quality solution when there are several different solutions for the problem.

Iteration	Bidirectional tree
K=0	Start 1
K=1 K=0	14 15 Goal 16
K=1	Start 1 2 3 4
K=2 K=1	11 12 13 14 15 Goal 16
K=2	Start 1 2 3 4 5 6 7 8 9
K=3 K=2 K=1	11 14 Goal 16

Scanned with CamScanner

We will compare searches discussed above on these parameters. Let us assume 'b' to be the branching factor and 'd' to be the depth of the tree in the worst case.

DFS: If the depth cut off is 'd', then the space requirement is of $O(d)$. The time complexity of DFS to depth 'd' is of $O(b^d)$ in the worst case. The DFS requires some cut-off depth. If branches are not cut off and duplicates are not checked for. the algorithm may not even terminate. We cannot say on optimality of solution.

BFS: In BFS, all the nodes up to depth 'd' must be generated in the worst case. At level i there will be b^i nodes generated. So, total number of nodes generated in the worst case is

$$1+b+b^2+b^3+\dots+b^{d-1} \cong O(b^d)$$

Space complexity in the worst case is also $O(b^d)$. The solution obtained using BFS is optimal but it may take higher computational time. It will terminate and find solution if it exists.

DFID: Space requirement of DFID is of $O(d)$, as search tree is generated using DFS method. The time requirement for DFID will be based on number of nodes generated during search. The total nodes in tree are calculated as follows:

- Nodes at depth d are generated once during the final iteration of the search.

Scanned with CamScanner

- Nodes at depth $d-1$ are generated twice.
- Nodes at depth $d-2$ are generated thrice and so on.

Thus, the total number of nodes generated in DFID to depth d is

$$N = b^d + 2b^{d-1} + 3b^{d-2} + \dots db$$

$$= b^d[1 + 2b^{-1} + 3b^{-2} + \dots db^{1-d}]$$

$$= b^d[1 + 2x + 3x^2 + 4x^3 + \dots + dx^{d-1}] \quad \{if\ x = b^{-1}\}$$

N converges to $(b^d) * (1-x)^{-2}$ for $|x|<1$. Since $(1-x)^{-2}$ is a constant and independent of 'd', we can say that $N \propto O(b^d)$ if $b>1$. So time complexity is of $O(b^d)$. Since DFID also moves level wise, it finds the optimal solution.

The following table summarizes the result.

performance comparision.

Search technique	Time	Space	Solution
DFS	$O(b^d)$	$O(d)$	—
BFS	$O(b^d)$	$O(b^d)$	optimal
DFID	$O(b^d)$	$O(d)$	optimal
Bi-directional	$O(b^{d/2})$	$O(b^{d/2})$	—

Above mentioned searches are blind and are not of much use in real life applications. There are problems where combinatorial Explosion takes place as the size of the search tree increases, such as travelling salesman problem.

Scanned with CamScanner

19) State and Explain travelling salesman problem?

Traveling salesman problem:

Statement: In traveling salesman problem (TSP), one is required to find the shortest route visiting all the cities once and returning back to starting point Assume that there are 'n' cities and the distance between each pair of the cities is given.

The problem seems to be simple, but deceptive. The TSP is one of the most intensely studied problems in computational mathematics. and yet no effective solution method is known for the general case.

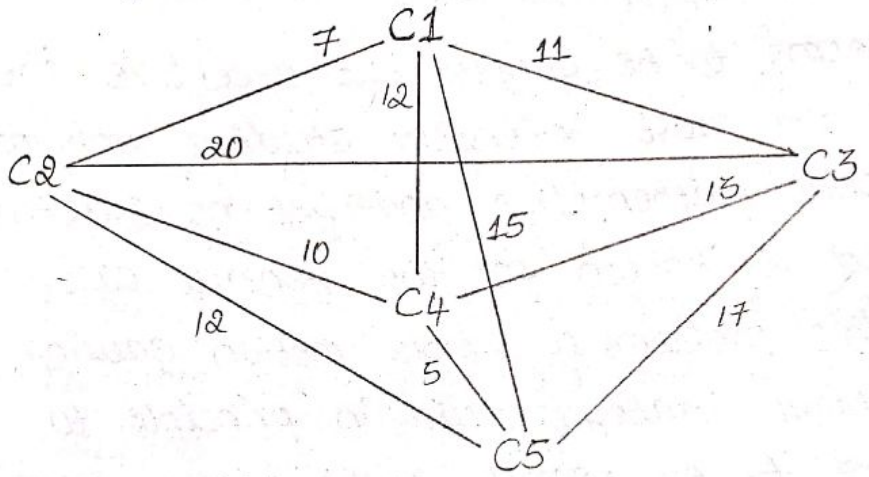
In this problem, a simple motion causing and systematic control strategy could, in principle, to applied to solve it. All possible paths of the search tree are explored and the shortest path returned. This will require (n-1)! paths to be examined for 'n' cities. If number of cities grows, then the time required to wait a salesman to get the information about the shortest path is not a practical situation This phenomenon is called combinatorial explosion.

Above mentioned strategy could be improved little bit using the following techniques.

- Start generating complete paths, keeping track of the shortest path found so far.
- Stop exploring any path as soon as its partial length becomes greater than the shortest path length found so far.

Scanned with CamScanner

This method is efficient than the first one but still requires exponential time that is directly proportional to some number raised to 'n'. Let us consider an example of five cities and see how we can find solution to this problem using above mentioned technique. Assume that C1 is the start city.



$D(C_1, C_2) = 7$; $D(C_1, C_3) = 11$; $D(C_1, C_4) = 12$;

$D(C_1, C_5) = 15$; $D(C_2, C_3) = 20$; $D(C_2, C_4) = 10$;

$D(C_2, C_5) = 12$; $D(C_3, C_4) = 13$; $D(C_3, C_5) = 17$;

$D(C_4, C_5) = 5$;

Graph for traveling salesman problem.

The possible paths generated using modified graphs approach upto some level. Some of the partial paths are pruned if the distance computed is less than minimum computed distance so far between any pair of cities. Initially, first complete path is taken to be the minimum and ✓ is put along with the distance, and if the distance is greater than the previously calculated minimum,

Scanned with CamScanner

then x is put to show pruning of that path.

continue till all the paths have been explored. In this case, there will be 4! = 24 possible paths. We notice that out of 13 paths 5 paths are partially evaluated. But still it requires exponential time.

Therefore, some kind of thumb rules or heuristic techniques may be thought of and applied. furthermore, there may be more than one ways of solving the problem and one would like to exercise a choice between various solution paths. based on some criteria of goodness or on some thumb rule. The following sections describe various heuristic search techniques.

So we need some intelligent methods which can make use of problem knowledge and improves the search time substantially.

paths explored. Assume C1 to be the start city		Distance
1. C1 → C2 → C3 → C4 → C5 → C1 7 20 18 5 15 27 40 45 60	current best path	60 ✓ ×
2. C1 → C2 → C3 → C5 → C4 → C1 7 20 17 5 12 27 44 49 61		61 ✓ ×
3. C1 → C2 → C4 → C3 → C5 → C1 7 10 18 17 15 17 40 57 72		72 ×
4. C1 → C2 → C4 → C5 → C3 → C1 7 10 5 17 11 17 22 39 50	current best path, cross path at S.NO 1	50 ✓ ×
5. C1 → C2 → C5 → C3 → C4 → C1 7 12 17 13 12 19 36 49 61		61 ×

Scanned with CamScanner

6. C1 → C2 → C5 → C4 → C3 → C1 7 12 5 13 11 19 24 37 48	current best path, cross path at S.NO 4	48 ✓
7. C1 → C3 → C2 → C4 → C5 7 20 10 5 37 47 52	(not to be Expanded further) partially Evaluated	52 X
8. C1 → C3 → C2 → C5 → C4 11 20 12 5 37 49 54	(not to be Expanded further) partially Evaluated	54 X
9. C1 → C3 → C4 → C2 → C5 → C1 11 13 10 12 15 24 34 46 61		61 X
10. C1 → C3 → C4 → C5 → C2 → C1 11 13 5 12 7 24 29 41 48	same as current best path at S.NO 6	48 ✓
11 C1 → C3 → C5 → C2 11 17 12 38 50	(not to be Expanded further) partially Evaluated	50 X
12 C1 → C3 → C5 → C4 → C2 11 17 5 10 38 43 53	(not to be Expanded further) partially Evaluated	53 X
13 C1 → C4 → C2 → C3 → C5 12 10 20 17 22 42 55	(not to be Expanded further) partially Evaluated	59 X
continue like this		

Scanned with CamScanner

20) Write notes on heuristic search techniques?

Heuristic technique is a criterion for determining which among several alternatives will be the most effective to achieve some goal. This technique improves the efficiency of a search process. possibly by sacrificing claims of systematic and completeness It no longer guarantees to find the best solution but almost always finds a very good solution. Using good heuristics, we can hope to get good solution to hard problems in less than Exponential time. There are two types of heuristics, namely,

- General purpose heuristics that are useful in various problem domains.
- Special purpose heuristics that are domain specific.

General purpose Heuristics:

A general purpose heuristics for combinatorial problem is nearest neighbor algorithms that work by selecting the locally superior alternative for such algorithms, it is often possible to prove an upper bound on the Error. It provides reassurance that we are not paying too high a price in accuracy for speed. In many AI problems, it is often difficult to measure precisely the goodness of a particular solution. for real world problems, it is often useful to introduce heuristics on the basis of relatively unstructured knowledge. It is impossible to define this knowledge in such a way that mathematical analysis can be performed. In AI approaches, behaviour of algorithms is analyzed by running

Scanned with CamScanner

them on computer as contrast to analyzing algorithm mathematically. There are atleast many reasons for such adhoc approaches in AI.

- It is fun to see a program do something intelligent than to prove it.
- Since AI problem domains are usually complex, it is generally not possible to produce analytical proof that a procedure will work.
- It is not even possible to describe the range of problems well enough to make statistical analysis of problem behaviour meaningful.

However, it is important to keep performance in mind while designing algorithms. One of the most important analysis of the search process is to find number of nodes in a complete search tree of depth 'd' and branching factor 'f'. that is f*d. This simple analysis motivates to look for improvements on the Exhaustive search procedures. The searches which use some domain knowledge are called Informed search strategies.

21) Explain branch and bound search technique?

In branch and bound search method, cost function is designed that assigns cumulative Expense to the path from start node to the current node x by applying the sequence of operators. While generating a search space, a least cost ~~partial~~ path, obtained so far is expanded at each iteration till we reach to

Scanned with CamScanner

goal state. Since branch and bound search expands the least cost partial path, it is sometimes called a uniform cost search. for eg, in traveling salesman problem, $g(x)$ may be the actual distance travelled from start to current node x,

During search process, there can be many incomplete paths contending for further considerable. The shortest one is always extended one level further, creating as many new incomplete paths there are branches. These new paths along with old ones are sorted on the values of cost function 'g' and again the shortest path is extended. Since the shortest path is always chosen for extension the path first reaching to the goal is certain to be optimal but it is not guaranteed to find the solution quickly. The following algorithm is simple to give you an idea about how it works.

<u>Algorithm:</u>

Input : START and GOAL states

Local variables: OPEN, CLOSED, NODE, SUCCS, FOUND;

Output : yes or NO

Method :

- Initially store the start node with $g(root) = 0$ in a OPEN list; CLOSED = ϕ ; FOUND = false;
- While (OPEN $\neq \phi$ and FOUND = false) do

  {

  - remove the top element from OPEN list and call it NODE;

- if NODE is the goal node. then FOUND = true
  else.

  {
  - put NODE in CLOSED list;
  - find succs of NODE, if any, and compute their 'g' values and store them in OPEN list;
  - sort all the nodes in the OPEN list based on their cost-function values;

  }

}/* End While */

- if FOUND = true then return yes otherwise
  return NO;
- Stop

In branch and bound method, if $g(x) = 1$ for all operators, then it degenerates to simple breadth first search. from AI point of view, it is as bad as depth first and breadth first. this can be improved if we augment it by dynamic programming. that is, delete those paths which are redundant. We notice that algorithm generally requires generate solution and test it for its goodness solution can be generated using any method on some heuristics. Skeleton of algorithm for 'generate and test' strategy is as follows:

start

- Generate a possible solution
- Test if it is a goal

Scanned with CamScanner

⊙ if not go to start else quit

end.

22) Elaborate idea about hill climbing?

Hill climbing:-

Quality measure turns Depth first search into hill climbing. It is an optimization technique that belongs to the family of local searches. It is a relatively simple technique to implement as a popular first choice is explored. Although more advanced algorithms may give better results, there are situations where hill climbing works well. Hill climbing can be used to solve problems that have many solutions but where some solutions are better than others. Traveling salesman problem can be solved with hill climbing. It is easy to find the solutions that will visit all the cities, but this solution will probably be very bad compared to the optimal solution. If there is some way of ordering choices so that the most promising node is explored first, then search efficiency may be improved. Moving though a tree of paths, hill climbing proceeds in depth first order, but the choices are ordered according to some heuristic value for eg., in traveling salesman problem. straight line distance between two cities can be heuristic measure of remaining distance.

Scanned with CamScanner

Algorithm:-

Input: START and GOAL states

Local variables : OPEN, NODE, SUCCS, FOUND;

Output : yes or NO

Method:

- Store initially the start node in a OPEN list (maintained as stack) ; FOUND = false;
- While (OPEN ≠ empty and Found = false) do

  {
  - remove the top element from OPEN list and call it NODE;
  - If NODE is the goal node, then FOUND = true

    Else
    - find succs of NODE, if any;
    - Sort succs by estimated cost from NODE to goal state and add them to the front of OPEN list;

  } /* End While */
- if FOUND = true then return yes otherwise return NO;
- Stop

Problems with hill climbing:-

There are few problems with hill climbing. The search process may reach to a position that is a solution but from there no more improves the situation. This will happen if we have reached local maximum, a plateau or a ridge.

Scanned with CamScanner

Local Maximum:- It is a state that is better than all its neighbours but not better than some other states which are far away. from this state all moves looks to be worse. In such situation, backtrack to some earlier state and try going in different direction to find a solution.

plateau:- It is a flat area of the search space where all neighbouring states has the same value. It is not possible to determine the best direction In such situation make a big jump to some direction and try to get to new section of the search space.

Ridge:- It is an area of search space that is higher than surrounding areas but that cannot be traversed by single moves in any one direction. It is a special kind of local maxima. Here apply two or more rules before doing the test, i.e., moving in several directions at once.

23) Discuss the algorithm of beam search?

Beam search:- Beam search is a heuristic algorithm in which W number of best nodes at each level always expanded. It progresses level by level and moves downward only from the best W nodes each level. Beam search uses breadth-first search to build its search tree. At each level of tree it generates all successors of the states at the current level, sorts them in order of increasing

Scanned with CamScanner

heuristic values. However, it only considers a W number of states at each level. other nodes are ignored. Best nodes are decided on the heuristic cost associated with the node. Here W is called width of beam search. If B is the branching factor, there will be only W * B nodes under consideration at any depth but only W nodes will be selected. If beam width is smaller, the more states are pruned. If W=1, then it becomes hill climbing search where always best node is chosen from successor nodes. If beam width is infinite, then no states are pruned. and beam search is identical to breadth first search. The beam width bounds the memory required to perform the search, at the expense of risking termination or completeness and optimality. The reason for such risk is that the goal state potentially might have been pruned.

Algorithm:-

Input : START and GOAL states.

Local variables: OPEN, NODE, SUCCS, W-OPEN, FOUND;

Output : yes or NO

Method:

- NODE = Root-node ; FOUND = false;
- if NODE is the goal node, then Found = true else find succs of NODE, if any with

Scanned with CamScanner

its estimated cost and store in OPEN list;

- While (FOUND = false and not able to proceed further) do

  {

  - sort OPEN list;
  - select top W elements from OPEN list and put it in W-OPEN list and empty OPEN list;
  - for each NODE from W-OPEN list

    {

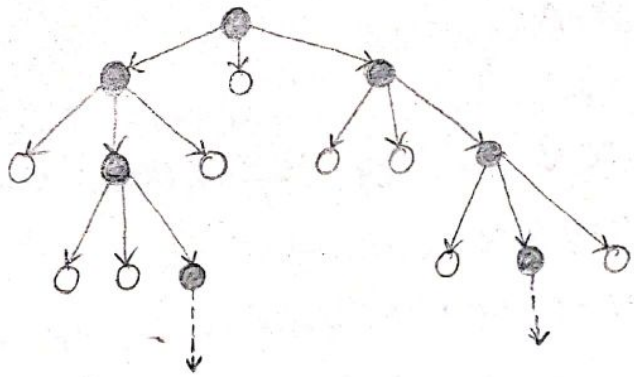
    - if NODE = Goal state then FOUND = true else find succs of NODE, if any with its estimated cost and store in OPEN list;

    }

  } /* End While */

- if FOUND = true then return yes otherwise return NO;
- stop

⊙ The search tree generated using Beam search algorithm, assume W=2 and B=3 is given below. Here, black nodes are selected based on their heuristic values for further expansion.



Scanned with CamScanner

24) Describe best first search Method?

Best first search is based on Expanding the best partial path from current node to goal node. Here forward motion is from the best open node so far in the partially developed tree. The cost of partial paths is calculated using some heuristic.

If the state has been generated Earlier and new path is better than the previous one, then change the parent and update the cost.

It should be noted that in hill climbing, sorting is done on the successors node, whereas in the best first search, sorting is done on the entire list. It is not guaranteed to find an optimal solution, but generally it finds some solution faster than solution obtained from any other method. The performance varies directly with the accuracy of the heuristic Evaluation function.

Algorithm:)

Input: START and GOAL states

Local variables: OPEN, CLOSED, NODE, FOUND;

Output: yes or No

Method:

- initialize OPEN list by root node: CLOSED = ∅;
  FOUND = false;
- While (OPEN = ∅ and FOUND = false) do
  {

Scanned with CamScanner

- if the first element is the goal node, then FOUND = true, else remove it from OPEN list and put it in CLOSED list;
- add its successor, if any, in OPEN list;
- sort the entire list by the value of some heuristic function that assigns to each node the estimate to reach the goal node;

}

- if FOUND = true then return yes otherwise return NO;
- stop

25) State and Explain A* algorithm?

A* Algorithm is a combination of 'branch and bound' and 'best search' methods combined with the dynamic programming principles. It uses a heuristic evaluation function usually denoted by f(x) to determine the order in which the search visit the nodes in the tree. The heuristic function for a node N is defined as follows:

$$f(N) = g(N) + h(N)$$

The function g is a measure of the cost of getting from the start node to the current node N, i.e., it is sum of costs of the rules that were applied along the best path to the current node. The function h is an estimate of additional cost of getting from the current node N to the

Scanned with CamScanner

goal node. The function h is an estimate of additional cost of getting from the current node. This is the place where knowledge about the problem domain is exploited. Generally A* algorithm is called OR graph/tree search algorithm.

A* algorithm incrementally searches all the routes starting from the start node until it finds the shortest path to a goal. Starting with a given node, the algorithm expands the node with the lowest f(x) value. It maintains a set of partial solutions. Unexpected leaf nodes of expanded nodes are stored in a queue with corresponding of values. This queue can be maintained as priority queue.

<u>Algorithm:</u>

Input: START and GOAL states

Local variables: OPEN, CLOSED, Best_Node, SUCCS, OLD, FOUND;

Output: yes or NO

Method:

- initialization OPEN list with start node; CLOSED = ∅; g = 0, f = h, FOUND = false;
- While (OPEN = ∅ and FOUND = false) do

  {

  - remove the node with the lowest value of f from OPEN list and store it is in CLOSED list. call it as a Best_Node;

Scanned with CamScanner

- if (Best_Node = goal state) then FOUND= true else

{

- generate the succs of the Best-Node;
- for each succ do

  {

  - establish parent link of succ;
  - compute g(succ) = g(Best_Node) + cost of getting from Best_Node to succ;
  - if succ ∈ open

    {

    - call the matched node as OLD and add it in the successor list of the Best-Node;
    - ignore the succ node and change the parent of OLD, if required as follows;
      - if g(succ) ~~closed~~ < g(OLD) then make parent of OLD to be Best_Node and change the values of g and f for else ignore;

    }

  - if succ ∈ closed then

    {

    - call the matched node as OLD and add it in the list of the Best-Node successors;

Scanned with CamScanner

- ignore the succ node and change the parent of OLD, if required as follows:
  - if $g(succ) < g(OLD)$ then make parent of OLD to be Best-Node and change the values of g and f for OLD and propogate the change to OLD's children using depth first search else ignore;

  }

- If succ $\notin$ OPEN or CLOSED

  {

  - add it to the list of Best-Node's successors;
  - compute $f(succ) = g(succ) + h(succ)$;
  - put succ on OPEN list with its f value

  }

 }

}

}

- if FOUND = true then return yes otherwise return NO;
- Stop

Let us consider an eg of eight puzzle again and solve it by using A* algorithm. The since evaluation function f(x) is:

Scanned with CamScanner

$f(x) = g(x) + h(x)$, Where

$h(x)$ = the number of tiles not in their goal position in a given state x

$g(x)$ = depth of node x in the search tree

Given

Start state

3	7	6
5	1	2
4	□	8

Goal state

5	3	6
7	□	2
4	1	8

The search tree using A* algorithm for eight puzzle person. It should noted that the quality of solution will depend on heuristic function. This simple heuristic not be used to solve harder eight puzzle problems. Let us consider following puzzle and try solving using earlier heuristic function.

Start state

3	5	1
2	□	7
4	8	6

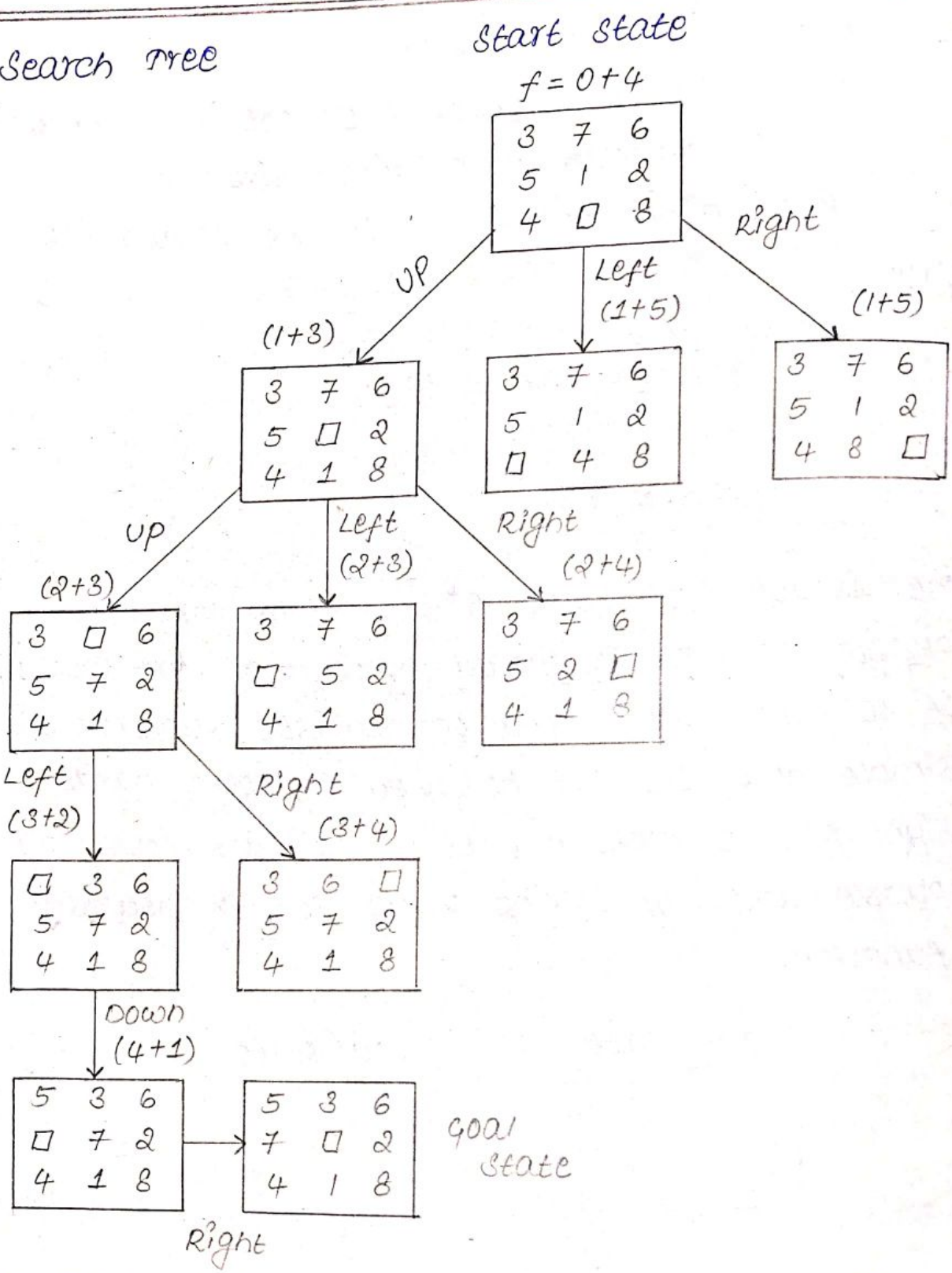
Goal state

5	3	6
7	□	2
4	1	8

P.T.O

Scanned with CamScanner

Search Tree



- A better estimate of h function might be as follows. The function g may remain same.
- h(x) = the sum of the distances of the tiles (1 to 8) from their goal position in a given state x.

Scanned with CamScanner

- Here start state has h(start_state) =
  $3+2+1+0+1+2+2+1 = 12$
- for the sake of brevity, search tree has been omitted.

26) Explain the concept of iterative deepening A*?

Iterative Deepening A* (IDA*) is a combination of the depth first iterative deepening and A* algorithm. Here the successive iterations are corresponding to increasing values of the total cost of a path rather than increasing depth of the search. Algorithm works as follows:

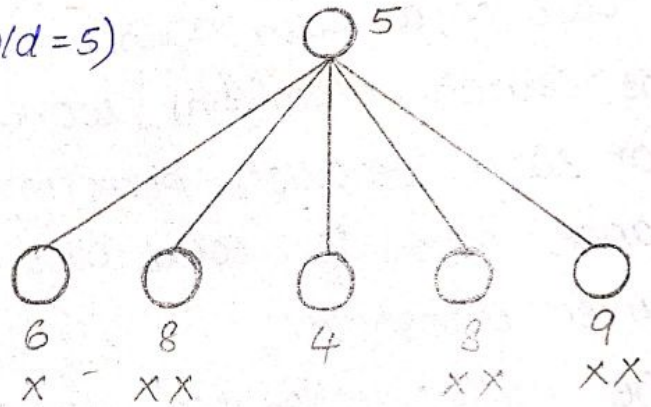
- For each iteration, perform a DFS pruning off a branch when its total cost (g+h) exceeds a given threshold.
- The initial threshold starts at the estimate cost of the start state and increases for each iteration of an algorithm.
- The threshold used for the next iteration is the minimum cost of all values exceeded the current threshold.
- These steps are repeated till we find a goal state.

Let us consider an eg to illustrate the working of IDA* algorithm. Initially, the threshold value is the estimated cost of the start node. In the first iteration, threshold=5,

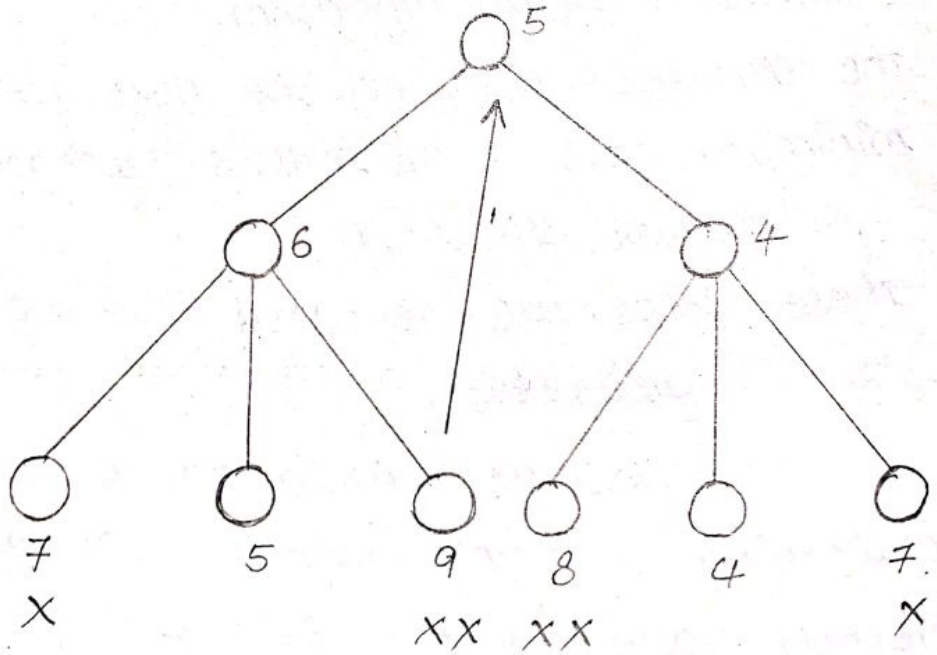
Scanned with CamScanner

Now, we generate all the successors of start node and compute their estimated values as 6, 8, 4, 8 and 9. The successors having values greater than 5 are to be pruned. Now for next iteration, we consider the threshold to be the minimum of the pruned nodes value, that is, threshold = 6, and the node with 6 value along with node with value 4 are retained for further expansion.

1st iteration (threshold = 5)

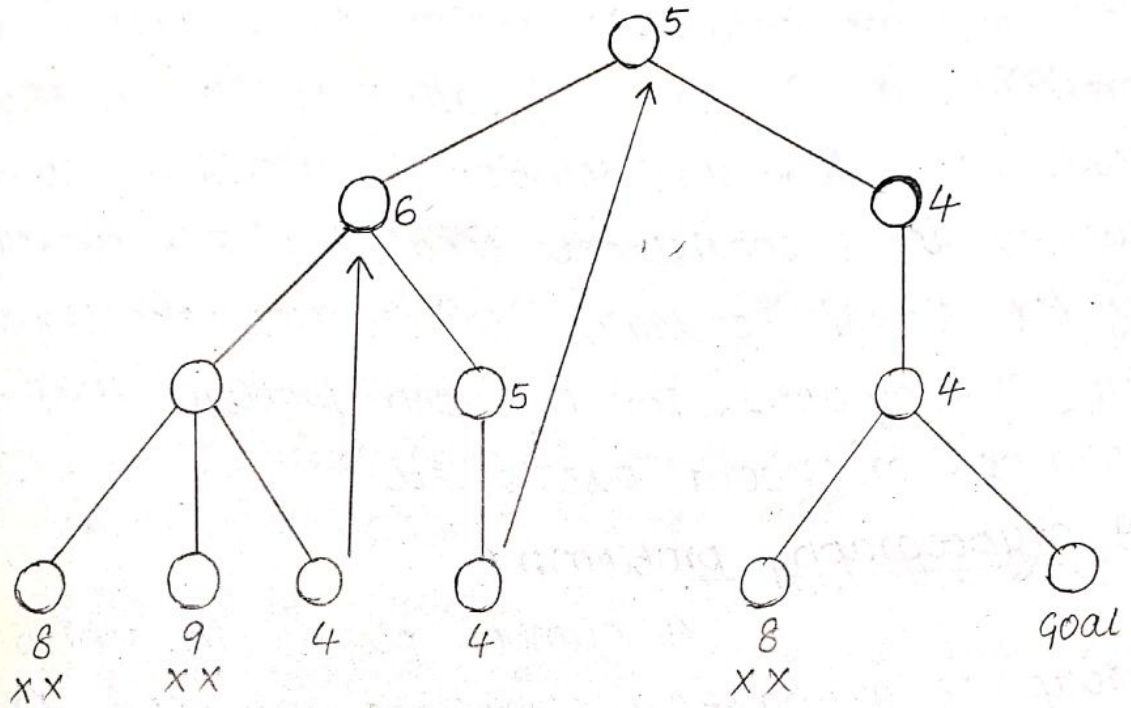


2nd iteration (threshold = 6)



Scanned with CamScanner

3rd iteration (Threshold = 7)



The IDA* will find a solution of least cost or optimal solution, if an admissible monotonic cost function is used IDA* not only finds cheapest path to a solution but uses far less space than A*, and it expands approximately the same number of nodes as that of A* in a tree search. An additional benefit of IDA* over A* is that it is simplier to implement, as there are no open and closed lists to be maintained. A simple recursion performs DFS inside an outer loop will handle iterations.

27) Write notes on constraint satisfaction?

Many AI problems can be viewed as problems of constraint satisfaction in which the goal is to solve some problem state that satisfies a given set of constraints instead of

Scanned with CamScanner

finding optimal path to the solution. Such problems are called constraint satisfaction (CS) problems. search can be made easier in those cases in which the solution is required to satisfy local consistency conditions. For eg some of the simple constraint satisfaction problems are cryptography, the n-Queen problem, map coloring, crossword puzzle, etc

A cryptography problem:-

A number puzzle in which a group of arithmetical operations has some or all of its digits replaced by letters and the original digits must be found. In such a puzzle each letter represents a unique digit. Let us consider the following problems in which we have to replace each letter by a distinct digit (0-9) so that the resulting sum is correct.

```
    B A S E
  + B A L L
  ---------
  G A M E S
```

The n-Queen problem:-

The condition is that on the same row, or column, or diagonal no two queens attack each other.

Scanned with CamScanner

A map colouring problem:-

Given a map, color regions of map using three colours, blue, red, and black such that no two neighbouring countries have the same colour.

In general, we can define a constraint satisfaction problem as follows:

- a set of variables $\{x_1, x_2, .., x_n\}$, with each $x_i \in D_i$ with possible values and
- a set of constraints, i.e., that are assumed to hold between the values of the variables.

The problem is to find, for each i, $1 \leq i \leq n$, a value of $x_i \in D_i$, so that all constraints are satisfied. A CS problem is usually represented as an undirected graph, called constraint graph in which the nodes are the variables and the edges are the binary constraints. We can easily see that a CSP can be given an incremental formulation as standard search problem.

- Start state: the empty assignment, i.e., all variables, are unassigned.
- Goal state: all the variables are assigned values which satisfy constraints.
- Operator: assigns value to any unassigned variable, provided that is does not conflict with previously assigned variables.

Scanned with CamScanner

Every solution must be a complete assignment and therefore appears at depth n if there are variables. Furthermore, the search tree extends only to depth n and hence depth-first search algorithms are popular for CSPs.

Many design tasks can also be viewed as constraint satisfaction problems. Such problems do not require new search methods but they can be solved using any of the search strategies which can be augmented with the list of constraints that change as parts of the problem are solved. The following algorithm is applied for the CSP. This procedure can be implemented as a DF search.

Algorithm:-

- until a complete solution is found or all paths have lead to dead ends.

  {

  - select an unexpanded node of the search graph:
  - apply the constraint inference rules to the selected node to generate all possible new constraints;
  - if the set of constraints contain a contradiction, then report that this path is a dead end.

Scanned with CamScanner

- if the set of constraints ~~contain~~ describes a complete solution, then report success;
- if neither a contradiction nor a complete solution has been found, then apply the problem space rules to generate new partial solutions that are consistent with the current set of constraints, Insert these partial solutions into the search graph.

}

- stop.

28) solve the crypt Arithmetic puzzle for base and ball?

<u>crypt Arithmetic puzzle:)</u>

<u>problem statement:)</u> solve the following puzzle by assigning numeral (0-9) in such a way the each letter is assigned unique digit which satisfy the following addition:

```
  B A S E
+ B A L L
---------
G A M E S
```

⊙ constraints : No two letters have the same value.

⊙ Initial problem state

G=? ; A=? ; M=? ; E=? ; S=? ; B=? ; L=?

⊙ Apply constraint inference rules to generate

Scanned with CamScanner

the relevant new constraints.

- Apply the letter assignment rules to perform all assignments required by the current set of constraints. then choose another rules to generate an additional assignment, which will, in turn, generate new constraints at the next cycle.
- At each cycle, there may be several choices of rules to apply.
- A useful heuristics can help to select the best rule to apply first

For Example:-

if a letter that has only two possible values and another with six possible values, then there is a better chance of guessing right on the first than on the second.

C4	C3	C2	C1	
	B	A	S	E
+	B	A	L	L
G	A	M	E	S

- constraints Equation are:

$E + L = S \rightarrow C_1$

$S + L + C_1 = E \rightarrow C_2$

$2A + C_2 = M \rightarrow C_3$

$2B + C_3 = A \rightarrow C_4$

Scanned with CamScanner

We can easily see that G has to be non-zero digit, so the value of carry C_4 should be 1. and hence $G=1$.

1. $G = C_4 \Rightarrow G = 1$
2. $2B + C_3 = A \rightarrow C_4$
   - Since $C_4 = 1$, therefore, $2B + C_3 > 9 \Rightarrow B$ can take values from 5 to 9.
   - Try the following steps for each value of B from 5 to 9 till we get a possible value of B.
   - If $B=5$ → if $C_3 = 0 \Rightarrow A = 0 \Rightarrow M = 0$ for $C_2 = 0$ or $M = 1$ for $C_2 = 1$ ✗
     → if $C_3 = 1 \Rightarrow A = 1$ ✗ (as $G=1$ already)
   - for $B=6$ we get similar contradiction while generating the search tree.
   - If $B=7$, then for $C_3=0$, we get $A=4 \Rightarrow M=8$ if $C_2=0$ that leads to contradiction later, so this path is pruned, if $C_2=1$ then $M=9$
3. Let us solve $S+L=C_1=E$ and $E+L=S$
   - Using both equations, we get $2L+C_1=0 \Rightarrow L=5$ and $C_1=0$
   - Using $L=5$, we get $S+5=E$ that should generate carry $C_2=1$ as shown

Scanned with CamScanner

- So $S+5>9 \Rightarrow$ possible values for E are $\{2, 3, 6, 8\}$ (with carry bit $C_2=1$)
- If $E=2$ then $S+5=12 \Rightarrow S=7$ (as $B=7$ already)
- If $E=3$ then $S+5=13 \Rightarrow S=8$
- Therefore $E=3$ and $S=8$ are fixed up

4. Hence we get the final solution as given below and on backtracking, we may find more solutions. In this case we get only one solution.

$G=1$; $A=4$; $M=9$; $E=3$; $S=8$; $B=7$; $L=5$

29) Compare and contrast game problem and state space problem?

It should be noted that there is a natural correspondence between games and state space problems.

For example:-

In state space problems, we have a start state, intermediate states, rules or operators and a goal state. In game problems also we have a start state, legal moves and winning positions.

- Comparisions between state space problems and Game problems

P.T.O

Scanned with CamScanner

State Space problems	Game problems
States	Legal board positions
Rules	Legal moves
Goal	Winning positions

30) Develop a game tree for NIM game with MAX playing first?

The Nim game is believed to have originated in china. but the exact location of its origin is not certain. charles L. Bouton of Harvard University developed the complete theory of the Nim game in the year 1901.

The Game, there is a single pile of matchsticks (>1) and two players. Moves are played / made by the players alternately. In a move, each player can pick up a maximum of half the number of match sticks in the pile. Whoever takes the last matchstick loses.

We will develop complete game tree with either MAX as first player.

- The convention used for drawing a game tree is that each node contains the total number of sticks in the pile and is labelled as W or L in accordance with the status labelling procedure.

Scanned with CamScanner

Level	complete game tree for NIM with MAX playing first
MAX	7 → L
MIN	6 → L; 5 → L; 4 → L
MAX	5 → W; 4 → W; 3 → L; 2 → W
MIN	4 → L; 3 → W; 2 → L; 1 → W
MAX	3 → L; 2 → W; 1 → L
MIN	2 → L; 1 → W
MAX	1 → L

Game tree for NIM with MAX playing first.

Scanned with CamScanner

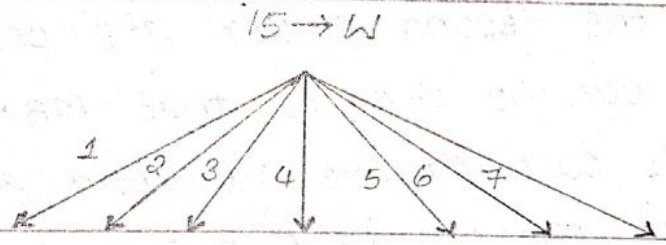
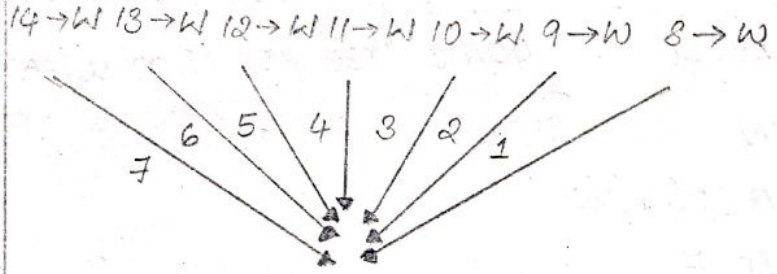
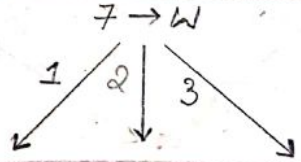
31) Discuss the cases of NIM game?

Cases of NIM Game:-

<u>case 1:-</u>

If MAX is the first player and $N \notin \{3, 7, 15, 31, 63, \ldots\}$, then MAX will always win.

⊙ consider a pile of 29 sticks and let MAX be the first player. The complete game tree for this case. it can be seen that MAX always wins. This can be validated for any number of sticks $\notin \{3, 7, 15, 31, \ldots\}$, thus, in this case, we can conclude by observing the figure that MAX is bound to win irrespective of how MIN plays

Level	Game tree for Nim with MAX playing first
MAX picks up 14 sticks	
MIN can pick up 1 to 7 sticks	
MAX picks up sticks in such a manner that 7 sticks are left	
MIN can pick up 1 to 3 sticks	

Scanned with CamScanner

MAX picks up n sticks, so that 3 sticks are left.	6 → W 5 → W 4 → W 3 2 1
MIN Has to pickup 1 stick	3 → W 1
MAX picks up 1 stick	2 → W 1
MIN	1 → W

<u>Case 2:)</u>

If MAX is the second player and $N \in \{3, 7, 15, 31, 63, \ldots\}$, then Max will always wins.

- consider a pile of 15 sticks and Let MAX be the second player. The complete game, it can be observed that MAX always wins.
- This case can be validated for any number of sticks $\in \{3, 7, 15, 31, 63, \ldots\}$

Level	Game tree for Nim with MAX Playing second
MIN can pickup 1 to 7 sticks	15 → W 1 2 3 4 5 6 7

Scanned with CamScanner

MAX picks up in such a way that 7 sticks are left	14 13 12 11 10 9 8 7 6 5 4 3 2 1
MIN can pick 1 to 3 sticks	7 → W 1 2 3
MAX picks up in such a way that 3 sticks are left	6 → W 5 → W 4 → W 3 2 1
MIN can only pick 1 stick	3 → W 1
MAX picks up 1 stick so that 1 stick is left for MIN	2 → W 1
MIN	1 → W

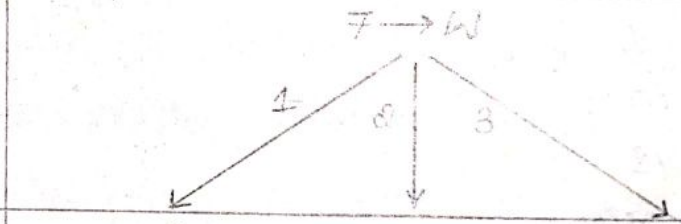
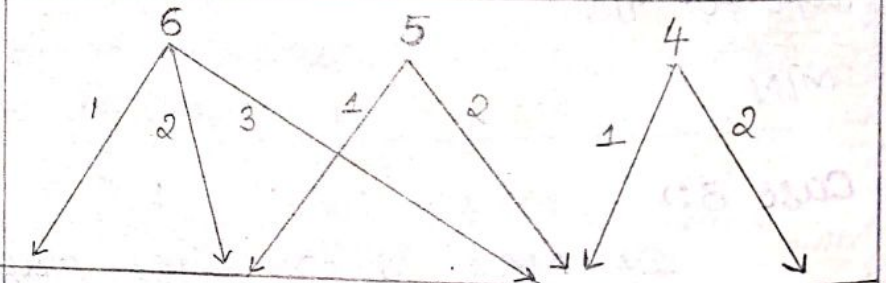
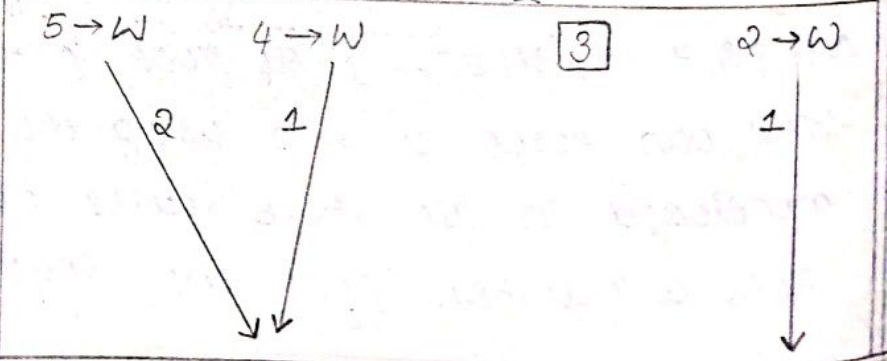
Case 3:)

If MAX is the first player and $N \in \{3, 7, 15, 31, 63, \ldots\}$ at root of the game, then MAX can force a win using the strategy mentioned in all above cases except when MAX gets a number from the sequence....

Scanned with CamScanner

$\{3, 7, 15, 31, 63, \ldots\}$ as its turn.

- Assume that $N = 15$, the MAX wins in all cases except when it gets 7 matchsticks in its turn. in above case
- MAX can even win the game when it gets 7 sticks at its turn in the game for all values except when it gets 3 sticks at its turn in the game.
- Therefore, we can conclude that if MAX is playing with M sticks. $\in \{3, 7, 15, 31, 63\}$ at any point in the game. it can win for all cases except for the case when it gets value 3. In these cases, MAX can win if opponent is playing without any strategy.

Level	Game tree for NIM with MAX gets 7.
MAX can pickup 1 to 3 sticks	
MIN	
MAX picks up sticks in such a way that 3 sticks are left for MIN	

Scanned with CamScanner

MIN	$3 \rightarrow W$ $\xrightarrow{1}$ $\quad 1 \rightarrow W$
MAX picks up only 1 stick	$2 \rightarrow W$ $\xrightarrow{1}$
MIN	$1 \longrightarrow W$

Case 4:)

If MAX is the second player and $N \notin \{3, 7, 15, 31, 63, \ldots\}$ then MAX can force a win using the above mentioned strategy in all cases except when it gets a number from the sequence $\{3, 7, 15, 31, 63, \ldots\}$ as its turn.

Level	complete game tree for Nim with MIN playing first
MIN can pickup 1 to 14 sticks.	$29 \rightarrow W$ $\xrightarrow{1}$ $28 \rightarrow W$, $\xrightarrow{2}$ $27 \rightarrow W$, $\xrightarrow{3}$ $26 \rightarrow W$, $\xrightarrow{4}$ $25 \rightarrow W$, ... $\xrightarrow{12}$ $17 \rightarrow W$, $\xrightarrow{13}$ $16 \rightarrow W$, $\xrightarrow{14}$ $\boxed{15}$
MAX picks up sticks in such a way that 15 sticks are left for MIN.	$\xrightarrow{13}$, $\xrightarrow{12}$, $\xrightarrow{11}$, $\xrightarrow{10}$, $\xrightarrow{2}$, $\xrightarrow{1}$

Scanned with CamScanner

MIN can pickup 1 to 7 sticks	15→W 1 → 14→W 2 → 13→W 3 → 12→W 4 → 11→W 5 → 10→W 6 → 9→W 7 → 8→W
MAX pickup sticks in such a manner that 7 sticks are left for MIN	14→W: 7 13→W: 6 12→W: 5 11→W: 4 10→W: 3 9→W: 2 8→W: 1
MIN can pickup 1 to 3 sticks	7→W 1 → 6→W 2 → 5→W 3 → 4→W
MAX pickup sticks in such a way that 3 sticks are left for MIN	6→W: 1 5→W: 2 4→W: 3
MIN can only pickup 1 stick	3→W 1 → 2→W
MAX	2→W

Scanned with CamScanner

32) Write notes on the process of using evaluation function?

The process of evaluation of a game is determined by the structural features of the current state. The steps involved in the evaluation procedure are as follows:

- The first step is to decide which features are of value in a particular game.
- The next step is to provide each feature with a range of possible values.
- The last step is devise a set of weights in order to combine all the feature values into a single value.

In the absence of a practical way of evaluating the exact status of successor game states, we have to resort to heuristic approxima-tion. It is important to understand that certain features in a game position contribute to its strength. While others tend to weaken it. A proper static evaluation function can convert all judge-ments about board situations into a single overall quality number. Therefore, the purpose of evaluation function is to provide the best judgement regarding a position in the game in terms of the probability that the MAX player has a greater chance winning from this position relative to other similar positions

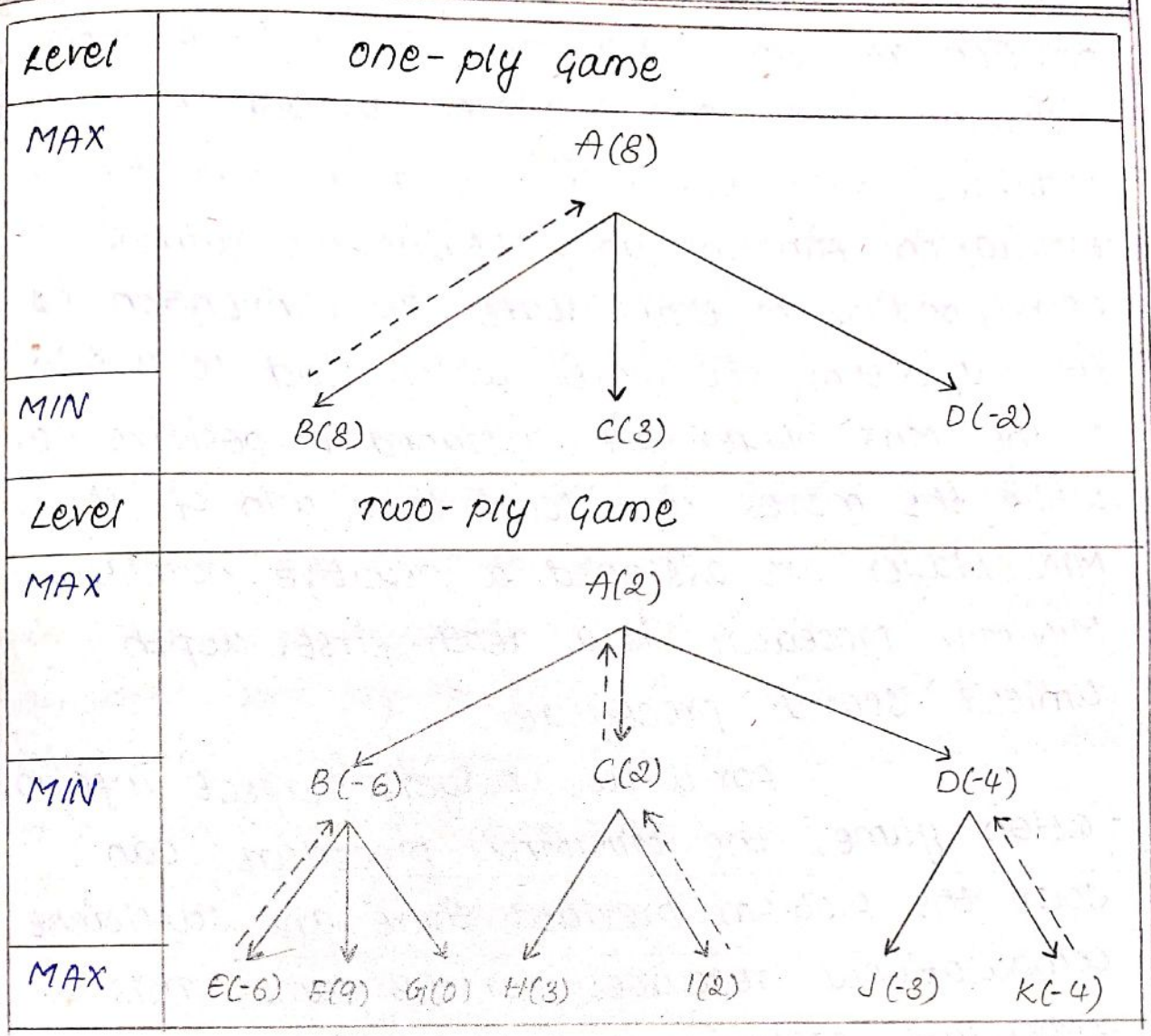
Scanned with CamScanner

The best Evaluation function based on the Experience of Expects who are well versed with the game.

Evaluation function represent estimates of a given situation in the game rather than accurate calculations. This function provides numerical assessment of how favourable the game state is MAX. We can use a convention in which a positive number indicates a good position for MIN while a negative number indicates a bad position. The general strategy for MAX is to play in a manner that it maximizing its winning chances. While simultaneo-usly minimizing the chance of the opponent. The heuristic values of the nods are determined at some level and then the state accordingly propogate up to the root. The node which offers the best path is then choosen to make a move. Let us assume the root node to be a MAX node. consider one ply and two ply games.

P.T.O

Scanned with CamScanner



The procedure through which the scoring information travels up the game tree is called MINIMAX procedure. This procedure represent a recursive algorithm for choosing the move in two player game. In this, a value is associated with each position or state of the game value is computed using an evaluation function and it denotes the extent to which it would be favourable for a player to reach that position the player is then required to make a move with maximizes the minimum value of the

Scanned with CamScanner

position resulting from the opponent's possible following moves. The MINIMAX procedure evaluates each leaf node using a heuristic evaluation function and obtains the values corresponding to that state. By convention of this algorithm, the moves which lead to a win of the MAX player are assigned a positive no, while the moves that lead to a win of the MIN player are assigned a negative number. MINIMAX procedure is a depth-first, depth limited search procedure.

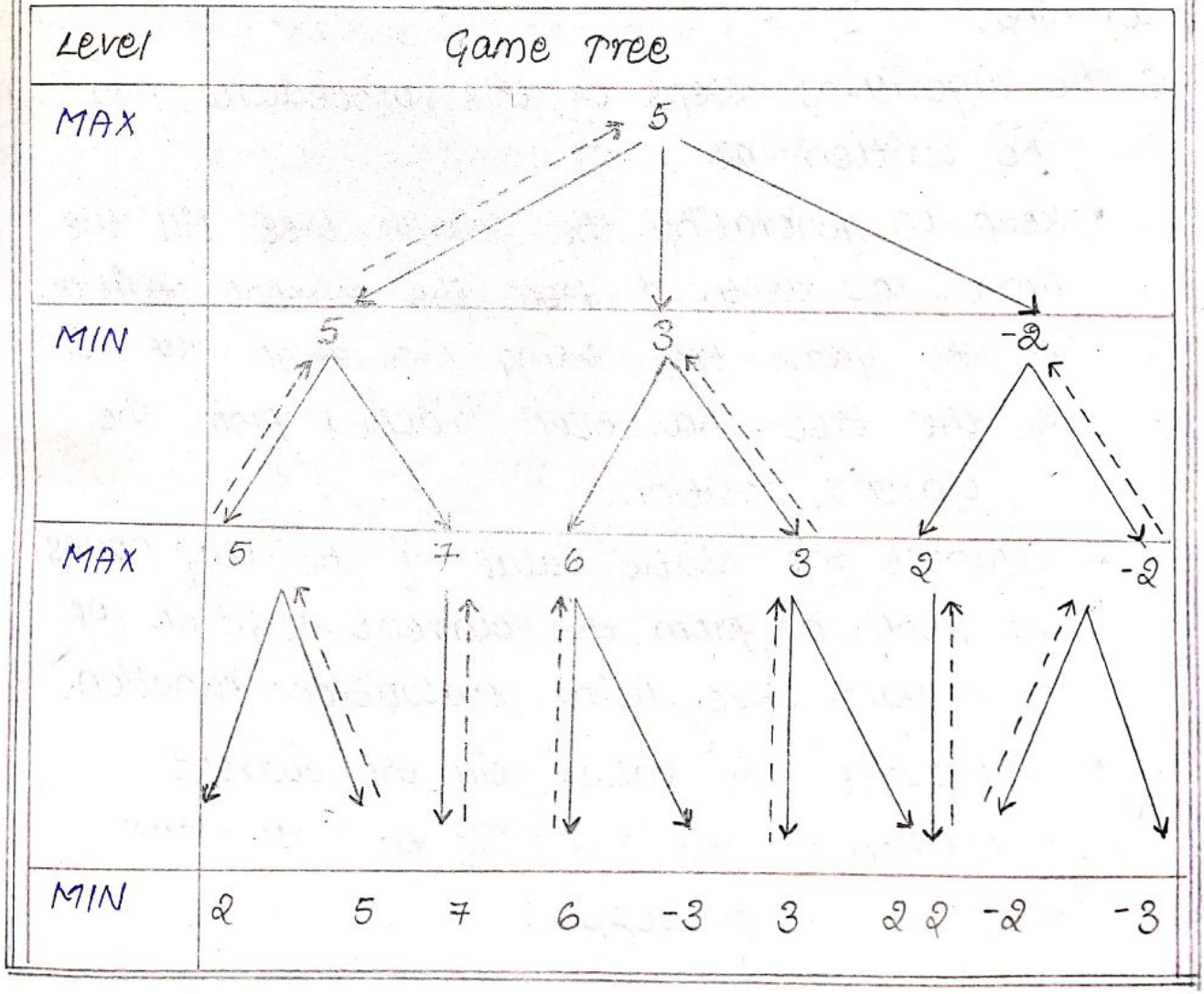
For a two player, perfect inform-ation game, the MINIMAX procedure can solve the problem provided there are sufficient computational resources for the same. This procedure assumes that each player takes the best option in each step. MINIMAX procedure starts from the leaves of the tree and then proceeds upwards towards the root.

38) Generate a game tree using MINIMAX procedure?

Lack of sufficient computational resources prevent the generation of a complete game tree, hence the search depth is restricted to a constant. The estimated scores generated by a heuristic evaluation function for leaf

Scanned with CamScanner

nodes are propagated to the root using MINIMAX procedure can solve the problem provided. Which is recursive algorithm where a player tries to maximize its chances of a win while simultaneously minimizing that of the opponent. The player hoping to achieve a positive number is called the maximizing player, while the opponent is called the minimizing player. At each move, the MAX player will try to take a path that leads to a large positive number; on the other hand, the opponent will try to force the game towards situations with strongly negative static evaluations this game tree is generated.



Scanned with CamScanner

Using MINIMAX procedure up to a depth of three. At MAX level, Maximum value of its success nodes is assigned, whereas at MIN level, minimum value of its succesor nodes is assigned.

The MAX node moves to a state that has a score of 5 in the example considered above. After the MIN will get a chance to play a move. Whenever MAX gets a chance to play, it will generates game tree of depth 3 from the state generated by MIN player in order to decide its next move. The process will continue till the game ends with, hopefully, MAX player winning.

⊙ The algorithms steps of this procedure may be written as

- Keep on generating the search tree till the limit, say depth d ~~from the current position of the game tree using evaluation function.~~ of the tree. has been reached from the current position.
- Compute the static value of the leaf nodes at depth d from the current position of the game tree using evaluation function.
- Propagate the values till the current position on the basis of the MINIMAX strategy.

Scanned with CamScanner

MINIMAX strategy:-

The steps in the MINIMAX strategy as written as follows:

- If the level is minimising level, then
  - Generate the successors of current position.
  - Apply MINIMAX to each of successors.
  - Return the minimum of the results.
- If the level is a maximising level, then
  - Generate the successors of current position.
  - Apply MINIMAX to each of these successors.
  - Return the maximum of the results.

MINIMAX Algorithm:-

```
MINIMAX (pos, Depth, player)
{
 • If DEPTH (pos, Depth) then return ({val = EVAL
        (pos, player), path = Nil})
  else
  {
   • SUCC_List = GEN(pos);
   • If SUCC-List = Nil then return ({val= EVAL
        (pos, player), path= Nil})
    else
    {
     • Best_val = Minimum value returned by
                EVAL function;
     • For each succ ∈ SUCC-List DO
       {
```

Scanned with CamScanner

```
• SUCC-Result = MINIMAX (SUCC, Depth +1,
                        ~ player);
• NEW-value = - val of SUCC-Result;
    • If NEW-value > Best-val then
      {
        • Best-val = NEW-value;
        • Best-path = Add (SUCC, path of SUCC-
                            Result);
      };
  };
  • Return ({val = Best-val, path = Best-path});
  }
 }
}
```

34) Explain the steps of Alpha-Beta pruning?

The strategy used to reduce the number of tree branches explored and the number of static evaluation applied is known as alpha beta pruning. This procedure is also called backward pruning, which is modified depth-first generation procedure. The purpose of applying this procedure is to reduce the amount of work done in generating useless nodes and is based on common sense or basic logic.

The alpha beta pruning procedure

Scanned with CamScanner

requires the maintenance of two threshold values: one representing a lower bound (α) on the value that a maximizing node may ultimately be assigned and another representing upper bound (β). on the value that a minimizing node may be assigned. Each MAX node has an alpha value, which never decreases and each MIN node has a beta value, which never increases. These values are set and updated when the value of a successor node is obtained, The search is depth first and stops at any MIN node whose beta value is smaller than or equal to the alpha value of its parent, as well as at any MAX node whose alpha value is greater than or equal to the beta value of its parent.

Levels	first step
MAX	A
MIN	(≤ 2) B
MAX	(2) D

Let us consider the systematic development of a game tree and propagation of α and β values using alpha-beta (α-β)

Scanned with CamScanner

pruning algorithm up to second level stepwise in depth first order. The MAX player expands root node A to B and suppose MIN player expands B to D.

Assume that the evaluation function generates $\alpha = 2$ for state D. At this point, the upper bound value $\beta = 2$ at state B and is shown as ≤ 2.

After the first step, we have to backtrack and generate another state E from B in the second step. The state E gets $\alpha = 7$ and since there is no further successor of B, the β value at state B becomes equal to 2. Once the β value is fixed at MIN level, the lower bound $\alpha = 2$ gets propagated to state A as ≤ 2.

In the third step, expand A to another successor C, and then expand C's successor to F with $\alpha = 1$, we note that the value at C is ≤ 1 and the value of a root A cannot be less than 2; the path from A through C is not useful and thus further expansion of C is pruned. Therefore there is no need to explore the right side of the tree fully as that result is not going to alter the move decision. Since there are no further successors of A, the value of root is fixed as 2, that is $\alpha = 2$,

Scanned with CamScanner

Levels	Second step
MAX	(≥2) A
MIN	(=2) B
MAX	(2) D (7) E

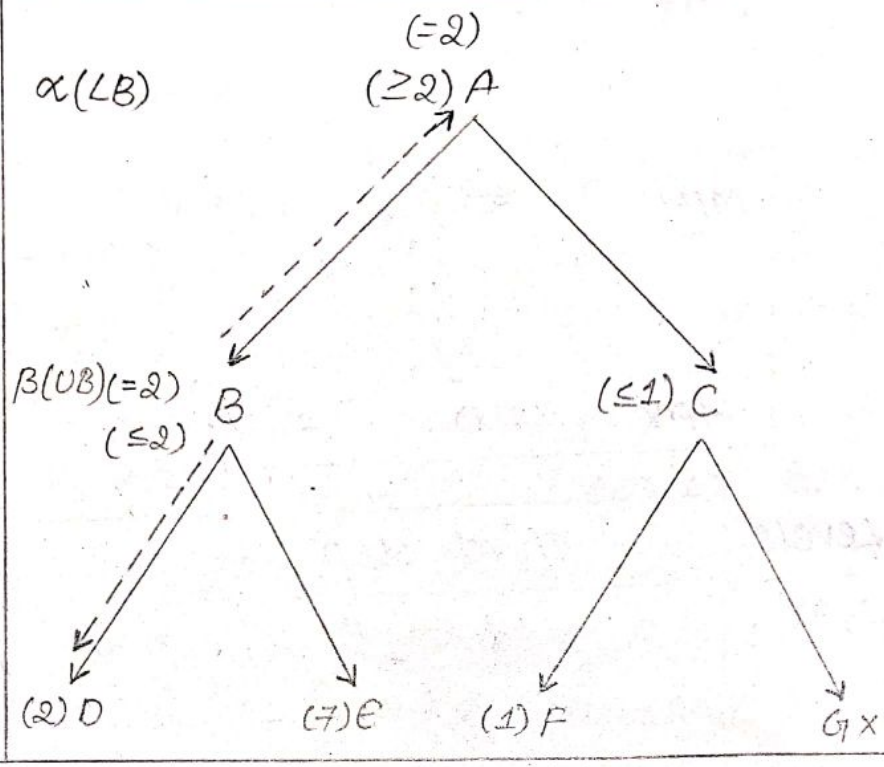
Levels	Third step
MAX	(=2) A
MIN	(=2) B (≤1) C
MAX	(2) D (7) E (1) F G x

⊙ The complete diagram of game tree gene-
-ration using (α-β) pruning algorithm.

P.T.O

Scanned with CamScanner

Levels	Game tree upto second level using α-β pruning algorithm
MAX	α(<B) (=2) (≥2) A
MIN	β(UB)(=2) (≤2) B (≤1) C
MAX	(2) D (7) E (1) F G ×



35) What are the refinements of α-β pruning?

In addition to α-β pruning, a number of modifications can be made to the MINIMAX procedure in order to improve its performance. one of the important factors that need to be considered during a search is when to stop going deeper in the search tree.

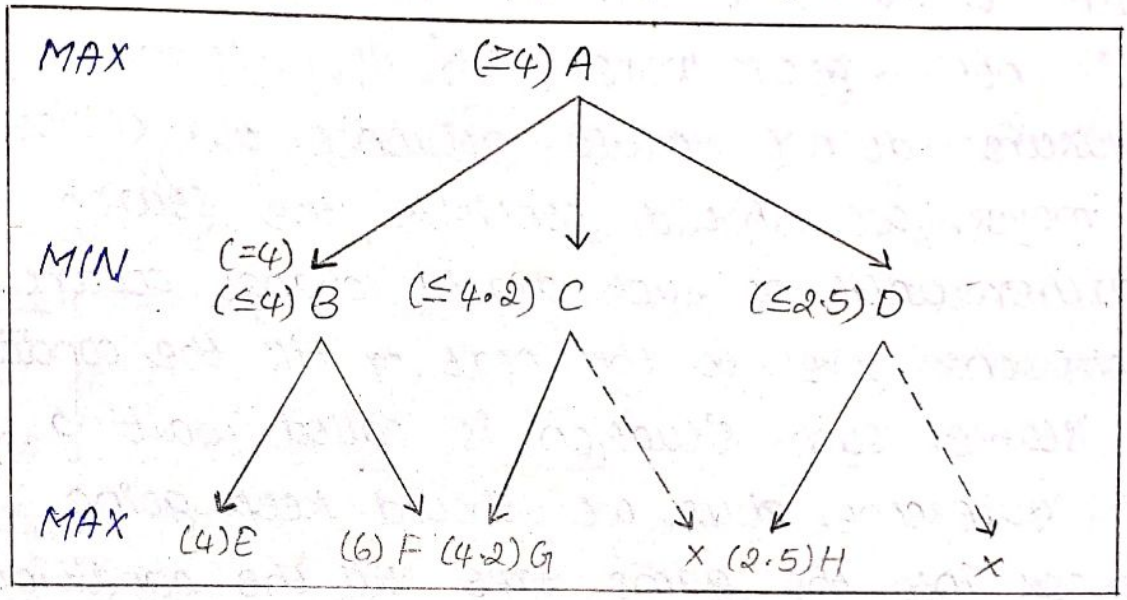
Further the idea behind α-pruning procedure can be extended by cutting-off additional paths that appear to be slight improvements over paths that have already been explored.

Pruning of slightly Better paths:-

We see that the value 4.2 is only slightly better than 4, so we

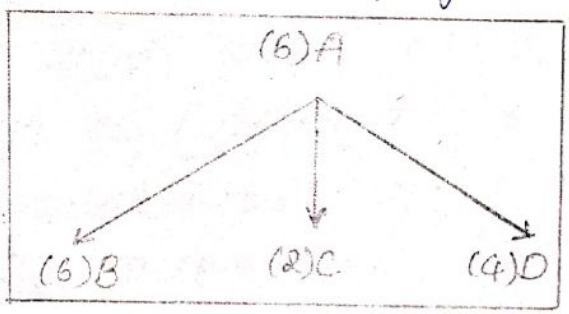
Scanned with CamScanner

may terminate further exploration of node c. terminating exploration of a sub tree that offers little possibility for improvement over known paths is called futility cut off.

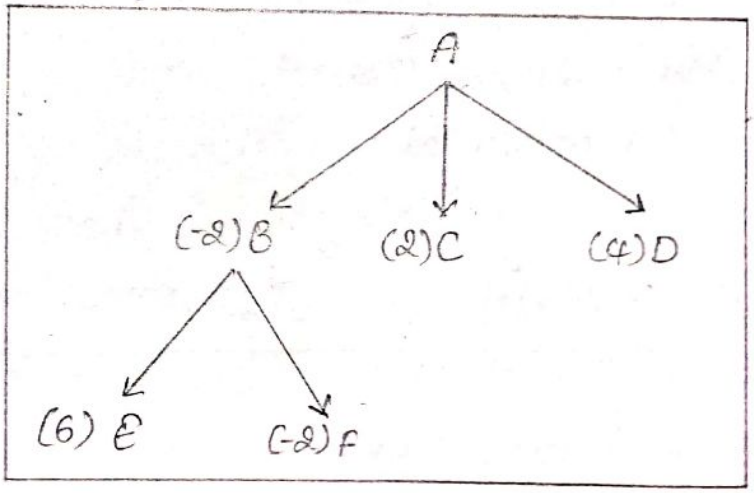


<u>Waiting for Quiescence:-</u>

consider a one level deep game tree.

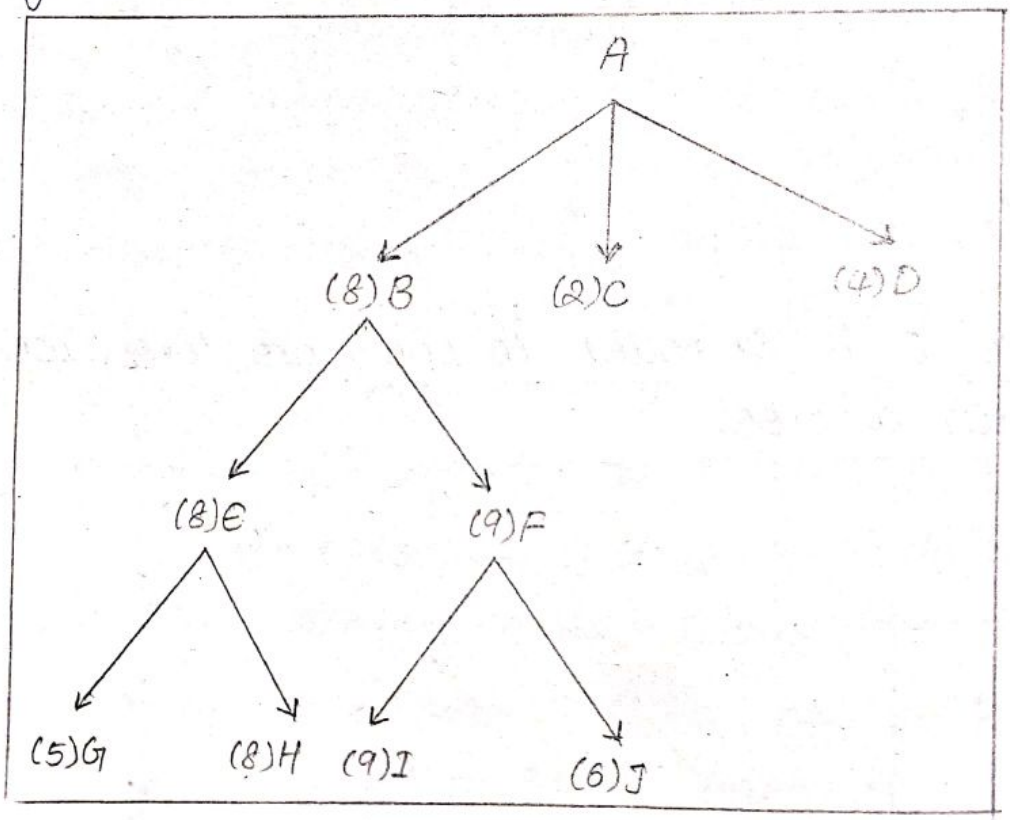


If node B is extended to one more level, we obtain a tree.



Scanned with CamScanner

We can see that our estimate of the worth of node B has changed. We can stop exploring the tree at this level and assign a value of -2 to B, and therefore, decide that B is not a good move. Such short-term measures do not unduly influence our choice of moves. We should continue the search further until no such drastic change occurs from one level to the next or till the condition is stable. Such situation is called waiting for quiescene. Thus, we should keep going deeper into the game tree till the condition is stable and then make a decision regardi-ng the move.



Scanned with CamScanner

36) Write notes on iterative deepening?

Rather than searching till a fixed depth in a given game tree, it is advisable to first search only one-ply, then apply MINIMAX to two-ply, then three-ply till the final goal state is searched. There is a good reason why iterative deepening is popular in case of games such as chess and others programs. In competitions, there is an average amount of time allowed per move. That idea that enables us to conquer this constraint is to do as much look-ahead as can be done in the available time. If we use iterative deepening, we can keep on increasing the look-ahead depth until we run of time. We can arrange to have a record of the best move for a given look ahead even if we have to interrupt our attempt to go one level deeper this could not be done using depth first search.

37) Write notes on propositional calculus?

propositional calculus refers to a language of propositions in which a set of rules are used to combine simple propositions to perform compound propositions with the help of certain logical operators. These logical operators are often called connectives;

Scanned with CamScanner

Examples of some connectives are not ($\sim$), and ($\wedge$), or ($\vee$), implies ($\rightarrow$), and equivalence ($\leftrightarrow$) In PC, it is extremely important to understand the concept of a well-formed formula. A well formed formula is defined as a symbol or a string of symbols generated by the formal grammar of a formal language. The following are some important properties of a well formed formula in PC.

- The smallest unit is considered to be a well formed formula.
- If α is a well-formed formula, then $\sim\alpha$ is also a well formed formula.
- If α and β are well formed formulae, then $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are also well-formed formulae.

A propositional expression is called a well formed formula. if and only if it satisfies the above properties.

38) Explain the concept of propositional calculus logic with an example?

Propositional logic deals with the validity, satisfiability, and unsatisfiability of a formula and the derivation of a new formula using equivalence laws. Each row of a truth table for a given formula α is

Scanned with CamScanner

called its interpretation under which the value of a formula may be either true or false. A formula α is said to be a tautology if and only if the value of α is true for all its interpretations. Now, the validity, satisfiability and unsatisfiability of a formula may be determined on the basis of the following conditions.

- A formula α is said to be valid if and only if it is a tautology.
- A formula α is said to be satisfiable if there exist atleast one interpretation for which α is true.
- A formula α is said to be unsatisfiable if the value of α is false under all interpretations.

39) Enumerate the rule of Natural deduction system?

Natural Deduction system (NDS) is thus, called because of the fact that it mimics the pattern of natural reasoning. This system is based on a set of deductive inference rules. Assuming that $A_1, \ldots A_k$, where $1 \leq k \leq n$, are a set of atoms and α_j, where $1 \leq j \leq m$, and β are well-formed formulae, the inference rules may be stated below.

Scanned with CamScanner

Rule Name	Symbol	Rule	Description
Introducing $\wedge$	$(I:\wedge)$	If $A_1, \ldots, A_n$ then $A_1 \wedge \ldots \wedge A_n$	If $A_1, \ldots, A_n$ are true, then their conjunction $A_1 \wedge \ldots \wedge A_n$ is also true
Eliminating $\wedge$	$(E:\wedge)$	If $A_1, \ldots, A_n$ then $A_i (1 \leq i \leq n)$	If $A_1 \wedge \ldots \wedge A_n$ is true, then any A_i is also true.
Introducing $\vee$	$(I:\vee)$	If any $A_i (1 \leq i \leq n)$ then $A_1 \vee \ldots \vee A_n$	If any $A_i (1 \leq i \leq n)$ is true, then $A_1 \vee \ldots \vee A_n$ is also true.
Eliminating $\vee$	$(E:\vee)$	If $A_1 \vee \ldots \vee A_n$ $A_1 \rightarrow A, \ldots, A_n \rightarrow A$ then A	If $A_1 \vee \ldots \vee A_n$, $A_1 \rightarrow A$, $A_2 \rightarrow A, \ldots,$ and $A_n \rightarrow A$ are true, then A is true.
Introducing $\rightarrow$	$(I:\rightarrow)$	If from $\alpha_1, \ldots \alpha_n$ infer B is proved then $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow B$ is proved	If given that $\alpha_1, \alpha_2, \ldots,$ and α_n are true and from these we deduce B then $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow B$ is also true
Eliminating $\rightarrow$	$(E:\rightarrow)$	If $A_1 \rightarrow A, A_1,$ then A	If $A_1 \rightarrow A$ and A_1 are true then A is also true This is called Modus ponen rule

Scanned with CamScanner

Introducing $\leftrightarrow$	$(I: \leftrightarrow)$	If $A_1 \rightarrow A_2$, $A_2 \rightarrow A_1$ then $A_1 \leftrightarrow A_2$	If $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$ are true then $A_1 \leftrightarrow A_2$ is also true.
Eliminating $\leftrightarrow$	$(E: \leftrightarrow)$	If $A_1 \leftrightarrow A_2$ then $A_1 \rightarrow A_2$, $A_2 \rightarrow A_1$	If $A_1 \leftrightarrow A_2$ is true then $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$ are true.
Introducing $\sim$	$(I: \sim)$	If from A infer $A_1 \wedge \sim A_1$ is proved then $\sim A$ is proved	If from A, a contradiction is proved then truth of $\sim A$ is also proved.
Eliminating $\sim$	$(E: \sim)$	If from $\sim A$ infer $A_1 \wedge \sim A_1$ is proved then A is proved	If from $\sim A$, a contradiction is proved then truth of A is also proved.

40) prove that $A \wedge (B \vee C)$ is deduced from $A \wedge B$?

<u>Solution:)</u>

The theorem in NDS can be written as from $A \wedge B$ infer $A \wedge (B \vee C)$ in NDS. we can prove the theorem as follows

P.T.O

Scanned with CamScanner

Description	formula	comments
Theorem	from $A \wedge B$ infer $A \wedge (B \vee C)$	To be proved
Hypothesis (given)	$A \wedge B$	1
$E: \wedge(1)$	A	2
$E: \wedge(1)$	B	3
$I: \vee(3)$	$B \vee C$	4
$I: \wedge(2, 4)$	$A \wedge (B \vee C)$	proved

If we assume that $\alpha \rightarrow \beta$ is true, then we can conclude that β is also true if α is true. It can be represented in the form of a theorem of NDS as from α. infer β and if we can prove the theorem then we can conclude the truth of $\alpha \rightarrow \beta$. The converse of this is also true.

4) prove the theorem infer

$$[(A \rightarrow B) \wedge (B \rightarrow C)] \rightarrow (A \rightarrow C)$$

The theorem infer $[(A \rightarrow B) \wedge (B \rightarrow C)] \rightarrow (A \rightarrow C)$ is reduced to the theorem from $(A \rightarrow B), (B \rightarrow C)$ infer $(A \rightarrow C)$ using deduction theorem. further to prove $A \rightarrow C$, we will have to prove a sub theorem from A infer C.

Scanned with CamScanner

Description	formula	comments	
theorem	from $A \to B$, $B \to C$ infer $A \to C$	To be proved	
Hypothesis 1	$A \to B$	1	
Hypothesis 2	$B \to C$	2	
sub-theorem	from A infer C	3	
Hypothesis	A		3.1
$E: \to (1, 3.1)$	B		3.2
$E: \to (2, 3, 2)$	C		3.3
$I: \to (3)$	$A \to C$	proved	

42) Mention the rules of semantic tableau?

Here α and β are two formulae.

Rule No	Tableau tree	Explanation
Rule 1	$\alpha \wedge \beta$ is true if both α and β are true $\alpha \wedge \beta$ α β	A tableau for a formula $(\alpha \wedge \beta)$ is constructed by adding both α and β to the same path.
Rule 2	$\sim(\alpha \wedge \beta)$ is true if either $\sim\alpha$ or $\sim\beta$ is true $-(\alpha \wedge \beta)$ $-\alpha$ $\quad$ $-\beta$	A tableau for a formula $\sim(\alpha \wedge \beta)$ is constructed by adding two new paths one containing $\sim\alpha$ and the other containing $\sim\beta$

Scanned with CamScanner

Rule 3	$\alpha \vee \beta$ is true, if Either α or β is true $\alpha \vee \beta$ $\alpha \quad \beta$	A tableau for a formula $(\alpha \vee \beta)$ is constructed by adding two new paths one containing α and the other containing β
Rule 4	$\sim(\alpha \vee \beta)$ is true if both $\sim\alpha$ and $\sim\beta$ are true $\sim(\alpha \vee \beta)$ $\sim\alpha$ $\sim\beta$	A tableau for $\sim(\alpha \vee \beta)$ is constructed by adding $\sim\alpha$ & $\sim\beta$ on the same path.
Rule 5	$\sim(\sim\alpha)$ is true then α is true $\sim(\sim\alpha)$ α	A tableau for $\sim(\sim\alpha)$ is constructed by adding α on the same path.
Rule 6	$\alpha \rightarrow \beta$ is true then $\sim\alpha \vee \beta$ is true $\alpha \rightarrow \beta$ $\sim\alpha \quad \beta$	A tableau for a formula $\alpha \rightarrow \beta$ is constructed by adding two new paths: one containing $\sim\alpha$ and the other containing β
Rule 7	$\sim(\alpha \rightarrow \beta)$ true then $\alpha \wedge \sim\beta$ is true $\sim(\alpha \rightarrow \beta)$ α $\sim\beta$	A tableau for a formula $\sim(\alpha \rightarrow \beta)$ is constructed by adding both α and $\sim\beta$ to the same path

Scanned with CamScanner

Rule 8	$\alpha \leftrightarrow \beta$ is true then $(\alpha \wedge \beta) \vee (\sim\alpha \wedge \sim\beta)$ is true $\alpha \leftrightarrow \beta$ $\alpha \wedge \beta$ $\quad$ $\sim\alpha \wedge \sim\beta$	A tableau for a formula $\alpha \leftrightarrow \beta$ is constructed by adding two new paths: one containing $\alpha \wedge \beta$ and other $\sim\alpha \wedge \sim\beta$ which are further Expanded.
Rule 9	$\sim(\alpha \leftrightarrow \beta)$ is true then $(\alpha \wedge \sim\beta) \vee (\sim\alpha \wedge \beta)$ is true $\sim(\alpha \leftrightarrow \beta)$ $\alpha \wedge \sim\beta$ $\quad$ $\sim\alpha \wedge \beta$	A tableau for a formula $\sim(\alpha \leftrightarrow \beta)$ is constructed by adding two new paths: one containing $\alpha \wedge \sim\beta$ and the other $\sim\alpha \wedge \beta$ which are further Expanded.

43) Construct a semantic tableau for a formula $(A \wedge \sim B) \wedge (\sim B \rightarrow C)$

The construction of the semantic tableau for the given formula $(A \wedge \sim B) \wedge (\sim B \rightarrow C)$

Description	Formula	Line number
Tableau root	$(A \wedge \sim B) \wedge (\sim B \rightarrow C)$	1
Rule 1(1)	$A \wedge \sim B$	2
	$\sim B \rightarrow C$	3
Rule 1(2)	A	4
	$\sim B$	5

Scanned with CamScanner

Rule 6(3) Rule 3(6)	$\sim(\sim B)$ C B $\surd$ (open) x (closed) $\{B, \sim B\}$	6

46) Show that a set $S = \{\sim(A \vee B), (B \rightarrow C), (A \vee C)\}$ is consistent.

The set S can be shown to be consistent.

Description	Formula	Line number
Tableau root	$\sim(A \vee B) \wedge (B \rightarrow C) \wedge (A \vee C)$	1
Rule 1 (1)	$\sim(A \vee B)$	2
	$(B \rightarrow C)$	3
	$(A \vee C)$	4
Rule 4(2)	$\sim A$	
	$\sim B$	
Rule 3(4)	A C	
	$x\{A, \sim A\}$	
Rule 6(3)	$\sim B$ C	
	$\surd$ $\surd$	

Scanned with CamScanner

47) Explain the steps for conversion of a formula to CNF form.

Any formula in propositional logic can be easily transformed into its equivalent CNF representation by using the equivalance laws

- Eliminate double negation signs by using

$$\sim(\sim A) \cong A$$

- Use De Morgan's Laws to push $\sim$ (negation) immediately before the atomic formula.

$$\sim(A \wedge B) \cong \sim A \vee \sim B$$

$$\sim(A \vee B) \cong \sim A \wedge \sim B$$

- Use distributive law to get CNF

$$A \vee (B \wedge C) \cong (A \vee B) \wedge (A \vee C)$$

- Eliminate $\rightarrow$ and $\leftrightarrow$ by using the following Equivalence laws

$$A \rightarrow B \cong \sim A \vee B$$

$$A \leftrightarrow B \cong (A \rightarrow B) \wedge (B \rightarrow A)$$

48) Convert the formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ into its equivalent CNF representation?

The given formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ can be transformed into its CNF representation in the following

$$(\sim A \rightarrow B) \wedge (C \wedge \sim A) \cong (\sim(\sim A) \vee B) \wedge (C \wedge \sim A)$$

Scanned with CamScanner

$\cong (A \vee B) \wedge (C \wedge \sim A)$ $\{as \sim A \rightarrow B \cong (\sim A) \vee B\}$

$\cong (A \vee B) \wedge C \wedge \sim A$ $\{as \sim(\sim A) \cong A\}$

The set of clauses in this case is written as $\{(A \vee B), C, \sim A\}$

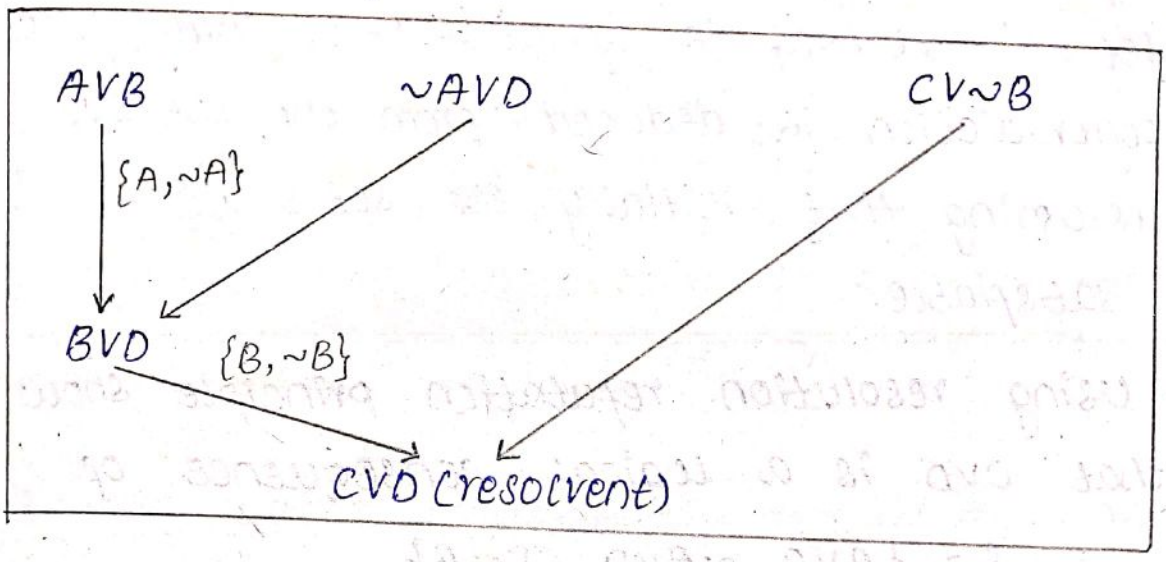
49) How to resolve clauses? Explan it?

Two clauses can be resolved by eliminating complementary pair of literals, if any, from both new clause is constructed by disjunction literals in both the clauses. Therefore the two clauses C_1 and C_2 contain a complementary pair of levels $\{L, \sim L\}$, then these clauses may be resolved together by deleting L from C_1 and $\sim L$ from C_2 and constructing a new clause by a disjunction of the remaining literals in C_1 and C_2. The new clause is called resolvent of C_1 and C_2. The clauses C_1 and C_2 are called parent clauses of the resolved clause. The resolution tree is a inverted binary tree with the last node being a resolvent, which is generated as a part of the resolution process.

Scanned with CamScanner

50) Find resolvent of the clauses in the set $\{A \vee B, \sim A \vee D, C \vee \sim B\}$

The method of resolution is shown



We can clearly see that CVD is a resolvent of the set $\{A \vee B, \sim A \vee D, C \vee \sim B\}$

51) State the results of resolution process?

- If c is a resolvent of two clauses C_1 and C_2 then c is called a logical consequence of the set of the clauses $\{C_1, C_2\}$. This is known as resolution principle.
- If a contradiction is derived from a set S of clauses using resolution then S is said to be unsatisfiable. Derivation of contradiction for a set S by resolution method is called a resolution refutation of S.
- A clause c is said to be a logical consequence of S if c is derived from S.

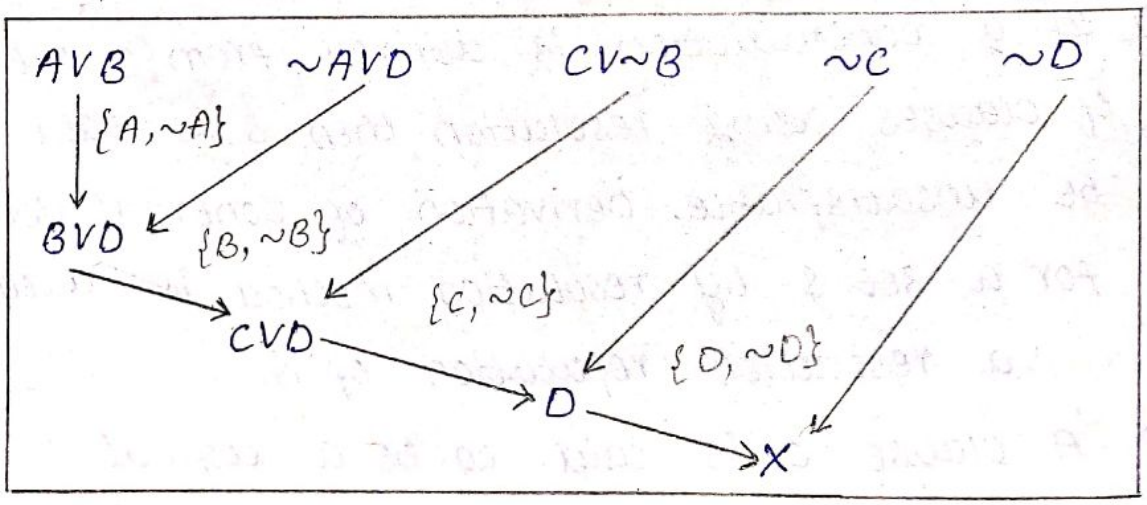
Scanned with CamScanner

⊙ Alternatively, using the resolution refutation concept, a clause c is defined to be a logical consequence of S if and only if the set $S' = S \cup \{\sim c\}$ is unsatisfiable, that is, a contradiction is deduced from the set S', assuming that initially the set S is satisfiable.

52) Using resolution refutation principle show that $C \vee D$ is a logical consequence of

$$S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$$

To prove the statement, first we will add negation of the logical consequence, that is $\sim(C \vee D) \cong \sim C \wedge \sim D$ to the set S to get $S' = \{A \vee B, \sim A \vee D, C \vee \sim B, \sim C, \sim D\}$ Now, we can show that S' is unsatisfiable by deriving contradiction using the resolution principle.



We can conclude that $(C \vee D)$ is logical consequence.

Scanned with CamScanner

53) Enumerate the logical predicate calculus?

predicate calculus has three more logical notions in addition to propositional calculus.

- <u>Term</u>:) A term is defined as either a variable, or constant. or n- place function. A function is defined as a mapping that maps n terms to a single term. An n- place function is written as $f(t_1, \ldots, t_n)$ where $t_1, \ldots t_n$ are terms.
- <u>predicate</u>:) A predicate is defined as a relation that maps n terms to a truth ~~tatte~~ value. {true, false}.
- <u>Quantifiers</u>:) Quantifiers are used with variables; there are two types of quantifiers namely universal quantifiers, $\forall$ (for all), and Existential quantifiers, $\exists$ (there Exists).

54) Write notes on Well formed formula?

<u>Well formed formula</u>:)

In predicate calculus, Well formed formula, Where p is a predicate symbol and $t_1, \ldots t_n$ are the terms.

- If α and β are well formed formulae, then $\sim(\alpha)$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are well formed formulae.

Scanned with CamScanner

- If α is a well formed formula and x is a free variable in α, then $(\forall x)\alpha$ and $(\exists x)\alpha$ are both well formed formulae. Here, α is in scope of quantifier $\forall$ or $\exists$. Scope of variable x is defined as the part of an expression where all occurrences of x have the same value.
- Well formed formulae may be generated by applying the rules described above a finite number of times.

55) First order predicate calculus is a formal language? Explain?

If the quantification in predicate formula is only on simple variables and not on predicates or functions then it is called first order predicate calculus. On the other hand if the quantification is over first order predicates and functions, then it becomes second order predicate calculus. For example $\forall P(P(x)) \leftrightarrow P(y))$ is a second order predicate statement, whereas $\forall x \forall y (P(x) \leftrightarrow P(y))$ is a first order predicate statement.

The first order predicate calculus is a formal language in which a wide variety of statements are expressed.

Scanned with CamScanner

the formulae in predicate calculus are formed using rules similar to those used in propositional calculus. When inference rules are added to first order predicate calculus, it becomes first order predicate logic.

56) Evaluate the truth value of an FOL formula $\alpha : (\forall X)(\exists Y) P(X, Y)$ under the following interpretation I: ⊙ $D = \{1, 2\}$

⊙ $P(1, 1) = F$, $P(1, 2) = T$, $P(2, 1) = T$, $P(2, 2) = F$

Let us denote true by T and false by F. For $X = 1$, then $\exists\, 2 \in D$ such that $P(1, 2) = T$ and for $X = 2$, then $\exists\, 1 \in D$ such that $P(2, 1) = T$, Hence α is true under interpretation I.

57) Write about prenex normal form?

Prenex Normal form:-

A closed formula α in FOL is said to be in PNF if and only if α is represented as $(Q_1 X_1)(Q_2 X_2) \ldots (Q_n X_n) M$, where Q_K are quantifiers ($\forall$ or $\exists$), X_K are variables, for $1 \le K \le n$, while M is a formula free from quantifiers. The list of quantifiers $[(Q_1 X_1) \ldots (Q_n X_n)]$ is called prefix and M is called the matrix of a formula α. Here, M is assumed to be represented in CNF notation.

Scanned with CamScanner

58) Show that $(\exists x)P(x) \wedge (\exists x)q(x) \neq (\exists x)(P(x) \wedge q(x))$ under the following interpret-ation $D = \{1, 2\}$

$P(1) = T$, $P(2) = F$

$q(1) = F$, $q(2) = T$.

We can easily see that the formula $(\exists x)P(x) \wedge (\exists x)q(x)$ is true under the interpretation I, whereas $(\exists x)(P(x) \wedge q(x))$ is false under I. Therefore, we can conclude that $(\exists x)\alpha(x) \wedge (\exists x)\beta(x) \neq (\exists x)(\alpha(x) \wedge \beta(x))$

Similarly

$$(\forall x)\alpha(x) \vee (\forall x)\beta(x) \neq (\forall x)(\alpha(x) \vee \beta(x))$$

can be shown using the same interpretation.

59) Explain the steps of skolemization procedu-re?

- Scan prefix from left to right till we obtain the first existential quantifier
  - If Q_1 is the first existential quantifier then choose a new constant $c \notin \{$set of constants in $M\}$. Replace all occurrence of x_1 appearing in matrix M by c and delete $(Q_1 x_1)$ from the prefix to obtain new prefix and matrix
  - If Q_r is the first existential quantifier

Scanned with CamScanner

and $Q_1 \cdots Q_{r-1}$ are universal quantifiers appearing before Q_r then choose a new $(r-1)$ place function symbol $f \notin$ {set of functions appearing in M}. Replace all occurrence of x_r in M by $f(x_1, \ldots, x_{r-1})$ and remove (Q_r, x_r) from prefix.

- Repeat the process till all existential quantifiers are removed from M.

60) Write the concept of clauses in FOL?

A clause is defined as a closed formula written in the form $(L_1 \vee \cdots \vee L_m)$, where each L_i is a literal and all variables occurring in $L_1, \ldots, L_m$ are universally quantified. The scope of variables appearing in a clause is the clause itself.

Let $S = \{C_1, \ldots, C_m\}$ be a set of clauses that represents a standard form of a given formula α.

- A formula α is said to be unsatisfiable if and only if its corresponding set S is unsatisfiable.
- S is said to be unsatisfiable if and only if there $\nexists$ no interpretation that satisfies all the clauses of S simultaneously.

Scanned with CamScanner

- S is said to be satisfiable if and only if each clause is satisfiable i.e., ∃ ~~are~~ an interpretation that satisfies all the clauses of S simultaneously.
- Alternatively, an interpretation I is said to model S if and only if I models each clause of S.

6) Elaborately describe resolution refutation method in FOL?

Resolution refutation method in FOL is used to test unsatisfiability of a set of clauses corresponding to the predicate formula. It is an extension of the resolution refutation methods described earlier for propositional logic. A deduction of a contradiction from a set S of clauses is called a resolution refutation of S. The resolution principle basically checks whether a contradiction is contained in or derived from S.

Resolution for the clauses containing no variables is simple and is similar to that used in propositional logic, but it becomes complicated when clauses contain variables. In such cases before resolution two complementary literals are resolved after proper substitutions so that both the literals

Scanned with CamScanner

have same arguments.

62) Find the resolvent of two clauses CL_1 and CL_2. Where p, q and r are predicate symbols, x is a variable and f is a unary function.

$$CL_1 = p(x) \vee q(x)$$

$$CL_2 = \sim p(f(x)) \vee r(x)$$

If we substitute f(a) for x in CL_1 and a for x in CL_2, where a is a new constant from the domain then we obtain

$$CL_3 = p(f(a)) \vee q(f(a))$$

$$CL_4 = \sim p(f(a)) \vee r(a)$$

We observe that clauses CL_3 and CL_4 have complementary literals as p(f(a)) and ~p(f(a)). The resolvent is generated by taking the disjunction of literals from parent clauses after eliminating pair of complementary literals therefore we get resolvent of CL_3 and CL_4 as

$$CL = q(f(a)) \vee r(a).$$

Here we notice that CL_3 and CL_4 do not have variables. These are called ground instances of CL_1 and CL_2. In general we substitute f(x) for x in CL_1, then we get another clause as follows:

$$CL_1' = p(f(x)) \vee q(f(x))$$

Scanned with CamScanner

Then, $CL' = \text{Resolvent}(CL_1', CL_2) = [q(f(x)) \vee r(x)]$

We notice that CL is an instance of CL', which is general form and can be obtained from CL' by substituting a for x.

63) Show that the formula $\alpha : (\forall x)(P(x) \wedge \sim [q(x) \rightarrow P(x)]$ is unsatisfiable?

To prove the above statement, we need to convert α into a set of clauses with the help of equivalence laws.

$$P(x) \wedge \sim [q(x) \rightarrow P(x)] \cong P(x) \wedge \sim [\sim q(x) \vee P(x)]$$
$$\cong P(x) \wedge \sim \sim q(x) \wedge \sim P(x)$$
$$\cong P(x) \wedge q(x) \wedge \sim P(x)$$

The set of clauses is written as $S = \{P(x), q(x), \sim P(x)\}$. Since there is a contradiction in S itself, S is unsatisfiable and consequently α is unsatisfiable.

64) Show that q(a) is a logical consequence of formulae α and β where

$\alpha : (\forall x)[P(x) \rightarrow q(x)]$

$\beta : P(a)$

We need to convert α and β into a set of clauses using equivalence laws. The set of clauses, S may be written as

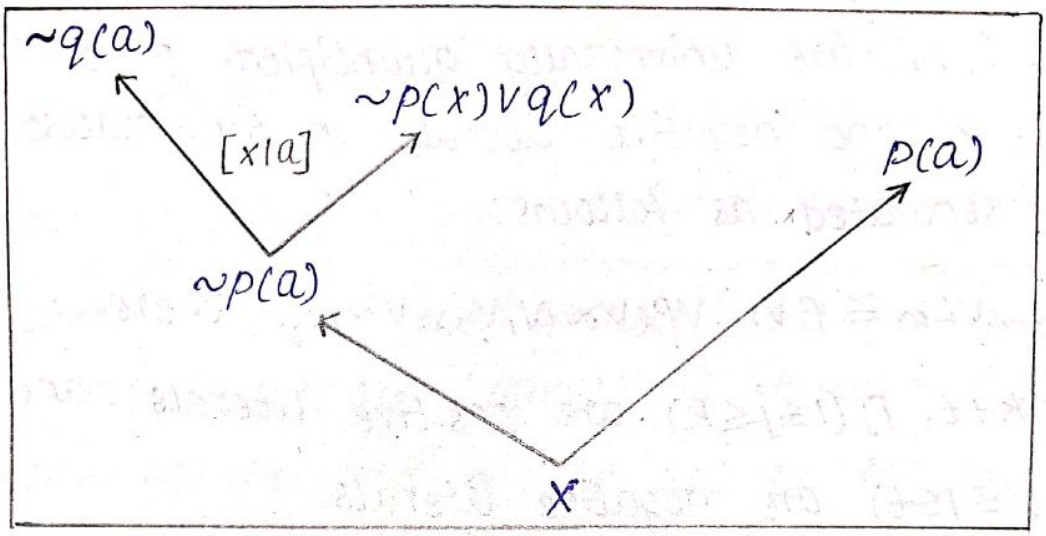
$$S = \{\sim P(x) \vee q(x), P(a)\}$$

Scanned with CamScanner

When we add the negation of goal, that is $\sim q(a)$. to S, we get the new set, S', where

$S' = \{\sim p(x) \vee q(x), p(a) \sim q(a)\}$

the resolution tree is



65) Write short notes on logic programming?

Logic programming was first introduc-ed by Kowalski in 1974 and colmerauer, et al. in 1973. it is based on FOL. In logic programming, a program is usually a collection of program clauses. A clause of FOL is represented in clausal notation in logic programming. clausal notation is written in the form $P_1, \ldots, P_K \leftarrow N_1, \ldots N_1$, where P_i $(1 \le i \le k)$ are positive literals and $N_j (1 \le j \le t)$ are negative literals. The goals or queries are solved, derived or proved from a logic program using resolution principle.

Scanned with CamScanner

66) convert a clause into clausal notation?

A clause in FOL denotes a closed formula of the form $L_1 \vee \ldots \vee L_m$, where each $L_k (1 \leq k \leq n)$ is a literal, which is free from quantifiers, and all the variables occurring in $L_1, \ldots, L_m$ are universally quantified. Now, the positive and negative literals in the clause can be separated as follows:

$L_1 \vee \ldots \vee L_m \cong P_1 \vee \ldots \vee P_k \vee \sim N_1 \vee \ldots \vee \sim N_t$ where $m = k+t$, $P_j (1 \leq j \leq k)$ are positive literals and $N_p (1 \leq j \leq t)$ are negative literals.

Further,

$(P_1 \vee \ldots \vee P_k \vee \sim N_1 \vee \ldots \vee \sim N_t) \cong (P_1 \vee \ldots \vee P_k) \vee \sim (N_1 \wedge \ldots \wedge N_t)$

$\cong (N_1 \wedge \ldots \wedge N_t) \rightarrow (P_1 \vee \ldots \vee P_k)$, $\{$as $P \rightarrow Q \cong \sim P \vee Q\}$

⊙ The clausal notation is written in the following form:

$(P_1 \vee \ldots, \vee P_k) \leftarrow (N_1 \wedge \ldots \wedge N_t)$

(or)

$P_1, \ldots, P_k \leftarrow N_1, \ldots, N_t$

It must be noted that the interpretation of $P \leftarrow N$ is same as that $N \rightarrow P$. Here, the symbol is read as 'implied by' The literals denoted by P_j (where $i \leq j \leq k$)

Scanned with CamScanner

are positive and those denoted by N_i (where $1 \le i \le t$) are negative. In clausal notation, all variables are assumed to be universally quantified. Negative literals N_p ($1 \le i \le t$) are called antecedents, which positive literals P_j, ($1 \le j \le k$) are called consequents. Commas in antecedents and consequents in denote conjunction and disjunction respectively.

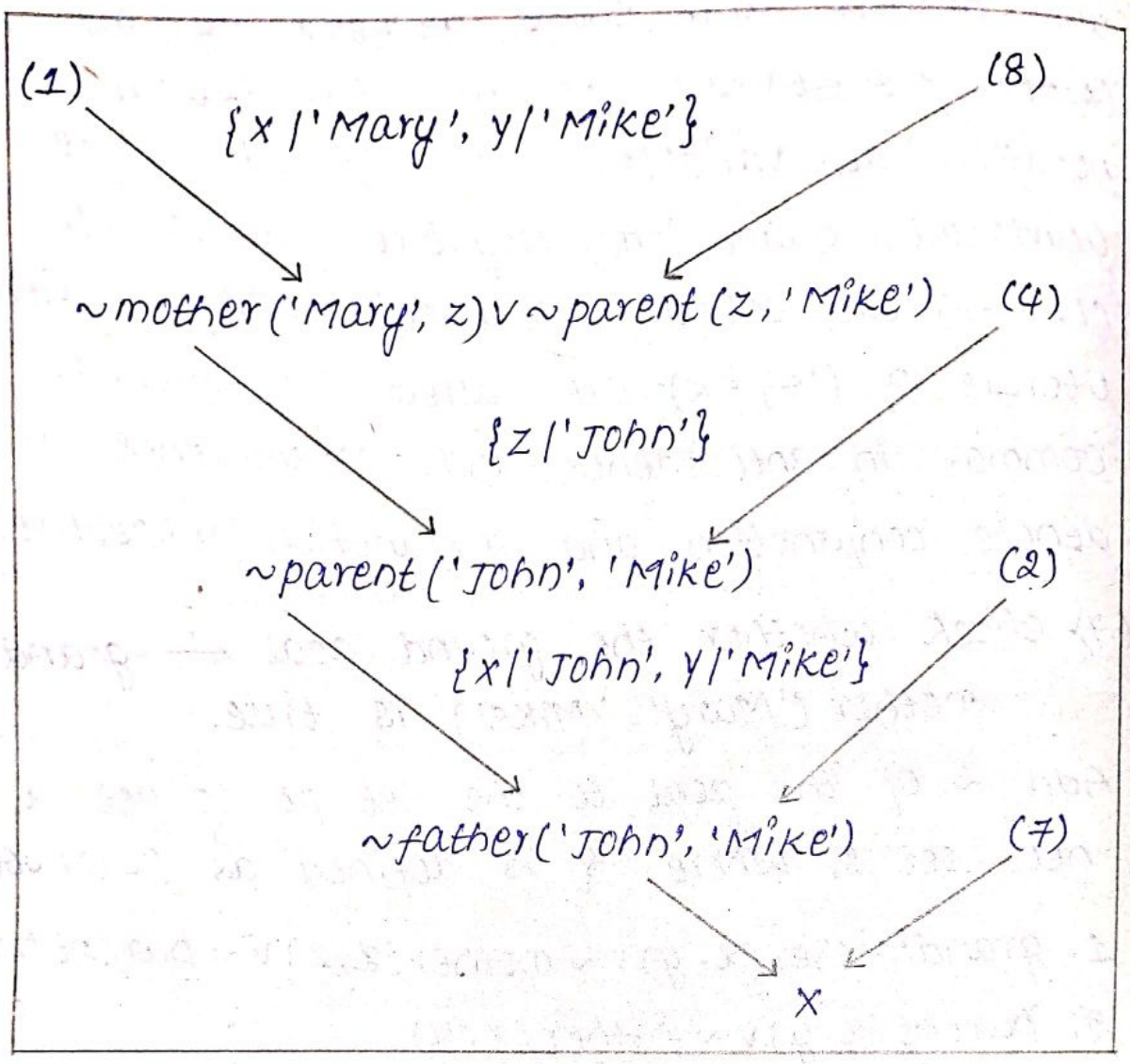
67) Check whether the ground goal ← grand mother ('Mary', 'Mike') is true.

Add ~ of the goal to the set PL to get a new set S, where S is defined as $PL \cup \{\sim G\}$

1. grandmother(x, y) V ~ mother(x, z) V ~ parent(z, y)
2. parent(x, y) V ~ father(x, y)
3. parent(x, y) V ~ mother(x, y)
4. mother ('Mary', 'John')
5. mother ('Tina', 'Kittu')
6. mother ('Kittu', 'Mita')
7. father ('John', 'Mike')
8. ~ grandmother ('Mary', 'Mike')

The resolution tree is generated. Since a contradiction is deduced, we can say that the goal is proved.

Scanned with CamScanner



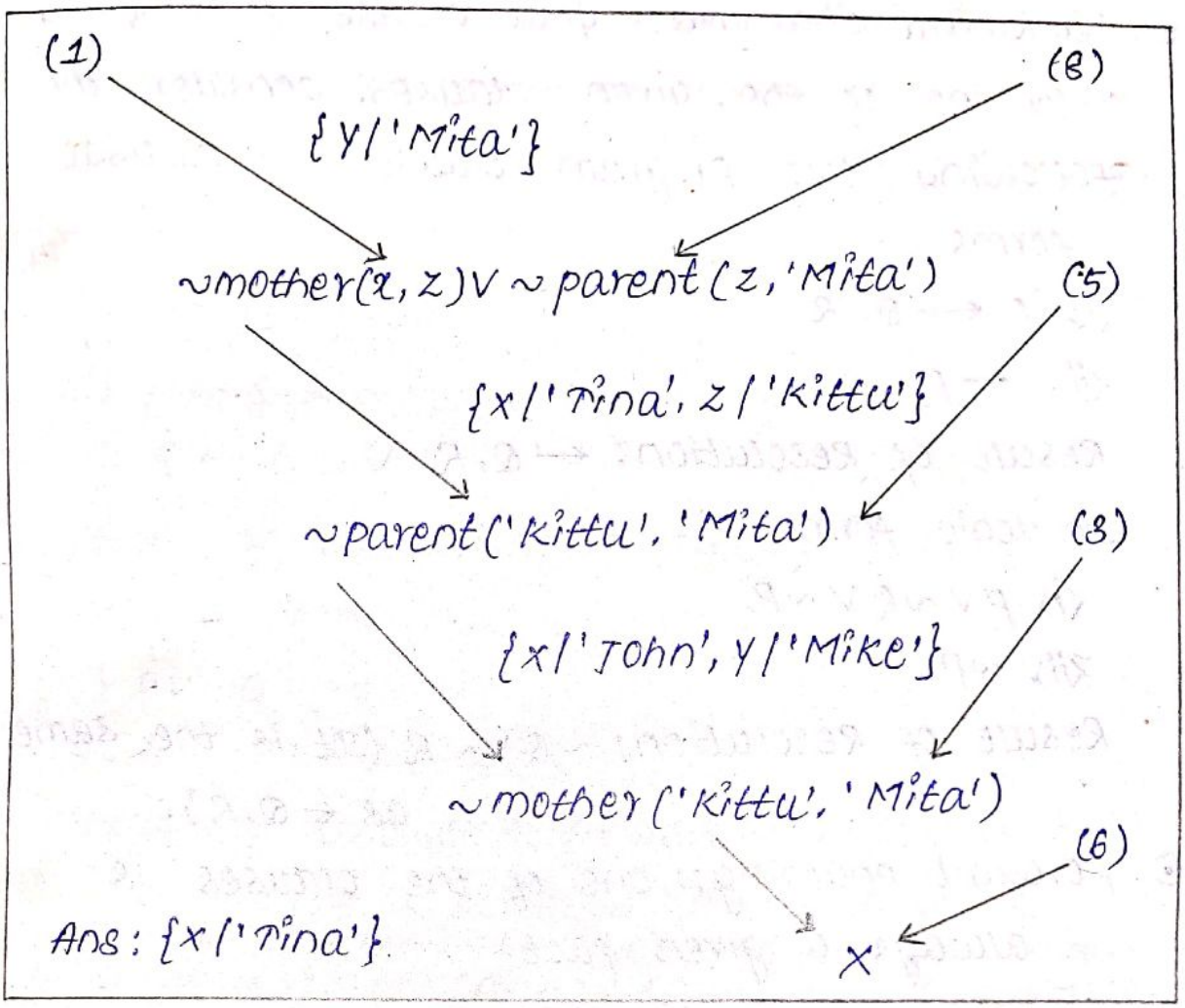
We can clearly see from the resolution tree that a contradiction is deduced, so we conclude the goal grandmother('Mary', 'Mike') is true.

68) Check whether the non ground goal ← grandmother(x, 'Mita') is true,

In the set of clauses, replace the last clause with ~grandmother(x, 'Mita'). In the process of resolution, a contradiction is deduced and variable x is bound to 'Tina'. Hence, we conclude that 'Tina' is the grandmother of 'Mita'.

Scanned with CamScanner

The resolution tree is:



69) Illustrate three types of resolution methods with Examples?

Resolution is a powerful inference technique which takes two clauses as input and produces a new clause as output. The output clause is called resolvent. If the resolvent obtained is a fact, then it is said that we have proved a fact and if the resolvent consists of an empty clause. There are three important cases of resolution used in reasoning, namely, backward chaining, forward chaining.

Scanned with CamScanner

and rule collapsing.

- Backward chaining:> goal clause is resolved with one of the given clauses. consider the following two program clauses in clausal forms

  (i) $P \leftarrow Q, R$

  (ii) $\leftarrow P$

  Result of Resolution: $\leftarrow Q, R$

  In logic form

  (i) $P \vee \sim Q \vee \sim R$

  (ii) $\sim P$

  Result of Resolution: $\sim Q \vee \sim R$ (It is the same as $\leftarrow Q, R$).

- Forward chaining:> one of the clauses is always a given fact.

  In clausal form

  (i) $P \leftarrow Q, R, S$

  (ii) R

  Result of Resolution: $P \leftarrow Q, S$

  In logic form

  (i) $P \vee \sim Q \vee \sim R \vee \sim S$

  (ii) R.

  Result of Resolution: $P \vee \sim Q \vee \sim S$ (It is same as $P \leftarrow Q, S$).

Scanned with CamScanner

⊙ Rule collapsing:> consider the following two program clauses. The head of clause <ii> is present in the body of clause <i>. Rule <i> is reduced by replacing R in its body by T, U, V.

In clausal form

<i> $P \leftarrow Q, R, S$

<ii> $R \leftarrow T, U, V$

Result of Resolution: $P \leftarrow Q, T, U, V, S$.

In logic form

<i> $P \vee \sim Q \vee \sim R \vee \sim S$

<ii> $R \vee \sim T \vee \sim U \vee \sim V$

Result of Resolution: $P \vee \sim Q \vee \sim T \vee \sim U \vee \sim V \vee \sim S$

(It is same as $P \leftarrow Q, T, U, V, S$)

45) Show that $\alpha : (A \wedge B) \wedge (B \rightarrow \sim A)$ is unsatisfiable using the tableau method.

It can be proven that $\alpha : (A \wedge B) \wedge (B \rightarrow \sim A)$ is unsatisfiable.

The semantic tableau for $\alpha : (A \wedge B) \wedge (B \rightarrow \sim A)$ has been drawn.

P.T.O

Scanned with CamScanner

Description	Formula	Line number
Tableau root	$(A \wedge B) \wedge (B \rightarrow \sim A)$	1
Rule 1 (1)	$A \wedge B$	2
	$B \rightarrow \sim A$	3
Rule 1 (2)	A	4
	B	5
Rule 6(3)	$\sim B$ $\sim A$	
	$\times\{B, \sim B\}$ $\times\{A, \sim A\}$	

UNIT II

KNOWLEDGE REPRESENTATION

70) Enumerate knowledge representation approaches?

AI programs use structures known as knowledge structures to represent objects, facts, relationships and procedures. The main function of these knowledge structures is to provide expertise and information so that a program can operate in an intelligent way. Knowledge structures are usually composed of both traditional and complex structures

Scanned with CamScanner

such as semantic network, frames, scripts, conceptual dependency structures and so on. A knowledge base is a special type of database that holds knowledge of the domain. knowledge bases share some of the functions of the database systems such as storing, updating and retrieving information. however they are much powerful and more structured

71) With an Example Explain knowledge Representation in a table?

Example:-

A fact John is a male whose age is 38 years and who is a graduate with a salary of Rs 20000 may be easily represented as one row in a table with attributes {Name, Age, sex, Qualification, and salary}

Name	Age	sex	Qualification	Salary
John	38	Male	Graduate	20000
Mike	25	Male	undergraduate	15000
Mary	30	female	PhD	25000
James	29	Male	Graduate	18000.

Scanned with CamScanner

72) Represent any knowledge as Logic?

Inferential capability can be achieved if knowledge is represented in the form of formal logic. For Example, knowledge regarding mortality such as All humans are mortal cannot be represented using relational approach; instead, it can be Easily represented in predicate logic as follows:

$$(\forall x)\, human(x) \leftarrow mortal(x)$$

If we have a fact John is human, then we can Easily infer that John is mortal.

73) Explain how to represent knowledge as semantic Network.

The basic idea applied behind using a semantic network is that the meaning of a concept is derived from its relationship with other concepts, and that the information is stored by interconnecting nodes with labelled arcs.

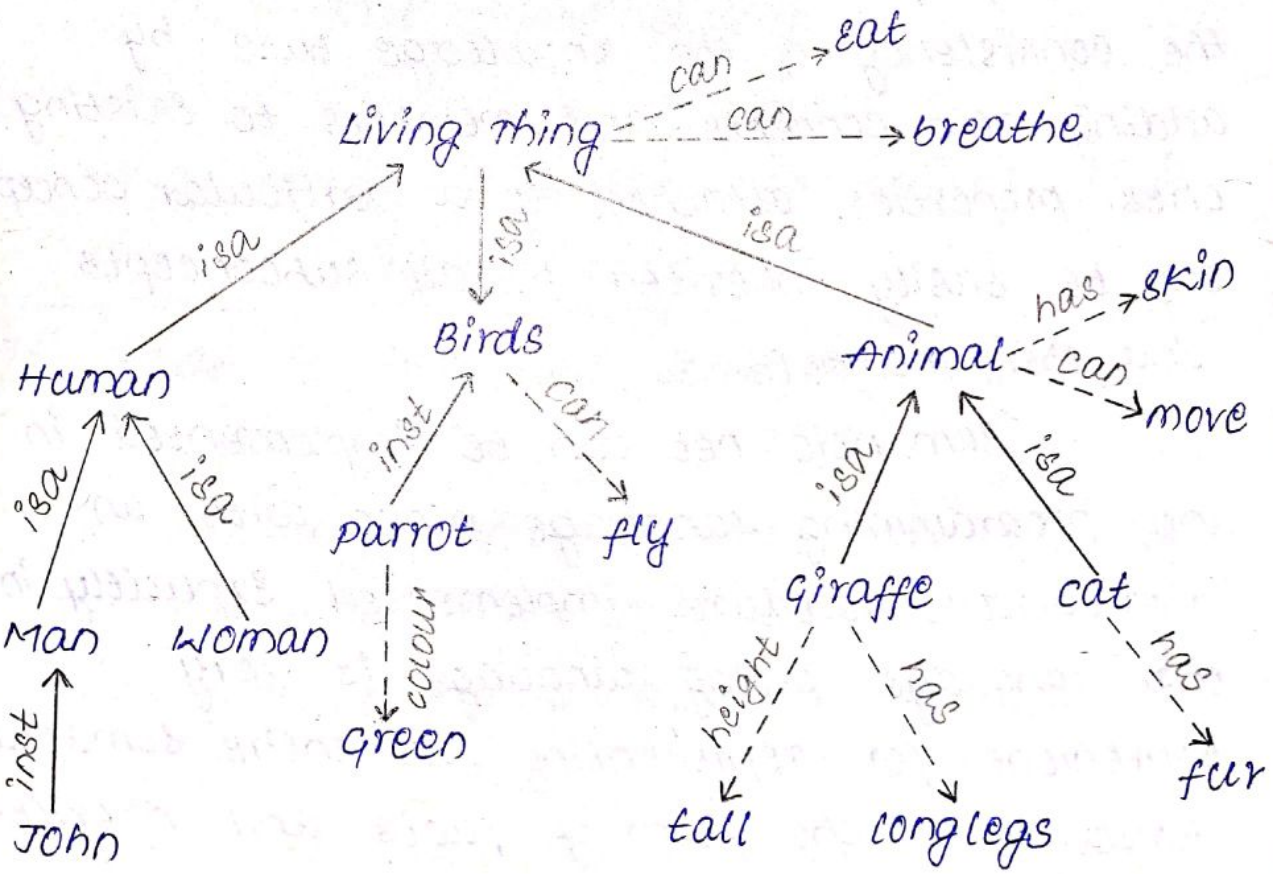
- Every human, animal and birds are living things who can breathe and Eat. All birds can fly. Every man and woman are human who have two legs. A cat has fur and is an animal. All animals have

Scanned with CamScanner

skin and can move. A giraffe is an animal and has long legs and is tall. A parrot is a bird and is green in colour. John is a man

We can represent such knowledge using a structure called a semantic network. It is conveniently represented in a graphical notation where nodes represents concept or objects and arcs represent relation between two concepts.

- isa: this relation connects two classes, where one concept is a kind or subclass of the other concept.
- inst: this relation relates specific members of class.



Scanned with CamScanner

74) How is inheritence implemented in semantic Net.

Hierarchial structure of knowledge representation allows knowledge to be stored at the highest possible level of abstraction which reduces the size of the knowledge base. Since semantic net is stored as hierarchial structure. the inheritence mechanism is inbuilt and facilitates inferencing of information associated with nodes in semantic net. It is natural tool for representing taxonomically structured information and ensures that all the members and sub-concepts of a concept share common properties. It also helps us to maintain the consistency of the knowledge base by adding new concepts and members to existing ones. properties attached to a particular concept can be easily inherited by all subconcepts and their members.

Semantic net can be implemented in any programming language along with an inheritance procedure implemented explicitly in that language. prolog language is very convenient for representing an entire semantic structure in the form of facts and inherite-
-nce rules.

Scanned with CamScanner

property Inheritance Algorithm

Input: object and property to be found from semantic Net;

output: returns yes, if the object has the desired property else returns false;

procedure:

- Find an object in the semantic net;
- Found = false;
- While [(object ≠ root) OR Found] DO

  {

  - If there is an attribute attached with an object then Found = true;

    else {object = isa(object, class) or object = inst (object, class)}

  };

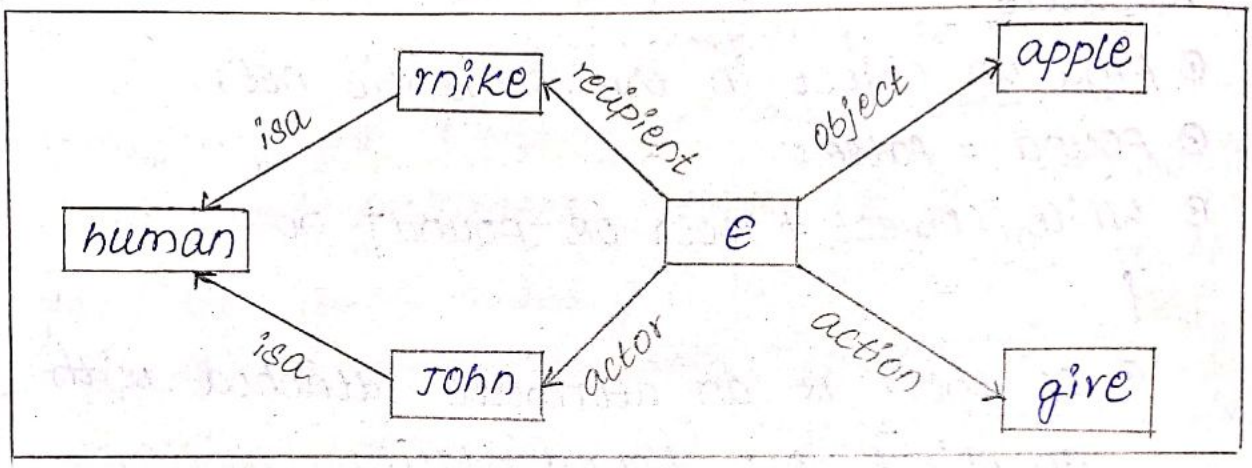
  - If Found = true then report 'yes' else report 'NO';

75) What is Extended semantic Networks for K.R?

Logic and semantic networks are two different formalisms that can be used for knowledge representations. Simple sematic network is represented as an directed graph whose nodes represent concepts or objects and

Scanned with CamScanner

arcs represent relationships between concepts or objects. It can only express collections of variable free assertions. The english sentences john gives an apple to mike and john and mike are human may be represented in semantic network.

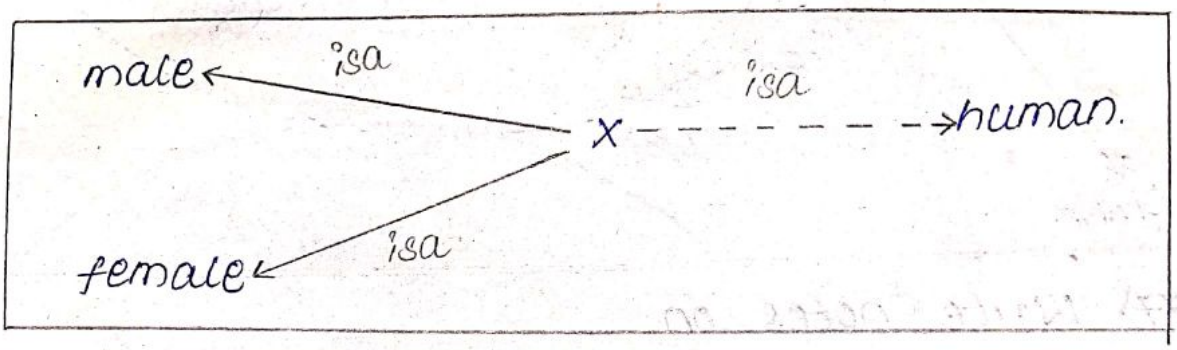


In this we cannot express clausal form of logic. To overcome this shortcoming, R Kowalski and his colleagues proposed an extended semantic network that combines the advantages of both logic and semantic networks. ESNet can be interpreted as a variant syntax for the clausal form form of logic, it has the same expressive power as that of predicate logic with well defined semantics, inference rules, and a procedural interpretation. It also incorporates the advantages of using binary relation as in semantic network rather than n-ary relations of logic.

Scanned with CamScanner

⊙ The clausal rule male (x), female (x) ← human(x) can be represented using binary representation as isa (x, male), isa (x, female) ← isa (x, human) and subsequently in ESNet as

ESNet Representation



Q6) Represent the following clauses in Esnet.

recipient (e, x) ← action (e, take), actor (e, x)

object (e, apple).

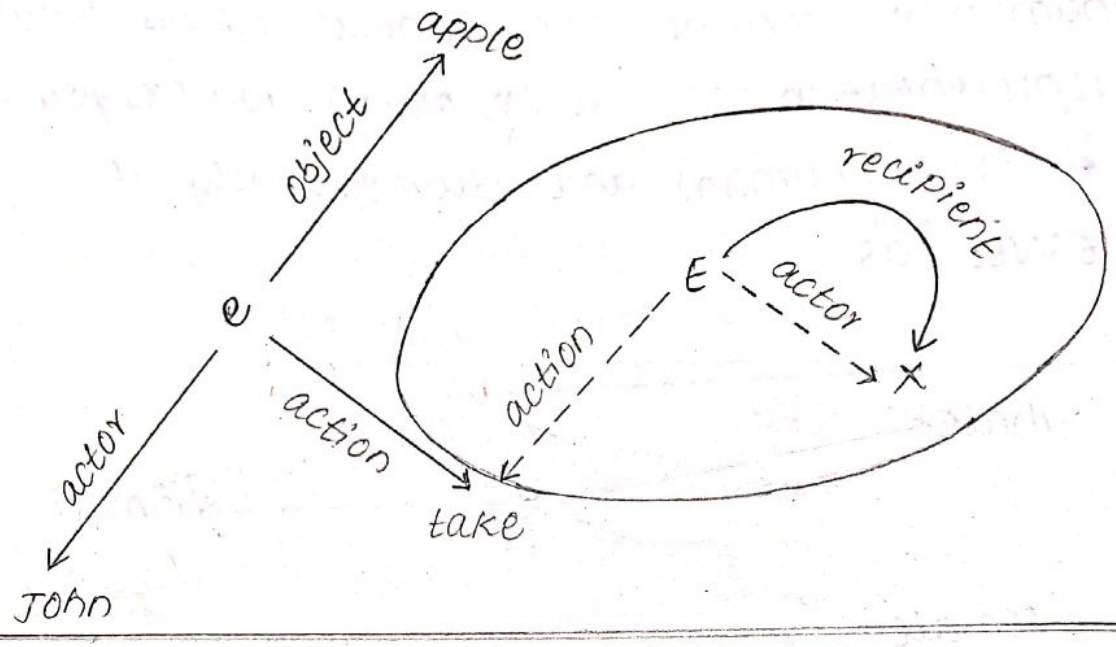
action (e, take).

action (e, John).

Here, e is a variable for some event and 'e' is an actual event. ESNet representation of the clauses is shown below,

the partition is shown to delimit the clause pictorially by enclosing the clausal rule in an ellipse.

Scanned with CamScanner



77) Write notes on

(a) forward Reasoning inference

(b) Backward Reasoning inference.

(a) <u>Forward Reasoning inference</u>:-

Given an esnet, apply the following reduction using modus ponen rule of logic {i.e., given (A ← B) and B, then conclude B}.

For eg:- consider the following set of clauses:

isa(X, human) ← isa(X, man)

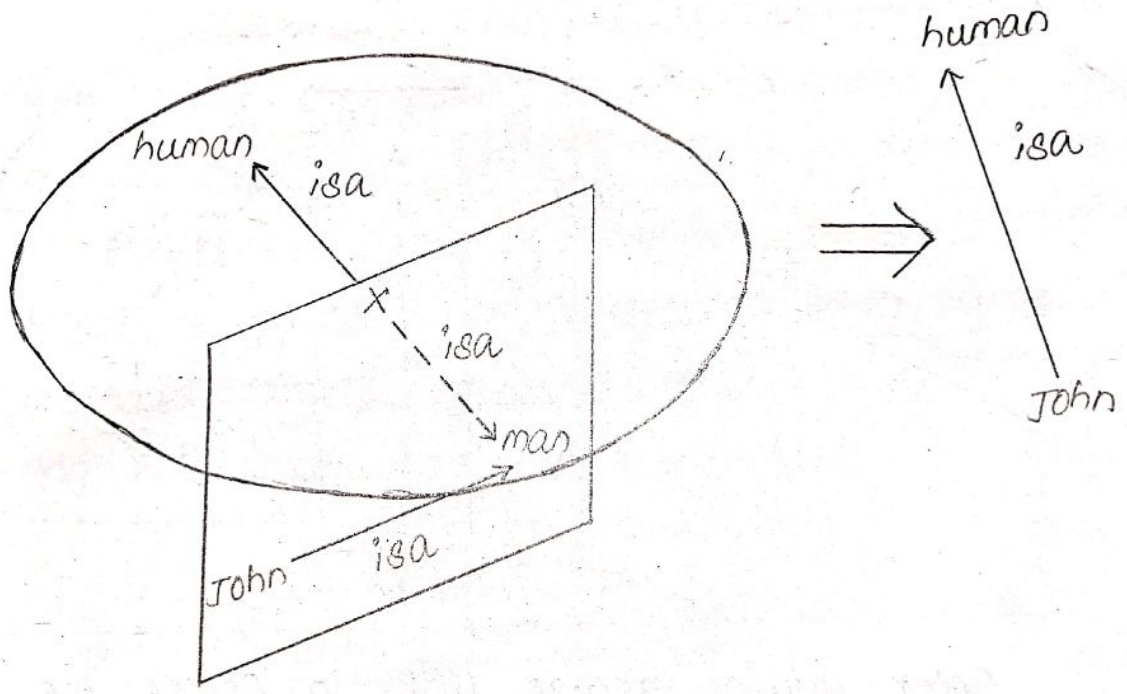
isa(john, man).

Using modus ponen rule of logic, we can easily derive that isa(john, human) holds true. Now, let us derive or inference isa(john, human) using esNet representation. The new insertion isa(john, human) is

Scanned with CamScanner

inferred by elimination of assertion and their denial links with the help of appropriate substitution. Here isa(x, human) is an assertion link and isa(x, man) is denial link.

Given set of clauses	Inferencing
isa(x, human) ← isa(x, man) isa(john, man)	isa(john, human)



Contradiction after x is bound to john

Here the contradiction is enclosed in a rectangular box.

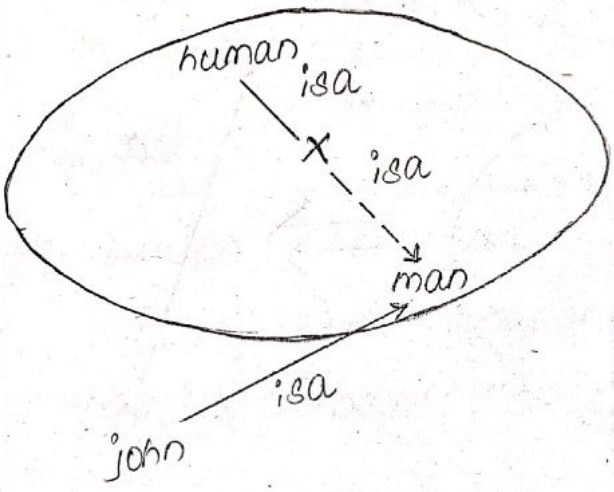
(b) <u>Backward Reasoning inference:</u>

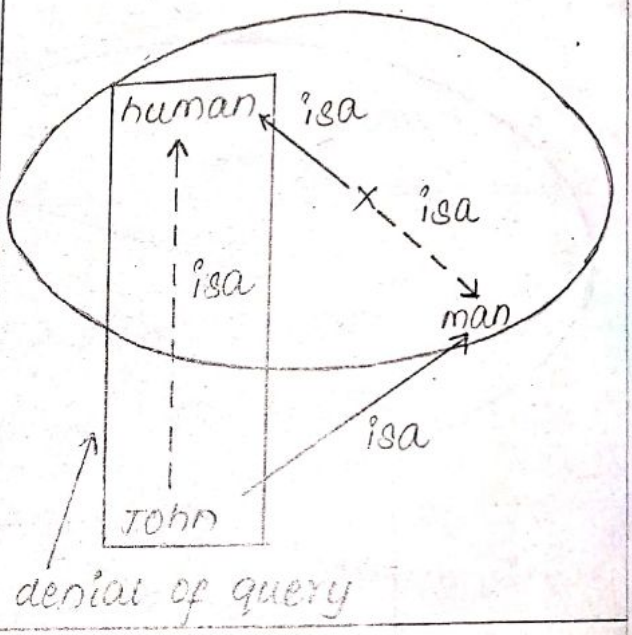
In backward inferencing mechanism, we can prove a conclusion or goal from a given esnet by adding the denial of the conclusion to the network and show that the resulting set of clauses in the network gives contradiction.

Scanned with CamScanner

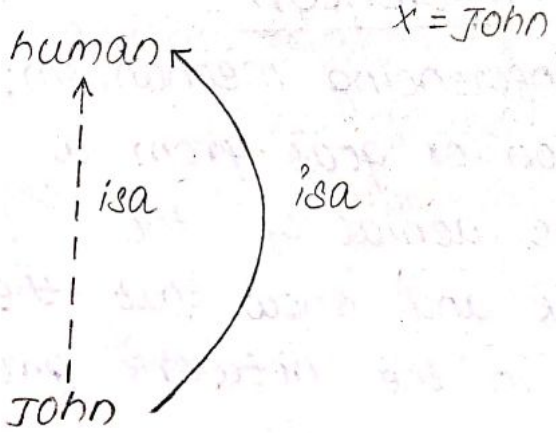
This is done by performing successive steps of resolution until an explicit contradiction is generated.

Given set of clauses	prove conclusion
isa(x, human) ← isa(x, man) isa(john, man)	Query: isa(john, human)





After adding denial link in ESNet, we get the reduction in ESNet, by elimination of assertion and their denial link with the help of appropriate substitution.



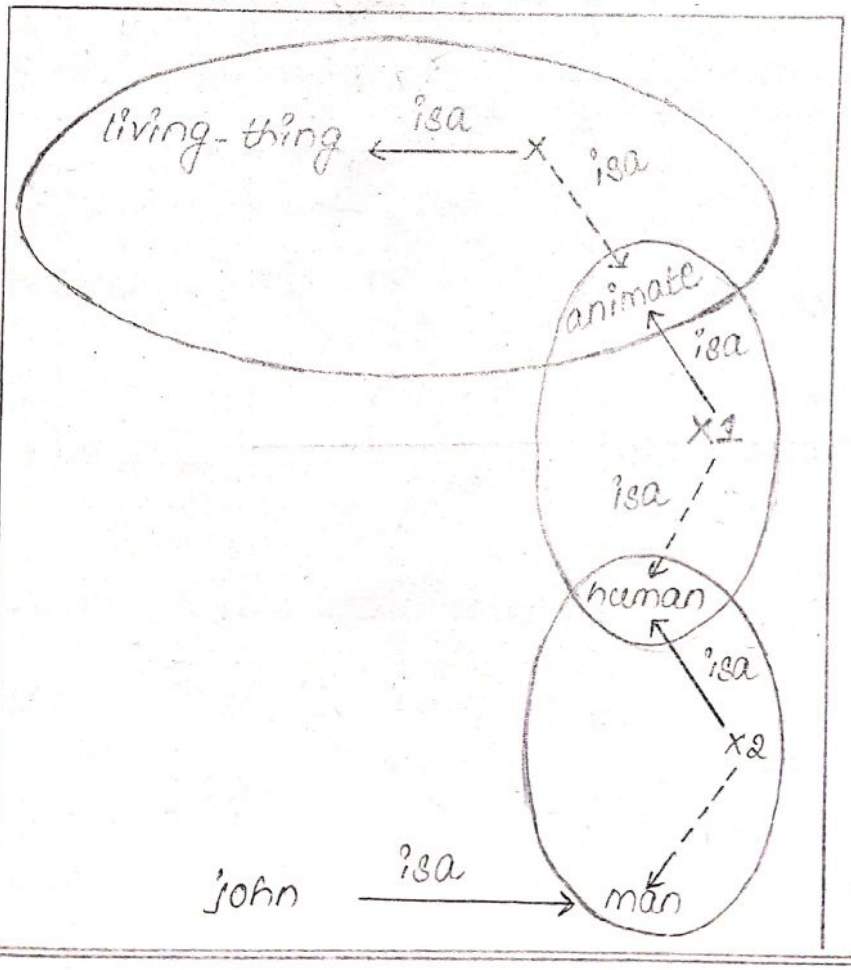
Contradiction or Empty network is generated. Hence, isa(John, human) is proved.

Scanned with CamScanner

7b) Represent the following Example in claused form and perform forward and backward reasoning.

isa(x, living-thing) ← isa(x, animate)
isa(x, animate) ← isa(x, human)
isa(x, human) ← isa(x, man)
isa(John, man)

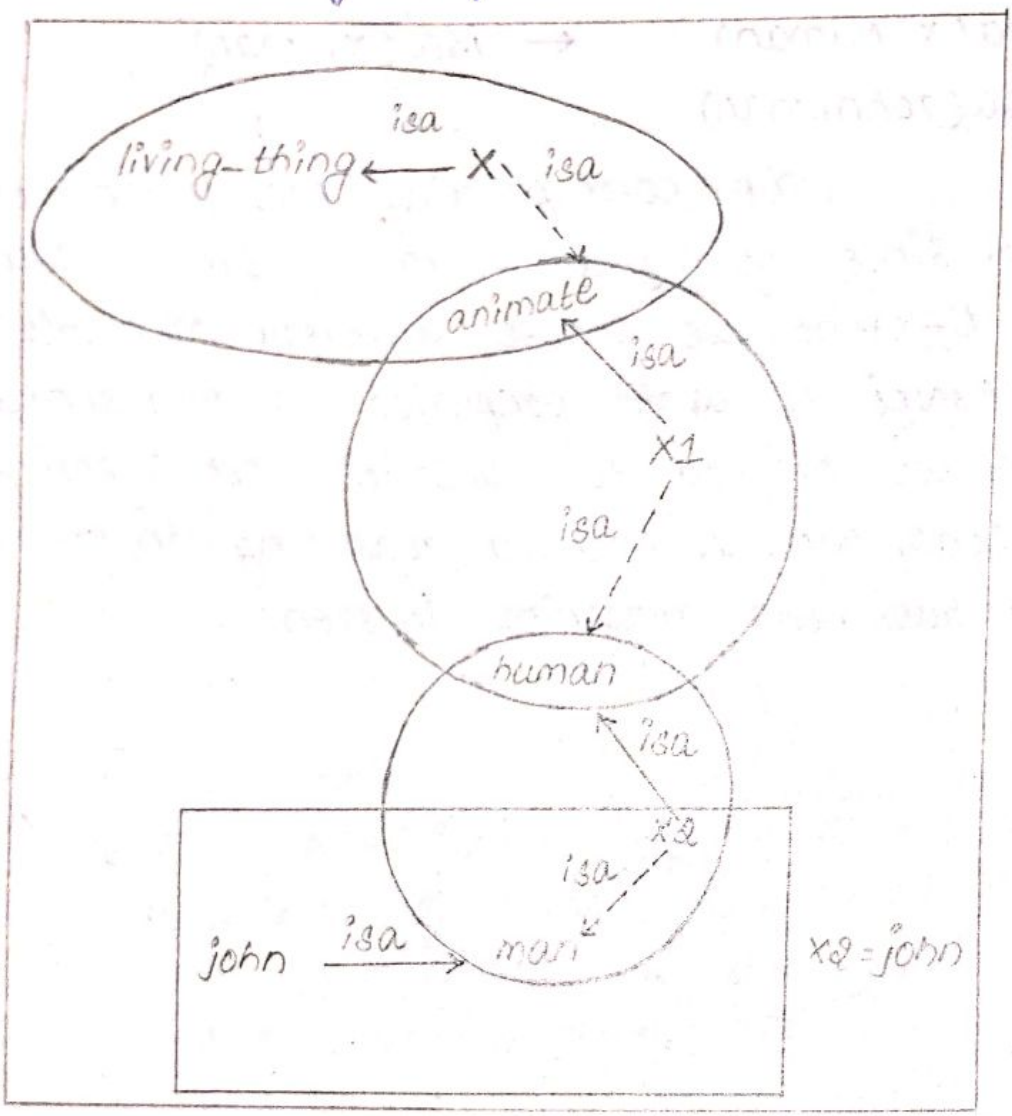
The corresponding ESNet representation. Since the variables in different clauses are different, we choose different variables in in ESNet to avoid confusion in representation Now, we proceed to illustrate both inferencing methods, namely, forward reasoning inference and backward reasoning inference.



Scanned with CamScanner

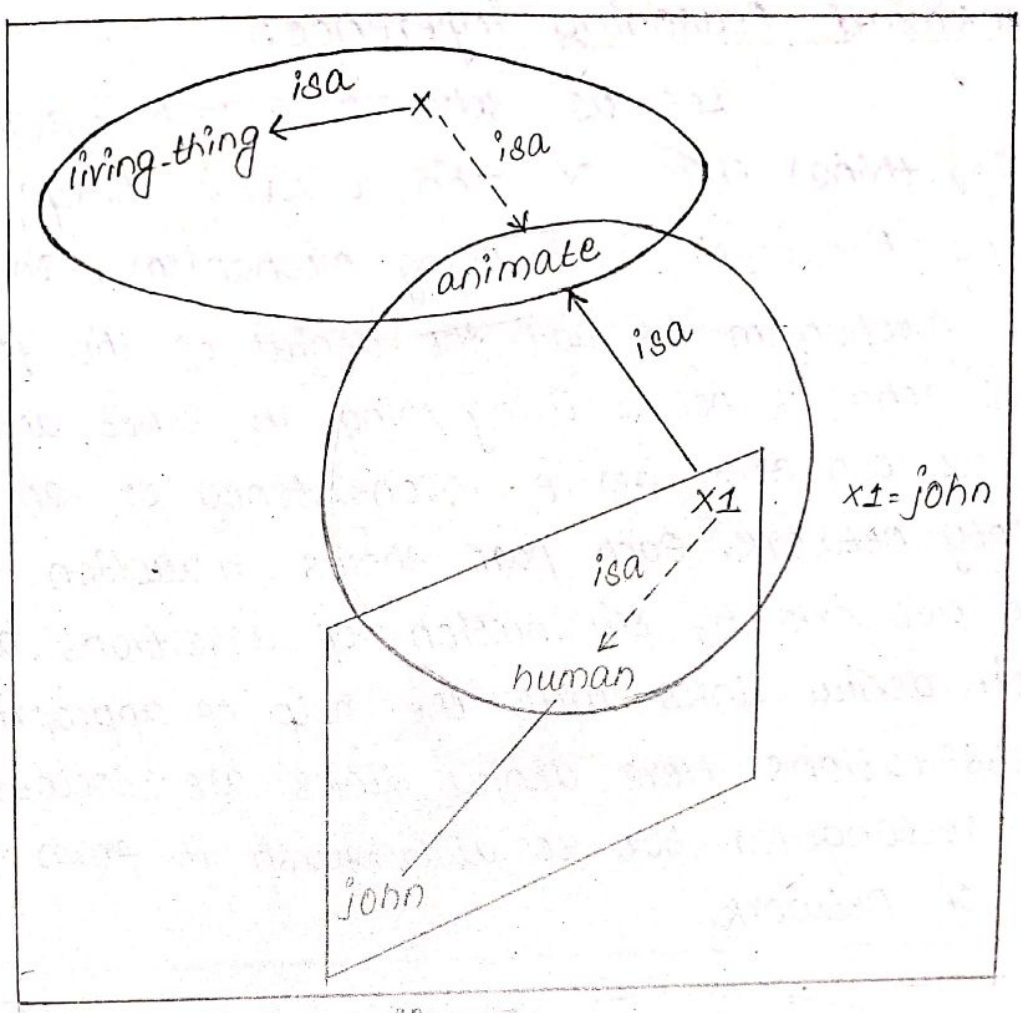
Forward Reasoning Inference:-

The new assertion john is human can be inferred or derived. the steps for inferring that john is animate. the assertion John is a living-thing can be inferred.

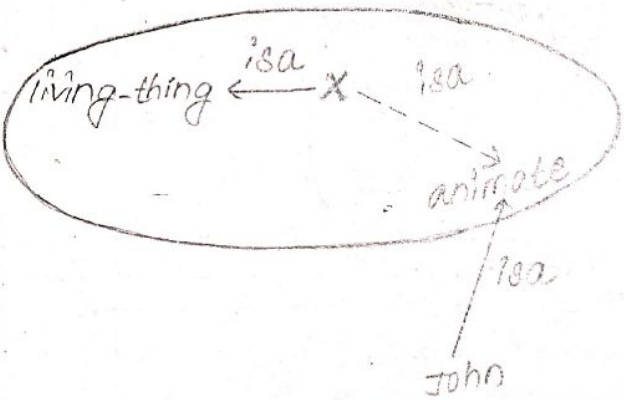


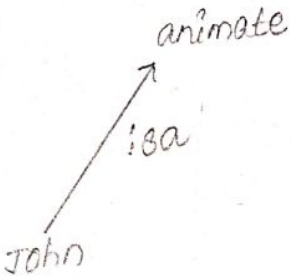
(i)

Scanned with CamScanner



(ii)





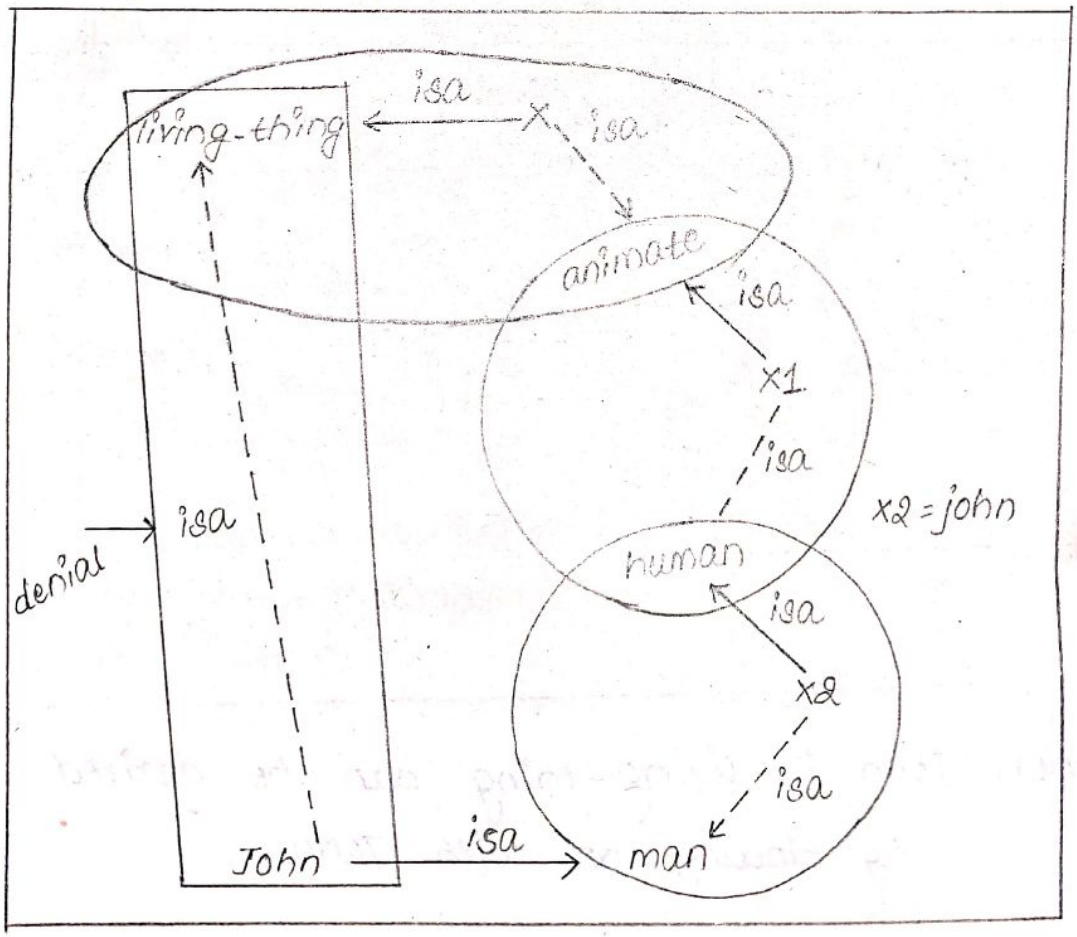
John is animate:
isa (John, animate) is inferred

Further, John is living-thing can be derived by binding x with John.

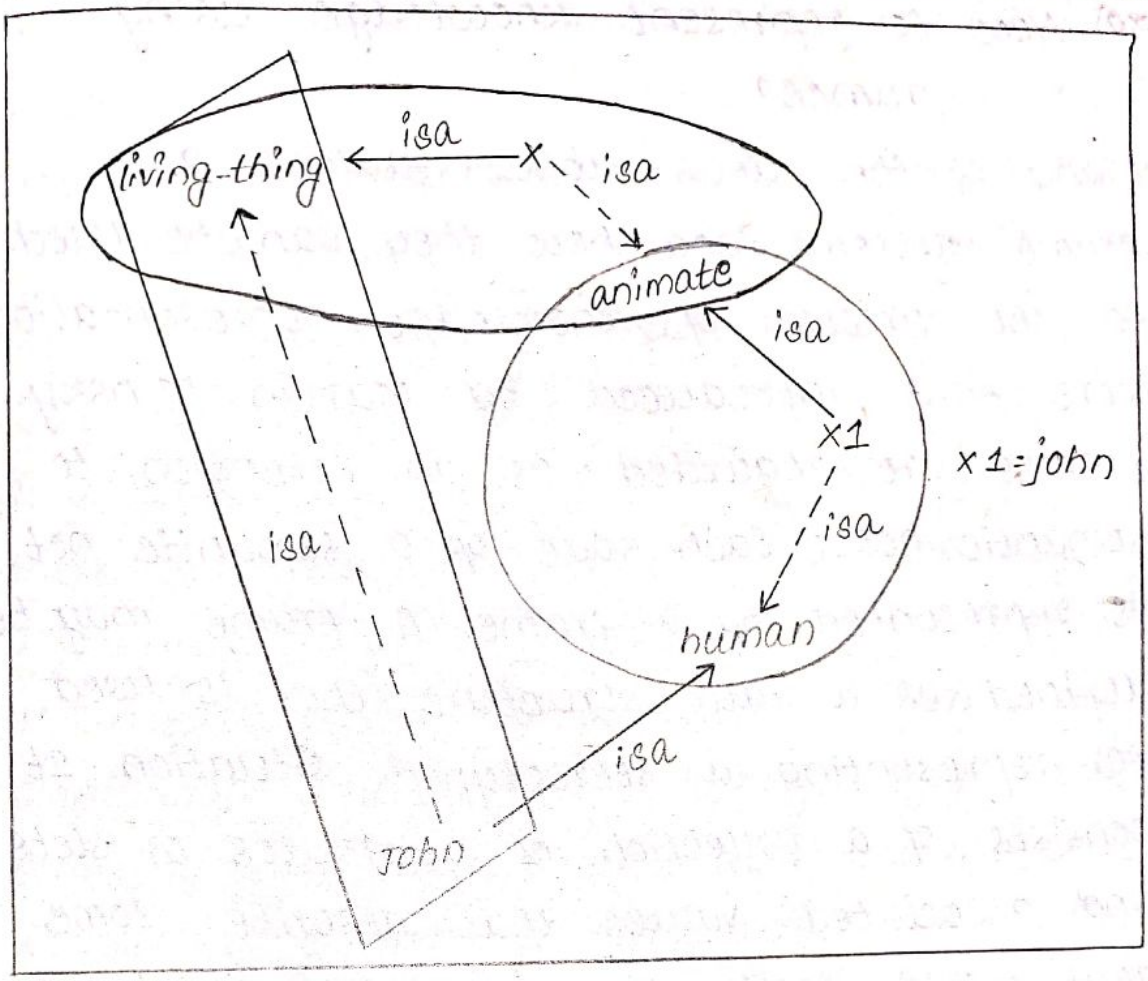
Scanned with CamScanner

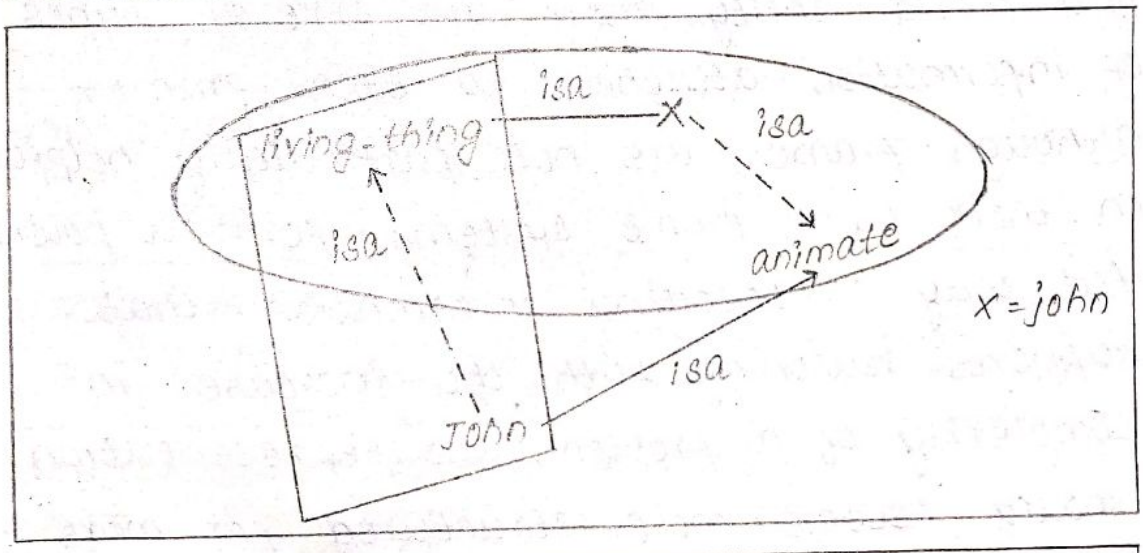
Backward Reasoning inference:-

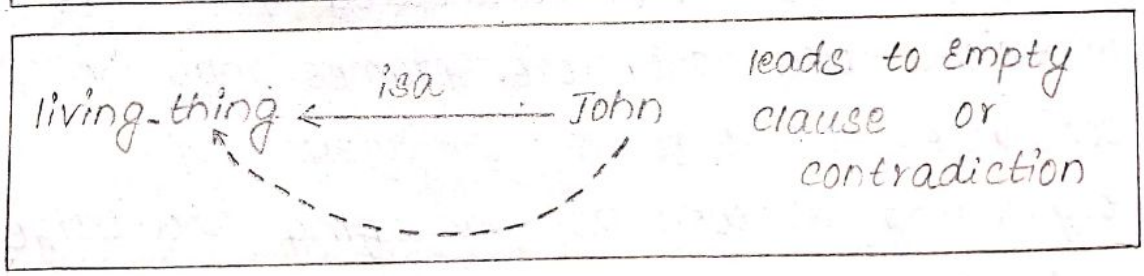
Let us solve a query isa(john, living-thing) (i.e., Is John a living thing?) using backward inferencing mechanism. In this mechanism, we add the denial of the fact that John is not a living-thing in esnet and see if we can ~~drive~~ derive inconsistency or an Empty network. Each part shows reduction in the network by elimination of assertions and their denial links with the help of appropriate substitutions. Here denial links are enclosed in rectangular box to distinguish it from rest of network.



Scanned with CamScanner







Hence, we can prove the query that john is an animate.

Scanned with CamScanner

79) How to represent knowledge using frames?

Many of the ideas about frames and frame systems and how they can be used for the process of knowledge representation were first introduced by Marvin Minsky. frames are regarded as an extension to semantic nets; each node of a semantic net is represented by a frame. A frame may be defined as a data structure that is used for representing a stereotyped situation. It consists of a collection of attributes or slots and associated values that describe some real-world entity. There are several types of information attached to each frame. Although frames are not particularly helpful on their own, frame systems form a powerful way of encoding information that supports reasoning. With the increase in complexity of a problem, the representation should become more structured for more beneficial use. Therefore, frames may be considered to represent the ways of organizing as well as packaging knowledge in a more structured form.

Scanned with CamScanner

frames are slightly similar to the concept of class of object oriented paradigm; class also contains attributes and methods. In frames, it consist of attributes or slots; slots are described with attribute-value pairs <slot-name, value>. slots are generally complex structures that have ~~effects~~ facets describing their properties. The value of a slot may be a primitive, such as a text string constant or an integer, or it may be another frame. so, slots may contain value, refer to other and some systems support procedural attachments as well. The general structure of a frame is given.

frame name
slot-filler
default values
constraints on values within the slots of a frame
pointers (links) to other frames
ako (a-kind-of or subclass)
inst (instance)
instantiation procedure
inheritance procedure
default inference procedure
triggers

Scanned with CamScanner

A frame may contain as many <slots-filler> pairs as required to describe an object. fillers are also known as facets. Each slot may contain one or more facets or fillers.

Lists of facets in a frame.

facet Name	Description
value	value of the slot
default	default value of the slot and it may be redefined by lower classes.
range	the range of integer or Enumera-ted values that a slot can have.
demons	procedural attachments such as if-needed, if-deleted, if-added.
other	may contain rules, other frames, semantic net, or any type of other information.

frames in a network of frames are connected using the links. other required information may be made available using slot-value pair concept.

- AKO
- Inst
- part-of

Scanned with CamScanner

80) Discuss inheritance in frames?

Inheritance is defined as a mechanism which is utilized for passing knowledge from one frame to other frames down through the taxonomy from general to specific frame. It is a good way of obtaining information that is not stored in the place we first looked. It leads to cognitive economy, where information is only stored in one place, while it can be retrieved from different parts of the network.

Attachment of Demons:)

Demons allow us to invoke rules within the frames and are considered to be a powerful style of programming. These can be attached with a slot along with other required facets. for eg., an attached demon if-needed may be used for monitoring the behaviour of the system. Another demon if-added may be used to validate data when the data is added in the value slot. for eg., if value of slot age is to be entered for a patient of child hospital, it will check whether it lies in the specified range.

Another use of attaching demons is that they allow dynamic information

Scanned with CamScanner

retrieval and storage. Here, one may get the value directly from the slot or may have to dynamically calculate the value of the required slot on the basis of other values.

UNIT - III

EXPERT SYSTEM AND APPLICATIONS

8) Explain the phases in building Expert systems?

The different interdependent and overlapping phases involved in building a Expert system may be categorized as follows

- **Identification phase:-**

In this phase, the knowledge Engineer determines important features of the problem with the help of the human domain Expert. The parameters that are determined in this place include the type and scope of the problem, the kind of resources required, and the goal and objective of the ES.

- **conceptualization phase:-**

In this phase, knowledge Engineer and domain Expert decide the

Scanned with CamScanner

concepts, relations and control mechanism needed to describe the problem solving method. At this stage, the issue of granularity is also addressed, which refers to the level of details required in the knowledge.

- **Formalization phase:-**

This phase involves expressing the key concepts and relations in some framework supported by ES building tools. formalized knowledge consists of data structures inference rules, control strategies and languages required for implementation.

- **Implementation phase:-**

During this phase, formalized knowledge is converted to a working computer program, initially called prototype of the whole system.

- **Testing phase:-**

This phase involves evaluating the performance and utility of prototype system and revising the system, if required. The domain expert evaluates the prototype system and provides feedback, which helps the knowledge engineer to revise it.

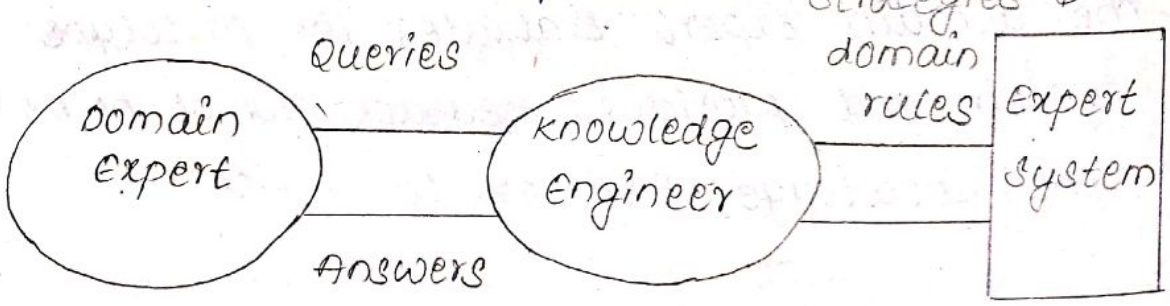
Scanned with CamScanner

Q2) Describe the concept of knowledge engineering?

The whole process of building an ES is often referred to as knowledge engineering. It typically involves a special form of interaction between ES builder, or the knowledge engineer, one or domain experts and potential users. Although there are different ways and methods of knowledge engineering, the basic approach remains the same. The tasks and responsibilities of a knowledge engineer involve the following.

- Ensuring that the computer has all the knowledge needed to solve a problem.
- choosing one or more forms to represent the required knowledge.
- Ensuring that the computer can use the knowledge efficiently by selecting some of the reasoning methods.

Interaction between knowledge engineer and the domain experts.



Scanned with CamScanner

The main role of the knowledge engineer begins only once the problem of some domain for developing an ES is decided. The job of the knowledge engineer involves close collaboration with the domain experts and the end users. The knowledge engineer may or may not have any knowledge of the application domain initially; the next step of process involves a more systematic interviewing of the expert. The knowledge engineer will then extract general rules from the discussion and interview held with expert and get them checked by the experts for correctness. The engineer then translates the knowledge into a computer usable language and designs an inference engine, which is a reasoning structure that uses the knowledge appropria-tely.

In order to develop the initial prototype, the knowledge engineer will have to take provisional decisions regarding appropriate knowledge representation, and inference methods.

Scanned with CamScanner

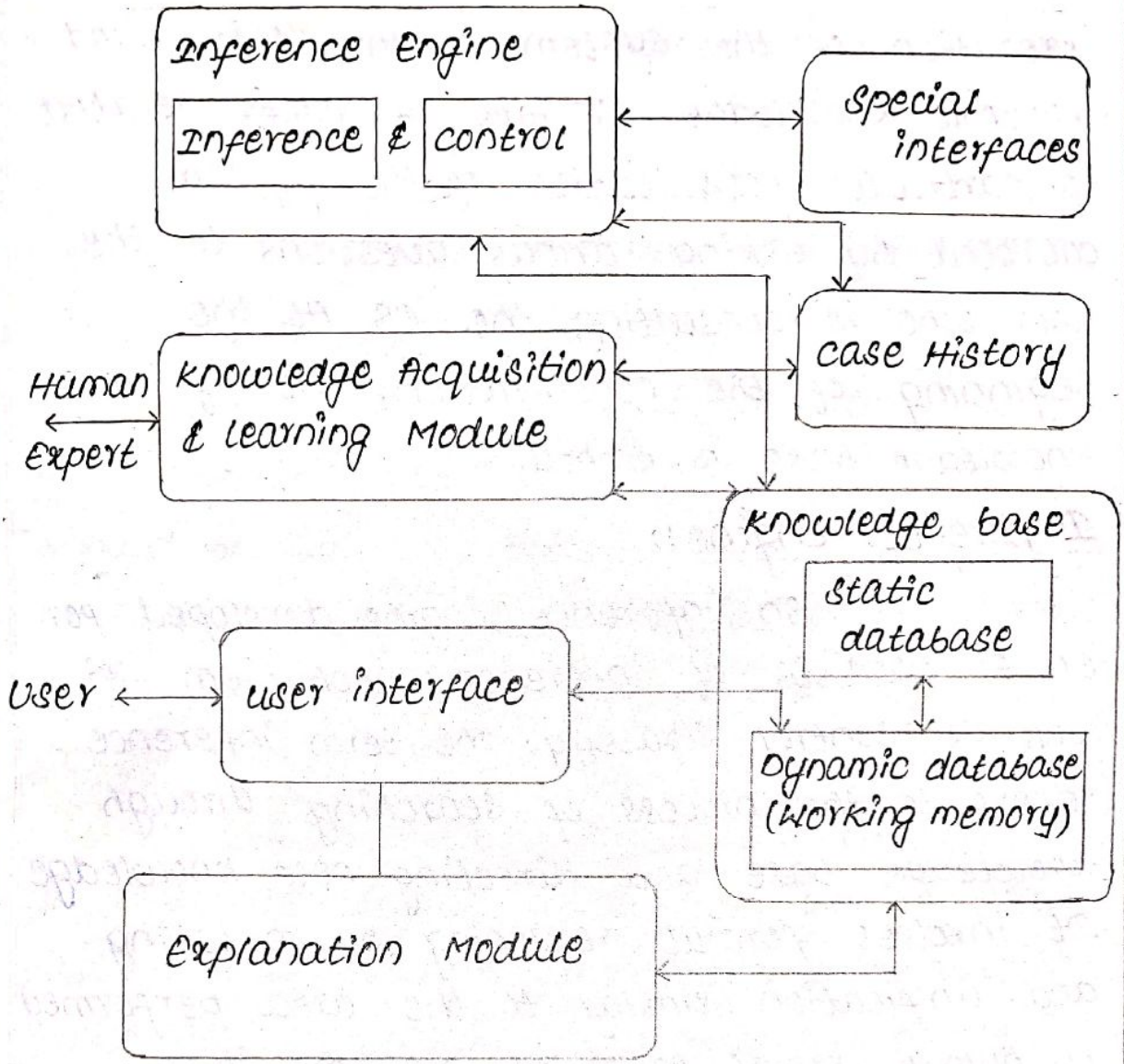
8) With a neat figure, explain the architecture of an expert system?

Expert system architecture:-

Expert system architecture may be effectively described with the help of diagram as given in below. Which contains important components of system. the user interacts with the system through a user interface which may use menus, natural language, or any other style of interaction then, an inference engine is used to reason with the expert knowledge as well as the data specific to the problem being solved. case specific data includes both data provided by the user and partial conclusions along with certainty measures based on this data. In a simple forward chaining rule based system, case specific data will be included in working memory. Generally, all expert systems possess an explanation subsystem. Which allows the program to explain its reasoning to user. Some systems also have a knowledge acquisition module that helps the expert or knowledge engineer to easily update.

Scanned with CamScanner

Expert system



~~8b) compare expert system versus traditional system?~~

Knowledge base :-

Knowledge base of an ES consists of knowledge regarding problem domain in the form of static and dynamic databases. Static knowledge consists of rules and facts, or any other form of knowledge edge representation

Scanned with CamScanner

which may be compiled as a part of the system and does not change during the execution of the system. on the other hand dynamic knowledge consists of facts related to particular consultation of the system collected by asking various questions to the user who is consulting the ES. At the beginning of the consultation, the dynamic knowledge base is empty.

<u>Inference Engine</u>:-

An inference Engine developed for an ES consists of inference mechanism as well as control strategy. The term inference refers to the process of searching through knowledge base and directing new knowledge It involves formal reasoning by matching and unification similar to the area performed by human expert to solve problems in a specific area of knowledge using modus power rule.

The other important inference mechanism that we are familiar with is backward chaining and the another one is forward chaining.

Scanned with CamScanner

Knowledge Acquition:-

Knowledge present in an ES may be obtained from many sources such textbooks, reports, case studies, Empirical data and domain experts. Interation between the knowledge Engineer and the domain Expert involves prolonged series of intense systematic interviews or using a quetionnaire. The knowledge Engineer working on a system should be able to Extract Expert systems methods, procedures, strategies and thumb rules for solving the problem at hand.

Case history:-

Case history stores the files created by inference Engine using the dynamic database and is used by the learning module to Enrich its knowledge base. Different cases with solutions are stored in case Base system and these cases are used for solving the problem using case Base Reasoning (CBR).

User interfaces:-

User interface of an ES allows user to communicate with the system in an iteractive matter and helps the system in creating working knowledge for the problem

Scanned with CamScanner

that has to be solved. knowledge may be entered using some editor or specialized designed user interface. The function of the user interface is to present questions and information to the user and supply the responses of the user to the interface. Engine.

Dialogue Module of a user Interface

system	Do you know AI
User	Yes

Explanation Module:-

Most of the ES has Explanation facilities that Enable users to query the System about why it asked some questions and how it reached some conclusion, these modules are called How and why at a solution to any particular problem. These questions are answered by referring to the system goals, the rules being used, and any Existing problem data.

Special interfaces:-

Special interfaces may be used in ES for performing specialized activities, such has handling uncertainty in knowledge

Scanned with CamScanner

these interfaces form a major area of expert system research which involves methods for reasoning with uncertain data and uncertain knowledge. A point to be kept in mind regarding knowledge is that it is generally incomplete and uncertain. To deal with uncertain knowledge, a confidence factor or a weight may be associated with a rule.

84) Compare Expert System versus traditional system?

The basic difference between an ES and a traditional system is that an ES manipulates knowledge whereas a traditional system manipulates data. The distinction between these systems lies in the manner in which the problem related Expertise is coded into them. In traditional application problem Expertise is encoded in program as well as in the form of data structures. On the other hand, in the ES approach, all problem related Expertise is encoded in data structures only and not in the programs. Traditional computer programs perform tasks using conventional decision making logic, which is often embedded as a part of code in the form of a basic algorithm containing little knowledge for

Scanned with CamScanner

solving that specific problem. Hence, if the knowledge changes the program has to be rebuilt. However in expert systems, small fragments of human experience are collected into a knowledge base which are used to reason through a problem. A different problem, within the domain of the knowledge base, can be solved using the same program without having to reprogram the system.

Another advantage of expert systems over traditional systems is that they allow the use of confidences or certainty factors. This is similar to human reasoning where one cannot always conclude things with 100% confidence

Further, conventional programs are designed to always produce correct answers whereas expert systems are designed to behave like human experts and may sometimes produce incorrect results. In the following subsections, we will discuss some important characteristics of expert systems their evaluations, advantages and disadvantages, and the languages that can be used for their development.

Scanned with CamScanner

86) Discuss the applications of Expert systems?

Applications may be categorized in the following:

<u>Diagnosis</u>:- The Expert systems belonging to this perform the task of inferring malfunctioning of system from observations such Expert systems use situation descriptions, behaviour, characteristics, or knowledge about component design to determine the probable cause of system malfunction. These systems may also be used for diagnosis of faulty modules in large signal switching networks and for finding faults in computer software system.

<u>planning and scheduling</u>:-

The Expert systems of this class help in designing actions and plans before actually solving a given problem. They analyze a set of one or more potentially complex and interacting goals in order to determine a set of actions that are needed to achieve these goals. They provide a detailed temporal ordering of these actions in the form of plans, taking into account personnel and other constraints.

Scanned with CamScanner

Example:-

- Airline scheduling of flights.
- Manufacturing job shop scheduling.
- Manufacturing process planning.

Design and Manufacturing:-

This is one of the most important areas for ES applications. Here, a solution to a problem is configured by a given set of objects under a set of constraints. configuration applications were pioneered by computer companies to ~~facilities~~ facilitate the manufacturing of semicustom minicomputers.

- gene colouring
- Modular home building
- Manufacturing etc.

prediction:-

The Expert systems of this class perform the task of interpreting and inferring situation description data of any domain such as geological data, cencus data, medical data etc

- Interpreting radar images to classify ships etc.,
- Interpreting intelligent sensors reports to perform situation assessment.

Scanned with CamScanner

Financial Decision Making:-

The financial services industry has been a prominent user of ES techniques. Such systems assist insurance companies to asses the risk presented by the customer and to determine a price for the insurance. Bankers use Expert systems for assitance in determining whether to grant loans to businesses and individuals. A typical applications are.

- foreign exchange trading
- formulating financial strategies
- giving advices etc.,

process monitoring and control:-

Expert systems belonging to this class analyze real-time data from physical devices with the goal by comparing observations to Expected outcomes, predicting trends and controlling for both optimality and failure correction.

- Steel making and oil refining industries
- nuclear reactor.
- patient monitoring systems in hospitals

and so on.,

Instruction:-

An ES can offer tutorials and instructions to students by incorporating a student's

Scanned with CamScanner

behaviour and learning capability models and can also evaluate a student's acquired skills.

Debugging:-

The systems of this class prescribe remedies for malfunctioning devices. They may suggest the type of maintenance needed to correct faulty equipments.

Knowledge publishing:-

This is a relatively new but a potentially important area now-a-days. The primary function of the ES in this field is to deliver knowledge to user that is in context of the user problem.

8) Enumerate the shells, tools of Expert systems?

ACQUIRE:-

It is primarily a knowledge acquisition system and an ES shell, which provides a complete development environment for the building and maintenance of knowledge based applications. ACQUIRE_SDK is a software development kit, which provides callable libraries for systems such as MS-DOS and SCO UNIX, Windows, Windows NT, Windows 95 and win 32. It is also used in custom

Scanned with CamScanner

controls for visual Basic.

ARITY:-

Expert Development package is an ES that was developed by Arity corporation, MA to perform the task of integrity / integrating rule based and frame based representations of knowledge with various types of certainty factors.

ART:-

It is called automated reasoning tool that is an ES shell. Which is based on LISP. It supports rule-based reasoning, hypothetical reasoning and case-based reasoning.

CLIPS:-

It is used to represent C Language Integrated production system. It is a public domain software tool that is used for building Expert systems. It provides a cohesive tool for handling a wide variety of knowledge and provides support to three different programming disciplines, namely, rule based, object oriented and procedural. CLIPS is probably the most widely used ES tool because it is fast, Efficient and free.

FLEX:-

Flex is a hybrid ES which is implemented in prolog. It was originally developed by LPA

Scanned with CamScanner

on a pc. It is available from most of the major vendors under license. It is basically a toolkit available across a wide range of different hardware platforms and offers frames, procedures and rules integrated within a logic programming environment. It supports interleaved forward and backward chaining, multiple inheritance, and procedural attachment and also possesses an automatic question and answer system.

Gensym's G2:-

It offers a graphical object-oriented environment for the creation of intelligent applications that are able to monitor, diagnose and control dynamic events in various environments. G2 features a structured natural language that is utilized for creating rules, models and procedures.

GURU:-

It is an ES development environment and offers a wide variety of information processing tools combined with knowledge based capabilities such as forward chaining, backward chaining, mixed chaining, multi-value variables, and fuzzy reasoning.

Scanned with CamScanner

HUGIN SYSTEM:-

It is a software package for construction of model based expert systems in domains characterized by inherent uncertainity. The Hugin system contains an easy to use probability based deduction system, applicable to complex networks with cause effect causal relations subject to uncertainity.

Knowledge Craft:-

It is an expert system development toolkit for scheduling, design and configuration applications.

K-vision:-

It is a knowledge acquisition and visualization tool. It runs a windows, DOS and UNIX work stations.

88) Write notes on probability theory?

The term probability is defined as a way of turning an opinion or an expectation into a number lying between 0 and 1. It basically reflects the likelihood of an event, or a change that a particular event occur. Assume a set S consisting of independent event representing all possible outcomes of a random experiment. Every non empty subset A of sample space S is called an event, the empty set is called an impossible event.

Scanned with CamScanner

Whereas 8 is called a random Event. Let us denote the probability of an Event A by P(A), it is defined as follows:

$$P(A) = \frac{\text{Number of outcomes favourable to A}}{\text{Total number of possible outcomes}}$$

89) Illustrate joint probability with Example?

joint probability is defined as the probability of occurence of two independent Events in conjunction. i.e., join probability refers to the probability of both Events occurring together. The joint probability of A and B is written as $P(A \cap B)$ or $P(A \text{ and } B)$ It may defined as given below

$$P(A \text{ and } B) = P(A) * P(B)$$

Example:-

Consider an Example of tossing of two fair coins separately. The probability of getting a head H on tossing the first coin is denoted by $P(A) = 0.5$, and the probability of getting a head on tossing the second coin is denoted by $P(B) = 0.5$. The probability of getting H on both the coins is called joint probability is represented as $P(A \text{ and } B)$

$$P(A \text{ and } B) = P(A) * P(B)$$

Scanned with CamScanner

$= 0.5 * 0.5$

$= 0.25$

Similarly, the probability of getting a head H on tossing one or both coins can be calculated. It is called union of probabilities P(A) and P(B), and is denoted by P(A∪B); or P(A or B).

- $P(A \text{ or } B) = P(A) + P(B) - P(A) * P(B)$

$= 0.5 + 0.5 - 0.25$

$= 0.75.$

90) Describe and prove conditional probability?

The concept of conditional probabity relates the probability of one event to the occurence of another. It is defined as probability of the occurence of an event H provided an event E is known to have occurred. It is denoted by P(H/E) and represented as follows.

$$P(H/E) = \frac{\text{No. of events favourable to H which are also favourable to E}}{\text{No. of events favourable to E}}$$

$$= \frac{P(H \text{ and } E)}{P(E)}$$

$= P(E) = 0$ is not valid for this rule.

Scanned with CamScanner

If Events H and E defined on a sample space S of a random Experiment are independent, then $P(H/E) = P(H)$

proof:) $P(H/E) = P(H \text{ and } E)/P(E)$ (def)

$= P(H) * P(E)/P(E)$

$= P(H)$

Similarly $P(E/H) = P(E)$.

9i) We are given a probability of any person chosen at random being literate as 0.40 and probability of any person chosen at random having age > 60 years as 0.005. find the probability of the fact that a person chosen at random of age > 60 years is literate.

The probability that a given person chosen at random is both literate and has age > 60 years is calculated as follows:

[P(x is literate and the age of x > 60] =

P(x is literate) * P(Age of x > 60)

$= 0.40 * 0.005$

$= 0.002$

Then, the probability that a person chosen at random age > 60 years is literate is calculated using conditional probability

$$P(x \text{ is literate}/(\text{Age of } x>60)] = \frac{[(x \text{ is literate \& age of } x>60)]}{[P(\text{Age of } x>60)]}$$

$= 0.002/0.005 = 0.4$

Scanned with CamScanner

92) State and prove Baye's theorem?

Statement:)

Baye's theorem relates the conditional probabilities of events which allows us to express $P(H/E)$ in terms of probabilities $P(E/H)$, $P(E)$ and $P(H)$. Here, H denotes hypothesis, while E represents a new piece of evidence.

$$P(H/E) = \frac{P(E/H) * P(H)}{P(E)}$$

Proof:)

We can express $P(H/E)$ using conditional probability.

$$P(H/E) = P(H \text{ and } E) / P(E)$$

$$\Rightarrow P(H/E) * P(E) = P(H \text{ and } E) \quad \text{— ⓐ}$$

We get,

Similarly, $P(E/H)$ can be expressed as

$$P(E/H) = P(E \text{ and } H) / P(H)$$

$$\Rightarrow P(E/H) * P(H) = P(E \text{ and } H) \quad \text{— ⓑ}$$

We get, from ⓐ and ⓑ

$$P(H/E) * P(E) = P(E/H) * P(H)$$

Hence, we obtain

$$P(H/E) = \frac{P(E/H) * P(H)}{P(E)}$$

This is Baye's theorem.

Scanned with CamScanner

93) Suppose we are given the probability of Mike has a cold as 0.25, the probability of Mike was observed sneezing when he had cold in the past as 0.9 and the probability of Mike was observed sneezing when he did not have cold as 0.20. Find the probability of Mike having a cold given that he sneezes.

Let us represent Mike has a cold as an hypothesis, H and Mike sneezes as an evidence E. we have to calculate $P(H/E)$.

probability of hypothesis Mike has a cold is 0.25 as $P(H) = 0.25$.

probability of the fact that Mike was observed sneezing when he had cold is 0.9

$P(\text{Mike was observed sneezing} / \text{Mike has cold}) = P(E/H) = 0.90$

probability of the fact that Mike was observed sneezing when he did not have cold is 0.20

~~P(Mike has cold / Mike~~

$P(\text{Mike was observed sneezing} / \text{Mike doesnot have cold}) = P(E/\sim H) = 0.20$

We have to find the probability of Mike having cold, when he was observed sneezing. i.e.,

$P(\text{Mike has cold} / \text{Mike was observed sneezing}) = P(H/E)$

$$P(H/E) = \frac{P(E/H) * P(H)}{P(E/H) * P(H) + P(E/\sim H) * P(\sim H)}$$

Scanned with CamScanner

Let us compute

$$P(E/H) * P(H) + P(E/\sim H) * P(\sim H) = (0.90)(0.25) + (0.20)(0.75)$$

$$= 0.375$$

We obtain

$$P(H/E) = [(0.9 * 0.25)] / 0.375$$

$$= 0.6$$

∴ We conclude that Mike's probability of having cold given that he sneezes is equal to 0.6

Similarly,

$$P(H/\sim E) = [P(\sim E/H) * P(H)] / P(\sim E)$$

$$= [(1-0.9) * 0.25] / (1-0.375)$$

$$= 0.025 / 0.625$$

$$= 0.04.$$

94) With an Example? Explain rule based systems using probabilities.

Let us develop a simple rule based system for the diagnosis of malfunctioning of some equipment, say a landline telephone, using the concept of probabilities. As the first step, various situations under which the telephone does not work properly are identified with the help of experts of this domain.

For Eg., we can think of following reasons for the malfunctioning of telephones.

- Faulty instrument.
- Problem in Exchange
- Broken cable.

Scanned with CamScanner

Let us estimate probabilities of these situations and write the corresponding rules.

Rules regarding faulty telephone instrument

Rule 1:)

If the telephone instrument is old and has been repaired faulty several times in the past then it is 40% sure that the fault lies with the instrument.

telephone-not-working (0.4) = ask (tele-history)

Here, the ask predicate will ask questions to the user who wants diagnosis regarding whether the instrument is old, has been repaired several times in the past and checks if response is positive or negative.

Rule 2:)

If the instrument has fallen on the ground and broken, then it is 80% sure that the fault lies within the instrument

telephone-not-working (0.8) :) ask (telephone-broken)

Rule 3:)

If there are children in the house who play with the keypad of the telephone, with some probability, then it is 80% sure that the instrument is faulty because of excessive and unusual usage.

telephone-not-working (P) :) ask (children-present, P1)
ask (children-keypad, P2),
and-combination[(P1, P2, 0.8], P]

Scanned with CamScanner

This rule says that when both the conditions hold with some probabilities, then it is 80% sure that the instrument is faulty. finally, to get the cumulative probability for telephone not working because of the fault of instrument we need to combine probabilities of all rules using or-combination.

```
instrument-faulty(P) :- findall(X, telephone-not-
                        working(X), L),
                        or-combination(L, P).
```

Rules for Diagnosis of Instrument fault :-

The rule for diagnosis based on instrument fault which we assume to be 70% sure.

```
diagnosis('Instrument is faulty', P) :-
    instrument-faulty(P1), and-combination([P1, 0.7],
                                           P).
```

Rules regarding faults due to other reasons

Let us consider other kinds of telephone diagnosis such as problems with telephone Exchange or cable connection.

problem in Exchange :-

```
Exchange-probability(0.09) :- ask(Exchange-problem)
Exchange-probability(0.5) :- ask(connecting-switch-
                                 problem).
overall-Exchang-probability(P) :- findall(X, Exchange-
                                  probability(X), L), or-combination
                                  (L, P).
```

Scanned with CamScanner

Due to broken cable:-

cable-broken-probability(0.98) :- ask(cable-old),
ask(if-storm-recently)

cable-broken-probability(0.3) :- ask(recent-
furniture-rearranging)

overall-cable-probability(P) :- findall(x, cable broken.
probability(x), L), or-combination(L, P).

Diagnosis :- cable fault

diagnosis('cable-problem', P) :- overall-cable-
probability(P1),
and-combination([P1, 0.8], P).

Exchange problem:-

diagnosis('Exchange-problem', P) :-
overall-Exchange-probability (P1),
and-combination([P1, 0.8], P).

95) Mention the advantages and disadvantages of Bayesian Method:-

Advantages:-

- Bayesian method is based on a strong theoritical foundation in probability theory it is currently the most advanced of all certainity reasoning methods.
- This method has well defined semantics for decision making.

Scanned with CamScanner

Disadvantages:

- The system using Bayesian approach needs quite a large amount of probability data to construct a knowledge base.

96) Formally define Bayesian Network?

The bayesian belief network may be formally defined as an acyclic directed graph where the nodes of the graph represent evidence or hypotheses and an arc connecting two nodes represents dependence between them.

Example:

if $\exists$ an arc from node x to another node y, then x is called a parent of y and y is called a child of x. The set of parent nodes of a node x_1 is represented by parent_nodes(x)

97) Consider the following table of conditional probabilities:

P(A)	P(B)
0.3	0.6

A	P(C)
T	0.4
F	0.2

A	B	P(D)
T	T	0.7
T	F	0.4
F	T	0.2
F	F	0.01

Scanned with CamScanner

We require only 8 probabilities values to represent the Bayesian belief network. given in contrast to 16 values required in general for four variables {A, B, C, D} in case of joint distribution probability.

As we already know that joint prasability for four variables can be computed using the following formula.

$$P(A, B, C, D) = P(D/A, B, C) * P(C/A; B) * P(B/A) * P(A)$$

However, this can be simplified using Bayesian belief network as some of these terms will not be required. C is not dependent on B; D is not dependent on C. Hence, P(C/A, B) is reduced to P(C/A) and P(D/A, B, C) is reduced to P(D/A, B). Further, A and B are independent, therefore, P(B/A) is reduced to P(B). with these reductions, we obtain the joint probability as

$$P(A, B, C, D) = P(D/A, B) * P(C/A) * P(B) * P(A)$$

$$= 0.7 * 0.4 * 0.6 * 0.3$$

$$= 0.0504.$$

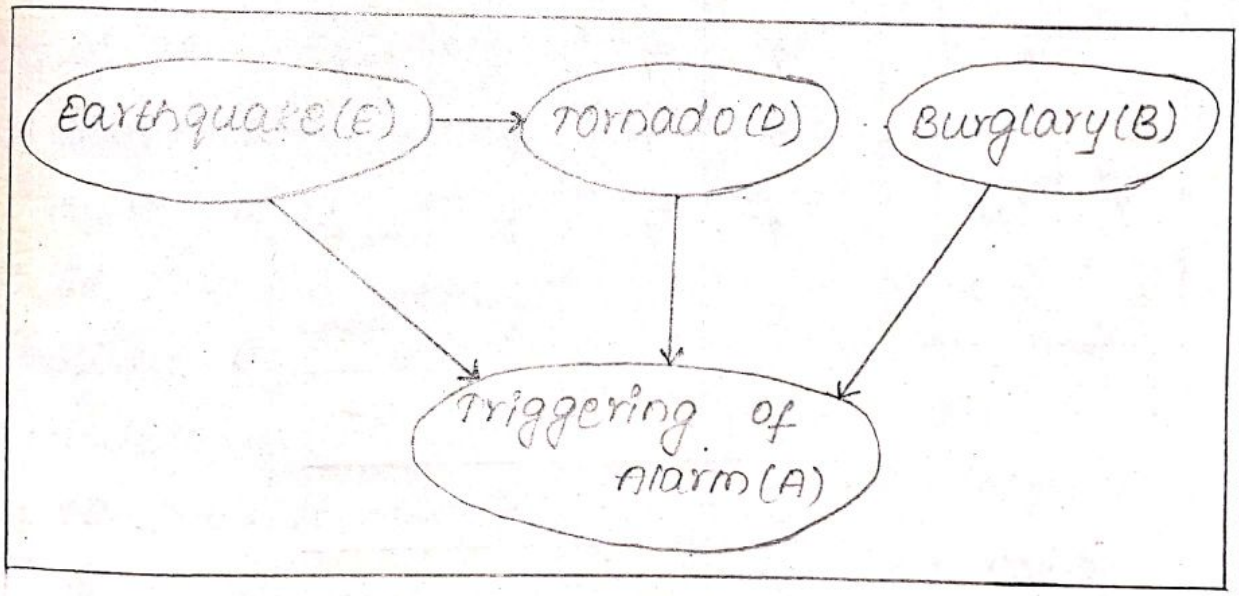
We notice that the above result can be obtained straightaway by using the following Bayesian network formula for joint probability.

Scanned with CamScanner

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid \text{parent-nodes}(X_i)).$$

98) consider an alarm system installed in a house and construct a simple bayesian Network of alarm system.

Consider an alarm system installed in a house that can be triggered by three Events, namely Earthquake, burglary and tornado. This can be modelled with the help of Bayesian network. All four variables have two possible values T and F. Here, the names of the variables can be abbreviated such that A = alarm, E = Earthquake, B = burglary, and D = tornado



Bayesian Network.

The directed acyclic graph given above illustrates the causality relationships

Scanned with CamScanner

that occur among the nodes. Here we notice that E and B are independent variables and A, D are dependent variables the probabilities values representing the complete Bayesian belief network. we know from the table that prior probability of Earthquake is 0.4 and if it is Earthquake then probability of tornado is 0.8 and if not, then probability of tornado is 0.5. similarly if it is Earthquake, then probabili-ty of ringing alarm is 0.85.

conditional probability tables.

P(E)	P(B)
0.4	0.7

E	P(D)
T	0.8
F	0.5

E	B	D	P(A)
T	T	T	1.0
T	T	F	0.9
T	F	T	0.95
T	F	F	0.85
F	T	T	0.89
F	T	F	0.7
F	F	T	0.87
F	F	F	0.8

the joint probability may be computed as

$P(E, B, D, A) = P(A/E, B, D) * P(D/E) * P(E) * P(B)$

Scanned with CamScanner

$= 1.0 * 0.8 * 0.4 * 0.7$

$= 0.214.$

once such a network has been constructed an inference engine can use it to maintain and propagate beliefs. Whenever some new information is recieved, the effects can be propagated through out the network until equilibrium probabilities are reached.

99) Explain Advantages and disadvantages of Bayesian Belief Network.

Advantages:)

- Since this model encodes dependencies among all variables, it can easily handle situations where some data entries are missing.
- It is intuitively easier for a human to understand direct dependencies and local distributions than complete joint distribution
- It can be used to learn casual relation-ships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention
- It is an ideal representation for combining prior knowledge and data because the model has both causal and probabilistic semantics.

Scanned with CamScanner

- Bayesian statistical methods in conjunction with Bayesian networks offer an efficient and principled approach for avoiding over-fitting of data.

Disadvantages:-

- The probabilities are described as a single numeric point value. This can be distortion of the precision that is actually available for supporting evidence.
- There is no way to differentiate between ignorance and uncertainty. These are distinct two different concepts and be treated as such.
- There exists the computational difficulty of exploring a previously unknown network.
- All branches must be calculated to find the probability of any branch of network. Even though the resulting ability to describe the network can be performed in linear time, this process of network ~~the resulting ability to describe the network~~ discovery is an NP hard task. It might be either too costly to perform or impossible, given the number and combination of variables.

Scanned with CamScanner

- The quality and extent of the prior beliefs used in Bayesian inference processing are major shortcomings.
- Reliability of Bayesian network depends on the reliability of prior knowledge.
- Selecting the proper distribution model to describe the data has a notable effect on the quality of the ~~network~~ resultant network. Therefore, selection of the statistical distribution for modeling the data is very important.

100) Explain about certainty factor theory?

Certainty factor theory provides another way of measuring uncertainty by describing a practical way of compromising on pure Bayesian system. Certainty factor is based on a number of observations. In traditional probability theory the sum of confidence for a relationship and against a relationship must add upto 1. However, in practical situation, it is often the case that an expert might have some confidence, say c, about some relationships being true and have no idea about the relationship being untrue. Further, confidence measures correspond to the informal evaluation that human ~~rights~~ experts attach to their conclusions, such as

Scanned with CamScanner

it is probably true or it is highly likely.

The certainty factor is based on some simplifying assumptions for creating confidence measures and for combining those confidences. These assumptions involve representing confidence for and confidence against separately by measure of belief and the measure of disbelief respectively. These two measures constrain each other where a given piece of evidence is either for or against a particular hypotheses the MB[H, E] is a measure of belief in the range [0, 1] in hypothesis H given the evidence E. if Evidence supports it fully then MB[H, E]=1 and it is zero. if the Evidence fails to support the hypothesis. Similarly MD[H, E] is a measure of disbelief in the range [0, 1] in hypothesis H given the Evidence E. It measures the extent to which the Evidence E. supports the negation of the hypothesis H. It is to be noted that MD is not compliment of MB.

The measure beliefs calculates the relative decrement of disbelief in a given hypothesis H due to some Evidence E. Measure of bilief may be intutively defined as follows

$$MB[H, E] = \frac{(1-P(H)) - (1-P(H/E))}{(1-P(H))}$$

Scanned with CamScanner

$$= \frac{P(H/E) - P(H)}{(1 - P(H))}$$

Here, $P(H)$ represents the probability of belief in hypothesis H and $(1 - P(H))$ represents probability of disbelief in hypotheses H without having any evidence. Further, $P(H/E)$ denotes the probability of belief in H after the knowledge of evidence E. In order to avoid getting a negative value of belief, we can modify the above definition to obtain positive value of measure. The modified definition of MB may be written as

$$MB[H, E] = \begin{cases} 1, & P(H) = 1 \\ \dfrac{Max\{P(H/E), P(H))\} - P(H)}{(1 - P(H))}, & \text{otherwise} \end{cases}$$

We can clearly see that MB takes values in the range $[0, 1]$

The measure of disbelief (MD) is similarly defined as the relative decrement of belief in a given hypothesis H due to some evidence E. It may be represented as follows.

$$MD[H, E] = \frac{P(H) - P(H/E)}{P(H)}$$

Alternatively, we can use the following definition for the measure of disbelief MD in order to

Scanned with CamScanner

get value of MD in the range [0,1].

$$MD[H,E] = \begin{cases} 1, & P(H)=0 \\ \dfrac{P(H)-\text{Min}\{P(H/E), P(H)\}}{P(H)}, & \text{otherwise} \end{cases}$$

Using MB and MD, we can define the certainty factor as given below.

$CF[H,E] = MB[H,E] - MD[H,E]$, where

$$-1 \le CF[H,E] \le 1.$$

positive certainty factor indicates evidence for the validity of the hypothesis, where evidence implies anything that is used to determine the truth of hypothesis. The belief in the hypothesis is directly proportional to the value of CF. For higher belief, there is higher confidence factor. If $CF = 1$, then the hypothesis is said to be true, while if $CF = -1$ the hypothesis is considered to be false. Moreover, if $CF = 0$, then there is no evidence regarding whether the hypothesis is true or false.

These measures satisfy the following properties

- If hypothesis H is true assuming E, then $MB[H,E] = 1$, $MD[H,E] = 0$. and $CF[H,E] = 1$.
- If hypothesis H is not true assuming E, then $MB[H,E] = 0$, $MD[H,E] = 1$, and $CF[H,E] = -1$

Scanned with CamScanner

- The sum of CF of belief and disbelief of hypothesis H, given Evidence E is 0, That is $CF[H, E] + CF[\sim H, E] = 0$

For computing CF in general, we need to determine the mechanism for handling the following three cases:

- certainty factor when there are two Evidences supporting hypothesis H. It is called incrementally acquired Evidence.
- certainty factor for combination of two hypotheses based on the same Evidence.
- certainty factor for chained rule.

The method for computing CF for each of the above mentioned cases is described

case 1:)

Incrementally Acquired Evidence:)

In this case, we have to compute certainty factor when there are two Evidences supporting hypothesis H, that is, CF (H, E_1 and E_2). In order to compute CF (H, E_1 and E_2), we first compute MB (H, E_1 and E_2) and MD (H, E_1 and E_2) as follows:

$$MB[H, E_1 \text{ and } E_2] = \begin{cases} 0 & \text{if } MD[H, E_1 \text{ and } E_2] = 1 \\ MB[H, E_1] + MB[H, E_2] * (1 - MB[H, E_1]) & \text{otherwise.} \end{cases}$$

Scanned with CamScanner

Similarly, MD is defined as follows

$$MD[H, e_1 \text{ and } e_2] = \begin{cases} 0, & \text{if } MB[H, e_1 \text{ and } e_2] = 1 \\ MD[H, e_1] + MD[H, e_2] * (1 - MD[H, e_1]) & \text{otherwise.} \end{cases}$$

Now, cf can be written as

$$cf[H, e_1 \text{ and } e_2] = MB[H, e_1 \text{ and } e_2] - MD[H, e_1 \,\&\, e_2]$$

101) Suppose we make an initial observation e_1, that confirms our belief in H with $MB[H, e_1] = 0.4$ and $MD[H, e_1] = 0$. consider a second observation e_2 that also confirms H with $MB[H, e_2] = 0.3$ and $MD[H, e_2] = 0.1$ compute $cf[H, e_1 \text{ and } e_2]$.

<u>Solution</u>:-

Given $MB[H, e_1] = 0.4$ and $MD[H, e_1] = 0$, we get $cf[H, e_1] = 0.4$. The measure of belief of H with combined observations e_1 and e_2 is determined in following manner.

$$MB[H, e_1 \text{ and } e_2] = MB(H, e_1) + MB(H, e_2) * (1 - MB(H, e_1))$$

$$= 0.4 + 0.3 * (1 - MB(H, e_1))$$

$$= 0.4 + 0.18$$

$$= 0.58$$

Similarly, $MD[H, e_1 \text{ and } e_2] =$

$$MD(H, e_1) + MD(H, e_2) * (1 - MD(H, e_1))$$

Scanned with CamScanner

$= MD(H, E_2)$

$= 0.1$

therefore, $CF[H, E_1 \text{ and } E_2] = 0.58 - 0.1 = 0.48$

102) Suppose we are given $MB[H_1, E] = 0.4$ and $MB[H_2, E] = 0.3$; $MD[H_1, E] = 0.1$ and $MD[H_2, E] = 0.2$ compute CF.

<u>solution:</u> As already defined

$MB[H_1 \text{ and } H_2, E] = Min(MB[H_1, E], MB[H_2, E]) = 0.3$

$MD[H_1 \text{ and } H_2, E] = Max(MD[H_1, E], MD[H_2, E]) = 0.2$

$MB[H_1 \text{ or } H_2, E] = Max(MB[H_1, E], MB[H_2, E]) = 0.4$

$MD[H_1 \text{ or } H_2, E] = Min(MD[H_1, E], MD[H_2, E]) = 0.1$

Using these values, we can compute $CF[H, E_1 \text{ and } E_2]$ and $CF[H, E_1 \text{ or } E_2]$ in the following manner:

$CF[H, E1 \text{ and } E2] = 0.3 - 0.2 = 0.1$

$CF[H, E1 \text{ or } E2] = 0.4 - 0.1 = 0.3$

103) Formalyze Dempster theory?

Let U be the universal set of all hypotheses, propositions, or statements under consideration. The power set P(U), is the set of all possible subsets of U, including the empty set represented by ϕ. The theory of Evidence assigns a belief mass to each subset of the power set. A function $m: P(U) \rightarrow [0,1]$ is called a basic belief assignment function. It satisfies

Scanned with CamScanner

the following two axioms:

- $m(\emptyset) = 0$
- $\sum m(A) = 1, \forall A \in P(U)$

Sometimes, m is also known as a basic probability assignment. The quantity m(A) is a measure of that portion of the total belief committed exactly to A but to no particular subset of A. The value of m(A) is called mass assigned to A on the unit intervals and pertains only to the set A and makes no additional claims about any subsets of A, each of which has, by definition, its own mass.

10c) Explain dempster's Rule of combination?

The original combination rule, known as Dempster's rule of combination, is a generalization of Baye's rule. This rule strongly emphasizes the agreement between multiple sources and ignores all the conflicting evidence through a normalization factor.

Assume that m_1 and m_2 are two belief functions used for representing multiple sources, in evidences for two differed hypotheses. Let $A, B \subseteq U$, such that $m_1(A) \neq 0$, and $m_2(B) \neq 0$. The Dempster's rule for combining two belief functions to generate

Scanned with CamScanner

a function m3 may be defined as

$$m3(\phi) = 0$$

$$m3(C) = \frac{\sum_{A \cap B = C} (m1(A) * m2(B))}{1 - \sum_{A \cap B = \phi} (m1(A) * m2(B))}$$

This belief function gives us a new value when applied on the set C, where $C = A \cap B$. The combination of two belief functions is called joint mass. Here m3 can also be written as $(m1 \circ m2)$. The Expression $[\sum_{A \cap B = \phi} (m1(A) * m2(B))]$ known as the normalization factor, is a measure of the amount of conflict between the two mass sets. The normalization factor has the effect of completely ignoring conflicts and attributing any mass associated with conflict to the null set. consequently, this operation yields counter intuitive results in the face of significant conflict in certain contexts. Now, we will illustrate the concept of belief function m through an abstract Example.

Scanned with CamScanner

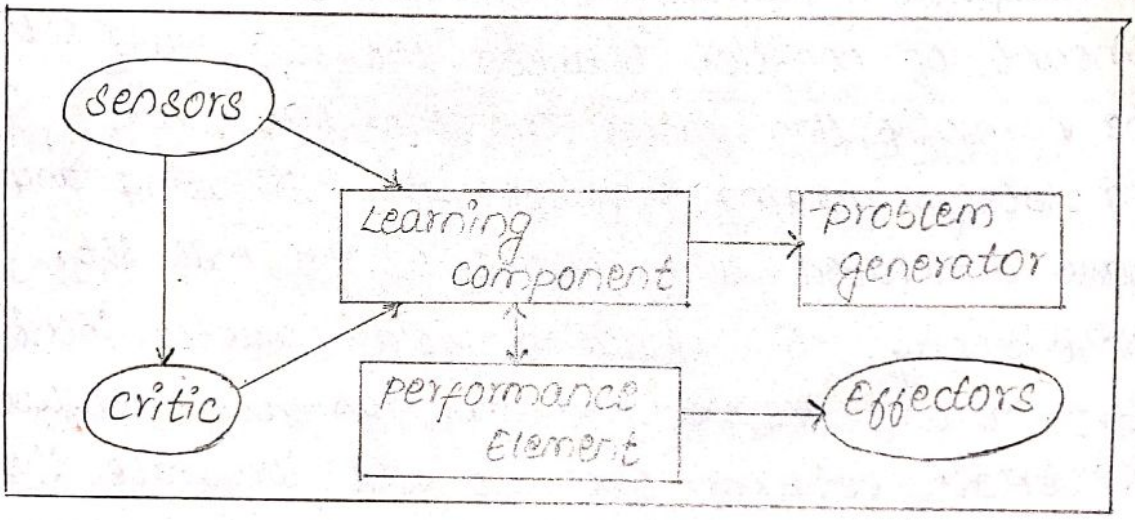
UNIT - IV

MACHINE LEARNING PARADIGMS.

10b) Write the components of learning system?

components of learning system:-

(i) Learning component.
(ii) performance element.
(iii) critic.
(iv) problem generator.
(v) sensors.
(vi) Effectors.



(i) Learning component:-

The basic purpose of learning component is to make changes or improvements to the system depending on its performance.

(ii) performance element:-

In a learning system, the performance element performs the task of choosing

Scanned with CamScanner

the actions that need to be taken.

(iii) critic :)

The job of the critic is to inform the learning component regarding its performance with respect to a fixed standard. Note that critic could be either a human or an automated component.

(iv) problem generator:)

problem generator is imperative to a learning system since it suggests problems or actions that would lead to generation of new examples or experiences, which will aid in further training of the system.

v) sensors and effectors:)

Both these components are external to the system. The system receives information or data from sensors, while the output is transmitted through effectors.

106) What is Rote learning?

Rote learning basically refers to the process of memorization. Hence, it requires saving knowledge so that it can be utilized again whenever needed. Rote learning involves one-to-one mapping from inputs to stored representation and is also known as learning by memorization, it uses association based storage and retrieval. Moreover, there is no

Scanned with CamScanner

repeated computation required, only an inference or a query is necessary. An Eg of rote learning is caching where large pieces of data or computed values are stored and recalled when required by a computation. Many AI programs use caching very effectively since it allows significant saving in time. Although memorization is a key requirement in learning and development of an intelligent program, it can be complicated subject. Inspite of being a basic and simple process, rote learning highlights some relevant issues.

- **Organization:-**

Knowledge should be stored in such a manner that accessing this stored knowledge is faster than resorting to re-computation. Organization of knowledge is achieved by employing techniques such as hashing, indexing, sorting and so on.

- **Generalization:-**

Since the number of stored objects can be quite large, we need to generalize some information to make the problem manageable.

- **Stability of Environment:-**

The method of rote learning does not work very effectively in a rapidly changing

Scanned with CamScanner

environment. In case there is a change in environment, the change has to be detected and recorded exactly.

107) What are learning Methods?

Some Basic learning Methods are

1. Rote learning
2. Learning by taking advice.
3. Learning by parameter Adjustment.
4. Learning by Macro-operators.
5. Learning by Analogy.
6. Supervised and Unsupervised learnings.
7. Neural Network Based learning.
8. Reinforcement learning.
9. Inductive learning.

108) What is Neural Network based learning?

A neural network is a computational model somewhat similar to the human brain. It contains several units that work in parallel with no central control. connections between all working units are weighted. and these weights can be modified by learning method. It represents a form of connectionist learning in which the data structure consists of a set of neurons connected by weighted links. Feed forward network is a kind of

Scanned with CamScanner

supervised learning where training eg's are given by external teacher to train the network. once it is trained, it can be used for any input instance. The training examples in the network are composed of 2 parts. an input vector and an output vector. associated with the input and output neurons respectively, In the process of learning, the input vector is provided at the input layer of the system along with the desired responses, one of each neuron, at the output layer. When a forward pass is done, the errors and discrepancies between the desired and actual responses for each neuron in the output layer are obtained.

109) Explain supervised concept learning?

In supervised concept learning, both the inputs and the outputs can be observed. Here, a set of (x, y) pairs are provided by a supervisor, where x denotes input, while y denotes the corresponding output. suppose, we are given a training set of positive and Negative Examples of a concept, now, we have to construct a description that will accurately classify whether future examples are positive or not.

Scanned with CamScanner

In other words, we have to learn some good estimation of function f, if we are given a training set $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$ where each y_i either positive or negative. supervised concept learning requires set of labelled instances. suppose we are given a set of examples of some concept or category, then we are required to determine if an eg is an instance of the concept or not. If it is an instance, we call it positive, otherwise it is called negative.

110) Explain unsupervised concept learning?

Unsupervised learning is a type of machine where manual labels of inputs are not used. It is therefore clearly distinguished from supervised learning and reinforcement learning approaches. In unsupervised learning a typical task is of classification or regression using a set of examples prepared by humans. In this kind of learning, there is no information available about the correct outputs. Unsupervised learners can learn to predict future percepts based upon present ones, but cannot learn which actions to take without a utility function. In this learning we are only given Inputs. In both supervised and unsupervised cases, the ultimate goal is to

Scanned with CamScanner

estimate f.

Examples of unsupervised learning include clustering and discovery. In clustering natural classes are formed for data instances and the method of classifying them are also devised. In discovery, a specific goal is not given.

Commonly used supervised learning algorithms among neural network models include self organizing map and adaptive resonance theory. The goal of learning in the self organizing map is to discover some unerlying structure of the data and cause different parts of the network to respond similarly to certain input patterns.

(iii) Explain about reinforcement learning?

In reinforcement learning, the decision-making system receives rewards or feedback for its action at the end of a sequence of steps. It is required to assign reward to steps while solving the credit assignment problem, this problem determines which steps should receive credit or blame for the final result. As opposed to supervised learning reinforcement learning takes place in an environment where the agent cannot

Scanned with CamScanner

directly compare the results of its action to a desired result. Instead it is given positive or negative feedback directly on the basis of its action. Reinforcement learning may cause a system to win or lose a game, or inform a system that it has made a good move or a poor one. Therefore the primary task of reinforcement learning is to obtain a successful function using these rewards.

The agents can learn basic kinds of information as follows.

(i) Utility function:

The agent learns the utility of being in various states, and chooses actions in order to maximize the expected utility of their outcomes. This requires the agent to keep a model of the environment.

(ii) Action value function:

The agent learns an action value function giving the expected utility of performing an action in a given state. This is called Q-learning. It is a model free approach.

Scanned with CamScanner

(12) Explain about Inductive learning?

Inductive learning is that process of learning where a system learns by Eg i.e., It tries to induce a general rule from a set of observed instances, it uses specific examples to reach to general conclusions. Humans often classify things in the world without knowing explicit rules. A teacher or tainer can aid in proper classification. This involves classification by assigning, to a particular input the name of a class to which it belongs,

The task of constructing class definitions is called induction or concept learning. It extrapolates from a given set of examples and can make accurate predictions about future examples. Most inductive learning is supervised learning, in which examples are provided with classifications. An alternative method to inductive learning is clustering. More formally, an Eg is a pair $(x, f(x))$, where x is the input and $f(x)$ is the output of the function applied to x. The task of pure inductive inference involves the following: given a set of Eg's of f, find a hypothesis h that approximates f.

The process of inductive

Scanned with CamScanner

learning can be described symbolically using predicates P and Q. If we observe repeated occurrence of events $P(v_1), P(v_2), \ldots, P(v_K)$ then we can generalize inductively that $\forall x\, P(x)$. similarly, if we observe implications $P(v_1) \rightarrow Q(u_1), P(v_2) \rightarrow Q(u_2) \ldots, P(v_K) \rightarrow Q(u_K)$ thus we can generalize by concluding $\forall x \forall y\, P(x) \rightarrow Q(y)$.

11B) Describe decision tree construction using greedy algorithm?

The decision tree is a hierarchial data structure implementing the divide and conquer strategy. This model is used in super-vised learning where the local region is identified using greedy algorithm, and consists of internal decision nodes and terminal leaves. Each decision node implements a test function, which is applied for each input and based on the output. the appropriate branch is chosen. The process starts from the root and is repeated recursively until leaf node is reached and the value stored in the leaf node gives the output corresponding to the input.

An algorithm called ID3 was developed by Quinlan which helps to learn object classification from labelled training eg's

Scanned with CamScanner

The basic algorithm for ID3 is based on earlier research on concept learner systems. The CLS algorithm starts with a set of n training objects. where each object has a set of m attributes. An attribute with small number of discrete values say K is selected as a root node of the tree. This node has K branches emanating from it where each branch corresponds to the value of selected attribute. The set of training objects are then parti-tioned into atmost K subsets depending on the attribute values of the objects. The same procedure is repeated recursively for each of these subjects using other (m-1) attributes to form lower level nodes and branches. The process stops when all the training objects have been divided into single class entities that become labelled leaf nodes of the tree. This structure is known as decision tree and is used to classify new unknown objects given a description consisting of its m attributes.

Algorithm:-

- Top-down construction of the decision tree by recursively selecting the 'best attribute' to use at the current node in tree.

Scanned with CamScanner

- Once the attribute is selected for the current node, generate children nodes, one for each possible value of the selected attribute.
- Partition the examples using the possible values of this attribute, and assign these subsets of the examples to the appropriate child node.
- Repeat for each child node until all examples associated with a node are either all positive or all negative.
- Choose the best attribute for a node. Some of the techniques for such selection may be listed as.
  - Random :- select any attribute at random.
  - Least-values :- choose the attribute with the smallest number of possible values.
  - Most-values :- choose the attribute with the largest number of possible values.

114) Explain about deductive learning?

Deductive learning involves working on already existing facts and knowledge and simply deducing new knowledge from knowledge already present in the system. Therefore, deductive learning or reasoning can be described as reasoning of the form if A then B. Thus, regarding deductive learning, we can say

Scanned with CamScanner

that it is not a form of learning where new knowledge is generated, in fact it simply refers to memorizing the logical consequences of what is already known to the system. therefore, deduction is merely an application of knowledge that leads to generation of new knowledge, however, this knowledge doesnot represent any new semantic information. the rule if then B represents the knowledge completely as the added knowledge., since any time the asser-tions (A) are true then the conclusion B is true as well.

15) Explain about clustering?

clustering is considered to be the most important unsupervised inductive learning problem. clustering can be defined as a process in which data or objects are arranged into groups in such a manner that all members of a group are similar in some way. therefore, a cluster may be defined as being a collection of similar objects, which are dissimilar to objects belonging to other clusters. the goal of clustering is to determine intrinsic grouping in a set of unlabelled data instances. the properties of

Scanned with CamScanner

clustering are:

- find a structure in a collection of unlabelled data.
- be scalable.
- be able to deal with different types of attributes
- be able to discover clusters with arbitary shape.
- have minimal requirements for domain knowledge to determine input parameters.
- have the ability to deal with noise.
- be insensitive to the order of input records
- have high dimensionality, interpretability, and usability.

11b) Describe the following

a) K-means b) Fuzzy-C-means

a) <u>K-means:</u>

K-means is one of the simplest unsupervised learning algorithms that can be used to solve the well-known clustering problem. The procedure follows fairly simple approach of classifying a given object set through a certain number of clusters, say K. if we assume that there are n data points, then K means the algorithm broadly is stated as follows.

- place K points into the space represented by the objects. These points represent initial group centroids.

Scanned with CamScanner

⊙ Repeat until the centroids do not have

* Assign each object to the group that has the closest centroid.
* When all objects have been assigned, recalculate the positions of the k centroids

This algorithm causes a seperation of objects into groups from which the metric to be minimized can be calculated. finally, the aim of this algorithm is to minimize can objective function. In this case, an objective function F is the squared error function and it may be defined as follows:

Let x_i^j be the ith data point in cluster j and c_j be the cluster centre, then we write

$$F = \sum_{j=1}^{K} \sum_{i=1}^{n} \left\| x_i^j - c_j \right\|^2$$

<u>Fuzzy c-means:</u>

Fuzzy c-means is a method of clustering in which an object is allowed to belong to two or more clusters. This method was initially developed by Dunn in 1973 and is frequently used in applications such as pattern recognition. Let us assume u_{ij} to represent the degree of membership of x_i (ith data) in the cluster j. and c_j to be the center of the cluster j.

Scanned with CamScanner

- Initialize $U^0 = [u_{ij}]$ matrix
- $K = 1$
- Repeat until $[\|U^{K+1} - U^K\| < \varepsilon]$
  - calculate the center vectors $C^K = [c_j]$ using U^K as follows

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^m \cdot x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

  - update U^K, U^{K+1} using the following formula

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left\{ \frac{[\|x_i - c_j\|]}{\|x_i - c_k\|} \right\}^{2/(m-1)}}$$

  - $K = K + 1$

this method of clustering is based on the minimization of the following objective function

$$F_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - c_j\|^2$$

117) Write about hierarchial clustering?

hierarchial clustering is a type of clustering that is based on the union of two nearest clusters. In this clustering, the starting condition is realized by setting every object as a cluster.

Scanned with CamScanner

After a few iterations, it reaches the final clusters. Given a set of n objects to be clustered, and an n*n distance or similarity matrix, the basic process of hierarchial clustering defined by S.C. Johnson is as follows.

- Start by assigning each object to a cluster, so that if we had n objects to start with, we now have n clusters, each containing just one object.
- Repeat until a single cluster of size n is formed.
  - find the closest pair of the clusters and merge them into a single cluster, so that now we have one cluster less.
  - compute distances or similarities between the new cluster and each of the old clusters.

There are number of methods that can be used for finding the closest pair of clusters

- Single-linkage clustering
- complete-linkage clustering
- Average-linkage clustering.

118) With regards with SVM Explain

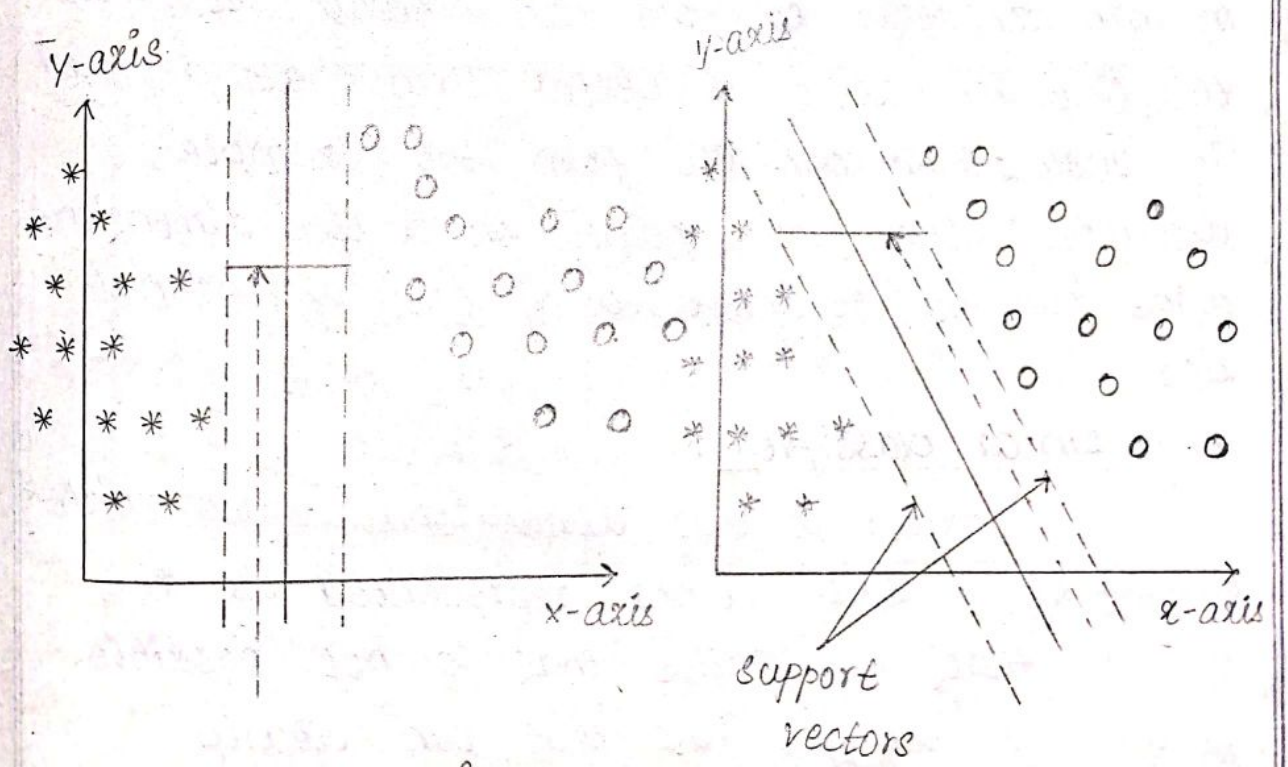
a) Linear classifier

b) Non linear classifier.

Scanned with CamScanner

a) Linear classifier:-

Let us consider a simple two dimensional eg to illustrate the concept of a support vector. Assume that we wish to classify a given cluster of data points expressed as pair of two predictor variables values into two categories. For this, the given data points are plotted in a two dimensional space with one predictor on the x-axis and the other on the y-axis. For the sake of graphical visualization, values of one category are represented using the symbol *, while values of the other category are represented using o.



case 1: Small Margin

case 2: Large Margin

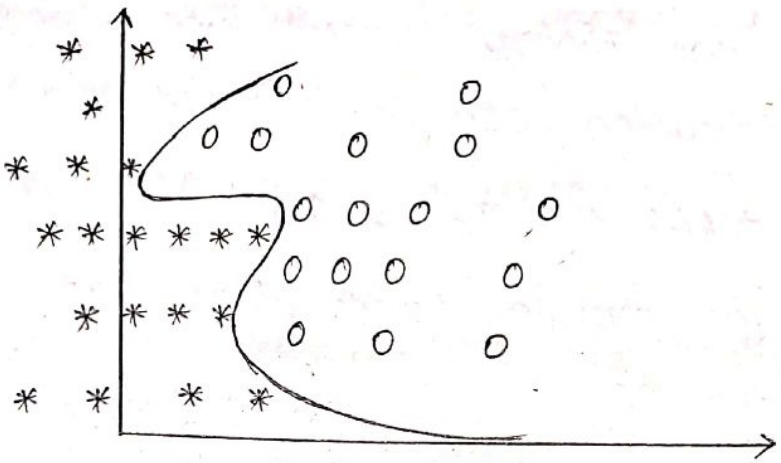
Scanned with CamScanner

As can be seen from the above cases of both categories are completely separated. In SVM analysis, a line is found such that it is able to separate the cases depending on their target categories. It is clear that an infinite number of such lines are possible. We have shown two cases of lines separating line mark the distance between the dividing line and the closest vectors to the line are support vectors and the distance between the two dashed lines is called the margin. In SVM analysis, we have to find the separator line in such a way that its support vectors have max margin. In this eg, we can clearly see that the line in case 2 is better than that depicted in case 1. We can see from the example discussed above that points on a two dimensional plane can be separated using one dimensional line.

Non Linear classifier:

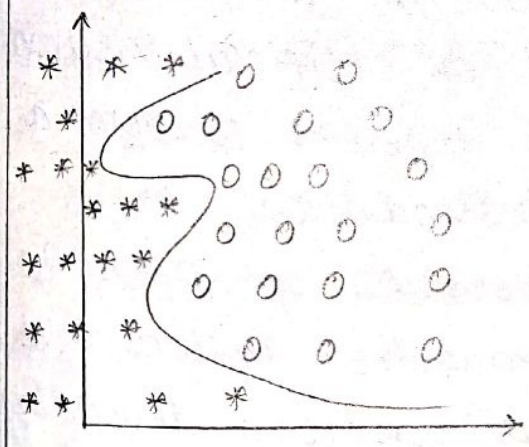
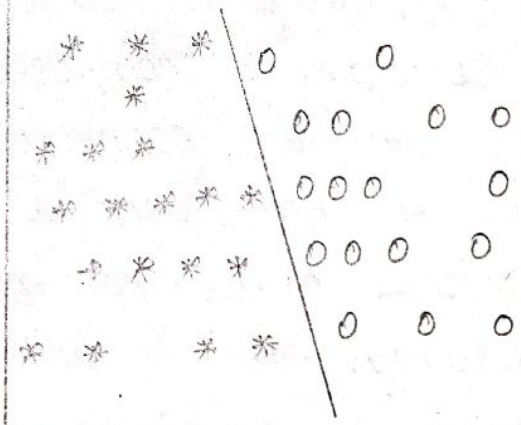
Consider a two dimensional space with two types of data points represented as * and o. Here we notice that is not possible to draw a straight line that can clearly separate the given data into two classes. A non linear curve has to be drawn to separate these points.

Scanned with CamScanner



separation of points using non linear curve.

In such situations, SVM uses a kernel function that transforms the given data into a higher dimension space and it becomes possible to perform the separation using lines instead of fitting non linear curves to the data.

Input Space	Feature space
	

Transformation into Higher Dimensional space.

A linear model of SVM is constructed using kernal function in the new space that represents a similar decision boundary in the original space.

Scanned with CamScanner

For the linearly separable case, a hyper plane separating the binary classes in three attribute situation can be represented as.

$$y = W_0 + W_1 x_1 + W_2 x_2 + W_3 x_3$$

The maximum margin hyper plane can be represented as

$$y = \sum \alpha_i y_i K(x(i) \cdot x).$$

11) Explain Neural Networks?

The term neural Network is generally used to describe a network of biological neurons that are functionally connected to the central nervous system of most living organisms. Group of neurons are assigned a specific physiological function. An artificial neural network is composed of artificial neurons or nodes that are connected together to form a network. An ANN uses a mathematical or computational model for processing of information. Usually, ANN is an adaptive system that is capable of modifying its structure depending on the external or internal information being transmitted through the system.

We can say that ANN is a machine learning approach that follows the same learning principle employed while performing a particular learning task.

Scanned with CamScanner

12) Enumerate Neural network architectures and explain single layer feed-forward Networks?

Neural Network architectures?

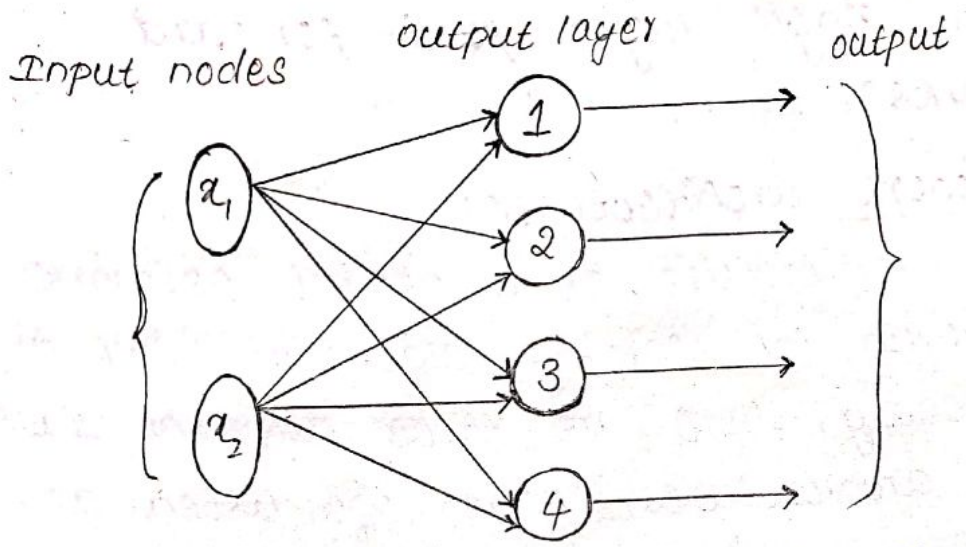
The architecture of a neural network is associated with the learning algorithm used to train it. Broadly, there are three different classes of network architectures, which are listed as

- Single Layer feed forward network.
- Multi-Layer-feed forward network.
- Recurrent Network.

Single Layer feed forward network.

Feed forward neural networks were the first and the simplest of ANNs to be developed. A single layer feed forward neural network is defined as a network where connections between various nodes do not form a directed cycle such that the information in this network moves only in the forward direction that is, from input nodes to output nodes, through a series of weights. There are no hidden layers It contains input and output layers only, each input node is connected to each output node with a link having weight. Input to each output node is weighted sum of input parameters, Thus the sum of products of weights and the input parameters is calculated at each node.

Scanned with CamScanner



Single layer feed forward Neural Network

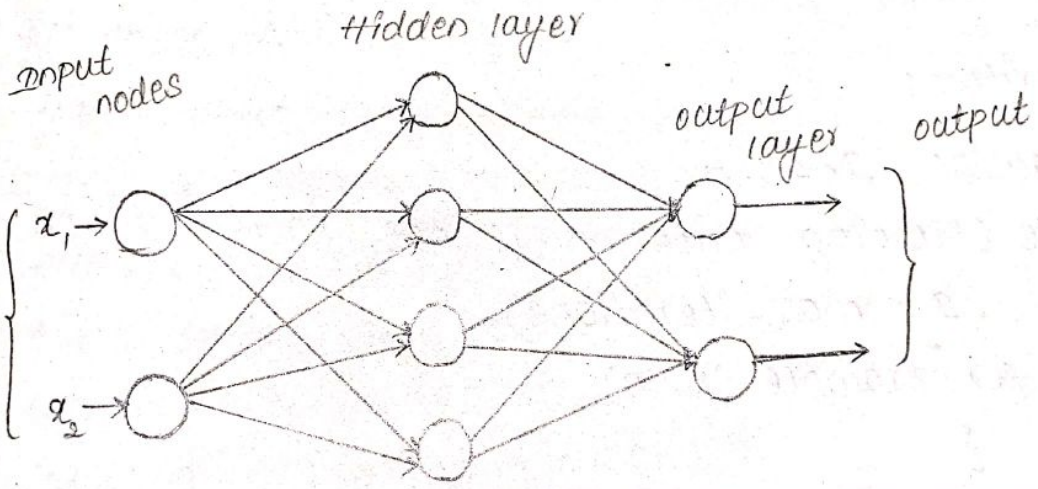
12) Write about Multi Layer feed forward Networks?

Multi layer feed forward network have to be used to handle data that is not linearly separable or not separable by hyper-plane. These contain hidden layers between input and output layers, which do not directly receive inputs or send outputs to the external environment. The architecture of an FFNN is shown below. It should be noted that each connecting link has a weight associated with it.

Multi layer feed forward networks continuously evolve with the help of learning techniques. One of the most crucial learning techniques is known as back propagation method.

Scanned with CamScanner

- The output values obtained for a known set of inputs is compared with the correct answer in order to determine the value of some pre-defined error function.
- This error is then fed back into the network using back propagation method.
- The algorithm utilizes this information and then adjusts the weights of all connections so that the corresponding error function is reduced to some extent.



122) Write about back propagation algorithm?

Back propagation consists of the repeated application of the following passes:

Forward pass:

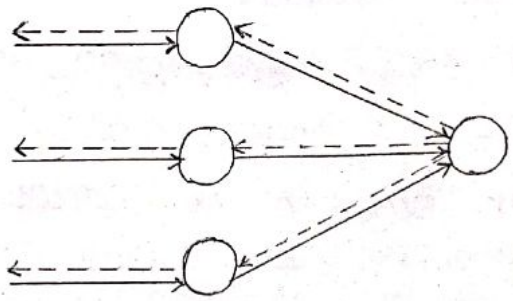
The network is activated for one example and the error of each neuron at the output layer is computed.

Backward pass:

The network is used for updating weights and propagated from last node to start node.

Scanned with CamScanner

Beginning at the output layer, the error is propagated backwards through the network, layer by layer. by recursively computing the local gradient of each neuron.



Back propagation Algorithm.

<u>Algorithm:</u>

```
Initialize I=1;
While (stopping criterion is not satisfied or
        I < max-iterations)
    for Example (x, D)
      {
        run the network with input x and
        compute the output y;
        update the weights in backward order.
        starting from those of the output
        layer computed using the delta rule
      };
      I= N+1;
End-While.
```

Scanned with CamScanner

123) Explain Radial basis function Network?

A function is said to be a radial basis function if its output depends on the distance of the input from a given stored vector. Networks that use such RBFs as activation functions are known as radial basis functions. The RBF neural network has an input layer a hidden layer and an ouput layer. In such RBF networks, the hidden layer uses neurons with RBFs as activation functions. The outputs of all these hidden neurons are combined linearly at the output node. These networks have a wide variety of applications such as.

- function approximation.
- time series prediction.
- control and regression.
- pattern classification tasks.
- for performing complex

124) What are the design issues of ANN?

Some of the design issues that are usually encountered while designing a neural network include:

- data representation.
- network topology.
- determination of network parameters and
- training and validation.

Scanned with CamScanner

Data representation:-

Data representation is a design issue that is problem specific. generally ANNs use continuous attributes, hence, symbolic attributes need to be encoded into continuous forms. Attributes of different types may have different ranges of the values which in turn affect the training process. To overcome this, each value is normalized to scale its values between 0 and 1 using the formula for each value of attribute x_j

$$x_j = (x_j - Min_j) / (Max_j - Min_j)$$

Network topology:-

Network topology may again be considered to be problem specific since the number of layers and neurons depends on the task at hand. In practice this issue is resolved by using trial and Error method. one of the following two strategies may be used

- start from a large network and successively keep removing neurons and links until network performance starts showing degradation
- start with a small network and introduce new neurons until performance becomes satisfactory.

Network parameters:-

The concept of network parameters involves a no. of steps.

Scanned with CamScanner

- proper initialization of weights.
- choosing appropriate learning rate.
- the training set of examples.

Initialization of weights:- Generally, initial weights are chosen randomly, with values typically lying between -1.0 and 1.0 or -0.5 and 0.5. However, in case some inputs are much larger than others the process of random initialization may bias the network so as to give greater importance to larger inputs.

- For weights from the input to the first layer

$$W_{ji} = \frac{\pm \sum_{i=1}^{N} \frac{1}{|X_i|}}{2N}$$

- For weights from the first to the second layer

$$W_{kj} = \frac{\pm \sum \frac{1}{\phi\left(\sum_{i=1}^{N} W_{ji} X_i\right)}}{2N}$$

Choice of learning rate:- The exact value of η depends on the problem. In general, values bet 0.1 and 0.9 are used in many applications other heuristic adapt η during the training of the network.

Training examples:- The heuristic that needs to be followed while training is that the no

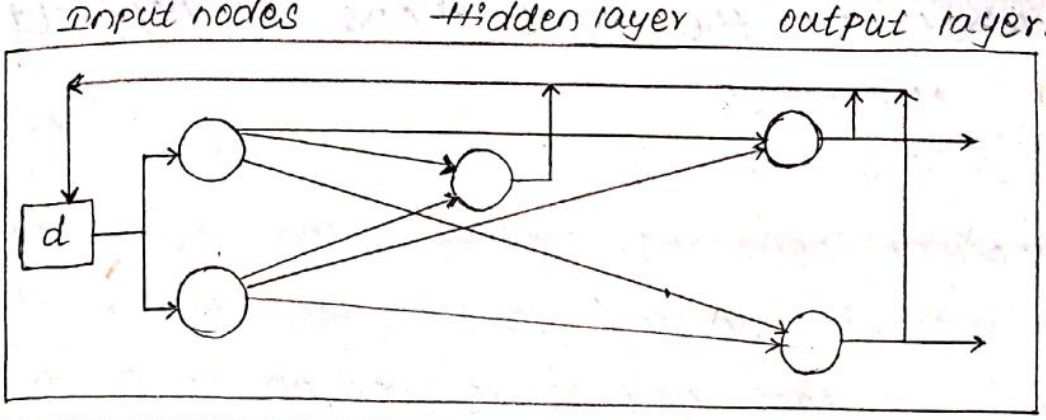
of training examples must be atleast five to ten times the number of weights of the network.

125) Explain about Recurrent Networks?

A feed forward network represents an acyclic network since data can pass from input to the output nodes but not vice versa once the FFNN is trained, its state gets fixed and does not modify when new data is presented to it. and it has no memory. These shortcomings of feed forward networks are resolved by another type of network called recurrent networks. These networks can have connections going back from output to input nodes and in, fact, can have arbitary connections between any nodes. In addition an internal state of recurrent networks can be modified as new sets of input data are presented. It also possesses a memory. which proves to be useful while solving problems. where the solution depends on all previous inputs and not just on the other inputs.

Recurrent networks can be trained by using back-propagation algorithm. once this training process is completed, the network becomes capable of performing a sequence of actions.

Scanned with CamScanner



126) Write notes on Boltzmann machines?

Boltzmann machine is a name given to a derivative of the Hopfield network, which is obtaining by combining a Hopfield network with simulated annealing search technique. Annealing is the process of gradually moving down from high temperature to low. This machine was proposed by Hinton and Sejnowski in 1986. The main point of difference in a Boltzmann machine is that the units contained in this machine update their individual binary states by a stochastic rule rather than a deterministic one. The probability of finding any given unit in the active state can be calculated by formula:

$$P = 1/(1+e^{\Delta E/T})$$

Where ΔE denotes the sum of the active input connections to the chosen unit and T denotes the temperature of network.

At high temperatures, units display random behaviour, while they behave as HN at low temperatures.

Scanned with CamScanner

127) What are the applications of hopfield Network?

The following are some applications:

- Hopfield networks can be used as a content addressable memory system by setting the state of each unit corresponding to a partial pattern. To retrieve a pattern, we only need to supply a portion of the pattern. The network will then settle into the stable state that best matches the partial pattern. This is known as associative memory since it recovers memories on the basis of similarity.
- These networks can be used to solve a wide variety of constraint satisfaction problems in which each unit is viewed as a hypothesis and weighted connections represent mutually supported hypothesis. As the network settles onto a stable state, it attempts to assign truth values to the various hypotheses, violating as few constraints as possible in the process.
- These networks can also be used for complex problems of optimization and pattern classification.

Scanned with CamScanner

UNIT - V

KNOWLEDGE REPRESENTATION TECHNIQUES

128) Describe about the case grammars?

Case grammar theory was proposed by the American linguist charles J. Fillmore in 1968 for representing linguistic knowledge that removed the strong distinction between syntactic and semantic knowledge of a language.

The case frame contains semantic relation rather than syntactic ones. The semantic roles such as agent, action, object and instrument are extracted from the sentence straightaway and stored in case frame which represents semantic form of a sentence. sentences having same syntactic parse structure but having different meanings are stored ~~different~~ differently in case frame.

- Agent (instigator of action):

Noun phrase of a sentence generally fills the Agent case if it describes the instigator of the action. Not all animate in the NP at subjects possible fill Agent case.

- Object (identity on which action is performed):

The noun being acted upon by action is used to fill up object case.

Scanned with CamScanner

- Dative: It basically refers to object of an action that is animate.
- Experiencer: sometimes we have sentences having intransitive verbs and then the subject of such sentences is experiencer.
- Beneficiary: It is animate person for whom a certain event is performed for its benefit.
- Instrument: If some instrument is used to perform an action, then it fills instrument case.

Case Frame	
Cases	Values
Action	Give
Agent	John
Objective	Apple
Beneficiary	Mike
Time	past
Location	kitchen

Mike was given an apple by John in kitchen

12a) Write notes on semantic web?

Current web technologies are based on the premise that humans can read, understand and operate information present on web in HTML. It contains virtually boundless information in the form of documents

Scanned with CamScanner

the search engines can search required documents but cannot understand the information stored in document. as it is not in structured form. these documents have to be read and interpreted by humans before any useful info can be extracted manually.

The semantic web is an extension of traditional web that provides a common framework allowing data to be shared and reused across different applications. It is a collaborative effort led by world wide web consortium with participation from a large number of reaserchers and industrial partners. The semantic web is about common formats for interchange of data and language for recording how the data relates to real world objects. The semantic web adds meaning or semantics to web content in order to make it easier to be used by both machines and humans. The semantic web in principle describes all the data and information, including those of the web itself, in the form of machine readable form called Meta Data. It allows the system to process the data automatically that becomes a part of web and may be processed independent of application. platform or domain. Therefore semantic web is about having data as well as documents on the web so that

Scanned with CamScanner

machines can process, transform, assemble and even act on the data in useful ways.

130) Discuss Resource Description Framework?

The basic data model used to build the semantic web is called the Resource Description Frame work that is a domain independent model for describing resources. A resource may be name, place, constant, web-pages, parts of web pages, or even physical objects and can be represented by a uniform resource identifier. The RDF uses URIs to represent data, usually in triple structures. The subject in triple structure is the URI of some resource, the predicate is also represented by an URI that expresses some relationship between the subject and the object. The object is either a literal or another resource represented by a URI.

It is a simple data model for referring to objects / resources and their relationships and can be represented in XML syntax. It is based on the idea of making statements about web resources in the form of subject predicate object expressions, called triples in RDF terminology.

Scanned with CamScanner

18) Explain the design of ontology?

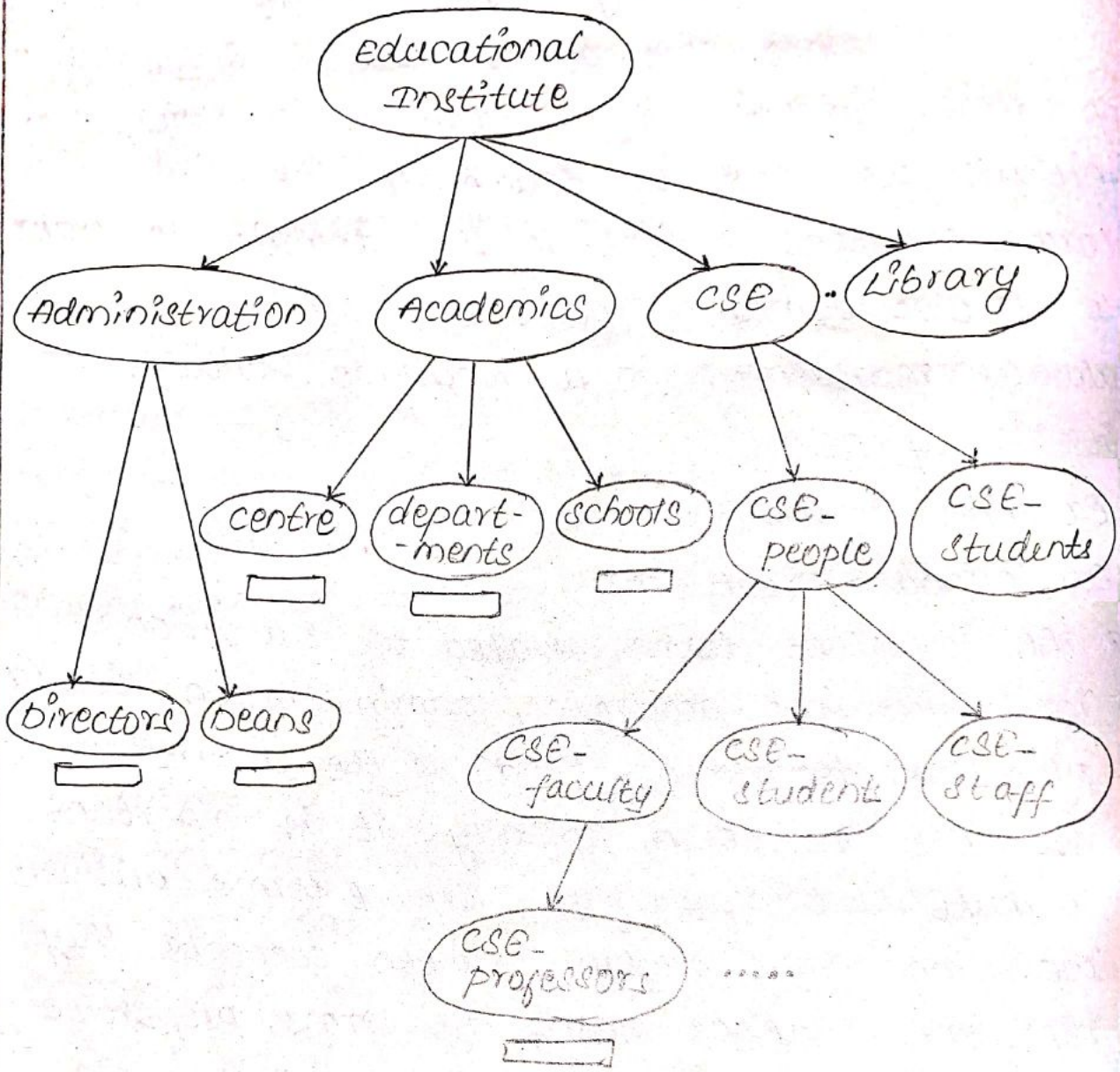
Since ontology is explicit specifications of terms present in the domain ~~in order~~ initially we need to enumerate the list of terms, relations present in the domain in order to develop domain specific ontology, ontology is always represented in a language suitable to its specification.

Example:

Educational Institutions:

- The important terms related to educational institutions are academics, administration, faculty professor, department, head of the department students, and so on. Initially, it is important to make a comprehensive list of terms without worrying about overlap between concepts they represent, relations among the terms, properties that the concepts may have, or whether the concepts are classes or slots.
- Define class hierarchy structure for these terms after enumerating all the possible terms of the domain
- one can design hiearchy structure based on the need.

Scanned with CamScanner



ontology structure.

- ontology is represented using ontology language such as RDF schema and OWL
- Each rectangular box contains the list of all instances of that class.

Scanned with CamScanner

132) Explain the phases of sentence Analysis.

A sentence is usually analyzed through different phases such as morphological analysis, syntactic analysis, semantic analysis, discourse and pragmatic analysis.

Morphological analysis:-

Morphological analysis process is carried out initially on the natural language sentences, this method tries to extract root word from declined or inflectional form of word after removing suffices and prefixes.

Syntactic analysis:-

This method of analysis uses the result of MAP to build a structural description of the sentence based on grammatical rules. This process is called parsing. A declarative representation of syntactic facts about the language and procedure compares the grammar against the input sentence to produce parse structures.

Semantic analysis:-

It creates a semantic structure by ascribing the literal meaning to a sentence using parse structure obtained in syntactic phrase. It maps individual words into corresponding objects in the knowledge base and combines the words with each other using

Scanned with CamScanner

semantic rules. Main purpose of semantic processing is the creation of target representation of the meaning of sentence.

Pragmatics analysis:-

It refers to intended meaning of a sentence used in different contexts. The context affects the interpretation of the sentence.

Discourse analysis:-

It refers to conversation between two or more individuals and the interpretation is based on the belief set at that time of conversation. Here the interpretation of spoken sentence will be based on the belief set of the people involved in conversation.

13) Write the structures and description of bottom up and top down parsing?

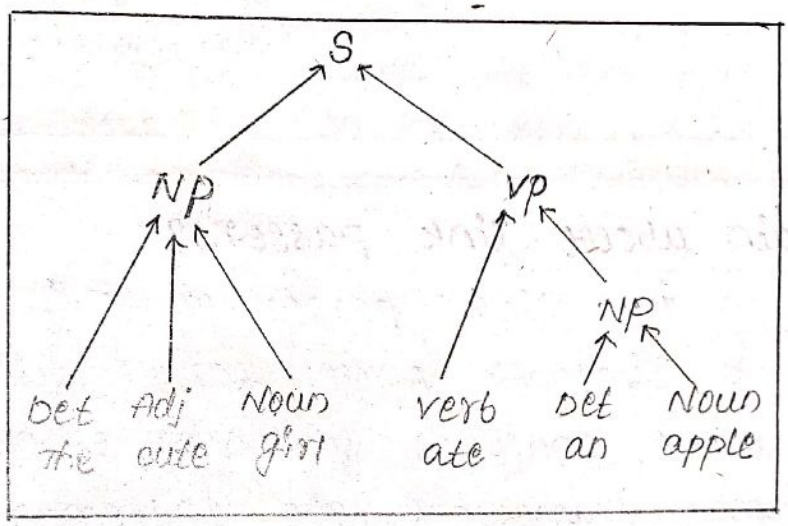
Bottom-up parsing:-

Let us consider the following sentence: The cute girl ate an apple and see its parse structure. The direction of the arrow shows the way in which the parse structure is made. Here the syntactic category to each word in the sentence is assigned before starting the actual parsing. The application of rules is quite clear from the parse

Scanned with CamScanner

structure and needs no Explanation.

Look-ahead parser is basically a bottom-up parser, where the parser when deciding how to interpret input word is allowed to look at the next k input items before making its decision for an appropriate rule.

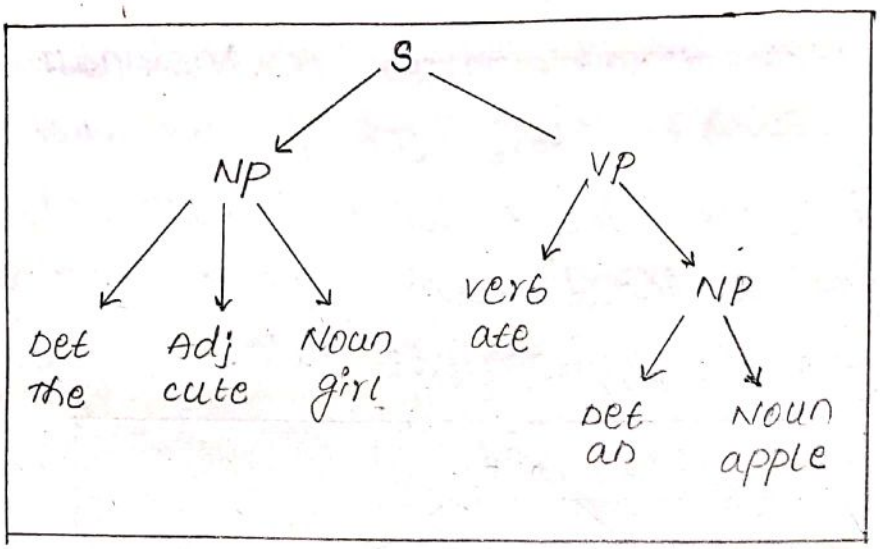


Top-Down parsing:-

To parse a sentence using top-down approach, we start from the start symbol S and apply grammar rules in the forward direction until the terminal symbols of the parse tree corresponds to the words in the sentence.

This process may require backtracking for trying alternative rules if matching with the words of the sentence fail.

Scanned with CamScanner



134) Explain about link passers?

The Link parser or Link grammar parser is a syntactic parser for English language which analyzes sentences using the Link grammar framework. The system assigns to a sentence a syntactic structure, which consists of a set of labelled links connecting pair of words. The parser also produces a 'constituent' representation of a sentence.

Link grammar is a theory proposed by Dary Temperley and Daniel Sleator in 1993 that builds relations between pairs of words, instead of constructing constituents in a tree kind hierarchy. In this representation, there are two basic parameters : directionality and distance.

- Basic rules for an SVO type language are given with simple labels as sub for subject, Det for determiner, obj for object

SVO Language

Link Grammar Rules	Interpretation of Rules
<Determiner> : Det+;	Determiner is defined as label Det connected to word to its right in a distance
<Noun-sub> : {Det-} & sub+;	subject Noun is defined as label sub connected to word on its right in a sentence and connected to label Det on left in a sentence.
<Noun-obj> : {Det-} & obj-;	object noun is defined as the last word in a sentence and labeled as obj is connected to label Det on left in a sentence.
<verb>: sub- & {obj+};	verb is defined as a word connected to label obj on right in a sentence and connected to label sub on left.

135) Explain about chart parsers?

chart parsers store parses of intermediate constituents to be reused while trying alternative parsing path. chart parsers store intermediate constituents that can be reused at the time of backtracking in bottom up parsing approach. A data structure called chart is maintained to keep a record of state of a bottom up parse ~~trans~~ traversed.

Scanned with CamScanner

This structure is a record of the positions of the words and new structure derived from the sentence. chart also maintains the record of rules that have been applied previously but are not complete. These rules are recorded as active arcs on the chart. parsing using chart parsing method is shown below. Let us consider CFG to be used for parsing the sentences.

Rules	Rule. NO	Dictionary words.
<S> → <NP> <VP>	1	<Det> → a/the/an
<NP> → <Det> <Noun>	2	<Noun> → girl/apple/song
<NP> → <Det> <Adj> <Noun>	3	<Adj> → cute/smart
<NP> → <Adj> <Noun>	4	<verb> → sings/ate
<VP> → <verb>	5	
<VP> → <verb> <NP>	6	

Grammar for a subset of English language

The chart parsing of the sentence. "The cute girl sings a song is shown below.

⊙ The words in the sentence are numbered with corresponding positions.

$_1$The $_2$cute $_3$girl $_4$sings $_5$a $_6$song $_7$.

Scanned with CamScanner

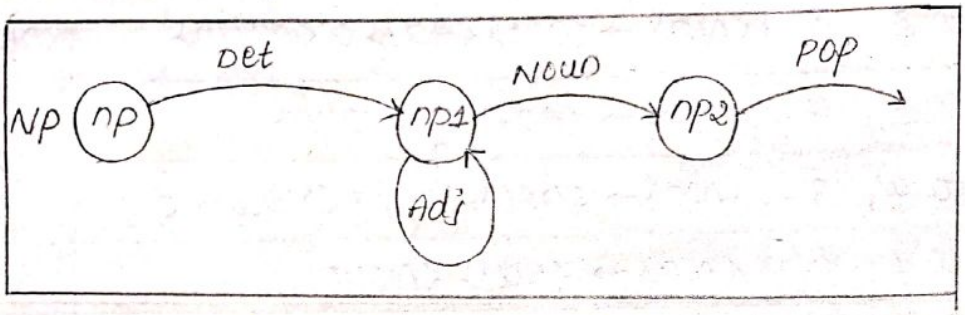
Token	Active arc	Rule No.	Status of Rule processing. '0' indicates position till last parse	Completed constituents
		1	<S> → 0 <NP> <VP>	
the	1 to 2	2	<NP> → <Det> 0 <Noun>	<Det>
	1 to 2	3	<NP> → <Det> 0 <Adj> <Noun>	<Det>
cute	1 to 3	3	<NP> → <Det> <Adj> 0 <Noun>	<Det> <Adj>
	2 to 3	4	<NP> → <Adj> 0 <Noun>	<Adj>
girl	1 to 4	3	<NP> → <Det> <Adj> <Noun> 0	<NP>
	2 to 4	4	<NP> → <Adj> <Noun> 0	<NP>
	1 to 4	1	<S> → <NP> 0 <VP>	<NP>
sings	4 to 5	5	<VP> → <verb> 0	<verb>
	4 to 5	6	<VP> → <verb> 0 <NP>	<verb>
a	5 to 6	2	<NP> → <Det> 0 <Noun>	<Det>
	5 to 6	3	<NP> → <Det> 0 <Adj> <Noun>	<Det>
song	6 to 7	2	<NP> → <Det> <Noun> 0	<NP>
	4 to 7	6	<VP> → <verb> <NP> 0	<VP>
	1 to 7	1	<S> → <NP> <VP> 0	<S>

18) Explain about ~~single~~ simple transition Network?

Transition Network is more convenient for visualizing the grammar. It consists of nodes and labeled arcs. parsing starts with the start node and traversal takes place through arcs. If the current word in the sentence is in the category on that arc then move to the next state in TN and continue the process

Scanned with CamScanner

till pop arc is reached. If there are no words left in the sentence at pop arc then it is assumed to be correctly parsed. Let us draw a network for NP. It consists of three nodes and arcs labeled by word categories. Det, Noun, Adj.

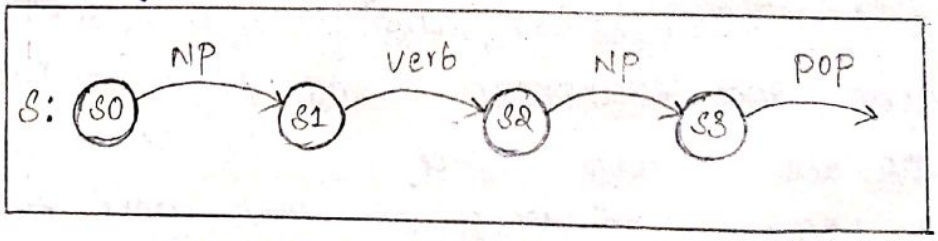


Transition Network

Simple transition Network formalism is not powerful enough to describe all languages that can be described by CFG.

18) Explain about Recursive Transition Network?

To get the descriptive power of CFG, you need a notion of recursion in network grammar. In RTN which is similar to a simple transition Network Except that it also allows arc labels that refer to other networks rather than word categories. The network for simple English sentences can be shown as.



Recursive Transition Network.

Scanned with CamScanner

Labels referring to another network are written in uppercase letters such as NP to differentiate it from category word that stars with capital letter such as 'verb' on the arcs of RTN. The arc from state S0 to state S1 can be followed only if the network for NP is successfully traversed to a pop arc in NP network. RTN may allow an arc labelled with its own name also. Any language represented by CFG can be represented by RTN and vice versa. Thus, they are equivalent in their generative capacity.

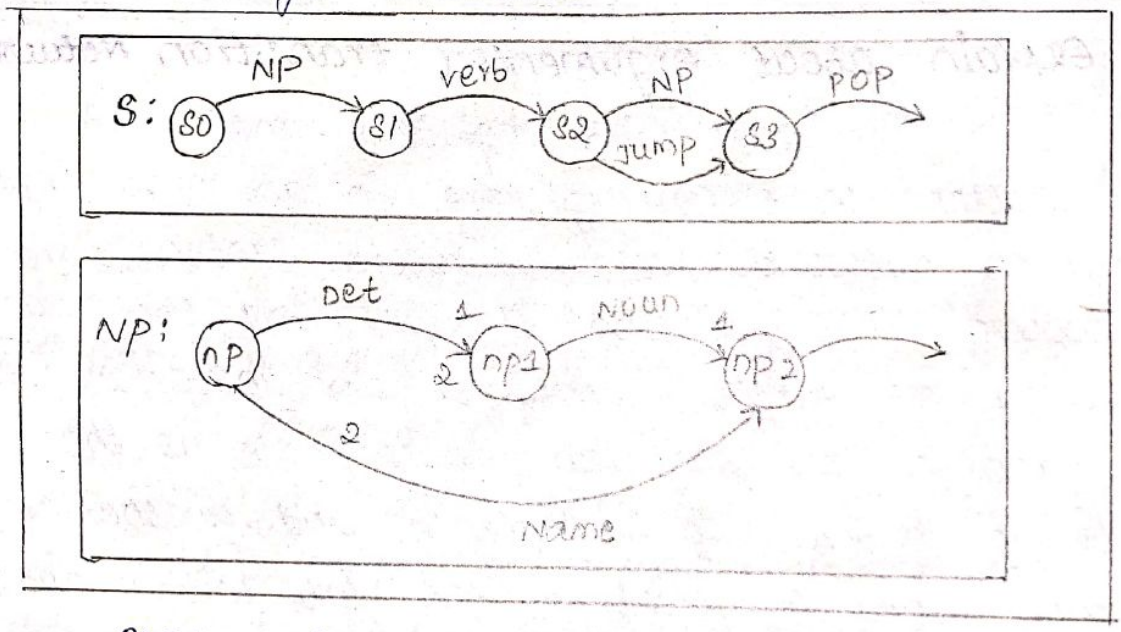
188) Explain about Argumented transition Network

RTN with registers and tests with actions on these registers in an Argumented Transition Network. We can record sentence structure while parsing through ATN that can be used for further analysis. For instance, we can identify one particular noun phase as the syntactic subject of a sentence and another NP as the syntactic object of the verb. Within noun phrase we might identify the determiner, adjective, head noun, and so on. Thus the sentence 'Jack found a bag' when parsed using ATN may produce following structure.

```
(S    (    SUBJ      (NP NAME     Jack)
           MAIN-V    found
           TENSE     PAST
           OBJ       (NP  DET  a
                          HEAD bag
           )                 )
)
```

Scanned with CamScanner

such a structure is created using ATN parser by allowing each network to have a set of registers. The registers are local to each network registers can be set to the values while parsing and these values are retrieved from registers. Thus, each time, a new network is pushed, a new set of empty registers are created. When the network is popped, the value collected in the registers is returned to the calling network and then these registers are no longer available



Argumented Transition Network

- NUM* – is the NUM register of structure in *
- NUM_{SUBJ} - is the NUB register of structure in $SUBJ
- The values of registers are often viewed as sets, and intersection, union of sets are allowed to combine the values of different registers.

Scanned with CamScanner

For the registers that may take a list of values, an append function is permitted.

⊙ Append (ADJS, *) returns the list of adjectives.

139) Explain about Definite clause grammar?

CFG is extended and argumented to another type of grammar called Definite clause grammar. parsing using DCG is one of the important applications of prolog and Logic programming. The DCG formalism is essentially independent of prolog and it would be possible to write compiler or interpreter for it in any programming language which permits the unification of arguments, but, parser for DCG can be easily implemented using prolog. The grammar rules in DCG consists of a left hand side corresponding to the particular part of ordinary prolog rules and right hand side corresponds to the condition part or body of prolog rule. The two sides of DCG rules are separated by an arrow symbol -->. In many prolog systems, DCG rules are transferred automatically into equivalent prolog rules by adding two difference arguments ~~DCG rules are transferred automatically into equivalent prolog~~ DCG grammar can produce parse structure and semantic representation also of a sentence. It is difficult to capture pragmatic and discourse information using DCG grammar.

Scanned with CamScanner

DCG Rules

```
sentence --> np, vp.
np       --> det, noun.
vp       --> verb, np.
vp       --> verb.
det      --> [a].
det      --> [an].
det      --> [the].
noun     --> [boy].
noun     --> [girl].
noun     --> [song].
noun     --> [apple].
verb     --> [sings].
verb     --> [Eats].
```

The DCG rules are translated into prolog rules by adding two difference - list arguments. The concept of difference argument is that there is a list between the differences of two lists.

The interpretation of this rule is that there is a sentence between the difference of two lists x and y. Here, the sentences is represented as a list of symbols.

(40) Elaborate on the construct of a parse structure?

For getting a parse structure after parsing a sentence using DCG, we incorporate an argument that will be unified with the entire tree structure.

Scanned with CamScanner

The complete sentence parse structure is made up of parse structures. is made up of parse structures of its constituents. Each word should have its corresponding parse encoded in lexicon. Let us see the method of constructing such structures. parse structures of sentence, noun and verb phrases are constructed as follows:

- P = sentence (NP, VP), where NP and VP are the parse structures of noun phrase and verb phrase respectively, P is representing parse structure of complete sentence.
- NP = np(D, N), Where D and N are the parse structures of determiner and noun.
- VP = vp(V, NP) Where V and P are the parse structures of verb and noun phrase.

DCG for producing parse structure.

DCG RULES		Rule Number
sentence (P)	--> sentence (N, P)	1
sentence (N, sentence (NP, VP)	--> np (N, NP), vp(N, VP)	2
np (N, np (D, W))	--> det(N, D), noun(N, W)	3
VP (N, vp (V, NP1))	--> verb(N, V), np(N, NP1).	4
det (singular, det(a))	--> verb(N, V)	5
det (singular, det(an))	--> [a]	6
det (-, det(the))	--> [an]	7
noun (singular, noun (girl))	--> [the]	8
noun (plural, noun (girls))	--> [girl]	9
noun (singular, noun (apple))	--> [girls]	10
verb (singular, verb(Eats))	--> [apple]	11
verb (plural, verb(Eat))	--> [Eats]	12
	--> [Eat]	13

Scanned with CamScanner

141) What are the approaches for creating semantic representation?

There are several approaches for creating semantic representation: they are

- Semantic Grammars.
- case grammar
- conceptual parser
- compositional semantic interpretation.

142) Explain semantic grammars?

combining syntactic and semantic processing involves collapsing them into a single uniform frame work. this results in semantic grammar that is similar to context-free grammar incorporating semantic information in the syntactic rules. semantic rules look like regular grammar except that it uses semantic categories for terminal symbols. consider the following simple syntactic grammar represented as context free like grammar. Here, the category symbol enclosed in ( ) indicates optional occurrence of category.

<S>	→	<NP> <VP>
<NP>	→	(<Det>) <Noun> / <Name>
<VP>	→	<verb> <NP> (<PP>)
<PP>	→	<prep> <NP>
<Det>	→	a / the / an
<Noun>	→	bus / train / man / ticket
<Name>	→	John / Mike / Delhi / Mumbai
<verb>	→	books / reserves / leaves / departs
<prep>	→	to / for / from

Scanned with CamScanner

If we modify the grammar given using semantic semantic information, then syntactically incorrect sentences could not be parsed. We clearly see that semantic grammar tends to be larger in size as compared to CFG and hence the efficiency of the parse degrades.

<S>	→ <NP-A> <VP-Res>
<S>	→ <NP-vehicle> <VP-Dep>
<NP-A>	→ (<Det>) <Noun-A> / <P-Name>
<NP-vehicle>	→ (<Det>) <Noun-vehicle> (<PP>)
<VP-Res>	→ <verb-Res> <NP-vehicle>
<VP-Res>	→ <verb-Res> <NP-NonA> for <NP-vehicle>
<VP-Dep>	→ <verb-Dep> <PP>
<PP>	→ to / from <Loc>
<Det>	→ a / the / man
<Noun-A>	→ man
<P-Name>	→ John / Mike
<Noun-NonA>	→ ticket
<Noun-vehicle>	→ bus / train / flight
<Loc>	→ Delhi / Mumbai
<verb-Res>	→ books / reserves
<verb-Dep>	→ leaves / departs

14) Explain semantic grammar using DCG formalism?

Using DCG formalism, we can get semantics in the form of first order predicate logic. For this, we have to include one more argument in 'sentence' predicate which captures

Scanned with CamScanner

the semantic structure called logical form of a sentence. We will omit other arguments such as number and phrase structure arguments discussed earlier for the sake of simplicity and will include only those arguments which are related to logical forms.

Example :-

- John is a man - man(John)
- Mary is a girl - girl (Mary)

Let us write simple DCG grammar for handling sentences of format 'noun is a noun'

DCG generating LF.

DCG Rules	Rule NO.
sentence (LF) -> np(N), vp(V), {LF = V(N)},	1
np (N) -> noun (N)	2
np (N) -> det, noun (N)	3
vp(V) -> aux-verb, np(V)	4
det -> [a]	5
aux-verb -> [is]	6
noun(man) -> [man]	7
noun(girl) -> [girl]	8
noun(crow) -> [crow]	9
noun(bird) -> [bird]	10
noun(John) -> [John]	11
noun(mary) -> [mary]	12
	13

Scanned with CamScanner

144) Explain about case frame representation for machine translation with Example?

Let us describe the case frame structure used in this system. various standard cases called main cases along with some more cases called Attribute case have been used. The attribute cases keep surface structure information.

Description of Each case:

Agent case: Noun phrase of subject part of sentence gives information related to agent case.

Example:

In the sentence "A very charming tall son of beautiful mary is Eating food" the agent case Entries of case structure are as follows.

AGENT : SON
AGENT ADJECTIVES: charming, tall
AGENT DETERMINER: a
AGENT QUANTIFIER: very
AGENT POSSESSOR : mary
AGENT POSSESSOR ADJECTIVE: beautiful.

Object case: The object case describes the object on which the agent is performing an action

Example: The object case for the sentence "I broke the beautiful ball of John's sister.

OBJECT: ball
OBJECT ADJECTIVE: beautiful
OBJECT DETERMINER: the
OBJECT POSSESSOR: sister
OBJECT POSSESSOR POSSESSOR: John

Scanned with CamScanner

Dative case: The dative and the object have a subtle difference between them. The dative is an animate whereas object is non animate noun on which the action is being performed

Example:
I kill the lonely dog
DATIVE : dog
Object ADJECTIVE : lonely
O.D : the

Instrument case: Instrument describes a tool material using which a particular action has been performed by an agent. The instrument is generally non-animate noun.

Example: Let the apple with her long knife
INSTRUMENT : Knife
I.A : long
I.P : her

Source case: The source case holds the place from ~~whether~~ where the actor moves

Example:
the beautiful office of Mike
SOURCE : office
S.O : the
S.P : Mike
S.A : beautiful.

Goal case: The goal case is the place to which the agent is moving.

Scanned with CamScanner

GOAL : Market
G.D : the
G.P : Delhi
G.A : big.

Beneficiary case: An animate noun who is the beneficiary of an action fills the beneficiary case.

Example:

cute and smart John.

BENEFICIARY : John

B.A : cute and smart.

145) Explain about Universal Networking language?

Project on Universal Networking language was initiated by Institute of Advanced Studies Established in 1995 by United Nations University. The institute INSA is an advanced research and Education institute with flexible and multi programs focusing interaction on social and natural systems. The UNL project started in 1996 and first version was proposed in 1999.

UNL is an electronic language that provides a uniform representation of a natural language. It intermediate language which helps understanding among different natural languages and is represented in the form of logical

Scanned with CamScanner

Expressions.

The objective of UNL is generic, language neutral formalism is systematic and coordi-nated way. UNL is a controlled language which has predefined vocabulary, building rules in an Electronic content mark up language. It basically provides a platform for language neutral mark-up language. Therefore the main objectives of UNL are to provide.

- Support for multilingualism
- concept oriented backbone derived from natural language communicative Elements.

Applications such as Machine translation System, Multilingual information service, Information retrieval system, UNL based search Engine. Expert system etc can be developed using UNL formalism.

UNL consists of vocabulary, syntax and semantics.

Scanned with CamScanner