# Table of Contents

CHAPTER ONE: Introduction to AI ..... 1-11

CHAPTER TWO: Problem Solving ..... 12-32

CHAPTER THREE: Search Techniques ..... 33-58

CHAPTER FOUR: Knowledge Representation, Inference and Meaning..... 59-99

CHAPTER FIVE: Structured Knowledge Representation.....100-115

CHAPTER SIX: Machine Learning .....116-143

CHAPTER SEVEN: Applications of AI .....144-207

Bibliography ..... 208

# Chapter ONE

# Introduction to AI

## Intelligence

The capacity for logic, understanding, self- awareness, learning, creativity and problem solving is known as intelligence. It can be described as the ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors within an environment or context.

Intelligence is most often studied in humans but has also been observed in both non-human animals and in plants. Intelligence in machine is called artificial intelligence, which is commonly implemented in computer system using programs and sometimes, specialized hardware.

## Artificial Intelligence

AI is branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.

## AI approaches

1. **Acting Humanly (Turing test)**

   This is a problem that has greatly troubled AI researchers for years. They ask the question "when can we count a machine being intelligence?" The most famous response is attributed to Alan Turing, a British mathematician and computing pioneer. The famous "Turing Test" was named after him. Till date, no program has yet passed Turing test

   **Factors to pass Turing Test**

   1. Speech Recognition
   2. Natural Language understanding and Generation
   3. Speech synthesis
   4. Knowledge representation
   5. Learning
   6. Automated reasoning and decision making

This test involves an interrogator who interacts with one human and one machine within a given time the interrogator how to find out which of the two the human is, and which one the machine.

**Critics of Turing Test**

1. Test is not reproducible, amenable or constructive to mathematical analysis as it is more important to study the underlined principle of intelligence than to duplicate example.
2. Trying to evaluate machine intelligence in terms of human intelligence is fundamental mistake. It focuses too much on the behavior of conversation.

**2. Thinking Humanly**

Thinking humanly means trying to understand and model how the human mind work.

There are (at least) two possible routes that humans use to find the answer to a question:

- We reason about it to find the answer. This is called introspection.
- We conduct experiments to find the answer, drawing upon scientific techniques to conduct controlled experiments and measures change.

The field of cognitive science focuses on modeling how people think.

**Critics:** Lacks scientific theories of internal activities of brain. Level of abstraction of the approach is argue too.

**3. Thinking Rationally**

Trying to understand how we actually think is one route to AI. But another approach is to model how we should think. The "thinking rationally" approach to AI uses symbolic logic to capture the laws of rational thought as symbols that can be manipulated. Reasoning involves manipulating the symbols according to well - defined rules, kind of like algebra. The result is an idealized model of human reasoning. This approach is attractive to theretists i.e. modeling how humans should think and reason in an ideal world.

**Critics:** It is not easy to take informal knowledge and state in the formal terms required by logical notation specially when knowledge is less than 100% certain.

4. **Acting Rationally**

Acting rationally means acting to achieve one's goals, give one's beliefs or misunderstanding about the world. An agent is a system that perceives an environment and acts within last environment. An intelligent agent is one that acts rationally with respect to its goals. For e.g. an agent that is designed to play a game should make moves that increase its chances of winning the game.

Achieving so - called "perfect rationality" making the best decision theoretically possible is not usually possible due to limited resources in a real environment (e.g. time, memory, computational power, uncertainly etc). The trick is to do the best with the information and resource you have.

**History of AI**

1. 1943 - The first work is now recognized as AI was done by Warren McCulloch and Walter Pits in 1943. They purposed a model of artificial neurons.
2. 1949 - Donald Hebb demonstrated an updating rule for modifying the connection strength between neurons. His rule is now called Hebbian learning.
3. 1950 - Alan Turning proposed Turing test.
4. 1955 - Allen Newell and Herbert A.Jimon created "First artificial intelligence program".
5. 1956 - The word "AI" first adopted by American Computer Scientist John McCarthy at the Dartmouth conference.
6. 1966 - The researches emphasized developing algorithms which can solve mathematical problems. Joseph Weizebaum created the first chatbot in 1966, which was named as ELIZA.
7. 1972 - The first intelligent humanoid robot was built in Japan which can named as WABOT-1.
8. 1980 - Expert system was introduced.
9. 1980 - The first national conference of the American Association of Artificial Intelligence was held at Standford University.
10. 1997 - IBM Blue beats world chess champion, Garry Kasparov, and became the first computer to beat a world chess champion.

11. 2002 - For the first time, AI entered the home in form of Roomba Vacuum cleaner.
12. 2006 - Companies like Facebook, Twitter and Netflix also started using AI.
13. 2011 - IBM's Waston Won Jeopardy, a quiz show where it had to solve the complex questions as well as riddles.
14. 2013 - Google has launched an Android app features "Google now" which was able to provide information to the user as a prediction.
15. 2014 - Chatbot " Eugene Goostman" won a competition in the famous "Turing test".
16. 2018 - The "Project Debater" from IBM debated on complex topics with two master debaters and also performed extremely well.

**Importance of AI**

1. It enable human capabilities - understanding, reasoning planning and perception.
2. It helps in analytical tasks like finding patterns clustering.
3. It provides abilities to create new opportunities is most business sectors and consumer applications.
4. It helps human to make complex decision.
5. It helps in fraud detection.

**AI Related Fields**

1. Robotics
2. Search
3. Planning
4. Machine Learning
5. Image Processing
6. Expert systems
7. Natural language processing

E.g: of AI

**ELIZA:** A program that simulated a psychotherapist interacting with a patient and successfully passed the Turing Test.

**Coded at MIT during 1964-1966 by Joel Weizenbaum.**

- First script was DOCTOR
- The script was a simple collection of syntactic patterns not unlike regular expression.
- Each pattern had a associated reply which might include bits of the input (after simple transformation (my → your)
- Weizenbaum was shocked at reactions:
  - Psychiatrists thought it has potential.
  - People unequivocally anthropomorphized.

**Applications of AI**

1. AI in Astronomy
   - can be very useful to solve complex universe problem.
2. AI in Healthcare
   - Helps to make a better and faster diagnosis than human.
3. AI in Gaming
   - AI machine can play games like chess etc.
4. AI in Finance
   - The Finance industry is implementing automation, chat box, adaptive intelligence and machine learning into financial process.
5. AI in Data security
6. AI in Social Media
7. AI in Travel and Transport
8. AI in Automotive Industry
9. AI in Entertainment
10. AI in E-commerce

**Knowledge**

Knowledge is a theoretical or practical understanding of a subject that helps us to solve problems in particular domain, to predict what will happen next and to explain why and how something has happened. Knowledge consists of information that has been interpreted, categorized, applied, experienced and revised. In general knowledge is more than just data, it consists of facts, ideas, beliefs, heuristics, association, rules, abstractions etc.

Knowledge is important for making intelligent agents.

## Learning

Learning denotes changes in a subsystem that enable a system to do the same task more efficiently the next time.

A computer program is said to learn from experience 'E' with respect to some class of tasks 'T' and performance measure 'P' if its performance at task in 'T', as measured by 'P' improve with experience 'E'.

**Old Question**

1. **Define Artificial intelligence. Justify that system that think rationally and act rationally" is a part of Artificial Intelligence.**

**(2076 Baisakh)**

⇒ **1st part:** See on page no. 1

**2nd part:** See on page no. 2 and 3

2. **What is an intelligent agent? How does learning agent work?**

**(2075 Bhadra)**

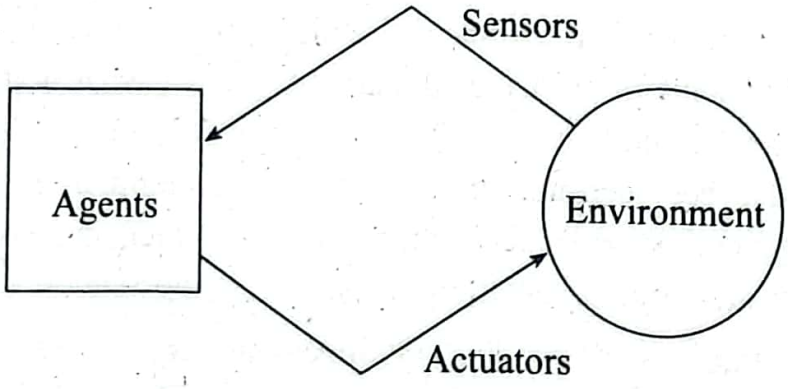
⇒ An intelligent agent is an autonomous entity which act upon an environment using sensors and actuators for achieving goals. An intelligent agent may learn from the environment to achieve their goals. A thermostat is example of an intelligent agent.

**Four main rules intelligent agent:**

1. An AI agent must have the ability to perceive the environment.
2. The observation must be used to make decisions.
3. Decision should result in an action.
4. The action taken by an AI agent must be a rational action.

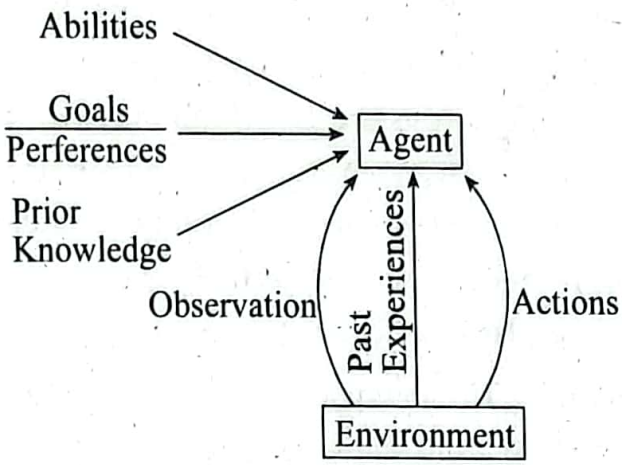
An AI system is composed of learning agent and its environment. The agents acts in their environment. The environment may contain other agent. An agent is anything that can be viewed as:

- Perceiving its environment through sensors and
- Acting upon the environment through actuators.



Architecture is the machinery that the agent executes on. It is a device sensors and actuators, for e.g.: A robotic car, A camera, A PC. Agent program is an implementation of an agent function. An agent function is a map from the percept sequence (history of all that an agent has perceived till date) to an action.

**E.g.:** A Robotic agent has cameras and infrared range finders which act as sensors and various motors acting as actuators.




3. **Define AI. What is the importance of Turing Test in AI? List the applications of AI.** **(2075 Bhadra)**

⇒ **1st part:** See on page no1

**2nd part:** See on old question no 10

**Applications of AI**

See on page no 5

4. **What is rational agent?" System that think like humans" and "system that act like humans" are the part of artificial intelligence. Justify these statement with practical example.**

**(2074 Bhadra)**

⇒ A rational agent is an agent which has clear preference, modus uncertainty, and acts in a way to maximize its performance measure with all possible actions.

**2nd part:** See on page no1

5. **If the Turing Test is passed does this show that computers exhibit intelligence? State your reasons.** **(2073Bhadra)**

⇒ See on page no1 and 2

6. **Define Artificial Intelligence(AI).Describe the importance and practical application of AI.** **(2073 Magh)**

⇒ **1st See on page no1**

**Importance of AI**

1. **Knowledge:** The ability to represent knowledge about the world. For software to possess knowledge, it must understand that: certain entities, facts and situations exist in the world; these entities have properties (including relationships to one another); and these entities and properties can be categorised.

2. **Reasoning:** The ability to solve problems through logical reasoning. To reason is to apply logic to derive beliefs, related ideas and conclusions from information. Reasoning may be deductive (derive specific conclusions from general premises believed to be true), inductive (infer general conclusions from specific premises) or abductive (seek the simplest and most likely explanation for an observation).

3. **Planning:** The ability to set and achieve goals. For software to be able to plan, it must be capable of specifying a future, desirable state of the world and a sequence of actions enabling progress towards it.

4. **Communication:** The ability to understand written and spoken language. To communicate with people, software must have the ability to identify, understand and synthesise written or spoken human language.

5. **Perception:** The ability to make deductions about the world based on sensory input. To perceive, software must be able to organise, identify and interpret visual images, sounds and other sensory inputs.

**Applications:** See on page no

7. **What is an Artificial Intelligence (AI)? Explain any two applications of Ai in real field.** **(2072 Ashwin)**

⇒ **1st part:** See on page no1

**2nd part:** See on old question no 15

8. **If the Turing Test is passed, does this show that computer exhibit intelligence? State your reasons.** **(2072 Magh)**

⇒ **1st part:** See on page no

**2nd part:** See on page no

9. **What is Artificial Intelligence(AI)?Discuss history of AI in brief.** **(2071 Bhadra)**

⇒ **1st part:** See on page no1

**2nd part:** see on page no3

10. **What is the importance of Turing Test in AI?** **(2071 Baisakh)**

⇒ Historically, the Turing Test is "important" because it was one of the first tests conceived that would allow us to gauge our progress in creating intelligent machines. Ask around and you "receive so many different definitions of "intelligence" that it is difficult to satisfy all of them with a test. The underlying belief behind Turing's test was that if a machine could fool a person into believing he/she was conversing with another person, then surely that machine possesses intelligence whatever that is.

11. **Distinguish between Knowledge and Learning? What does acting humanly refer to?** **(2071 Magh)**

⇒

| Knowledge | Learning |
|---|---|
| • Knowledge is the power of understanding of a subject. | • Learning is system according to which you learn from environment. |
| • If the information is stored in long term memory it becomes knowledge. | • Learning is getting information. |

**2nd part:** See on page no

12. **What is Artificial Intelligence (AI)?Discuss brief history of AI with Chronological development.** **(2070 Magh)**

⇒ **1st part:** See on page no1

**2nd part:** See on page no3

13. **Define AI. When a machine is said to be passed Turing test?** **(2070 Bhadra)**

⇒ **1st part:** See on page no1

**2nd part:** See on page no1 and 2

**14. Define Artificial Intelligence (AI)? Justify that "AI makes machine working more human friendly". (2070 Chaitra)**

⇒ **1st part:** See on page no1

Since robots tend to be strong, but rather stupid machines, it is common, and even prescribed by law, to position robots safely behind bars. However, there are more and more areas where robots operate in a human environment, as personal assistants like Siri and Alexa, care takers like Paro and Lea but also 'bots' for sensing and surveillance or as the heart of autonomous or self-driving cars like the Wepod, Waymo and Tesla's Autopilot. We are already surrounded by more or less autonomous systems but when these are provided with movement and autonomous decision-making, it is time to raise the question on how to design Human-Robot Systems so that they fit, physically, socially and otherwise into a human environment.

**15. Discuss any two fields of your daily life where artificial intelligence has been applied. (2069 Bhadra)**

⇒ **1. Artificial Intelligence in eCommerce:**

Artificial Intelligence technology provides a competitive edge to e-commerce businesses and is becoming readily available to companies of any size or budget. Leveraging machine learning, AI software automatically tags, organizes and visually searches content by labeling features of the image or video.

AI is enabling shoppers to discover associated products whether it is size, color, shape, or even brand. The visual capabilities AI is improving every year. By first obtaining visual cues from the uploaded imagery, the software can successfully assist the customer in finding the product they desire. Many e-commerce retailers are already becoming more sophisticated with their AI capabilities, and I only expect this to grow in the future.

**2 AI in Healthcare:**

In the year ahead, and particularly in the next five to ten years, artificial intelligence is going to have a big impact on the healthcare industry and the ways in which healthcare related companies utilize AI. Here is a short note from Dr. Jeff Dunn, CEO of Redivus Health. Redivus Health is a transformative mobile app used by healthcare providers to prevent medical errors by offering both clinical decision support during critical medical events as well as documenting those events electronically in real time.

AI presents opportunities for our application to take the data we have gathered from patients and be able to clinically innovate to improve patient outcomes to an even greater extent. AI improves reliability, predictability, and consistency with quality and patient safety. For us, AI, as applied to software, is used as a decision augmentation tool, but it should not have free reign without human interaction and guidance. While it can't replace doctors and nurses, it can make them more effective, efficient and happier on the job as it takes the cognitive burden off our providers – which increases confidence as well as reduces stress and anxiety.

⌘⌘⌘

Chapter TWO

# Problem Solving

**Problem solving refers to a intelligence which compromises a number of techniques such as algorithms, heuristics to solve a problem.**

**Steps performed by Problem - Solving Agent**

(i) **Goal Formation:** Helps to organize behaviour by isolating and representing the task knowledge necessary to solve problem.

(ii) **Problem Formulation:** Define the problem precisely with initial states, final state and acceptable solutions.

(iii) **Searching:** Find the most appropriate techniques 0 sequence among all possible techniques.

(iv) **Execution:** Once the search algorithm returns a solution to the problem, the solution is then executed by the agent.

**State Representation**

A state space essentially consists of set of nodes representing each state of the problem, acrs between node representing the legal moves from one state to another an initial state and a goal state.

**A problem can be defined by**

(i) Initial state

(ii) Actions (using success or function)

(iii) Goal test (to determine the goal test)

(iv) Path cost

A problem is when defined with these components is called well defined problem.

**For e.g.**

**The Eight tile problem Formulation:**

The eight tile puzzle consist of 3 by 3(3*3) square frame board which holds 8 movable tiles numbered 1 to 8. One square is empty allowing the adjacent tiles to be shifted. The objective of the puzzle is to find a sequences of tile movements that lead from a starting configuration to a goal configuration.

| | | |
|---|---|---|
| 3 | B | 1 |
| 6 | 2 | 5 |
| | 4 | 4 |

To be tranformed →

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 8 | | 4 |
| 7 | 6 | 5 |

The states of 8 tile puzzle are the different permutations of the tiles within the frame. Let's do a standard formulation of this problem now.

**States:** It specifies the location of each of the 8 tiles and the blank in one of the nice squares.

**Initial states:** Any state can be designated as initial state.

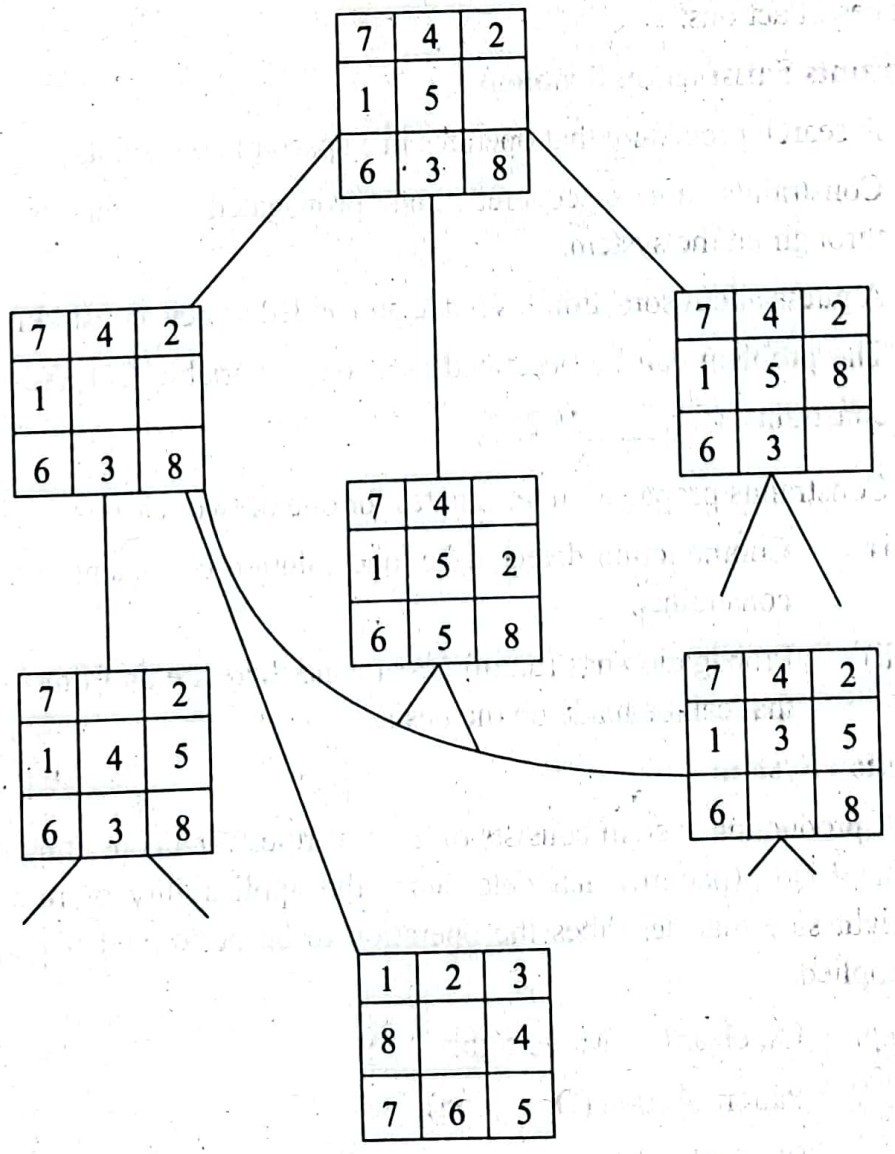
**Goal:** Many goal configurations are possible one such is shown in the figure.

**Legal moves (or state):** They generate legal states that result from trying the four actions.

- Blank moves left
- Blank moves right
- Blank moves up
- Blank moves down

**Path cost:** Each steps costs 1, so the path cost is the number of steps in the path.

**The tree diagram showing the search space is shown in figure.**



**Problem Classification**

(i) **Ignorable:** Intermediate actions can be ignored e.g: water - jug problem.

(ii) **Recoverable:** The actions can be implemented to go the initial state. e.g. - 8 puzzle game.

(iii) **Irrecoverable:** The actions cannot help to reach the precious state. e.g. tic-tac-toe.

(iv) **Decomposable:** The Problem can be broken into similar one e.g. Multiplayer game.

**Well - Defined Problems:**

A well defined problem is a problem in which the initial state or starting position, the allowable operations and the goal state are clearly defined.

A well - defined problem can be described by initial state. Operator or successor function. For any state X returns (X), the set of states reachable from X with one action. State - Space - all states reachable from initial by any sequences of actions.

**Constraints Satisfaction Problem**

- A search procedure that operates in a space of constraints.
- Constraints are discovered and propagated as far as possible throughout the system.
- A guess about something is made and added as new constraint.
- The problem can be described as a set of variable (X1, X2, ___) and constraints (C1, _____, C2).
- Constraints propagation terminates for one of two reasons.
  - i) Contradiction detected i.e. one solution consistent with known constraints.
  - ii) Propagation has run off stream and there are on further changes that can be made on the basis.

**Production System**

- A production system consists of a set of rules each consisting of a left hand side (pattern) that determines the applicability of rules and a right side that describes the operation to be performed if the rule is applied.

  e.g. [A, clean] ⇒ Move right

  Pattern Action (Operation)

  [A, dirt] ⇒ clean

- A production system may have one or more knowledge base that contain whatever information is appropriate for the particular.
- A control strategy that specifies the order in which the rules will be selected and a way of resolving the conflicts that arise when several rules matched at once i.e. it must have a rule applier for conflict resolution.
- The first requirement of a good control strategy is that it must cause motion.
- The second requirement of a good strategy is that it should have systematic.

**Game Playing**

Game Playing is an important domain of artificial intelligence. Games don't require much knowledge, the only knowledge we need to provide is the rules, legal moves and the conditions of winning or losing the game. Both players try to win the game. So, both of them try to make he best more possible at each turn. Searching Techniques like BFS (Breadth First Search) are not accurate for this as the branching factor s very high, So searching will take a lot of time. So, we need another search procedures that improve.

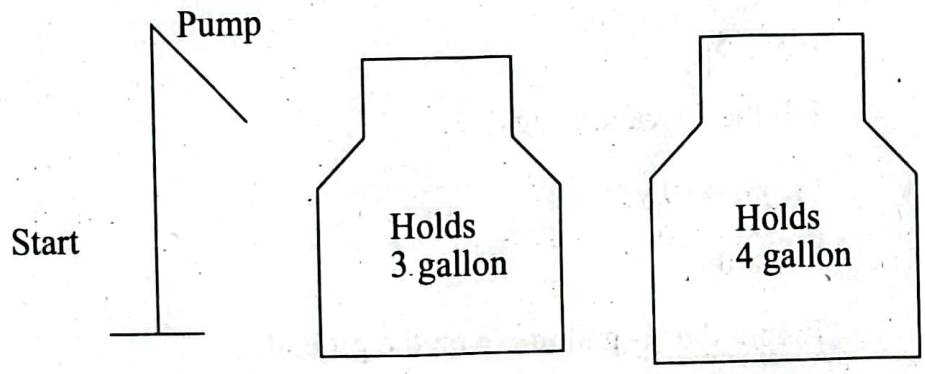
- Generate procedure so that only good moves are generated.
- Test procedures so that the best move can be explored first.

The most common search techniques is game playing is Min Max search techniques. It is depth - first depth - limited search procedure. It is used programs like chess and tic - tac - toe.

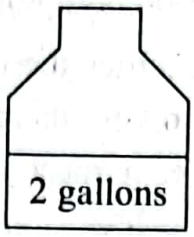
Well defined problems examples:

i) **Water Jug Problem**

You are given two Jugs, a 4 - gallon one and a 3 - gallon, and a pump which has unlimited water. How can you get exactly 2 gallons of water in the 4 - gallon jug?



**Goal:**




- **State Representation:** (x, y)
  - : contents of four gallon, $0 \leq x \leq 4$
  - : contents of three gallon , $0 \leq y \leq 3$
- **Initial state:** (0, 0)
- **Goal state:** (2, y) where $0 \leq y \leq 3$
- **Operators:**
  - Fill 3 - gallon from pump, Fill 4 - gallon from pump.
  - Fill 3 - gallon from 4 - gallon, Fill 4 - gallon from 3 - gallon.
  - Empty 3 - gallon into 4 - gallon, empty 4 - gallon into 3 - gallon.
  - Dump 3 - gallon down drain, dump 4 - gallon down drain.

**Production Rules for the water Jug Problem:**

1. $(x, y) \rightarrow (4, y)$

   If $x < 4$

   Fill the 4 - gallon jug.

2. $(x, y) \rightarrow (x,3)$

   If $y < 3$

   Fill the 3 - gallon Jug.

3. $(x, y) \rightarrow (0,y)$

   If $x > 0$

   Empty the 3 - gallon jug on the ground.

4. $(x, y) \rightarrow (x, 0)$

   If $y > 0$

   Empty the 3 - gallon jug on the ground.

5. $(x, y) \rightarrow (4, y-(4-x))$

   If $x + y \geq 4$ and $y > 0$

   Pour water from the 3 - gallon jug into the 4 - gallon jug until the 4 - gallon jug is full.

6. $(x, y) \rightarrow (x-(3-y), 3)$

   If $x + y \geq 3$ and $x > 0$

   Pour water from the 4 - gallon jug into the 3 - gallon jug until the 3 - gallon jug is full.

7. $(x, y) \rightarrow (0, x + y)$

   If $x + y \leq 4$ and $y > 0$

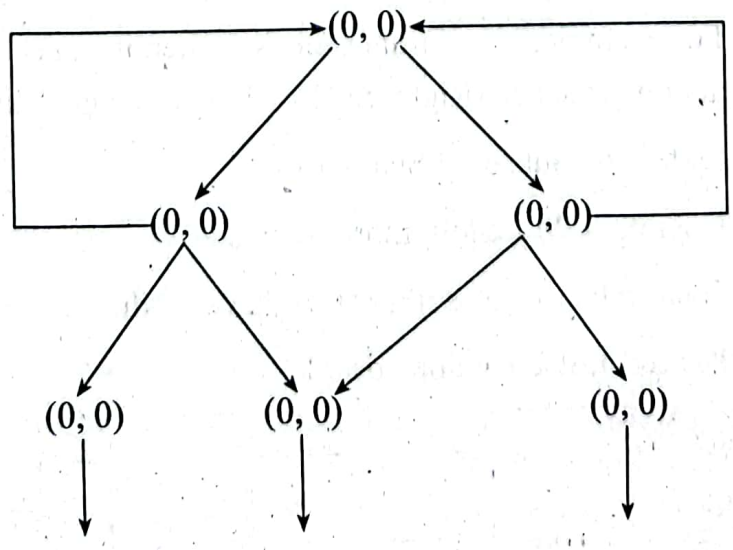
   Pour all the water from the 3-gallon jug into the 4-gallon jug.

8. $(x, y) \rightarrow (0, x + y)$

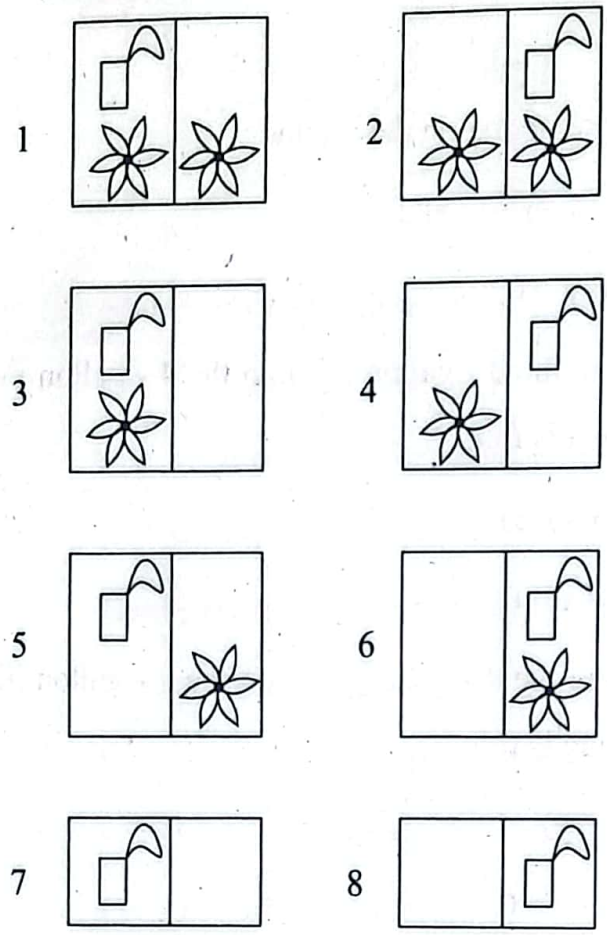
   If $x + y \leq 3$ and $x > 0$

   Pour all the water from the 4 - gallon jug into the 3 - gallon jug.

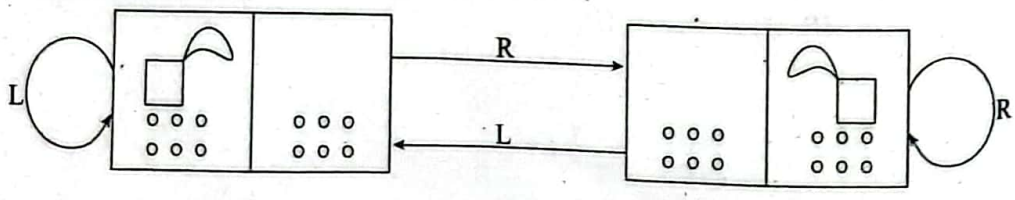
Search tree for water jug problem.

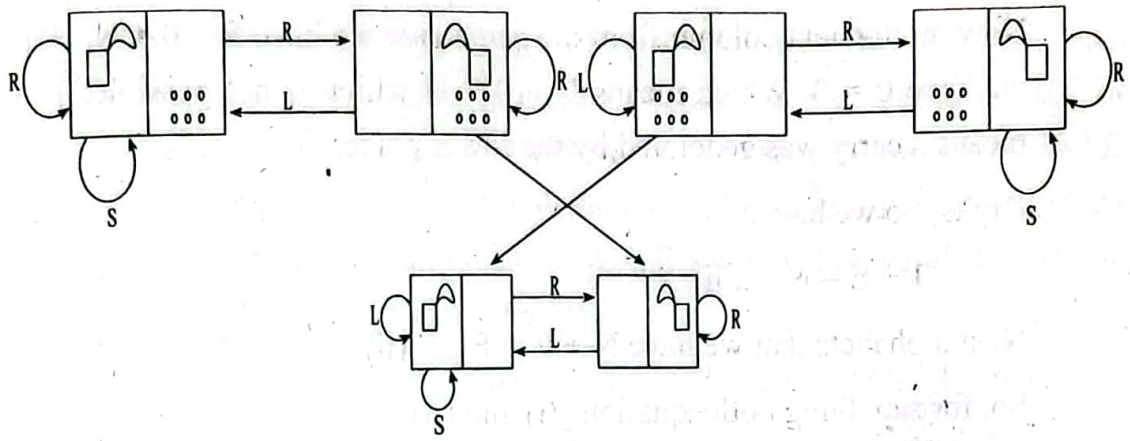


## ii) Vacuum World Problem




- We assume that the agent knows its location and the locations of all the pieces of dirt, and that the suction is still in good worker order.
- There are 8 possible world states.
- The agent ahs three possible actions in this version of the vacuum world: Left, Right and Suck.
- The goal is to clean up all the dirt.
- For example, if its initial state is 5, then it can calculate that the action sequence [Right, Suck] will get to a goal state.
- State sets: Subset of States 1 to 8.
- Operators: More left, more right, suck.
- Goal test: All states in state set have no dirt.
- Path cost: Each action costs 1.






**Examples of constraints satisfaction Problem**

1. Crypt - Arithmetic
2. Map colouring: Given a map, colour three regions in blue, red and black, such that no two neighbouring regions have the same colour.

**Crypt Arithmetic Problem**

The crypt arithmetic problem in Artificial Intelligence is a type of encryption problem in which the written message in an alphabetical from which is easily readable and understandable is converted into a numeric from which is neither easily readable nor understandable. In simpler works, the crypt arithmetic problem deals with the converting of the non - readable cipher text. The constraints which this problem follows during the conversion is as follows:

1. A number 0 - 9 is assigned to a particular alphabet.
2. Each different alphabet has a unique number.
3. All the same, alphabets have the same numbers.
4. The numbers should satisfy all the operations that any normal number does.

e.g.

i) **SEND + MORE = MONEY**

⇒ We first look for the MSB in the last word which is M in the world 'MONEY' here. It the letter which is generated by carrying. So carry generated can be only one. So, we have M = 1.

Now, we have S + M = 0 in the second column from the left side. Here M = 1. Therefore, we have S + 1 = 0. So, we need a number for such that it generates a carry when added with 1. And such a number is 9. Therefore, we have S = 9 and 0 = 0.

Now, in the next column from the same side we have E + 0 = N. Here we have 0 = 0. Which means E + 0 = N which is not possible. This means a carry was generated by the lower place.

Digits, So we have:

1 + E = N..... ii)

Next alphabets that we have N + R = E ..... (ii)

So, for satisfying both equations (i) and (ii)

We get E = 5 and N = 6.

Now, R should be 9, but 9 is already assigned to S. So, R = 8 and we have 1 as a carry which is generated from the lower place digits.

Now, we have D + 5 = Y and this should generate a carry. Therefore, D should be greater than 4. As 5, 6, 8 and 9 are already assigned, we have D = 7and therefore Y = 2.

Therefore, the solution to the given crypt- Arithmetic problem is.

S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2

```
  9 5 6 7
+ 1 0 8 5
---------
1 0 6 5 2
```

**ii)**

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

Since the result is one digit more than the numbers, it is quite obvious that there is a carry over and therefore G must be equal to 1. Now consider the part.

```
  S E
+ L L
-----
  E S
```

This suggests that S + L = E or 10 + E (with 1 as a carry) and E + L= S or 10 + S. Now this happens only when L= 5 and S ~ E (difference b/w S and E) = 5.

This gives us paired values of (0, 5), (1, 6), (2, 7), (3, 8) and (4, 9) as possible values for (E, S) or (S, E). But out of these (0, 5) and (1, 6) cannot be accepted as G = 1 and L = 5.

The first rule of cryptic equations like this is that different letter cannot have the same numerical value. So we are left with the possibilities of (2, 7), (3, 8) and (4, 9). We can also infer that S and E, E is the smaller value and S is the larger, because if F were larger, we would have a carry and then S + L = E would not be valid. This means that S + L = E has a carryover of 1.

Let these values remain for now, we shall come back to them later.

Now coming to the part

```
  B A
+ B A
-----
G A M
```

Since there is a carry obtained here, we can infer that B must be greater than or equal to 5. But, L = 5, therefore B > 5. Now we have two cases, If the sum A + A produces a carry, the A is odd, else A is even. Since there is carry over from S + L = E

M has to be an odd number.

Now we have gathered all the information we can and there is nothing else to do. To proceed we shall have to sue the trial and error method substituting values for the letters keeping all the above points in mind.

Let us assure E = 2 and S = 7 and B = 6. So we have,

```
  1
  6 A 7 2
+ 6 A 5 5
---------
1 A M 2 7
```

Now A can be either 2 or 3 depending on whether we have a carry from A + A or not. But, since E = 2, that means A must be 3 and therefore there is a carry. But, replacing the other A's in the equation with 2's gives us two contradiction. Firstly M shall become equal to 7 (S is already equal to 7) and A + A does not produce a carry. Therefore, our assumptions were wrong and we will have to try again for different values.

Now let us try for E = 3 and S = 8 and B = 7. We have,

```
  1
  7 A 8 3
+ 7 A 5 5
---------
1 A M 3 8
```

This gives us A = 4 or 5 based on whether there is a carry or not, but since already L = 5, must be equal to 4, therefore M = 9. We have obtained values for all unknowns without any contradictions and hence this is the solution.

Therefore,

G = 1 A = 4 B = 7 M = 9

E = 3 L = 5 S = 8

Hence,

```
  1
  7 4 8 3
+ 7 4 5 5
---------
1 4 9 3 8
```

**Some Practice Questions**

1. RIGHT + RIGHT = WRONG
2. WRONG + WRONG = RIGHT
3. LOGIC + LOGIC = PROLOG
4. CROSS + ROADS = DANGER
5. ONE + ONE = TWO
6. ONE + ONE + TWO = FOUR
7. KYOTO + OSAKA = TOKYO
8. APPLE + OSAKA = CHERRY
9. TWO = TWO = FOUR

## OLD QUESTION SOLUTION

1. **Solve following crypto-arithmetic problem.**

   **SEND+MORE=MONEY** **(2076 Baisakh)**

⇒ See on page no20

2. **What do you understand about well defined problems? Explain about the problems that can be solved using production rules with an example.** **(2075 Bhadra)**

⇒ 1st part see on page no12

2nd part see on page no15

**3. What is Problem Space? Solve the following cryptoarthimetic problem by showing all the steps.**

**BASE+BALL = GAMES** **(2075 Baisakh)**

⇒ Problem Space refers to the entire range of components that exist in the process of finding a solution to a problem. This range starts with "defining the problem" then proceeds to the intermediate stage of "identifying and testing" possible solutions" and ends with the final stage of "choosing and implementing a solution". Plus, it includes all of the smaller steps that exist between these identified stages.

2nd part see on page no

**4. Define constraint satisfaction problem (CSP).Solve the following crypto - arithmetic problem different letter denote different integer. SWIM + WEAR = RELAX** **(2074 Bhadra)**

⇒ 1st part see on page no15: Making equations

1. $M + R = X + 10C_1$
2. $I + A + C_1 = A + 20 \times C_2$
3. $W + E + C_2 = E + 10 \times C_4$
5. $C_4 = R$

Since, MSR cannot be 0 so, $C_1 = 1$

$\therefore$ $R = 1$

So, eq$^n$ iv) become: $S + W = C_3 = E + 10 + 1$

or, $S + W + C_3 = E + 10$

Attempt 1, take E = 9 (Max value)

$\Rightarrow$ $9 + W + C_3 = E + 10$ or, $W + C_3 = E + 1$

If $C_3 = 0$, $W = E + 1$.... (a)

From eq$^n$ (iii)

$W + E + C_2 = L + 10 \times C_1$

or, $E + L + E + C_3 = L + 10 \times 0$ or, $2E + 1 + C_2 = L$ —(b)

From eq$^n$ (a) and (b)

E must be less than such that L must not exceed 9.

Taking E = 3 such that W = 3+1 = 4

And, 9 is max $C_2$ must be 0

$\therefore \quad C_2 = 0 \qquad L = 7$

So, from eq$^n$ (ii) $I + A + C_1 = A + 10 \times O \Rightarrow I + C_1 = O$

This is only possible if $I = C_1 = O$

This makes,

$$M + R = X + 10 \times C_2$$

$$M + R = X$$

**Remaining values**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| I | R | A | E | W | | | L | | S |

As, R = 1

M + R x is valid if M = 5 and X = 6

So,

$$\begin{array}{r} 9\,4\,0\,5 \\ 4\,3\,2\,1 \\ \hline 1\,3\,7\,2\,6 \end{array}$$

Hence,

$$S = 9, W = 4, L = 7, X = 6, R = 1 \text{ and } I = 0.$$

5. **Solve the following puzzle by assigning numeral (0 - 9) in such that each letter is assigned unique digit which satisfy the following addition. ONE + ONE + TWO = FOUR** **(2073 Bhadra)**

$\Rightarrow$ We have Equation,

1. $E + O = R + C_1$
2. $N + N + W + C_1 = U + 10C_2$
3. $O + O + T + C_2 = O + 10\,C_3$

Assuming MSD cannot be 0

$\therefore \quad C_3 = 1$

$\therefore \quad F = 1$

Solving eq$^n$ (iii)

$$2 \angle 0 + T + C_2 = O + 10 \times 1$$

$\Rightarrow \quad O + T + C_2 = 10$

$\therefore \quad O + T + C_2 = 10 \ldots\ldots (v)$

$\Rightarrow \quad C_2 = 10 - O - T$

As, 0 is MSB, O cannot be zero.

So, let O = 2 So that, $C_2 = 10 - 2 - T$

$\therefore$ $C_2 = 8 - T$ As $C_2$ can be only 0, 1, 2 for sum 5 numbers.

Let, $C_2 = 1 \Rightarrow T = 7$

Such that,

$C_2 + 20 + T = 1 + 2 \times 2 + 7 = 12$ which is true $\therefore$ O = 2 and T = 7 and $C_2 = 1$

Such that from eq$^n$ (ii)

$2N + W + C_2 = U + 1O \times 1 \Rightarrow 2N + W + C_2 = U + 10$

Similarly, $C_1$ can be 0, 1 and 2

And eq$^n$ (i)

$2E + O = R + 10\,C_2$

or, $2E + 2 = R + 10C_2$

Let, E = 3 such that

$\Rightarrow$ $2 \times E + 2 = R + 10 \times C_1$

$\Rightarrow$ $2 \times 3 + 2 = R + 10 \times C_2$

$\Rightarrow$ $8 = R + 1\,0 \times C_2$

$\therefore$ $R = 8$ and $C_1 = 0$

Now, For eq$^n$ $2N + W + O = U + 10$

We have 0, 4, 5, 6 If $N = O \Rightarrow W = U + 10$

Which cannot be if N = 4,

$W = 10 + O - 8 = 2 + 0$

Not possible for any values.

If N = 6, $W = 10 + U - 12 = U - 2$

Also, not possible.

If N = 9, $W = 10 + O - 18 = U - 8$

Let, F = 4

$2 \times 4 + 0 = R + 10\,C_1$

$\Rightarrow$ $8 + 2 = P + 10\,C_1$

If, $C_1 = 1$

Such that R = 0

Now,

$$2N + W + C_1 = U + 10$$

or, $2N + W + 1 = U + 10$

$\Rightarrow$ $2N + W = U + 9$

if $N = 3$

If $U = 6$, $W = 9$

Thus

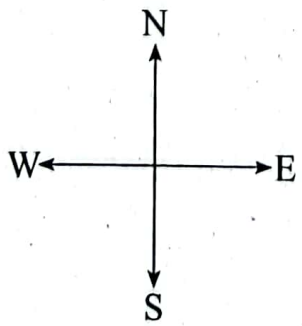
```
  T W O     2 3 4
  O N E     2 3 E
  O N E     7 9 2
 -------   -------
 F O U R   1 2 6 0
```

**6. A farmer has a goat, a wolf and a cabbage on the west side of a river. He wants to get all of his animals and his cabbage across the river onto the east side. The farmer has a row boat but he only has enough room for himself and one other thing. The wolf will eat the goat if they are left together alone. The goat will eat the cabbage if they are left together alone. How can the farmers get everything on the east side? (2073 Magh)**

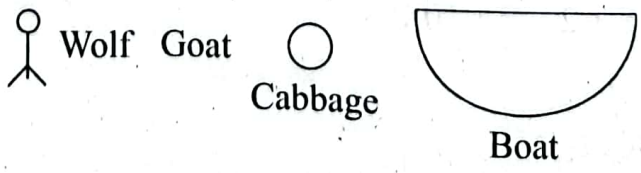
**i) Formulate this puzzle as search.**

**ii) Solve this problem - using search (any method) Draw the search tree and show the final solution.**

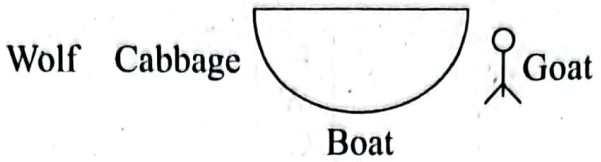
$\Rightarrow$ **i) One of the possible solution of the problem may be as follows:**



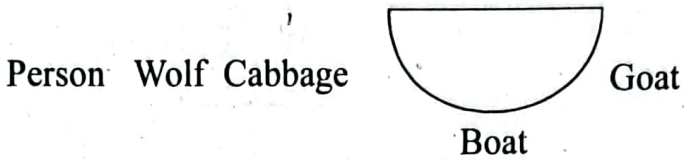

i) State (w, w, w, w)



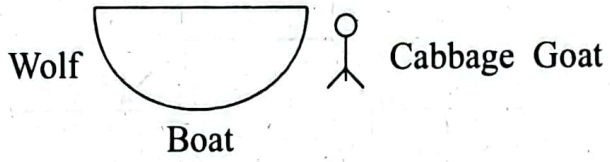
ii) State (e, w, e, w)



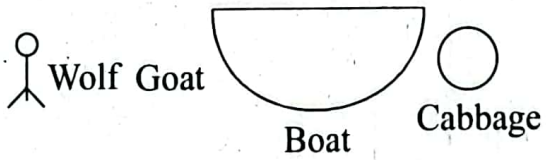

iii) State (w, w, e ,w)



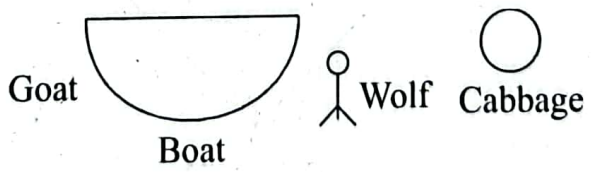

iv) State (e, w, e, e)



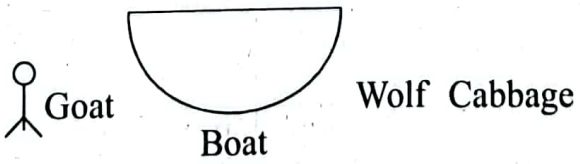

v) State (w, w, w, e)




vi) State (e, e, w, e)

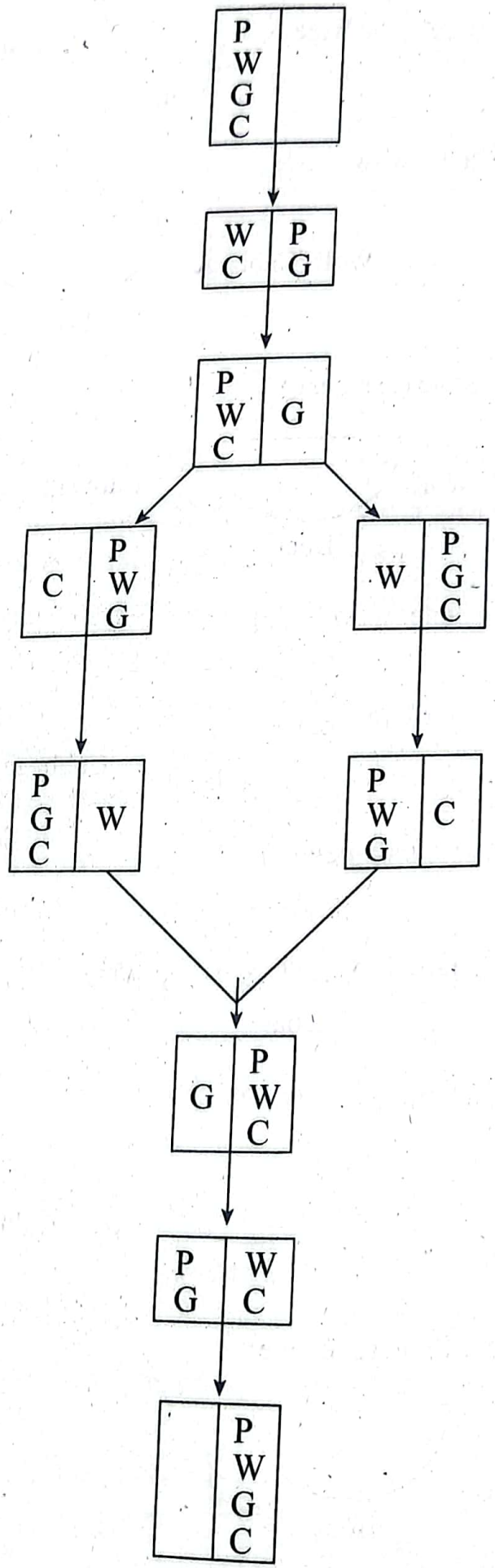



vii) State (w, e, w, e)




viii) State (e, e, e, e)

ii) **Search Tree**



**What is Constraint satisfaction problem? Solve the following Cryptoarthimetic problem.**

**SEND+MORE=MONEY** **(2072 Ashwin)**

⇒ 1st part see on page no15

2nd part see on page no

**What is a Constraint satisfaction problem? Solve the following crypto - arithmetic problem with necessary steps.**

**LOGIC + LOGIC = PROLOG** **(2072 Magh)**

⇒

$$\begin{array}{r} L\ O\ G\ I\ C \\ +\ L\ O\ G\ I\ C \\ \hline P\ R\ O\ L\ O\ G \end{array}$$

**1st part:** See on page no15

Here, P must be 1

Again $0 + 0 = 0$

There must not be carry to make sum of same digit to be same. This is only possible if $0 = 0$. As, $0 + 0 = 0$ does not create carry.

$$L + C = PR = 10 + R$$

$$\Rightarrow \quad 2L = R + 10$$

Again,

$$1 + 1 + (1/O) = 10 + O$$

The sum of two digit is always even thus, there must not be carry.

i.e. $I + I = 10 + O = 10$

$$\Rightarrow \quad 2I = 10$$

$$\therefore \quad I = 5$$

Also, a + a has carry from I + I ⇒ resulting a + a + 1 = 1 (without carry from $0 + 0 = 0$)

$$\Rightarrow \quad 2a + 1 = 1$$

$$\Rightarrow \quad a = \frac{L-1}{2}$$

Also,

$$c + c = a \text{ with not carry}$$

$$\Rightarrow \quad 2c = a$$

$$\Rightarrow \quad 2c = \frac{L-L}{2}$$

$$\Rightarrow \quad 4c + L$$

As, $L + L > 10$

$\Rightarrow \quad 2L = R + 10$

$L > 5$

$L = 7, 8, 9$

$\therefore$ If $L = 9$, $C = 2$ (As 0, 1 already)

As, $L + L = 10 + R$

$\therefore \quad R = 8$

$\therefore \quad a = \frac{L-1}{2} = \frac{9-1}{2} = 4$

$\therefore \quad L = 9 \qquad o = 0$

$a = 4 \qquad I = 5$

$c = 2$

$$\begin{array}{r} 9\,0\,4\,5\,2 \\ 9\,0\,4\,5\,2 \\ \hline 1\,8\,0\,8\,0\,4 \end{array}$$

**9. Solve the following cryptoarthimetic problem with necessary steps.**

**ONE+ONE+TWO=FOUR** **(2071 Magh)**

$\Rightarrow$ See on old question no 5

**10. Explain the role of production system with example.** **(2070 Magh)**

$\Rightarrow$ See on page no15

**11. Solve the following crypto - arithmetic problem where different letters denote different integer WRONG + WRONG = RIGHT. Explain the steps that you have a followed.** **(2070 Bhadra)**

$\Rightarrow$ We have

Carry = 0, 1

Variables = R, I, G, T, W, O, N

Numerical = 0, 1, 2 ----, 9

**Constraints:**

1. $2T = G + 10 + 1$ — (i)
2. $2H + C1 = N + 10C2$ — (ii)
3. $2a + C2 = O + 10\,C3$ — (iii)
4. $2I + C3 = R + 10C4$ — (iv)
5. $2R + C4 = W$ — (v)
6. Every variable should be unique value

Solving equation (5)

$2R + C4 = W$

As 2R + C4 do not result in carry i.e. carry = 0, the value of R may be 0, 1, 2, 3 and 4.

| R | Cy | W |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |
| 2 | 0 | 4 |
| 2 | 1 | 5 |
| 3 | 0 | 6 |
| 3 | 1 | 7 |
| 4 | 0 | 8 |
| 4 | 1 | 9 |

As MSB cannot be 0, R can be 1, 2, 3 ad 4.

Let, R – 1, cy = 0, W = 2

2I + C3 = 1 (from4)

I = 0 and C3 = 1

Solving equation — (iii)

C2 + 2C1 = 0 + 10 C3

or, C2 + 2C1 = 0 + 10

Here, the value of a must range from 5 to 9.

| G | Cy | W |
|---|---|---|
| 5 | 0 | 0 |
| 5 | 1 | 1 |
| 6 | 0 | 2 |
| 6 | 1 | 3 |
| 7 | 0 | 4 |
| 7 | 1 | 5 |
| 8 | 0 | 6 |
| 8 | 1 | 7 |
| 9 | 0 | 8 |
| 9 | 1 | 9 |

Assume, a = 7 and C2 ⇒ O = 4

Solving eq$^n$ (i)

$2T = a + 10C1$

or, $2T = 7 + 10C1$(not possible)

Here, 10C1 + 7yield odd number

Again,

Assume a – 6 and C2 = 1, O = 3

Solving equation (i)

$2T = a + 10C1$

or, $2T = 6 + 10C1$

If C1 = O, T = 3T [not possible]

If C1 = 1, T = 8

Solving equation (ii)

$2h + C1 = n + 10C^2$

or, $2h + 1 = n + 10$

or, $2h = n + 9$ — (vi)

Here, n must be odd for validity

So, the possible values of n are 5, 7 and 9.

If n = 5, 2h = 5 + 9

∴ h = 7

The solution is

```
 1 0 6 7 8
 1 0 6 7 8
 ---------
 2 1 5 5 6
```

**12. Solve the following crypto-arithmetic problem, where different letters denote different and identical letters denote same integer.**
**LOGIC+LOGIC=PROLOG** **(2069 Bhadra)**

⇒ See on old question no 8

⌘⌘⌘

# Chapter THREE

# Search Techniques

## Searching

Searching is the process of finding required nodes. Search process is carried out by constructing a search tree. It is performed through the state space.

## State space

A state space is usually implemented as a graph nodes corresponding to states, edges correspond to moves (start from an initial state, perform sequences of legal moves)

## Path

A path in the state space is simply any sequence of actions leading from one state to another.

## Steps in searching

1. Check whether the current state is the goal state or not?
2. Expand the current state to generate the new sets of states.
3. Choose one of the new states generated for search depending upon search strategy.
4. Repeat step 1 to 3 until the goal state is reached or there are no more state to be expanded.

## Searching is useful in the following problems:

i) Path finding Problems

E.g: eight puzzle, Travelling salesman problem.

ii) Two players games

E.g: chess and checkers.

iii) Constraint Satisfaction problem

E.g: eight queens.

## A search problem consists of:

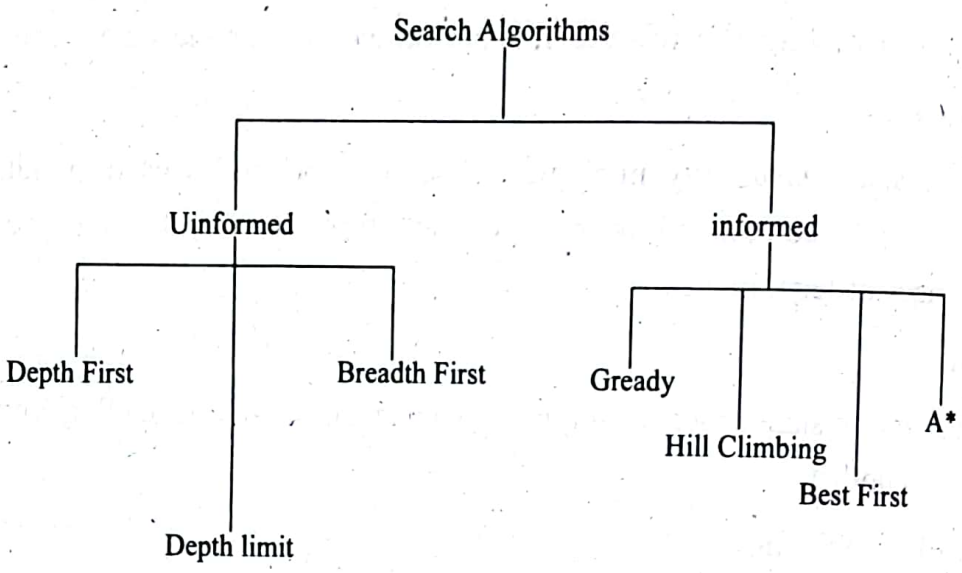
- **A state Space:** Set of all possible state where you can be.
- **A Start State:** The state from where the search begins.

- **A Goal Test:** A function that looks at the current state returns whether or not it is the goal state.

  The solution to search problem is a sequence of actions, called the plan that transforms the start state to the goal state.

  This plan is achieved through search algorithms

**Search Algorithm**




**Uninformed**

Uninformed search is a class of general - purpose search algorithm which operators in brute - force way. Uninformed search algorithm do not have additional information about state or space other than how to traverse the tree, so, it is also called blind search.

**i) Breadth First search**

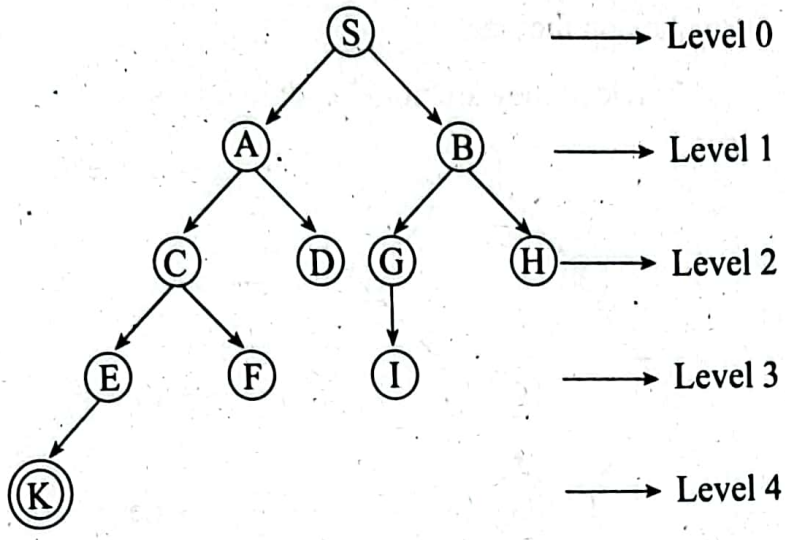
Breadth First search is the most common search strategy for traversing a tree or graph. This algorithm searches breadth wise in a tree or graph, so it is called breadth - first search.

BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level. Breadth - First search implemented using FIFO queue data structure.

**Algorithm**

1. Visit the adjacent unvisited vertex. Mark it as visited. Display it. Insert it in a queue.
2. If no adjacent vertex is found remove the first vertex from the queue.

3. Step (1) and (2) is required until the queue is empty.




Root Node: S
Goal Node: K

S---→A---→B---→C---→D---→G---→H---→E---→F---→I---→K

i) Time complexity

$T(b) = O(b^d)$

Where,

d = depth of solution

b = is a node at every state

ii) Space complexity: $O(b^d)$

iii) Completeness: Yes

iv) Optimally: BFS is optimal if path cost is non - decreasing function of the depth of the node.

**ii) Depth - First search**

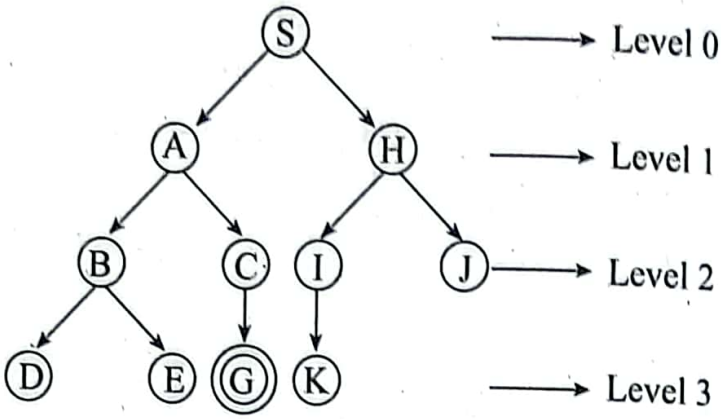
Depth - First search is a recursive algorithm for traversing a tree or graph data structure. It is called depth - first search because it starts from the root node and follows each path its greatest depth node before moving to the next path. DFS uses a stack structure for its implementation.

**Algorithm**

1. Put the start node on the stack.
2. While stack in not empty.
   - Pop the stack
   - If the top of stack is the goal, stop.

- Otherwise push the nodes connected to the top of the stack on the stack.

  (Provided they are not already on the stack)




Root Node: S
Goal Node: K

S---→A---→B---→D---→E---→C---→G

i) Completeness: Yes

ii) Time complexity $T(n)_o = (n^m)$

Where, m = maximum depth of any node

iii) Space complexity = $O(b^m)$

iv) Optimal = Non - optimal

**iii) Depth - limited Search Algorithm**

A depth - limited search algorithm is similar to depth - first search with a predetermined limit Depth - limited search can solve the drawback of the infinite path in the Depth - First search. In this algorithm, the node at the depth will treat as it has no successor or node further.

**Depth - limited search can be terminated with two conditions of failure:**

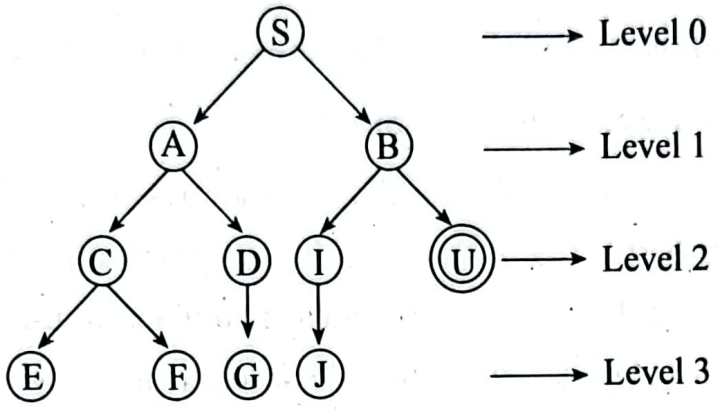
- **Standard failure:** It indicates that problem does not have any solution.
- **Cut – Off failure value:** It defines no solution for the problem within a given depth limit.

**Advantages**

Depth limited search is memory efficient.

**Disadvantages**

- Depth - limited search also has a disadvantages of incompleteness.
- It may not be optimal if the problem has more than one solution.




Order S---→A---→C---→D---→B---→I ---→J

i) Completeness: DLS search algorithm is complete if the solution is above the depth - limit.

ii) Time complexity: $O(b^l)$

iii) Space complexity: $O(b \times l)$

iv) Optimal: Depth - limited search can be viewed as a special case of DFS, and it is also not optimal even if $\pi > d$.

**Informed Search**

Informed search algorithm contains an array of knowledge such as how far we are from the goal, path cost, how to reach to goal node etc.

The informed search algorithm is more useful for large search space. Informed search algorithm uses the idea of heuristic, so it also called Heuristic search.

**Heuristic Function**

Heuristic is a function which is not used in Informed search, and it finds the most promising path. It takes the current states of the agent as its input and produces the estimation of how close agent is from the goal.

It is denoted by h(n).

Admissibility of the heuristic function given as:

$h(n) <= h^*(n)$

Where,

h (n) = heuristic cost

h*(n) = estimated cost

Hence, heuristic cost should be less than or equal to the estimated cost.

1. **Best First Search (Greedy Search)**

Greedy best - first search algorithm always selects the path which appears best at the moment. It is the combination of depth - first search and breadth - first search algorithm. It uses the heuristic function and search. In the best - first search algorithm, we expand the node which is closest to the goal node and the cost is estimated by heuristic function.

$F(n) = h(n)$

Where, h(n) = estimated cost from node n to the goal.

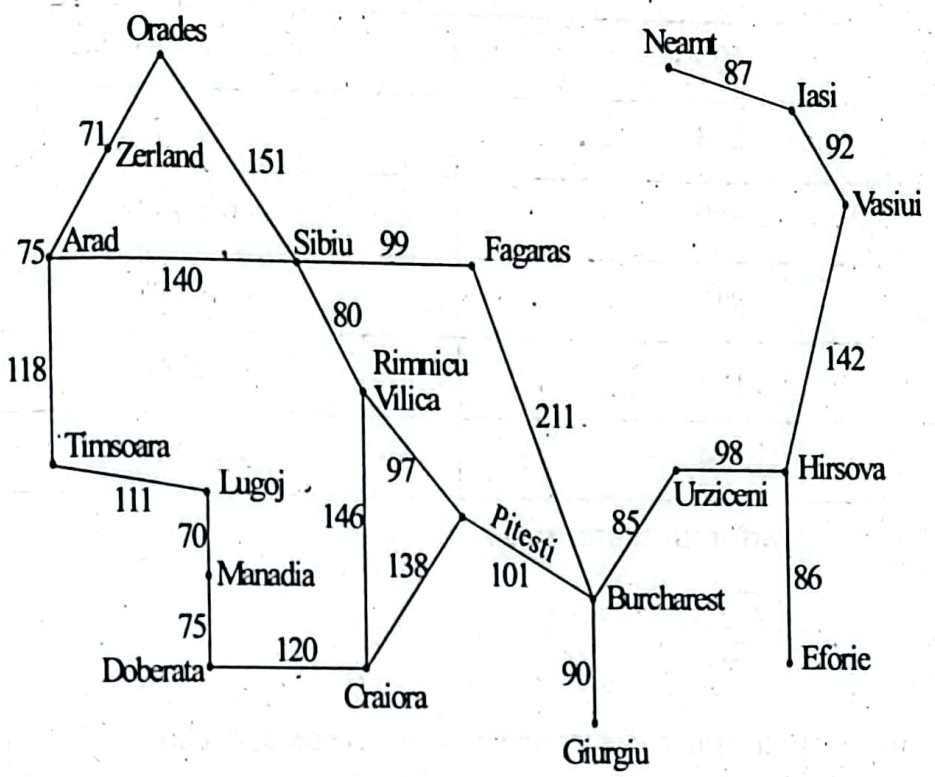
The greedy best first algorithm is implemented by the priority queue.

**Algorithm**

1. Place the starting node into the OPEN list.
2. If the OPEN list is empty, stop and return failure.
3. Remove the node n, from the open list which has the lowest value of h(n), and places it in the CLOSED list.
4. Expand the node n, and generate the successor of node n.
5. Check each successor of node n, and find whether any node is goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to step 6.
6. For each successor node, algorithm checks for evaluation function F(n), and the check if the node has been in either OPEN or CLOSED list. If the node has not been in list, then add it node has not been in list, then add it to the OPEN list.

7. Return to step2.

To find oath from Arab to Bucharest

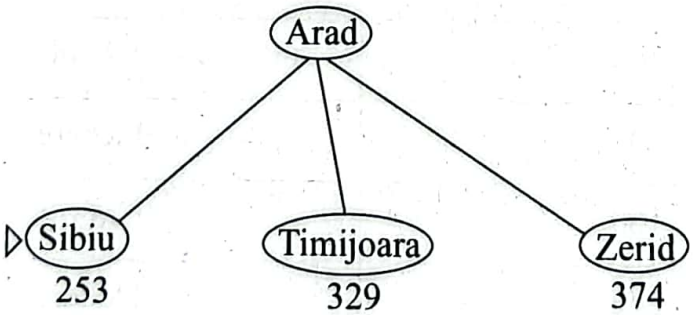



| Straight Distance to Brucharest | Place |
|---|---|
| 366 | Arad |
| 0 | Bucharest |
| 160 | Craiova |
| 242 | Doberta |
| 161 | Fforie |
| 178 | Fagaras |
| 77 | Giurgin |
| 151 | Hirsova |
| 226 | Iasi |
| 244 | Lugoj |
| 241 | Mehadia |
| 234 | Neamt |

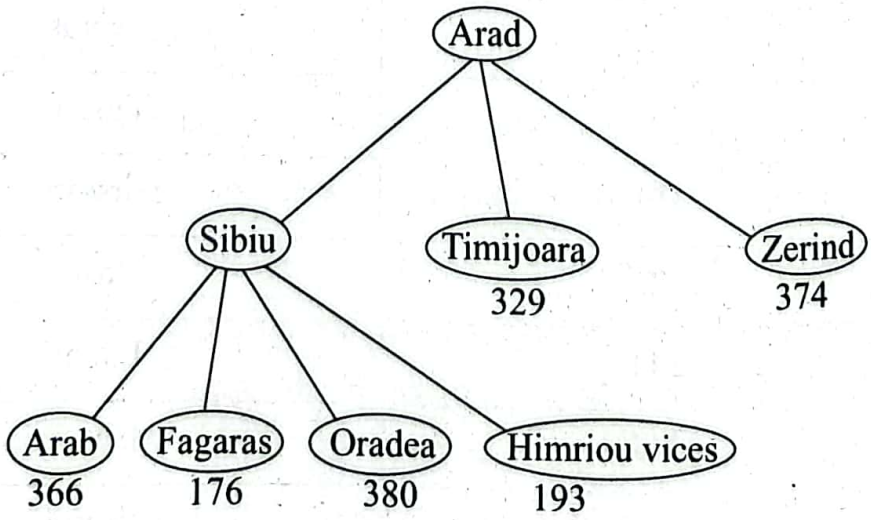
| | |
|---|---|
| 380 | Oradea |
| 98 | Pitesti |
| 193 | Rimnicu Vilcea |
| 253 | Sibiu |
| 329 | Timisoara |
| 80 | Uriziceni |
| 199 | Vaslui |
| 374 | Zerland |

i) Start from source node

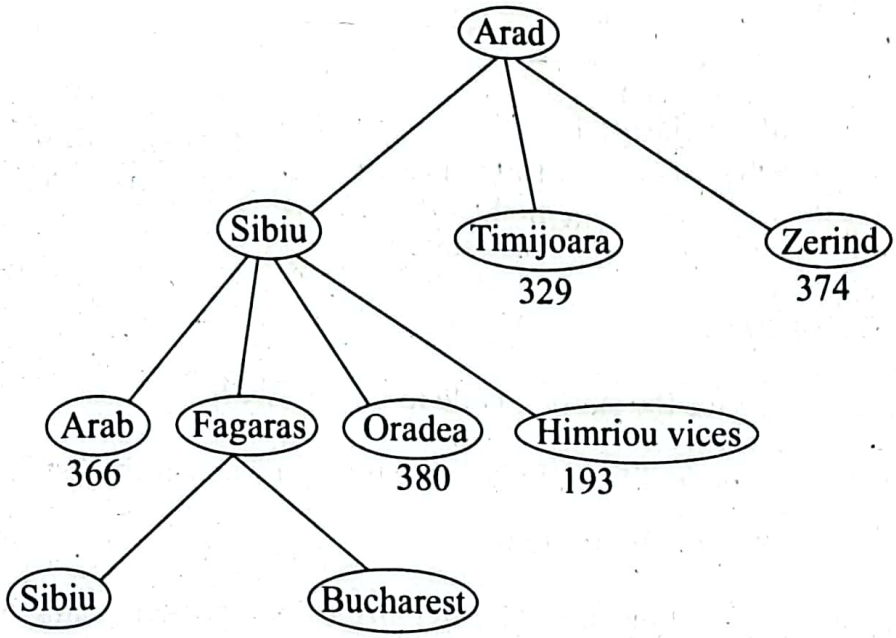
ii) Evaluate cities connected to source and choose the one with lowest (h) distance value.




iii) Continue evaluating cities with their distance to destination.



iv) Continue evaluating until destination is reacted.




Final solution is Arad → Sibiu → Fagaras → Bucharest

i) **Time complexity:** The worst case time complexity of greedy best first search is $O(b^m)$.

ii) **Space complexity:** The most case space complexity of greedy best first search is $O(b^m)$. Where, m is the maximum depth of the search space.

iii) **Complete:** Greedy best - first search is also incomplete even if the given state space is finite.

iv) **Optimal:** Greedy best first search algorithm is not optimal.

2. **A* Search Algorithm**

A* search is the most commonly known from of Best - First search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UGS and greedy best - first search, by which it solves the problem efficiently A* search algorithm finds the shortest path through the search space using the heuristic function,. This search algorithm expands less search tree and provides optimal result faster. A* algorithm is similar to UCG except that it uses g(n) + h(n) instead of g(n).

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence, we can combine both costs as following and this sum is called as a fitness number.

$$F(n) - g(n) + h(n)$$

- $F(n)$: Estimate cost of the cheapest solution
- $g(n)$: Cost to reach node n from start state
- $h(n)$: Cost to reach from node n to goal node

**Algorithm**

1. Place the starting node in the OPEN list.
2. Check if the OPEN list is empty or not, if the list is empty then return failure stops.
3. Select the node from the OPEN list which has the smallest value of evaluation function (g + h), if node n is goal node then return success and stop otherwise.
4. Expand node n and generate all of its successors, and put n into the closed list. For each successor $n^+$ check whether $n^+$ is already in the OPEN or CLOSED list, if not then compute evaluation function for $n^+$ and place into open list.
5. Else if node $n^+$ is already in OPEN and CLOSED then it should be attached to the back pointer which reflects the lowest $g(n^+)$ value.
6. Return to Step 2.

e.g:

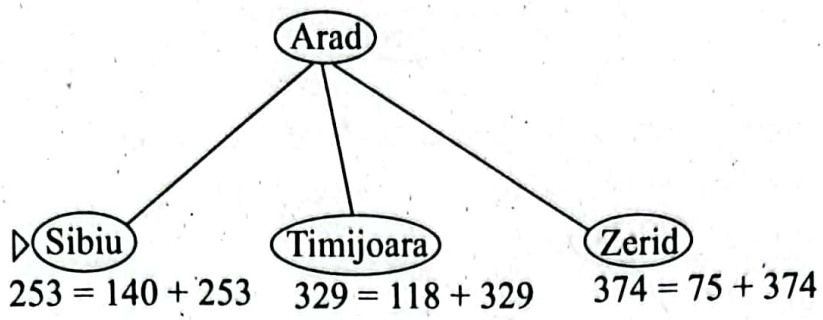
Find the path from Arad to Bucharest

i) Start with source node.

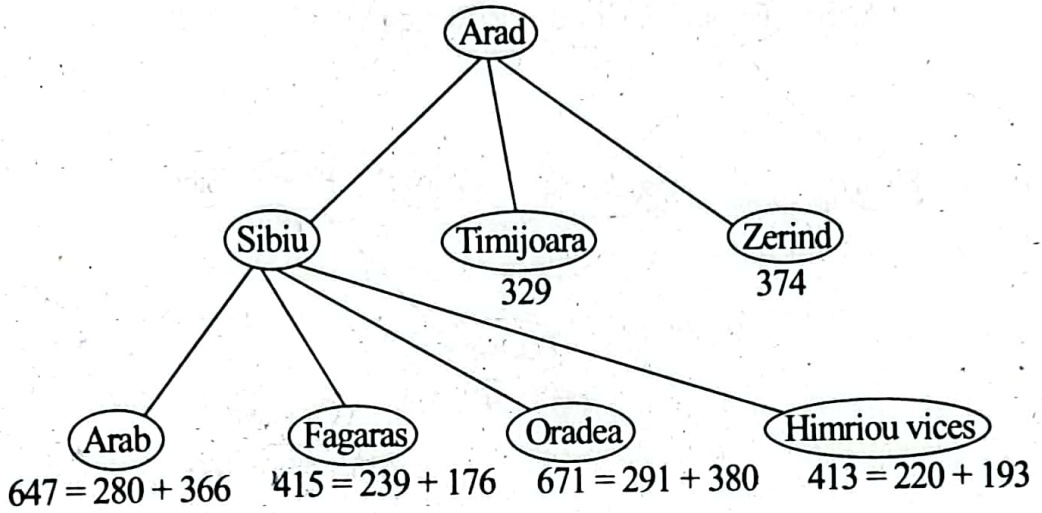
▷ Arad

$366 = 0 + 366$

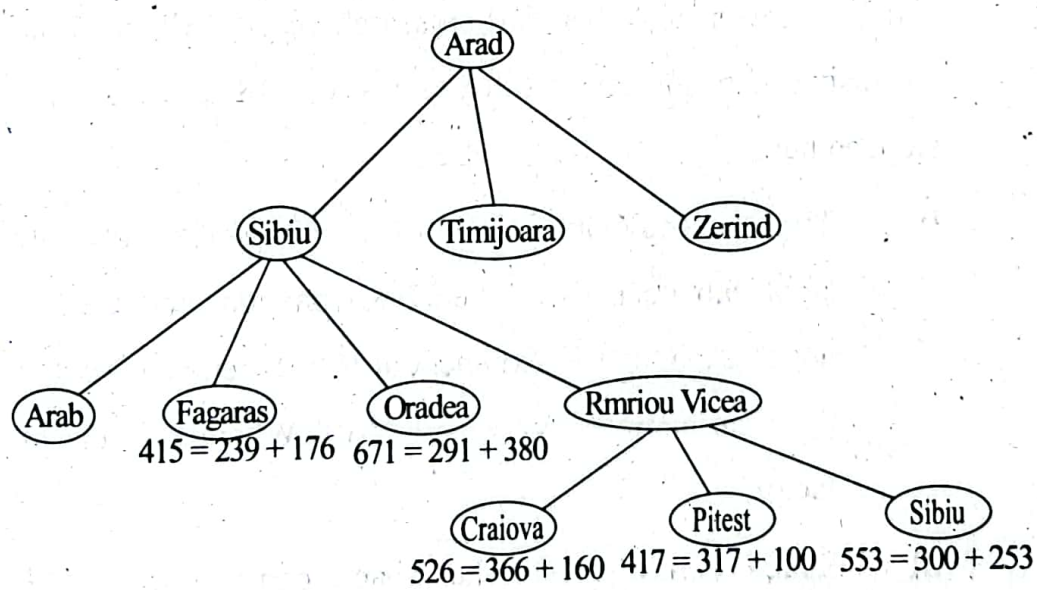
ii) Evaluate nodes converted to source. Evaluate $F(n) = g(n) + h(n)$ for each node. Select node with lowest $F(n)$ value. Sibiu node is chosen.



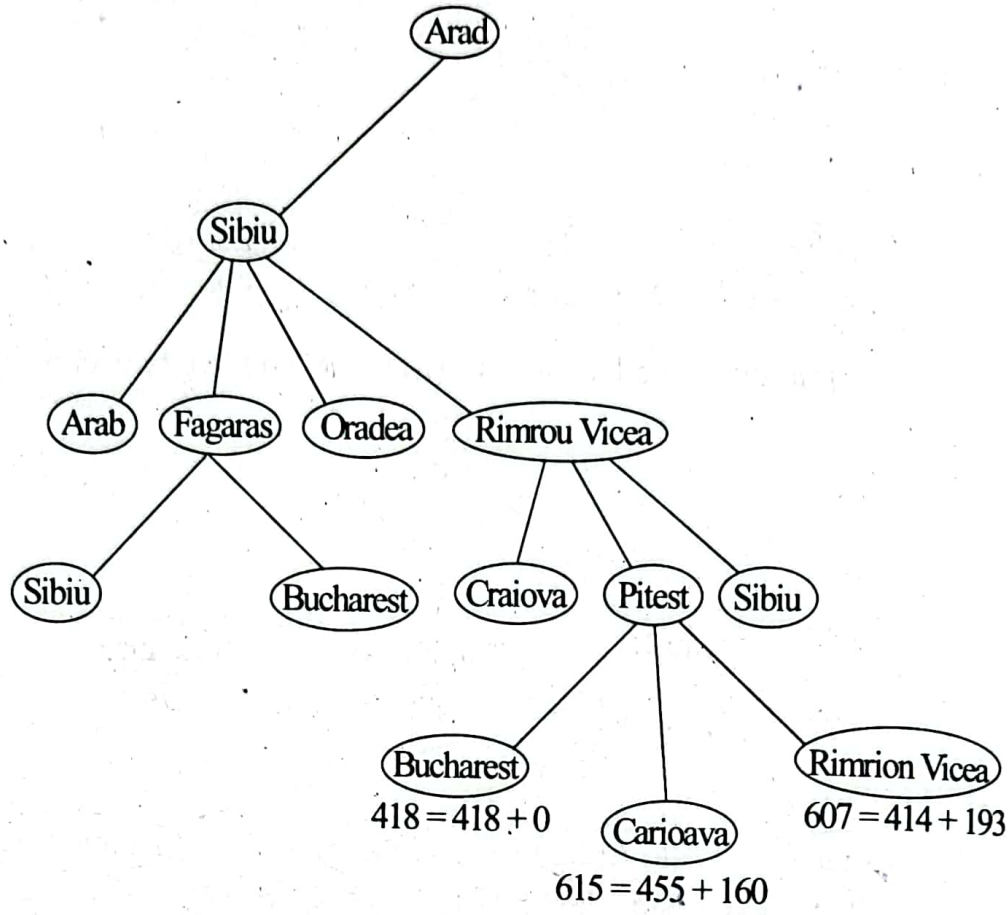

iii) Rimniou vice a has lowest f(n) value, so the city is chosen.




iv) In all nodes expanded Fagaras has lowest F(n) value, so this node is expanded next.




v) Expanding Fagaras node reaches our destination on Bucharest with F(n) value 450. So all the nodes with F(n) value expensive than 450 are avoided as evaluating them would result in cost of path expensive than our path already achieved . The nodes with F(n) value lower than 450 are evaluated next.




Evaluating next node with Petesti with cost 417 (lower than 450 – Arad – Sibiu – Fagaras – Bucharest) we reach Bucharest with cost 418 which is lower than our earlier search result. All other nodes are expensive than our new result i.e. cost of 418. Hence, our search is stopped here.

i) **Time complexity:** The time complexity of A* search algorithm depends on heuristic function, and the number of nodes expanded is exponential to the depth of solution d. So the time complexity is O (b ^ d). Where b is the branching factor.

ii) **Space Complexity:** The Space complexity of A* search algorithm is O (b ^ d).

iii) **Complete**: A* algorithm is complete as long as:

- Branching factor is finite.
- Cost at every action of fixed.

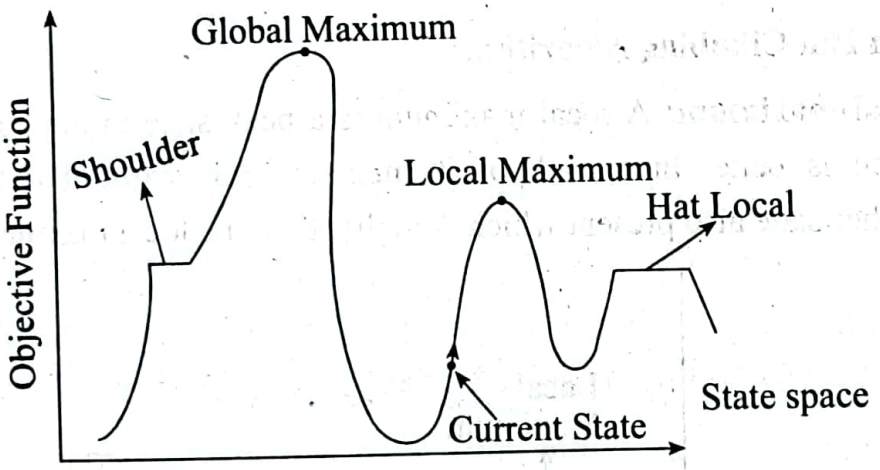
iv) **Optimal:** A* algorithm is optimal if it follows below two conditions.

a) **Admissible:** The first conditions requires for optimality is that h (n) should be an admissible heuristic for A* tree search. An admissible heuristic is optimistic in nature.

b) **Consistency:** Second required condition is consistency for only A* graph - search.

i) **Hill Climbing Algorithm**

Hill climbing algorithm is a search algorithm which continuously moves in the direction of increasing evaluation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbour has a higher value.




**Different region in State landscape**

i) **Local Maximum:** Local maximum is a state which is better than its neighbour states, but there is also another state which is higher than it.

ii) **Global Maximum:** Global maximum is the best possible state of state space landscape.

iii) **Current State:** It is a state in a landscape diagram where an agent is currently present.

iv) **Flax local maximum:** It is a flat space in the landscape where all the neighbour state of current states have the sane value.

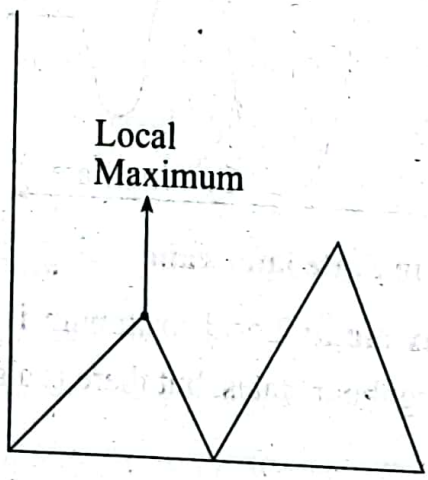
v) **Shoulder:** It is a plateau region which has an uphill edge.

**Algorithm**

1. Evaluate the initial state, if it is goal state the return success and stop.
2. Loop until a solution is found or there in no new operator left to apply.
3. Select and apply an operator to the current state.
4. Check new state.
   a) If it is goal state, then return success and quit.
   b) Else if it is better than the current state then assign state as a current state.
   c) Else if not better than the current state, then return to step 2.
5. Exit.

**Problem in Hill Climbing Algorithm**

1. **Local maximum:** A local maximum is a peak state in the landscape which is better than each of its neighbouring states, but there is another state also present which is higher than the local maximum.

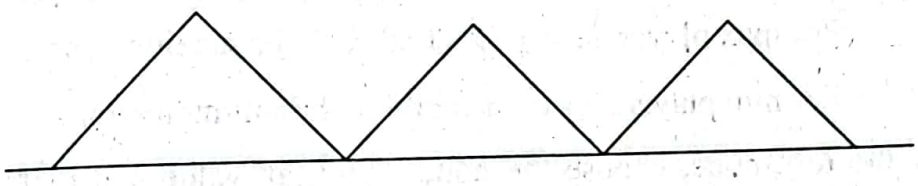



**Solution:** Backtracking technique can be a solution of the local maximum in state space representation. Create a list of promising path so that algorithm can back track the search space and explore other paths as well.

2. **Plateau:** A plateau is the area of the search space in which all the neighbour states of the current state contains the same value, because of this algorithm does not find any best direction to more. A hill - climbing search might be lost in the plateau area.

**Solution:** The solution for the plateau is to take big steps or very little steps while searching to solve the problem. Randomly select a state which is far away from the current state so it is possible that the algorithm could find non - plateau region.

3. **Ridges:** A ridge is special from of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope, and cannot be reacted in a single move.



**Solution:** With the use of bidirectional speed or by moving in different directions. We can improve this problem.

**Adversarial Search**

Competitive environment in which the agents are in conflict give rise to adversarial search, often known as game.

In AI, games means deterministic, fully observable environments in which there are two agents whose actions must alternate and in which utility values at the end of the games are always equal and opposite.

E.g: If first player wins, the other player necessarily loses.

Opposition between the agents utility functions makes the situation adversarial.

**Games**

A game can be formally be defined as a kind search problem with the following components.

- Initial state
- A successor function
- A terminal test
- A utility function

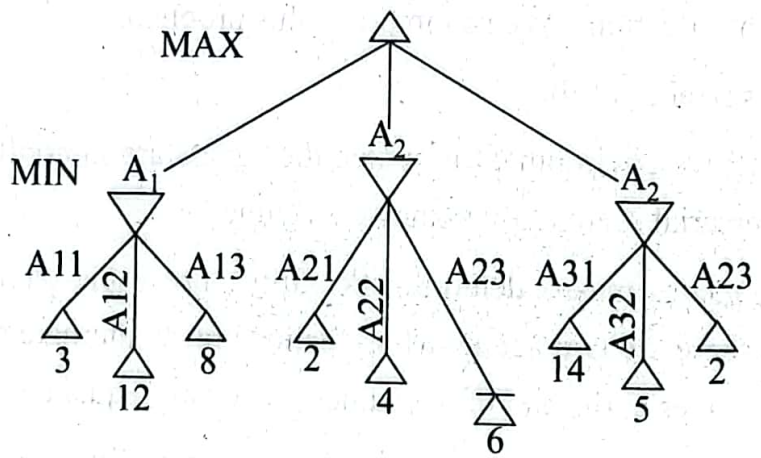
## Min – Max Algorithm

Min Max is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal more for player, assuming that your opponent also plays optimally. It is widely used in two player turn - based games such as tic - tac - toe, Backgammon, chess etc.

In minmax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the score possible.

**Algorithm**

1. Construct the complete game tree.
2. Evaluate scores for leaves using the evaluation function.
3. Back - up scores from leaves to root, considering the player type.
   - For max player, select the child with the maximum score.
   - For min player, select the child with minimum score.
4. At the root node, choose the node with max value and perform the corresponding move.




**Properties**

(i) Complete: Yes

(ii) Optimal: Yes

(iii) Time complexity: O ($b^m$)

(iv) Space complexity: O ($b^m$)

**Alpha - Beta Pruning**

Alpha - Beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm. As we have seen in the minimax search algorithm seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree.

Since we cannot eliminate the exponent but we can cut it half. Hence, there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called pruning. This involves two threshold parameter, so it is called alpha - beta pruning. It is also called as Alpha - Beta Algorithm.

**The two - parameter can be defined as:**

a) Alpha: The best (highest - value) choice we have found so far at any along the path of maximizer. The initial value of alpha is $-\infty$.

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

Condition for Alpha - Beta pruning.

$$\boxed{\alpha >= \beta}$$

**Key points about alpha - beta pruning.**

(i) The max player will only update the value of alpha.

(ii) The min player will only update the value of beta.

(iii) While back tracing the tree, the node values will be passed to upper nodes instead of values of alpha and beta.

(iv) We will only pass to alpha, beta values to the child nodes.

**Algorithm**

Function alpha beta (node, depth, α, β, maximizing player)

If depth = 0 or node is a terminal node return the heuristic value of node.

If maximizing player

For each child of node

α: = max (α, alphabet (child depth -1 α, β, FALSE))

If β ≤ α

Break (A β cut - off*)

returnα

Else,

For each child of node

β: = min (β, alpha beta (child, depth - 1, α, β, TRUE))

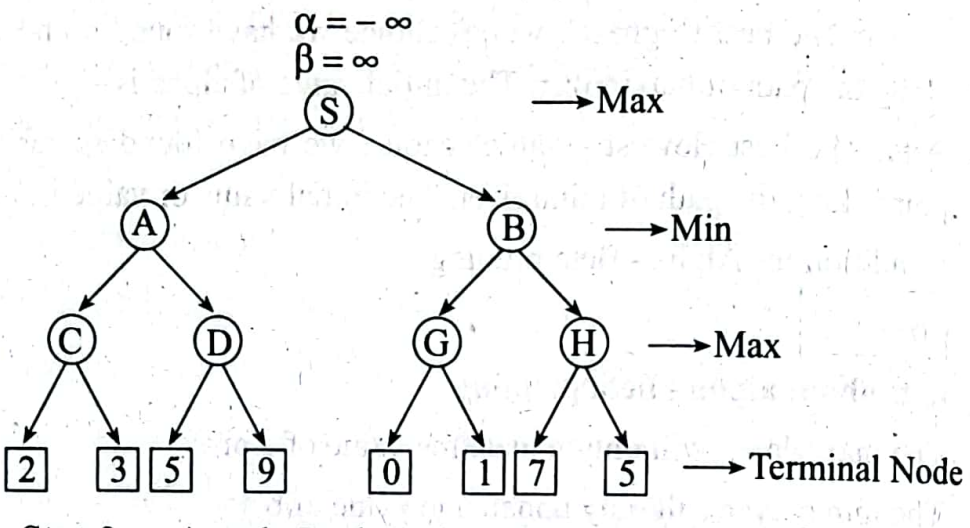
If β ≤ α

Break (* α cut-off*)

returnβ

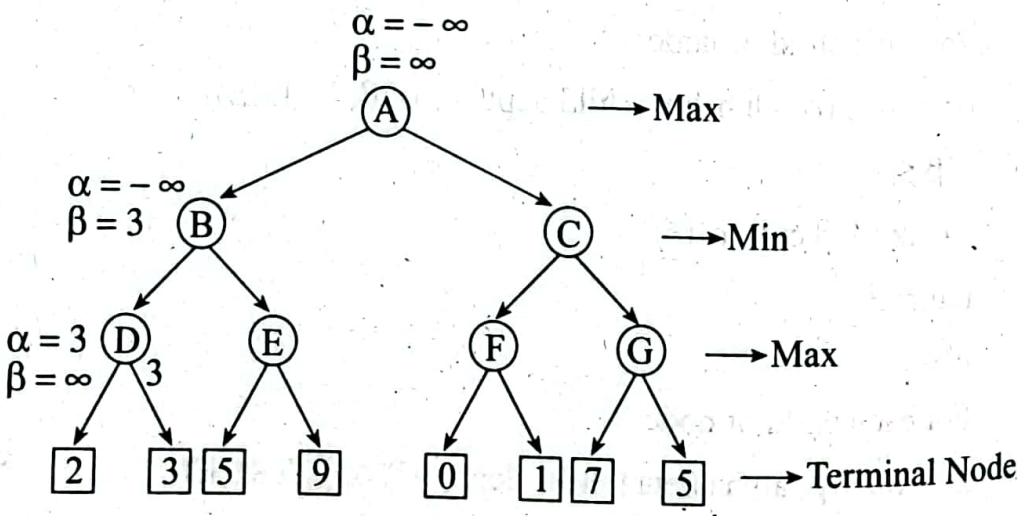
E.g.

**Step 1:** At First step, the max player will start first move from node A where $\alpha = -\infty$ and $\beta = +\infty$ values of alpha and beta passed down to node B where again $\alpha = -\infty$ and $\beta = +\infty$, αvδ node B passes the same value to its child D.



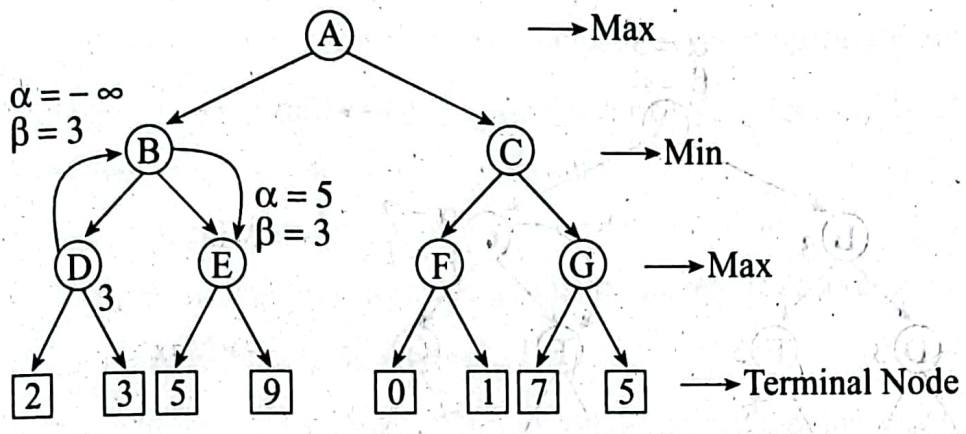

**Step 2:** A node D, the value of α will be calculated as its turn for max. The value of α is compared with firstly 2 and then 3, and the max (2, 3) = 3 will be the value of α at node D and value will also 3.

**Step 3:** Now algorithm backtrack to node β, where the value of β will change as this is return of Min, Now $\beta = +\infty$, will compare with the available subsequent nodes value i.e. min $(\infty, 3) = 3$ hence, at node β now $\alpha = -\infty$ and $\beta = 3$.




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

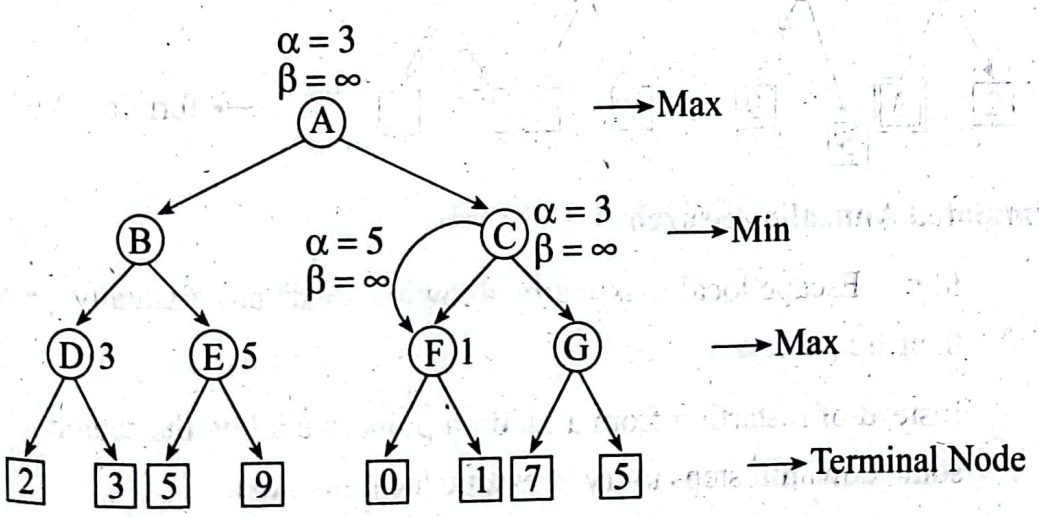
**Step 4:** At node E, Max will take its turn, and the value of alpha will change. The current value of alpha will be compared with 5, so max (-∞, 5) = 5, hence at node E $\alpha = 5$ and $\beta = 3$. Where $\alpha >= \beta$. So, the right successor of E will be pruned, and algorithm will not traverse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



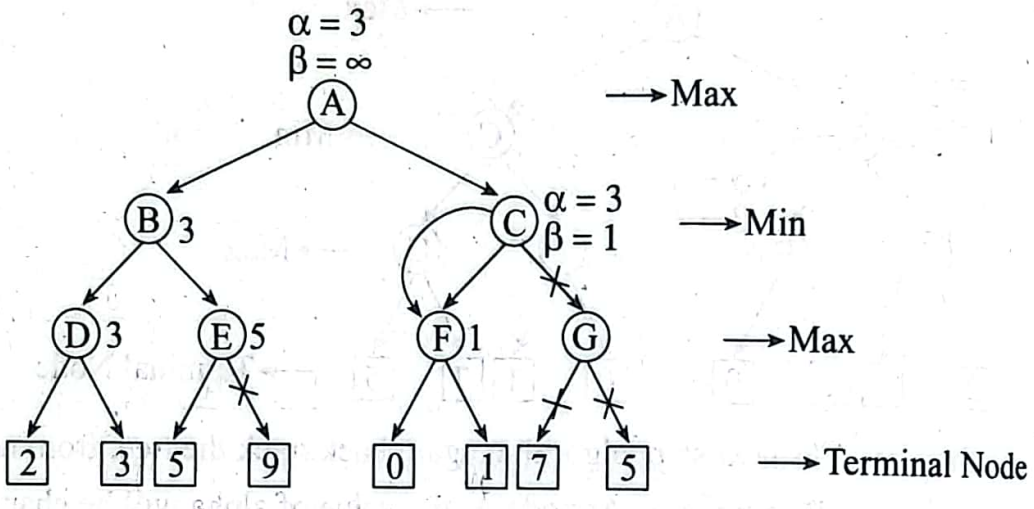

**Step 5:** At next step, algorithm again back track the tree, from node B to node A. At node A, the value of alpha will be changed the maximum available value is 3 as max (- ∞, 3) = 3 and $\beta = +\infty$, these two values now passes to right successor of A which is Node C.

AT node C, $\alpha = 3$ and $\beta = +\infty$, and the same values will be passed on to node F.

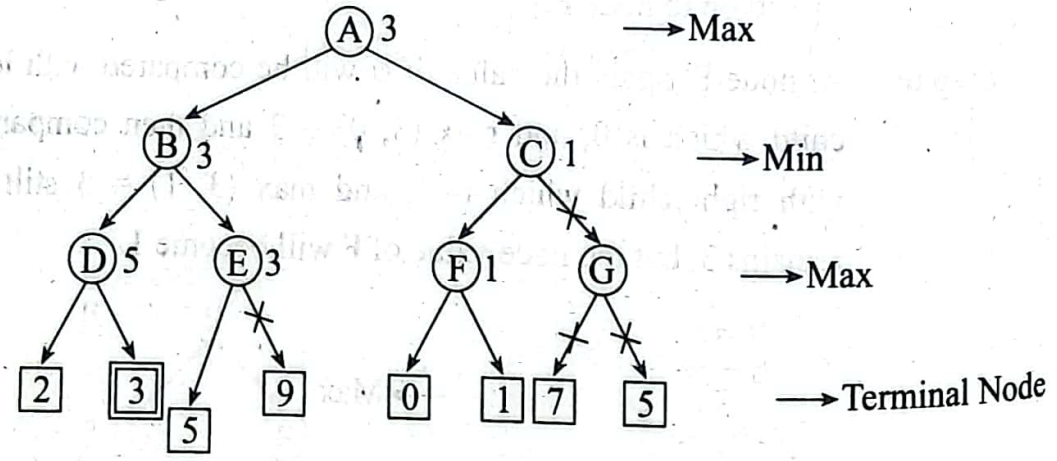
**Step 6:** At node F, again the value of $\alpha$ will be compared with left child which is 0, and max (3, 0) = 3 and then compared with right child which is 1, and max (3, 1) = 3 still $\alpha$ remains 3, but the node value of F will become 1.



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




**Step 8:** C now returns, the value of 1 to A here the best value of A is max (3, 1) = 3. Following is the final game tree which is the showing the nodes which are computed and nodes which has never completed. Hence, the optimal value for the maximizer is 3 for this example.




**Simulated Annealing Search**

- Idea: Escape local maxima by allowing "bad" but gradually decrease their frequency.
- Instead of restarting from a random point, we allow the search to take some downhill steps to try to escape local maxima.

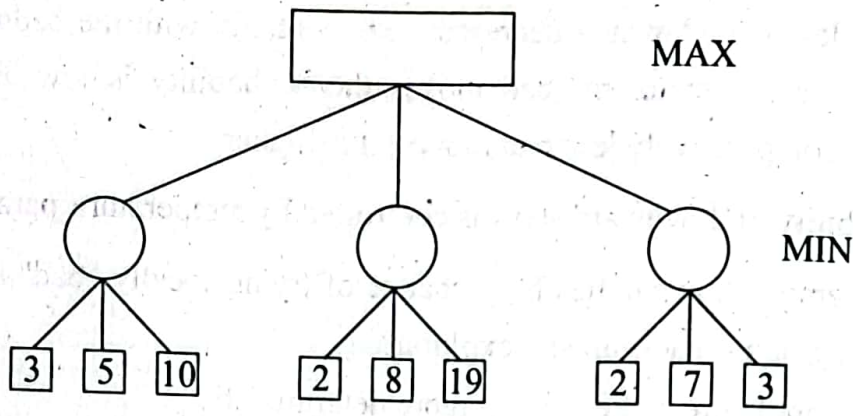
- A random pick is made for the move.
  - → If it improves the situation, it is accepted straight away.
  - → If it worsens the situation, it is accepted with some probability less than 1 which decreases exponentially with the badness of the move i.e. for bad moves the probability is low and for comparatively less bad moves, it is higher.
- Probability of downward steps is controlled by temperature parameter.
- High temperature implies high chance of trying locally "bad" moves, allowing non - deterministic exploration.
- Low temperature makes search more deterministic.
- Temperature begins high and gradually decreases according to pre determined annealing schedule.
- Initially we are willing to try out lots of possible paths, but over time we gradually settle in on the most promising path.
- If temperature is lowered slowly enough, an optimal solution will be found.
- In practice, this schedule is often too slow and we have to accept sub - optimal solutions.

**Local beam search**

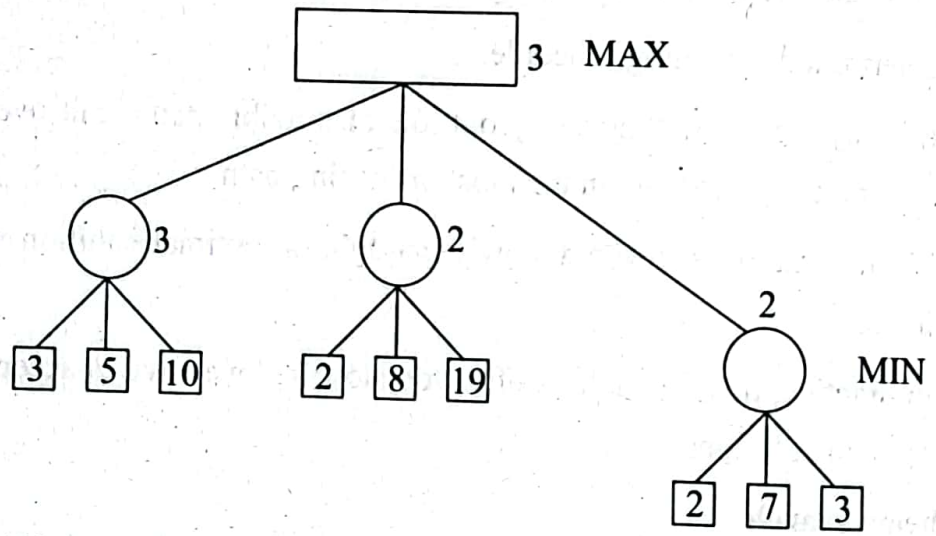
- A path based algorithm
- Keep tracks of K states instead of one.
  - → Initially: start with K randomly selected states.
  - → Next: Determine all successors of K states.
  - → If any of successor is goal →finished.
  - → Else select K best from successor and repeat.
- Major difference with random - restart search.
- Information is shared among K search threads.
- Provide a simple, efficient way to share some knowledge across a set of searches.
- Can quickly becomes concentrated in a small region of the state space.

## OLD QUESTION SOLUTION

1. **Discuss about alpha-beta purning algorithm. Find the value of min-max value using this concept in the following tree.** **(2076 Baisakh)**




⇒ 1st part see on page no 48




2. **Discuss about evaluation criteria for search algorithm. State the problems in hill climbing search algorithm.** **(2075 Bhadra)**

⇒ i) **Completeness:** A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.

ii) **Optimality:** If a solution found for an algorithm is guaranteed to be best solution (lowest path cost) among all other situations, then such a solution for is said to be an optimal solution.

iii) **Time Complexity:** Time complexity is a measure of time for an algorithm to complete its task.

iv) **Space Complexity:** It is maximum storage space required at any point during the search as the complexity of the problem.

**2nd part:** See on page no. 46

3. **Discuss the hill climbing search algorithms along with problem associated with it and discuss their solutions. Why simulated annealing is important? (2075 Baisakh)**

⇒ **1st part:** See on page no. 45 and 46

Simulated Annealing is a probabilistic technique for approximating the global optimum of a given function. Specially, it is a metaheuristic to approximate global optimization is large search space for an optimization problem. It is often used when the search space is discrete (e.g. the travelling salesman problem). For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to extract algorithms such as gradients descent, Branch and Bound.

4. **Why searching is important in problem solving? What is the drawbacks of greedy-best first search and how A* search algorithm solve it. Explain with example. (2074 Bhadra)**

⇒ Search is a universal problem - Solving technique. It involves systematic trial and error exploration of alternative solutions. Many problems don't have a simple algorithmic solution. Casting these problems as search problems if often the easiest way of solving them. Search is useful when the sequence of actions required to solve a problem in not known like:

- Path finding problems e.g: eight puzzle, Travelling Salesman problem.
- Two player games e.g: chess and checkers.
- Constraint satisfaction problems e.g: eight queens.

**2nd part:** See on page no41

5. **Explain the necessity of searching techniques in AI? Differentiate between breath first search and depth first with their performance criteria. (2073 Bhadra)**

⇒ 1st part see on old question no 4

| BFS | DFS |
|---|---|
| 1. BFS find the shortest path to the destination | 1. DFS goes to the bottom of a sub tree then backtracks. |

| | | | |
|---|---|---|---|
| 2. | It uses a queue to keep track of the next location to visit. | 2. | It uses a stack to keep track of the next location to visit. |
| 3. | BFS traverses according to tree level. | 3. | DFS traverses according to tree depth. |
| 4. | It is implemented using FIFO list. | 4. | It is implemented using LIFO list. |
| 5. | You can never be trapped into infinite loops. | 5. | You can be trapped into finite loops. |

**6.) Devise an example to show A* algorithm uses past cost and heuristic cost to generate best solution. (2073 Magh)**

⇒ See on page no41

**7. What is a searching? Explain Breadth First search and Depth First Search compare their performance criteria(2072 Ashwin)**

⇒ Searching

See on page no33

Breadth First Search

See on page no34

Depth First Search

See on page no35

Performance Calculation

See on old question no 9

**8. Searching is an important part of AI, justify it. Explain any two types of blind search and compare them in terms of space and time complexity. (2072 Magh)**

⇒ **1st part see on old question no 4**

**Two types of blind search**

See on page no 34 and 35

**Comparison:**

See on old question no 9

**9. What is the advantages of depth limit search? Compare it with other strategies. (2071 Bhadra)**

**Ans: Advantages of depth limited search:**

i) Depth limited search is better than DFS and requires less time and memory space.

ii) DFS assures that the solution will be found if it exists infinite time.

iii) There are applications of DLS in graph theory particularly similar to the DFS.

iv) In order to combat the disadvantages of DFS, we add a limit to the depth and our search strategy performs recursively down the search tree

**Comparison**

| Criteria | BFS | DFS | DLS |
|---|---|---|---|
| Time complexity | $O(b^{d+2})$ | $O(b^m)$ | $O(b^l)$ |
| Space complexity | $O(b^{d+1})$ | $O(b^m)$ | $O(b^l)$ |
| Optimality | Yes | NO | No |
| Completeness | Yes | No | No |

**10. How informed search is different from informed search? Explain min-max algorithm with suitable example, discuss how alpha - beta is different from min - max algorithm. (2071 Magh)**

⇒

| Informed Search | Uninformed Search |
|---|---|
| 1) It uses knowledge for the searching process. | 1) It doesn't use knowledge for. |
| 2) It finds solution more quickly. | 2) It finds solution slow as compared. |
| 3) It is high efficient. | 3) It is mandatory efficient. |
| 4) Cost is low. | 4) Cost is high. |
| 5) It consumes less time. | 5) It consumes moderate time. |
| 6) E.g: Greedy search, A* search, Graph search. | 6) E.g: DFS, BF etc. |

**Min-max:** See on page no

Yes, Alpha - Beta and mini max returns the same answer. All alpha - beta is prevent mini max from making calculations that are 100% guaranteed to NOT be an optimal state foe the current player (MAX or MIN)

You may however have equivalent actions for a given state. How your algorithm decides which equivalent actions to return depends on how it is implemented. It sets/unordered lists are used somewhere the order in which evaluations are made any change.

This, may also depends on what you do if Alpha/Beta value is equal to the current best option. Since equal value would not produce a better result, there's no point in exploring that path further. Therefore, you would just keep the "First best action encountered". However, with mini max, you explore everything anyways, so you may decide to keep the "last best" value. That's one action than Alpha - Beta. But they are still equivalent as far as your scoring function is concerned.

**11. Why searching is necessary in AI? (2070 Magh)**

⇒ See on old question no 4

**12. Differentiate between informed and blind search. How depth search is different to breadth first search. Compare with evaluation parameters. (2070 Bhadra)**

⇒ **1st part:** See on old question no 10

**2nd part:** See on page no5 3rd part see on old question no 9

**13. Compare the search strategy of beadth first search with depth first search. (2070 Chaitra)**

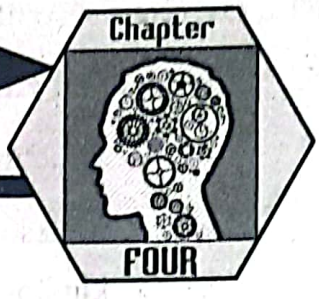
⇒ See on page no5

**14. Discuss the hill-climbing search algorithm along with the problem associated with it and discuss their solutions. (2069 Bhadra)**

⇒ See on page no46

⌘⌘⌘

# Knowledge Representation, Inference and Meaning




## Knowledge

Knowledge is awareness or Familiarity gained by experiences of facts, data and situations.

## Types of Knowledge in AI

1. **Declarative knowledge**
   - Declarative knowledge is to know about something.
   - It includes concepts, facts and objects.
   - It is also called descriptive knowledge and expressed in declarative sentences.
   - It is simpler than procedural knowledge.
2. **Procedural Knowledge**
   - It is also known as imperative knowledge.
   - Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
   - It can be directly applied to any task.
   - In includes rules, strategies, procedures, agendas etc.
   - Procedural knowledge depends on the task on which it can be applied.
3. **Meta - Knowledge**
   - Knowledge about the other types of knowledge is called Meta - Knowledge.
4. **Heuristic knowledge**
   - Heuristic knowledge is representing knowledge of some experts in a file or subject.
   - Heuristic knowledge is rules of thumb based on previous experiences awareness of approaches and which are good to work but not guaranteed.

5. **Structural Knowledge**

- Structural knowledge is basic knowledge to problem - solving.
- It describes relationships between various concepts such as kind of part of, and grouping of something.
- It describes, the relationship that exists between concepts or objects.

**Knowledge representation using Logic**

- Logic is defined as a formal language for expressing knowledge and ways of reasoning.
- Logic makes statement about the world which are true (or false).
- Compared to natural languages (expressive but context sensitive) and programming languages (good for concrete data structure but not expensive) logic combines the advantages of natural languages and formal languages.
- Logic is:
  - Concise
  - Unambiguous
  - Context insensitive
  - Expressive
  - Effective for inferences
  - A logic is defined by the following:

1. **Syntax** - describes the possible configurations that constitute sentences.
2. **Semantics** - determine what facts in the world the sentences refer to i.e. the interpretation. Each sentence makes a claim about the world.
3. **Proof Theory** - Set of rules for generating new sentences that are necessarily true given that old sentences are true. The relationship between sentences is called entailment. The semantics link these sentences (representation) to facts of the world. The proof can be used to determine new facts which follow from the old.

4. **A set of sentences** - A sentences is constructed from a set of primitives according to syntax rules.

5. **A set of interpretation** - An interpretation gives a semantic to primitives. It associates primitives with value.

**Types of Logic**

i) Propositional

ii) First - Order Predicate Logic

**Propositional Logic**

Propositional logic (PL) is the simplest form of logic where all the statements are made by propositions. A proposition is a declarative statement which is either true or false. It is a technique of knowledge representation in and mathematical form.

E.g:

i) It is Sunday

ii) The sun rises from west (false proposition)

iii) 3 + 3 = 7(False proposition)

iv) 5 is a prime number

**Syntax of propositional Logic**

a) **Atomic Proposition**

Atomic proposition are the simple proposition. It consists of a single proposition symbol. These are the sentences which must be either true or false.

E.g:

(i) 2 + 2 is 4, it is an atomic proposition.

(ii) "The sun is cold".

b) **Compound Proposition**

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

E.g:

(i) " It is raining today and street is wet"

(ii) "Ankit is a doctor and his clinic is in Mumbai"

**Logical connections**

Logical connections are used to connect two simpler propositions or representing a sentence logically. We can create compound propositions with the help of logical connectives.

i) **Negation:** A sentences such as $\neg$ p is called negation of P.

ii) **Conjuction:** A sentence which has $\wedge$ connective such as $P \wedge Q$ is called a conjuction.

E.g:

Rohan is intelligent and hard working. It can be written as,

P = Rohan is intelligent

Q = Rohan is hardworking

$\therefore P \wedge Q$

iii) **Disjunction:** A sentence which has V connective, such as $P \vee Q$ is called disjunction, where P and Q are the propositions.

E.g:

"Ritika is a Doctor or Engineers"

Here,

P = Ritika is a Doctor

Q = Ritika is a Engineer

So, we can write it as $P \vee Q$

iv) **Implication:** A sentences such as $P \rightarrow Q$, is called an implication. Implications are also known as if - then rules. It can be represented as

If it is raining, then the street is wet. Let, P = It is raining and Q = Street is wet. So, it is represented as $P \rightarrow Q$.

v) **Biconditional:** A sentence such as $P \leftrightarrow Q$ is a biconditional sentence, example.

If I breathing, then I am live

P = I am breathing

Q = I am alive

$\therefore P \leftrightarrow Q$

## Truth Table

In Propositional logic, we need to know the truth values of propositions in all possible scenarios. We can combine all the possible combination with logical connectives, and the representation of these combinations in a tabular format is called Truth Table.

Following are the truth table for all logical connectives:

**For Negation**

| P | ⌉P |
|---|---|
| True | False |
| False | True |

**For Conjuction**

| P | Q | P∧Q |
|---|---|---|
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**For Disjunction**

| P | Q | PvQ |
|---|---|---|
| True | True | True |
| False | True | True |
| True | False | True |
| False | False | False |

**|For Implication**

| P | Q | P → Q |
|---|---|---|
| True | True | True |
| True | False | False |
| False | True | True |
| False | False | True |

**For Biconditional**

| P | Q | P → Q |
|---|---|---|
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | True |

| P | Q | P ↔ Q |
|---|---|---|
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**Precedence of Connections**

(i) ( )

(ii) ⌉

(iii) ∧

(iv) V

(v) →

(vi) ↔

**Limitations of propositional Logic**

- We cannot represent relations like ALL, some, or none with propositional logic. E.g:
  - a) All the girls are intelligent
  - b) Some apples are sweet.
- Propositional logic has limited expressive power.
- In propositional logic, we cannot describe statements informs of their properties or logical relationships.

**Laws of Proposition**

i) **Commutatively**

- $P \wedge Q = Q \wedge P$ or
- $P \vee Q = Q \vee P$

ii) **Associativity**

- $(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$
- $(P \vee Q) \vee R = P \vee (Q \vee R)$

iii) **Identity element**

- $P \wedge True = P$
- $P \vee True = True$

iv) **Distributive**

- $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$
- $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$

v) **De - Morgan's law**

- $\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$
- $\neg (P \vee Q) = (\neg P) \wedge (\neg Q)$

vi) **Double negation elimination**

- $\neg (\neg P) = P$

**Inference rules**

Inference rules are the template for generating valid arguments. Inference rules are applied to derive proofs in artificial intelligence, and the proof is a sequence of the conclusion that leads to the desired goal.

| | Rule of inference | Tactology | Name |
|---|---|---|---|
| i) | $P \rightarrow q$<br>$P$<br>$\therefore q$ | $[p \wedge (p \rightarrow q)] \rightarrow q$ | Modus Porens |
| ii) | $\rightarrow q$<br>$P$<br>$\therefore q$ | $[p \wedge (p \rightarrow q)] \rightarrow p$ | Modus Tollen |
| iii) | $\rightarrow q$<br>$P$<br>$\therefore q \rightarrow r$ | $[(p \rightarrow q)] \vee (q \rightarrow r)] \rightarrow (p \rightarrow r)$ | Hypothetical Syllogism |
| iv) | $p \rightarrow q$<br>$p$<br>$\therefore q$ | $((p \vee q) \vee p \rightarrow q$ | Disjunctive Syllogism |
| v) | $p$<br>$\therefore p \vee q$ | $p \rightarrow (p \vee q)$ | Addition |
| vi) | $p \wedge q$<br>$\therefore p$ | $(p \wedge q) \rightarrow p$ | Simplification |
| vii) | $p$<br>$q$<br>$\therefore p \wedge q$ | $((p) \wedge (q)) \rightarrow (p \wedge q)$ | Conjuction |
| vii) | $p \vee q$<br>$p \vee r$<br>$\therefore q \vee r$ | $[((p \vee q) \wedge (q \vee r)) \rightarrow (p \vee r)$ | Resolution |

**First order Logic (Predicate Logic)**

- First order is logic another way of knowledge representation in artificial intelligence. It is an extension to propositional logic.
- FOL is sufficiently expensive to represent the natural language statements in a concise way.
- First - order logic is also known as predicate logic or first - order predicate logic.

- First - order logic does not only assume that the world contains facts propositional logic but also assumes the following things in the world.

  i) **Objects:** A, B, people, numbers, colours, wars etc.

  ii) **Relations:** It can be unary relation such as red, round, is adjacent, or n - any relation such as the sister of, brother of, has color, comes between.

  iii) **Function:** Father of, best friend, third inning of etc.

  As a natural language, First - order logic also has two main parts.

  a) Syntax

  b) Semantics

**Syntax of First - order Logic**

The syntax is FOL determines which collection of symbols is a logical expression in first - order logic. The basic syntactic elements of First - order logic are symbols. We write statements in short - hand notation in FOL.

**Basic elements of First - Order Logic**

| | |
|---|---|
| Constant | 1, 2, A, John, Mumbai etc. |
| Variables | X, Y, Z, a, b |
| Predicates | Brother, Father, > |
| Function | Start, Left, Lego F |
| Connections | $\wedge$, v, $\neg$, $\Rightarrow$, $\leftrightarrow$ |
| Equality | =, = |
| Quantifier | $\forall$, $\exists$ |

**Atomic Sentences**

- Atomic sentences are the most basic sentences of first - order logic. These sentences are formed from a predicate symbol followed by a parenthesis with a sequence of terms.

- We can sequence atomic sentence as predicate (term1, term2, ....., term n)

  E.g: Ravi and Ajay are brothers: $\Rightarrow$ Brothers (Ravi, Ajay)

  Chinky is a cat: $\Rightarrow$ cat (Chinky)

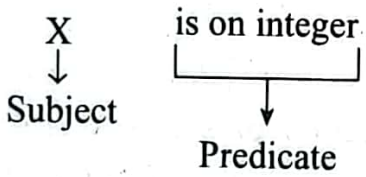
**Complex Sentences**

- Complex sentences are made by combining atomic sentence using connectives.

**First - order logic statements can be divided into two parts:**

- **Subject:** Subject is the main part of the statement.
- **Predicate:** A predicate can be defined as a relation, which binds two atoms together in a statement.

  **Consider the statement:** "X is integer" it consists of two parts, the first part X is the subject of the statement and second part "is an integer", is known as predicate.




**Quantifies in First - order Logic**

- A quantifies is a language element which generates quantification and quantification specifies the quantity of specimen in the universe of discourse.
- These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression.
  - **a)** Universal Quantifier (For all, everyone, everything)
  - **b)** Existential Quantifier (For some, at least one)

**Universal Quantifier**

Universal Quantifiers is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing.

The universal quantifiers is represented by a symbol ∀, which resembles an inverted A.

If X is a variable, the ∀x is read as:

- For all X
- For each X
- For every X

  e.g: All man drink coffee

⇒ Let, a variable X which refers to a cat so all X can be represented in UOD as below:

- $X_1$ drinks coffee

  ∧

- $x_2$ drinks

  ∧

- $x_3$ drinks milks

  ∧

- Universe of Discourse

- $x_n$ drinks milks

  ⇓

So, in shorthand notation, we can write it as:

∀x man, (x) → drink (x, coffee)

It will be read as: The are all x where x is man who drink coffee.

**Existential Quantifier**

Existential Quantifier are the type of quantifier which express that the statement within its scope is true for at least one instance of something. It is denoted by the logical operator ∃, which resembles as inverted E. When it is used with a predicate variable then it is called as existential quantifier.

If X is a variable, then existential quantifier will be ∃x or ∃(x). And it will be read as:

- There exists a 'X'
- For exam 'X'

- For at least one 'X'

  Example:

  - $x_1$ boys are intelligent

    $\vee$

  - $x_2$ is intelligent

    $\vee$

  - $x_3$ is intelligent

    $\vee$

    $X_1, X_2, X_3, X_4$
    $X_5, X_n$ Boys

    Universe of Discourse

  - $\vee$

  - $x_n$ is intelligent

    $\Downarrow$

  So, in short - hand notation, we can write it as:

  $\exists x$: boy (x) $\wedge$ intelligent (x)

  It will be read as: There are some X where X is a boy who is intelligent.

**Points to remember**

- The main connective for universal quantifier $\forall$ is implication $\rightarrow$.
- The main connective for existential quantifier $\exists$ is and $\wedge$.

**Some examples of FOLP using quantifiers.**

1. All birds Fly

   $\forall x$ bird (x) $\rightarrow$ Fly (x)

2. Every man respects his parent.

   $\forall x$ man (x) $\rightarrow$ respects (x, parent)

3. Some boys play cricket.

   $\exists x$ boy (x) $\rightarrow$ Play (x, cricket)

4. Not all students like both mathematics and science.

   $\neg \forall(x)$ [students (x) $\rightarrow$ like (x, mathematics) $\wedge$ like (x, science)]

5. Only one students failed in mathematics

   $\exists(x)$ [students (x) $\rightarrow$ Failed (x, mathematics)]

**Resolution**

Resolution is a theorem proving technique that proceeds by building reputation proofs i.e. proofs by contradictions.

Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Resolution is a single inference rule which can efficiently operate on the conjuction normal form or clausal form.

Clause: Disjunction of literals is called a clause.

**The resolution inference rule**

The resolution rule for first - order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assured to be standarized a part so that they share no variables.

$$\frac{I_{IV}.....V_{IK}.....m_{IV}.....V_{mn}}{SUBST(\theta I_{IV}.....VI_{i-1}.....VI_{i+1V}.....VI_{k}V_{min}..... V_{mj-iv}V_{mj+1v}....V_{mn})}$$

Where Ii and mj are complementary literals. This rule is also called the binary resolution rule because it only resolves exactly two literals.

E.g: We can resolve two clauses which are given below:

[Animal (g (x) ∨ Loses (F (x), x] and [ ⌉Loses (a, b) ∨ ⌉kills (a, b)]

Where two complimentary literals are: Loses (f (x), x) and ⌉loses (a, b)

These literals can be unified with unified θ = [a/f(x)], and b/x] and it will generate a resolvant clause.

[Animal (g(x) ∨ ⌉f(x), (x)]

**Steps for resolution**

1. Conversion of facts into first - order logic.
2. Negate FOL statements into CNF.
3. Negate the statement which needs to prove.
4. Draw resolution graph (unification)

   E.g:

   a) John likes all kind of food.
   b) Apple and vegetables are food.
   c) Anything anyone eats and not killed is food.
   d) Anil eats peanut and still alive.
   e) Harry eats everything that Anil eats.

   Prove by resolution that:

   John likes peanuts.

**Step 1: Conversion of facts into FOL.**

a) $\forall x$: Food (x) $\rightarrow$ likes (John, x)

b) Food (Apple) $\wedge$ food (vegetables)

c) $\forall x\ \forall y$: eats (x, y) $\wedge \neg$killed (x) $\rightarrow$ Food (y)

d) Eats (Anil, Peanut) $\neg$alive (Anil)

e) $\forall x$: eats (Anil, x) $\rightarrow$ eats (Harry, x)

f) $\forall x$: $\neg$killed (x) $\rightarrow$ alive (x)

g) $\forall x$: alive (x) $\rightarrow$ $\neg$Killed (x)) added predicates.

h) Likes (John, peanuts)

**Step 2: Conversion of FOL into CNF**

- **Eliminate all implication ($\rightarrow$) and rewrite**

  a) $\forall x$ $\neg$Food (x) $\wedge$ likes (John, x)

  b) Food (Apple) v Food (vegetables)

  c) $\forall x\ \forall y$ $\neg$[eats (x, y) v $\neg$Killed (x)] $\vee$ foodly)

  d) Eats (Anil, peanuts) v Alive (Anil)

  e) $\forall x$ $\neg$eats (Anil), x) $\wedge$ eats (Harry, x)

  f) $\forall x$ $\neg$[$\neg$ killed (x)] $\wedge$ alive (x)

  g) $\forall x$ $\neg$alive (x) $\wedge$ $\neg$killed (x)

  h) Likes (John, Peanuts

- **More Regation (7) inwards and rewrite**

  (a) $\forall x$ $\neg$ Food (x) $\vee$ likes (John, x)

  (b) Food (Apple) $\wedge$ Food (Vegetables)

  (c) $\forall x\ \forall y$ eats (x, y) $\vee$ killed (x) $\vee$ killed (x) $\vee$ food (y)

  (d) Eats (Anil, peanuts) $\wedge$ Killed (x) $\vee$ Food (y)

  (e) $\forall x$ $\neg$ eats (Anil, x) $\vee$ eats (Harry, x)

  (f) $\forall x$ $\neg$ killed (x) $\vee$ alive (x)

  (g) $\forall x$ $\neg$ alive (x) $\vee$ $\neg$ killed (x)

  (h) likes (John, Peanuts)

- **Rename variable or standardize variables**

  (a) $\forall x \neg$ Food (x) $\vee$ likes (John, x)

  (b) Food (Apple) $\wedge$ Food (vegetables)

  (c) $\forall y \forall z \neg$ eats (y, z) $\vee$ killed (y) $\vee$ Food (z)

  (d) eats (Anil, Peanuts) $\wedge$ alive (Anil)

  (e) $\forall w \neg$ eats (Anil, w) $\vee$ eats (Anil)

  (f) $\forall g \neg$ Killer (g) $\vee$ alive (Harry, w)

  (g) $\forall k \neg$ alive (k) $\vee \neg$ Killed (k)

  (h) likes (John, parents)

- **Eliminate existential instantiation quantifier by elimination.**

  In this step, we will eliminate existential quantifier $\exists$, and this process is known as skolemization. But in this example problem since there is no existential quantifier so all the statements will remain same in this step.

- **Drop Universal quantifier**

  (a) $\neg$ Food (x) $\vee$ likes (John, x)

  (b) Food (Apple)

  (c) Food (Vegetables)

  (d) $\neg$ eats (y, z) $\vee$ killed (y) $\vee$ Food (z)

  (e) eats (Anil, Peanuts)

  (f) alive (Anil)

  (g) $\neg$ eats (Anil, w) $\vee$ eats (Harry, w)

  (h) Killed (g) $\vee$ alive (g)

  (i) $\neg$ alive (g) $\vee$ alive (g)

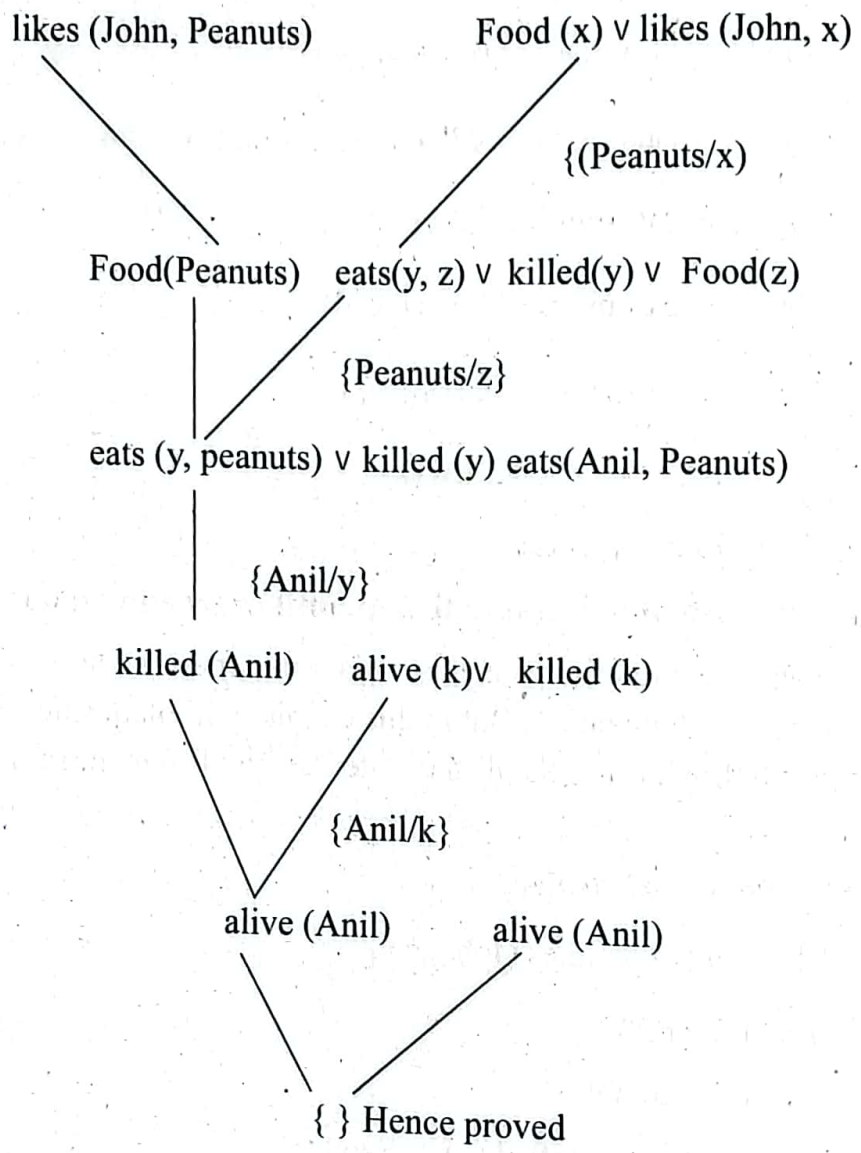
  (j) likes (John, peanuts)

**Step 3:**

Negate the statement to be proved. In this statement, we will apply negation to the conclusion statements, which will be written as $\neg$ likes (John, Peanuts).

**Step 4:**

Draw Resolution graph.




**Rule Based System**

Rule Based system (RBS) provide automatic problem solving tools for capturing the human expertise and decision making Rule based system is a set of "if - then" statements that uses a set of assertions, to which rules on how to act upon those assertions are created.

**Forward Chaining**

Forward chaining is also known as a forward deduction or forward reasoning method when using an inference engine. Forward chaining is a form of reasoning which start with atomic sentences in the knowledgebase and applies inference rules (Modus Ponens) in the forward directions to extract more data until a goal is reached.

The forward - chaining algorithm starts from known facts, triggers, all rules whose premises are satisfied and add their conclusion to the known facts. This repeated until the problem is solved.

**Properties of forward - chaining**

- It is a down - up approach, as it moves from bottom to top.
- It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.
- Forward - chaining approach is called as data - driven as we reach to the goal using available data.
- Forward - chaining approach is commonly used in the expert system such as CLIPS, business and production rule system.

E.g:

"As per law it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American Citizen".

1) **Facts conversion in FOL:**

- It is a crime for an American to sell weapons to hostile nations.

  American (p) ∨ Weapon (q) ∧ sells, (p, q, r) ∧ hostile (r) → criminal (p) .... (i)

- Country A has some missiles (p)

  owns (A, P) ∨ Missile (P)

  owns (A, T1) ... (ii)

  Missile (T1) ... (iii)

- All of the missiles were sold to country A by Robert.

  Missile (P) ∧ owns (A, P) → Sells (Robert, P, A) ... (iv)

- Missiles are weapons

  Missile (P) → weapons(P) as

- Enemy of America is known as hostile.

  Enemy (P, America) → Hostile(P) – (vi)

- Country A is an enemy of America

  Enemy (A, America)

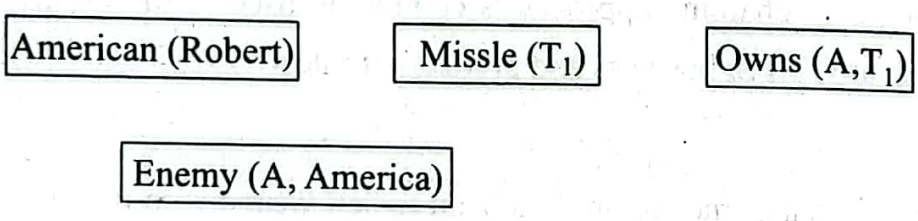
- Robert is American

  American (Robert) – (viii)

**Forward chaining proof:**

**Step 1:**

In the first step we will start with the known facts and will choose the sentences which do not have implications such as American (Robert), Enemy (A, America), owns (A, T1) and Missile (T1). All these facts will be represented as below.




**Step 2:**

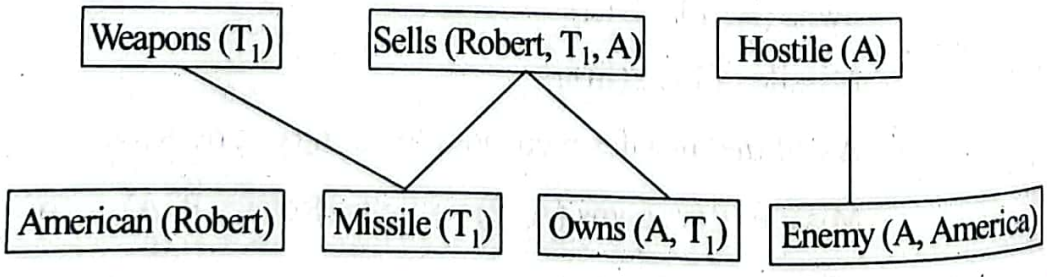
At the second step, we will see those facts which infer from available facts and with satisfied premises.

Rule (1) does not satisfy premises, so it will not be added in the first interaction.

Rule (2) and (3) are already added.

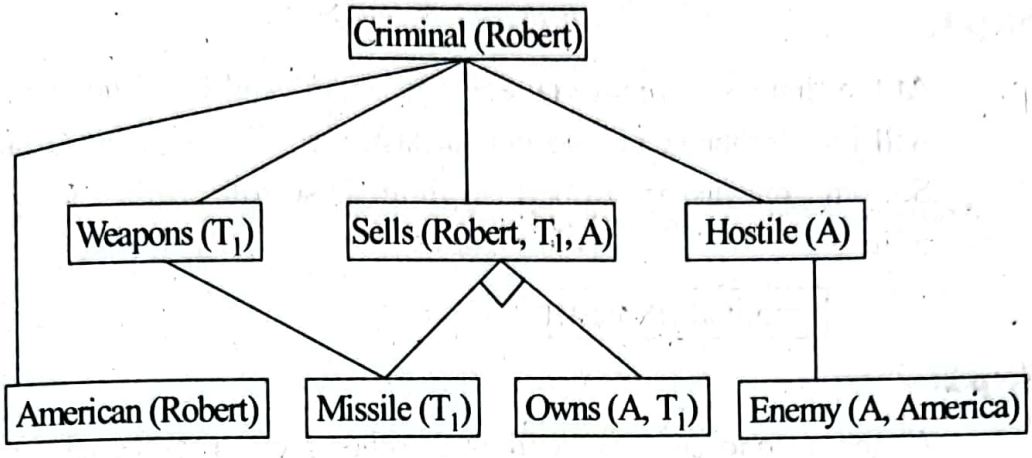
Rule (4) satisfy with the substitution {P/T1} so sells (Robert, T1, A) is added which infers from the conjunction of rule (2) and (3).

Rule (6) is satisfied with the substitution (P/A). So hostile (A) is added and which infers from rule (7).




**Step 3:**

At step - 3 as we can check rule (1) is satisfied with the substitution {P/Robert, 4/T1, r/A3, so we can add criminal (Robert) which infers all the available facts. And hence we reached our goal statement.




Hence, it is proved that Robert is criminal using forward chaining approach.

**Backward Chaining**

Backward - chaining is also known as a backward deduction or backward reasoning method using an inference engine. A back - ward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.

**Properties of Backward Chaining**

- It is known as a top - down approach.
- Backward - chaining is based on modus ponens inference rule.
- In backward - chaining, the goal is broken into sub - goal or sub - goals to prove the facts true.
- It is called a goal - driven approach, as a list of goals decides which rules selected and used.
- Backward - chaining algorithm is used in games theory, automated theorem proving tools, inference engines, Proof assistants and various AI applications.
- The backward - chaining method mostly used a depth - first search strategy for proof.

E.g:

**Backward - Chaining Proof:**

In backward chaining, we will start with our goal predicate, which is criminal (Robert) and then infer further rules.

**Step 1:**

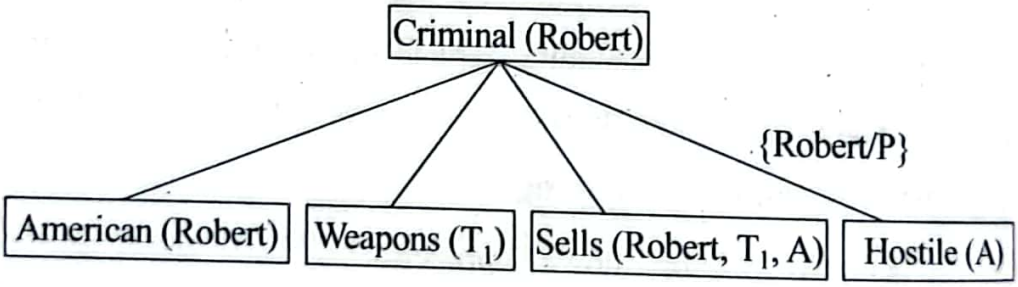
At the first step, we will take the goal fact. And from the goal fact, we will infer other facts, and gets at last, we will prove those facts true. So, our goal fact is "Robert is criminal" so following is the predicate of it.

Criminal (Robert)

**Step 2:**

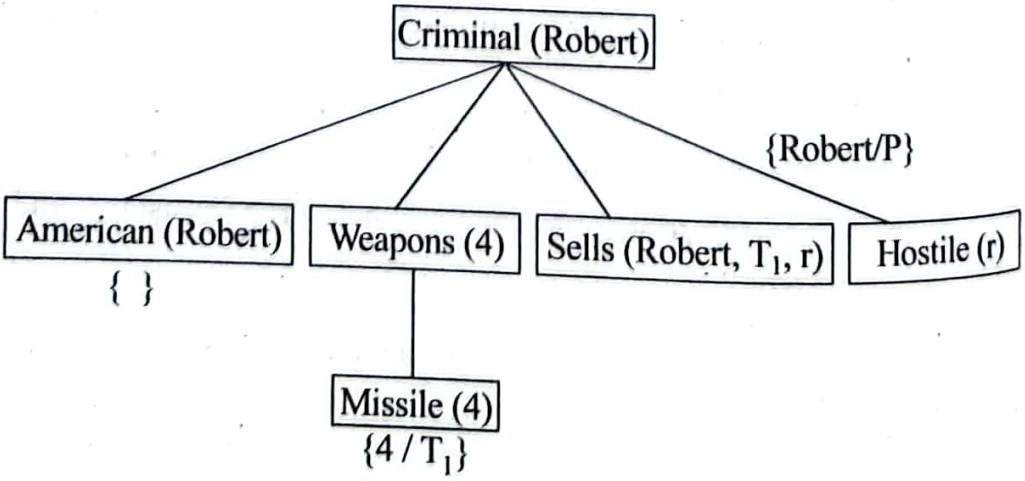
At the second step, we will infer other facts. From goal fact which satisfies the rule. So as we can see in Rule - 1, the goal predicate criminal (Robert) is present with substitution {Robert/P}. So we will add all the connective facts below the first level and will replace P with Robert.

Here, we can see American (Robert) is a fact, so it is proved here.



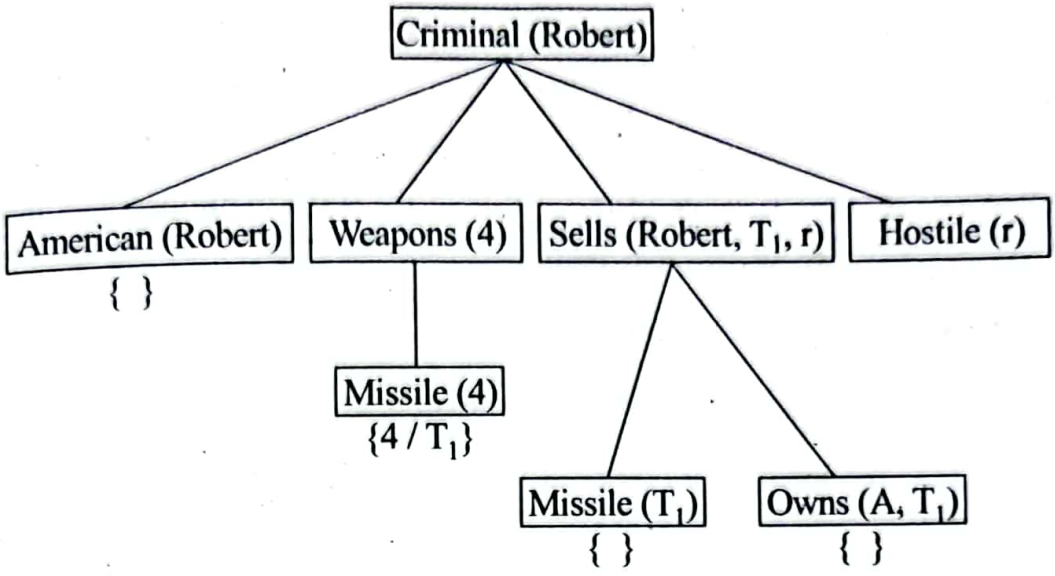

**Step 3:**

At step 3 we will extract further fact missile (4) Which infer from Weapon (4), as it satisfies Rule (5). Weapon (4) is also true with the substitution of a constant T1 as 4.



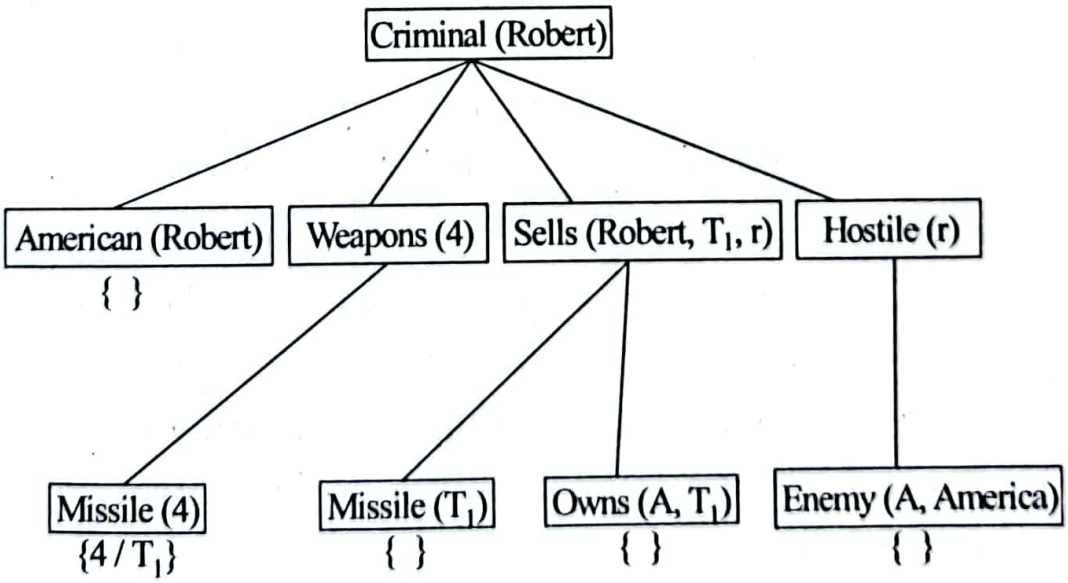

**Step 4:**

At step - 4, we can infer facts Missile (T1) and owns (A, T1) from sells (Robert, T1, r) which satisfies the rule - 4 with the substitution of A in the place of r. So, there two statements are proved here.




**Step 5:**

At step - 5 we can infer the fact Enemy (A, American) from Hostile (A) which satisfies rule -6. And hence all the statements are proved true using backward chaining.




**Stastical Reasoning Probability**

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainity in Knowledge. In Probabilistic reasoning, we combine Probability theory with logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of some one's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today" "behaviour of someone for some situations" "A match between two teams or two players". These are probable sentences for which we can assume that it will happen but not sure about it so here we use probabilistic reasoning.

**Need of probabilistic reasoning in AI.**

- When there are unpredictable outcomes.
- When specification or possibilities of predicate becomes too large to handle.
- When an unknown error occur during an experiment.

In Probabilistic reasoning, there are two ways to solve problems with uncertain knowledge.

- Bay's rule
- Bayesian statstics

**Baye's Theorem**

Baye's theorem is also known as Baye's rule. Bayes law or Bayesian reasoning which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Baye's theorem was named after the British mathematician Thomas Bayes. The Bayesian inference is an application of Baye's Theorem, which is fundamental to Bayesian statstics.

It is a way to calculate the value of P(B/A) with the knowledge of P(A/B).

Baye's theorem allows updating the probability prediction of an event by observing new information of the real world.

E.g: If cancer corresponds to one's age then by using Baye's theorem, we can determine the probability of cancer more accurately with the help of age.

Baye's theorem can be derived using product rule and conditional probability of event A with known event B.

**As from product we can write:**

$$P(A \wedge B) = P(A/B)\, P(B) \text{ or,}$$

Similarly, the probability of event B with known event A.

$$P(A \wedge B) = P(B/A)\, P(A)$$

Equating right hand side of both the equations, we get

$$\frac{P(B/A)\, P(A)}{P(B)} \quad ....(a)$$

The above equation (a) is called as Baye's rule or Baye's theorem. This equation is basic of most modern AI system for probabilities inference.

It shows the simple relationship between joint and conditional probabilities. Here, P(A/B) is known as posterior, which we need to calculate , and it will be read as probability of hypothesis A when we have occurred an evidence B.

P(B/A) is called the likelihood, in which we consider that hypothesis is ture, then we calculate the probability of evidence.

P(A) is called the prior probability, probability of hypothesis before considering the evidence.

P(B) is called marginal probability, pure probability of an evidence.

In the equation (a) in general, we can write P(B) = P(A) * P(B/A) hence the Baye's rule can be written as:

$$P(Ai/B) = \frac{P(Ai) * P(B/Ai)}{\sum_{i=1}^{k} P(Ai) * P(B/Ai)}$$

Where A, $A_2$, $A_3$ ___ An is a set of mutually exclusive and exhaustive events.

**Applying Baye's Rule:**

Baye's rule allows us to compute the single term P(B/A) in terms of P(A/B), P(B) and P(A). This is very useful in cases where we have a good probability of these three terms and want to determine the fourth one. Suppose we want to perceive the effect of some unknown cause, and want to compute that cause, then the Baye's rule becomes.

$$P(cause/effect) = \frac{P(effect/cause)\, P(cause)}{P(effect)}$$

1. **What is the probability that a patients has diseases meningitis with a stiff neck?**

⇒ A doctor is aware that disease meningitis causes a patient to have a stiff neck, and it occurs 80% of the time. He is also aware of some more facts, which are given as follows:

- The known probability that a patient has meningitis disease is 1/30,000.
- The known probability that a patient has a stiff neck is 2%.

Let, be the proposition that patient has stiff neck and be the proposition that patient has meningitis, so we can calculate the following as:

$P(a/b) = 0.8$

$$P(b) = \frac{1}{30,000}$$

$P(a) = 0.2$

$$P(b/a) = \frac{P(a/b)\,P(b)}{P(a)}$$

$$= \frac{0.8 * \frac{1}{30,000}}{0.02}$$

$$= 0.001333333$$

Hence, we can assume that 1 patient out of 750 patients has meningitis disease with a stiff neck.

2. **From a standard deck of playing cards, a single card is drawn. The probability that the card is king is 4/52, then calculate posterior probability P(King/Face) which means the drawn face card is a king card.**

⇒ $$P(King/Face) = P(King/Face) = \frac{P(Face/King) \times P(King)}{P(Face)} \quad ...(i)$$

P(King) : Probability that the card is king $= \frac{4}{52} = \frac{1}{13}$

P(Face) : Probability that a card is a face card $= \frac{3}{13}$

P(Face/King): Probability of face card when we assume it is a king = 1

Putting all values in eqn (i) we will get:

$$P(King/Face) = \frac{1 * (1/13)}{(3/13)} = \frac{1}{3}$$

It is a probability that a face card is a king card.

**Bayesian Belief Network**

A Bayesian network is a probabilistic graphical model represents a set of variables and their conditional dependencies using a directed a cyclic graph. It is also called a Baye's network, belief network, decision network or Bayesian model.

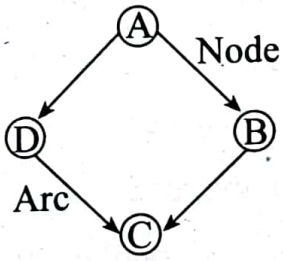
Bayesian network are probabilistic because these networks are built from a probability distribution and also use probability theory for prediction and anamoly detection.

Bayesian network can be used for building models from data and expert opinions, and it consists of two parts:

i) Directed Acyclic graph

ii) Table of conditional probabilities

**A Bayesian network graph is made up of nodes and Arcs (directed links) where:**




- Each mode corresponds to the random variables, and a variables can be continuous or discrete.
- Arc or directed arrows represent the causal relationship or conditional probabilities between random variables. These directed links or arrows convert the pair of nodes in the graph.

  These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other.
- In the above diagrams A, B, C and D are random variables represented by the nodes of the network graph.
- If we are considering node B, which is connected with node A by a directed arrow then node A is called the parent of node B.
- Node C is independent of node A.

The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed cyclic graph or DAG.

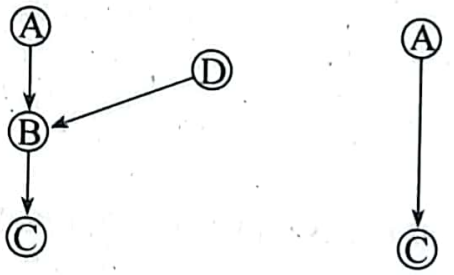
**The Bayesian network has many mainly two components:**

- Causal components
- Actual numbers

Each node in the Bayesian network has probability distribution P(xi / parent (xi)), which determines the effect of the parent on that node.

Bayesian network is based on Joint probability distribution and condition probability. So let's first understand the joint probability distribution.

E.g:




Let, P(A) denote the event that you wake up late in the morning.

P(B) denotes the event you are late for school.

P(C) denotes the event you are scolded by your teacher.

P(D) denotes the event there is a traffic jam on the road.

P(E) denotes the event your father boards the public bus every morning.

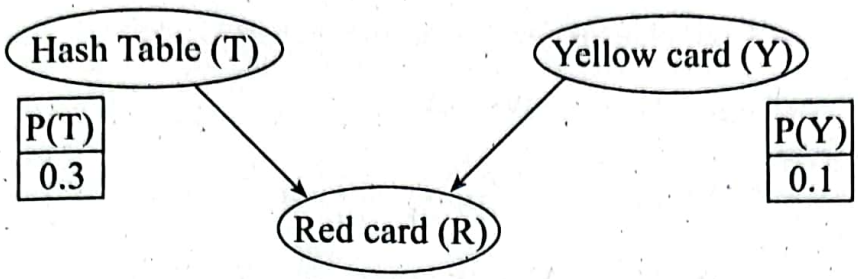
P(F) denotes the event your father goes to office.

So what is the probability that you are scolded at school? Now, you can easily find that out using the formula. So the final result becomes.

P(C/B), P(B/A,D), P(D), P(A)

Hey, but wait !!! what about P(E) and P(F) ?They are also present there in the model. Remember, two events which are not connected with each other with arcs directly or indirectly given any other node are called independent events and occurrence of one has no effect on the occurrence of another. We can safely predict from the model the value of P(C/F) i.e. the probability you being scolded because of the event that your father is going to Office in O.

The probability value for each node given their parent node is represented in the form of a table called conditional probability Table (CPT). The below examples is a representation of CPT.




| P(T) | P(Y) | P(R) |
|---|---|---|
| F | F | 0.02 |
| F | T | 0.2 |
| T | F | 0.5 |
| T | T | 0.97 |

*Fig: Conditional probability Table*

**OLD QUESTION SOLUTION**

1. **Consider the following aximos. All hounds howl at night. Anyone who has any cats will not have any mice. Light sleepers do not have anything which howls at night. John has either a cat or a hound. Prove: "If John is a light sleeper then John does not have any mice". By using resolution refutation. (2076 Baisakh)**

⇒ **The clauses written for the above aximos are shown below:**

1. $\forall x\ (HOW(x) \rightarrow HOWL(x))$
2. $\forall x\ \forall y\ (HAVE\ (x, y) \wedge (AT\ (Y) \rightarrow \exists z\ (HAVE\ (x, z) \wedge MOUSE\ (Z)))$
3. $\forall x\ (LS\ (x) \rightarrow \exists y\ (HAVE\ (x, y) \wedge HOWL\ (y))$
4. $\exists x\ (HAVE\ (JOHN, x) \wedge (AT\ (x) \vee HOUND\ (x))$
5. $LS\ (JOHN) \rightarrow \exists z\ (HAVE\ (JOHN, Z \wedge MOUSE\ (Z))$

The next is to transform each WFF into prenex Normal from, Skolemize and rewrite as clauses in conjuction normal form.

i) $\neg\ HOUND\ (x) \vee HOWL\ (x))$

ii) $\neg\ HAVE\ (x, y) \vee \neg\ (AT\ (Y) \vee \neg\ HAVE\ (x, z) \wedge MOUSE\ (z))$

iii) $\neg\ LS\ (x) \vee \neg\ HAVE\ (x, y) \vee \neg\ HOWL\ (Y)$

iv) $HAVE\ (John, a) \wedge (CAT(a) \vee HOUND\ (a))$

v) $LS\ (John) \wedge HAVE\ (John, b) \wedge MOUSE\ (b)$

The set of CNF clauses for this problem are as follows:

1. ⌉ HOUND ∨ HOWL (x)
2. ⌉ HAVE (x, y) ∨ ⌉ (AT (y) ∨ ⌉ HAVE (x, z) ∨ ⌉ MOUSE (z)
3. ⌉ LS (x) ∨ ⌉ HAVE (x, y) ∨ ⌉ HOWL (y)
4. i) HAVE (John, a)
   ii) CAT (a) ∨ HOUND (a)
5. i) LS (John)
   ii) HAVE (John, b)
   ii) MOUSE (b)

Now,

6. CAT (a) ∨ HOWL (a) From 1 and 4(b)
7. ⌉ HAVE (x, y) ∨ ⌉ CAT (Y) ∨ ⌉ (a, b) FROM 2 and 5 (c)
8. ⌉ HAVE (John, y) ∨ ⌉ CAT (y) From 7 and 5 (b)
9. ⌉ Have (John, a) ∨ HOWL (a) From 6 and 8
10. HOWL (a) From 4 (a) and 9
11. ⌉ LS (X) ∨ ⌉ HAVE (x, a) Form 3 and 10.
12. ⌉ LS (John) From 4(a) and 11
13. ⌉ From 5(a) and 12.

**2. Why CNF is necessary? "Everyone who loves all animals are loved by someone" represent this statement in FOPL and explain all the steps involved to convert it into CFN. (2075 Bhadra)**

⇒ Conjuction normal formal (CNF) is an approach to Boolean logic that expresses formulas as conjuction of clauses with an AND or OR. Each clauses connected by a conjuction, or AND must be either a literal or contain a disjunction or OR operator. CNF is useful for automated theorem proving.

**In FOPL**

∀x [∀y [Animal (y) → Loves (x, y) ]]→ ∃ z Loves (z, x)}]

Now, converting into CNF

i) Remove Implications

$\forall x [\neg [\forall y \neg \text{Animal}(y) \vee \text{Loves}(x, y)]] \vee [\exists z \text{ Loves}(z, x)]$

ii) Move negation inward

$\forall x[\exists y \text{ Animal}(y) \wedge \text{Loves}(x, y)] \vee [\exists z \text{ loves}(z, x)$

iii) Skolemize

$\forall x [\text{Animal}(F(x)) \wedge \neg \text{Loves}(x, F(x))] \vee [\text{Loves}(a(x), x)]$

iv) Drop universal quantifier

$[\text{Animal}(f(x) \wedge \neg \text{Loves}(x, F(x))] \vee [\text{Loves}(a(x), x)]$

v) Use Distribute Law

- Animal (F(x) Loves (a(x), x)
- $\neg$ Loves (x, F(x) Loves (a(x) ,x)

3. **Given premises" Every American who sells weapons to hostile nations is criminal. The country XYZ is enemy of America. All of its misallies in XYZ were sold by Donald, who is an American."Prove that Donald is a criminal by using FOPL based resolution refutation method.(2075 Baisakh)**

⇒ **Similar to page no74**

4. **Why CNF is required? Explain all the steps used to convert a quantified statement with suitable example.(2075 Baisakh)**

⇒ **1st part:** See on old question no 2

**2nd part:** See on old question no 19

5. **Assume the following facts**

- **John likes all kinds of food.**
- **Apples are food**
- **Chicken is food**
- **Anything anyone eats and isn't likes, killed by is food.**
- **Bill eats peanuts and is still alive**
- **Sue eats everything bill eats.**

**Prove that John likes using resolution**

**(2074 Bhadra, 2072 Ashwin)**

⇒ **In FOPL**

i) $\forall x$: Food (x) → likes (John, x)

ii) Food (apple)

iii) Food (chicken)

iv) $\forall x: (\exists y: \text{eats}(y, x) \wedge \neg \text{killed by}(y, x)) \rightarrow \text{Food}(x)$

v) A: eats (Bills, Peanuts)

B: alive (Bill)

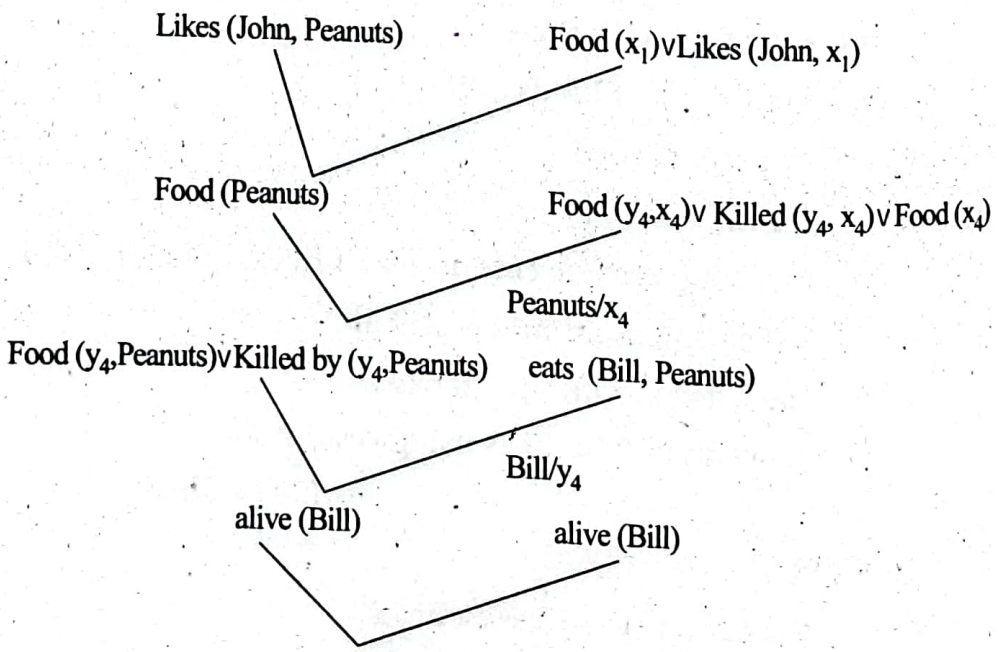
vi) $\forall x: \text{eats}(\text{Bill}, x) \rightarrow \text{eats}(\text{sue}, x)$

vii) $\forall x: \forall y: \text{alive}(x) \rightarrow \neg \text{killed by}(x, y)$

**Clause Form**

1. $\neg$ food (x1) $\vee$ likes (John, x1)
2. Food (apples)
3. Food (chicken)
4. $\neg$ eats (y4, x4) $\vee$ killed by (y4, x4) $\vee$ Food(x4)
5. Eats (Bill, Peanuts)
6. Alive (Bill)
7. $\neg$ eats (Bill, X7) $\vee$ eats (sue, x7)
8. $\neg$ alive (x8) $\vee$ $\neg$ killed by (x8, y8)

**Resolution Principle**



6. **Assume the following facts:**

i) **Horses, cows, pigs are mammals**

ii) **An offspring of a horse is a horse**

iii) **Blue beard is a horse**

iv) **Blue beard is Charlie's parent.**

v) **Offspring and parent are inverse relations.**

vi) **Every mammal has a parent.**

**Prove Charlie is a horse resolution refutation** **(2073 Bhadra)**

⇒ **Converting into FOPL**

a) Horse (x) → Mammal (x)

Cow (x) → Mammal (x)

Pig (x) → Mammal (x)

b) Offspring (x, y) ∧ Horse (y) → Horse (x)

c) Horse (Bluebeard)

d) Parent (Bluebeard, Charlie)

e) Offspring (x, y) → Parent (y, x)

Parent (y, x) → Offspring (x, y)

f) Mammal → Parent (a(x), x)

g) Negation of conclusion

¬ Charlie (Horse)

**Converting into CNF**

a) ¬ Horse (x) ∨ Mammal (x)

¬ Cow (x) ∨ Mammal (x)

¬ Pig (x) ∨ Mammal (x)

b) ¬ Offspring (x ,y) ∨

¬ Horse (y) ∨

Horse (x)

c) Horse (Bluebeard)

d) Parent (Blue beard, Charlie)

e) 1. ¬ Offspring (x, y) ∨ Parent (y, x)

2. ¬ Parent (y, x) ∨ Offspring (x, y)

f) ¬ Mammal (x) ∨ Parent (a(x), x)

g) ¬ Horse (Charlie)

**Now, using resolution principle**

1. Parent (a) (x), x) From a and F
2. Offspring (x, y) From 1 and e (2)
3. ⌉ Horse (y) ∨ Horse (x) From 2 and b
4. Horse (x) From 3 and c
5. □ From 4 and g

**7. Consider the following aximos:**

**i) Anyone whom many laws is football star.**

**ii) Any student who does not pass does not play.**

**iii) John is a student.**

**iv) Any student who does not study does not pass.**

**v) Anyone who does not play is not a football star.**

**Prove that "If John does not study, then Many does not love John" Resolution by Refutation. (2073 Magh)**

⇒ **Converting into FOPL**

i) ∀x (Loves (Many, x) → Star (x))

ii) ∀x (Student (x) ∧ ⌉ Pass (x) → ⌉ Plays(x)

iii) Student (John)

iv) ∀x (student (x) ∧ ⌉ Study (x) → Pass(x)

v) ∀x (⌉ play(x) → ⌉ Star(x)

vi) Negation of conclusion

⌉ [⌉ Study(John) → ⌉ Loves (Mary, John)]

**Now converting in CNF**

i) ⌉ Loves (Mary, x) ∨ Star(x)zs

ii) ⌉ Student (x) ∨ Pass (x) ∨ ⌉ Play (x)

iii) Student (John)

iv) ⌉ Student (x) ∨ Study (x) ∨ ⌉ Pass (x)

v) Play (x) ∨ ⌉ Star (x)

vi) ⌉ (Study (John) ∨ ⌉ Loves (Mary, John))

⌉ Study (John) ∨ Loves (Mary, John))

**Now, using resolution principle**

1. ⌉ Study (John) ∨ Star(x) Form (i) and (ii)
2. ⌉ Study (John) ∨ Play(x) Form 1 and v)
3. ⌉ Study (x) ∨ ⌉ Pass(x) ∨ Play (x) From 2 and iv)
4. ⌉ Pass (x) ∨ Play (x) From 3 and iii)
5. ⌉ Student (x) From 4 and ii)
6. □ From 5 and iii)

8. **A doctor is called to see a sick child. The doctor has prior information that 90% of sick children in that neighbourhood have the flu, while the other 10% are sick with measles. Let F stand for an event of a child being sick with FLU and M stand for an event of a child being sick with measles. Assume for simplicity that three no other maladies in that neighbourhood. A well known (and common) symptoms of measles is a rash and has probability of 0.95. However, very occasionally children with FLU also develop rash and has probability of 0.08 upon examining the child, the doctor finds a rash. What is the probability that the child has measles? (2073 Magh)**

⇒ P(M/R) = ?

Given,

P(F) = 0.9

Probability a child has flue

P(M) = 0.1

Probability of a child has Measles.

P(R/M) = 0.95

A child with rash and has measles

P(R/F) = 0.08

A child with rash flue.

Now,

Probability of measles with rash

$$P(M/R) = \frac{P(R/M) \times P(M)}{P(R/M) \times P(M) + P(R/F) \times P(F)}$$

$$= \frac{0.95 \times 0.1}{0.95 \times 0.1 + 0.08 \times 0.9} = \frac{95}{167}$$

$\therefore$ The required probability is $\frac{95}{167}$.

**9. What is knowledge, representation and reasoning? Describe forward chaining with practical example. (2072 Ashwin)**

⇒ Knowledge

See on page no 59

Representation

It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communication with humans in natural language.

**Reasoning**

Reasoning is an act of deriving a conclusion from certain premises using a given methodology.

**2nd part:** See on page no

**10. What is rule based reasoning? Explain Backward Chaining with examples. (2072 Magh)**

⇒ **1st part: See on page no: 2nd part: See on page no 76**

**11. Using the resolution Principle solve the following statements. Marcus was a man. Marcus was a roman. All men are people. Caesar was a ruler. All Romans were either loyal to Caesar or hated him (or both). Everyone is loyal to someone. People only try to assassinate rulers that they are not loyal to Marcus tried to assassinate Caesar. Find did Marcus Caesar? (2072 Magh)**

⇒ **In FOPL**

a) Man (Marcus)

b) Roman (Marcus)

c) $\forall x$ man (x) $\rightarrow$ Person(x)

d) $\forall x$ roman (x) $\rightarrow$ Loyal (x, Caesar) $\vee$ hate (x, Caesar)

e) $\forall X \exists Y$ Loyal (x, y)

g) $\forall x \forall y$ Person (X) $\wedge$ Try assain (x, y) $\rightarrow \neg$ Loyal (x, y)

h) Try assain (Marcus, Caesar)

i) $\neg$ Hate (X, Caesar) (Negation of conclusion)

**Into CNF**

a) Man (Marcus)

b) Roman (Marcus)

c) ($\neg$ man (x), person (x))

d) Ruler (Caesar)

e) ($\neg$ Roman(x), loyal (x, Caesar), hate (x, Caesar)

f) Loyal (x, F(x)

g) ($\neg$ person(x), $\neg$ rule(Y), $\neg$ try assain (X,Y) $\neg$ loyal (X,Y)

h) Try assain (Marcus, Caesar)

i) $\neg$ hate (x, science)

**Now, using resolution principle**

1. $\neg$ roman (x), loyal (x, Caesar) From i) and e

2. Loyal (Marcus, Caesar) From 1and b

3. $\neg$ Person (Marcus), Ruler (Caesar), $\neg$ try assasin (Marcus, Caesar) From 2and g

4. $\neg$ man (Marcus) , ruler(Caesar), $\neg$ try assasin (Marcus, Caesar) From 3 and c

5. $\neg$ ruler (Caesar), $\neg$ try assasin (Marcus, Caesar) From 4 and a

6. $\neg$ try assasin (Marcus, Caesar) From 5 and b

7. $\neg$ From 6 and h

Hence proved,

**12. If X is on the top of Y, Y supports X. If X is above Y and they are touching each other, x is on the top of Y. A cup is above a book. A cup is touching a book. Show that supports (book, cup) is true.**

**(2071 Bhadra)**

⇒ **Into FOPL**

i) $\forall x\ \forall y$ on (x, y) → support (y, x)

ii) $\forall x\ \forall y$ Above (x, y) ∧ touches (x, y) → on (x, y)

iii) cup(x)
Book (y)
Above(x, y)

iv) cup(x)
Book(y)
Touches(x, y)

v) Negation of conclusion:
¬ support (book, cup)
¬ support (y, x)

**Into CNF**

i) ¬ on (x, y) ∨ support (y, x)

ii) a) ¬ above (x, y)
b) ¬ touches (x, y)
c) on (x, y)

iii) Above (x, y)

iv) Touches (x, y)

v) ¬ support (y, x)

Now, using resolution principle

1. ¬ on (x, y) ∨ ¬ above i) and v)
2. ¬ on (x, y) ∨ ¬ above (x, y) From 1 and ii(a)
3. ¬ on (x, y) From 2 and iii)
4. ¬ on (x, y) ∨ ¬ touches (x, y) from 3 and iv)b
5. ¬ on (x, y) From 4 and iv
6. ¬ From 5 and ii) c

**13. Explain reasoning in belief network with suitable example.**

**(2071 Magh)**

⇨ See on page no 81

**14. A doctor knows that the diseases meningitis causes the patient to have a stiff neck 50% of the time. The doctor also knows that the probability that a patient has meningitis is 1/50,000 and the probability that any patient has a stiff neck is 1/20. Find the probability that a patient with a stiff neck has meningitis.**

**(2071 Magh)**

⇨ $P(S) = \frac{1}{20} = 0.05$

$P(S/M) = 50\% = 0.5$

$P(M) = \frac{1}{50,000} = 0.00002$

Now, required probability is

$$P(M/S) = \frac{P(M) \times P(S/M)}{P(S)} = \frac{0.00002 \times 0.5}{0.05} = 0.0002$$

**15. Compare the following:**

**a) Forward versus backward chaining** **(2071 Magh)**

⇨ 1. Forward chaining starts from known facts and applies inference rule to extract move data unit it reaches to the goal whereas backward chaining starts from the goal and works backward through inference rules to find the required facts that support the goal.

2. Forward chaining is bottom - up approach whereas backward chaining in top - down approach.

3. Forward chaining is known as data - driven inference technique where as backward chaining is known as goal - driven technique.

4. Forward chaining tests for all the available rules whereas backward chaining only test for few required rules.

**16. Why do we need FOPL? State any three rules of inference.**

**(2070 Magh)**

⇨ **1st part:** See on page no66

**2nd part:** See on page no 65

**17. Explain backward chaining with suitable example and compare with forward chaining.** **(2070 Magh)**

⇒ **1st part:** See on page no 76

**2nd part:** See on old question no 15

**18. All over smart persons are stupid. Children of over smart persons are naughty. Ram is children of Hari. Hari is over smart. Show that Ram is naughty using FOPL based solution method.**

**(2070 Bhadra)**

⇒ **In FOPL**

i) $\forall x$ [ Over smart (x) $\rightarrow$ Stupid (x)]

ii) $\forall x$ (Over smart(x) $\wedge$ Children (y, x) $\rightarrow$ Naughty (y))

iii) Children (Ram, Hari)

iv) Over smart (Hari)

**In CNF**

i) $\neg$ Over smart (x) $\vee$ Stupid (x)

ii) $\neg$ Over smart (x) $\vee \neg$ Children (y, x) $\vee$ Naughty (y)

iii) Children (Ram, Hari)

iv) Over smart (Hari)

**Now, using resolution principle**

i) $\neg$ Over smart (x) $\vee$ Naughty (y) From (i) and (ii)

ii) Naughty (y) From (i) and (iv)

Hence, Ram is naughty proved

**19. Explain the step involved in conjuction normal form (CNF) with suitable example** **(2070 Bhadra)**

⇒ 1. Eliminate implications $\leftrightarrow, \rightarrow$

2. Reduce the scope of $\neg$

De - Morgan laws

i) $\neg (x \vee y) \equiv \neg x \wedge \neg y$

ii) $\neg (x \wedge y) \equiv \neg x \vee \neg y$

3. **Standardize variables:** Rename variables so that each different variables in the set of WFFs have a different symbol.

4. **Skolemization:** Eliminate existential quantifiers and replace constant or skolem function as appropriate.

5. Convert into prerex from by moving all universal quantifiers to the beginning of WFFs.

   WFFs Prerex form = Prefix (string of quantifiers) + Matrix (Quantifier - free formula)

6. Drop universal quantifiers.

7. Use distributive laws and equivalent rules of propositional logic to transform the matrix to CNF.

Example conversion to CNF

$[(\forall x)\ Q(x)] \rightarrow (\forall x)\ (\forall y)\ [\exists z\ P(x, y, z) \rightarrow \forall u\ R(x, y, u, z)]$

1. **Eliminate Implications** $\leftrightarrow, \rightarrow$

   $\neg\ [(\forall x)\ Q(x)] \vee (\forall x)\ (\forall y)\ [(\exists x)\ [\neg\ P(x, y, z) \vee (\forall u)$

   $R(x, y, u, z)$

2. **Reduce Scope of negations**

   $[(\exists x)\ \neg\ Q(x)] \vee (\forall x)\ (\forall y)\ (\exists z)\ [\neg\ P(x, y, z) \vee (\forall y)\ R(x, y, z)]$

3. **Standardize Variables**

   $[(\exists w)\ Q(w)] \vee (\forall x)\ (\forall y)\ [(\exists z)[\neg\ P(x, y, z) \vee (\forall x)\ R(x, y, u, z)]$

4. **Skolemization**

   $\neg\ Q(A) \vee (\forall x, y)\ [\neg\ p\ (x, y, F(x, y)) \vee (\forall u)\ R(x, y, u, F(x, y)]$

5. **Prerex Form**

$(\forall x, y, u)\ [\neg Q(A) \vee \neg (x, y, F(x, y)) \vee R(x, y, u, F(x, y))]$

6. **Drop Universal Quantifiers**

$\neg Q(A) \vee \neg (x, y, F(x, y)) \vee R(x, y, u, F(x, y))$

7. **Convert to CNF**

WFF already in CNF

20. **Convert the following sentences into FOPL and hence into CNF.**

⇒ i) Everyone likes someone

ii) All the student who visited science museum are not engineering student.

iii) Sumit likes all the fruits that are rich vitamin A.

iv) Shyman likes all the movies Kirshna likes (2070 Chaitra)

**In FOPL**

i) $\forall x\ \exists y$ like (x, y)

ii) $\forall x\ \forall y$ (student (x, y) $\wedge$ visit (x, science) $\rightarrow \neg$ student (x, engineering)

iii) $\forall x$ Sumit (x) $\rightarrow$ Likes (x, fruit rich in vitamin A)

iv)

Movies (x) $\wedge$ Likes (Krishna, x) $\rightarrow$ Likes (Shyam, z)

**Into CNF**

i) Like (x, y)

ii) $\neg$ student (x, y) $\vee \neg$ (x, science) $\vee \neg$ student (x, engineering)

iii) $\neg$ Sumit (x) $\vee$ Likes (x, fruit rich in vitamin A)

iv) $\neg$ movies (x) $\vee$ Likes (Krishna, x) $\vee$ Likes (Shyam, z)

21. **Write short notes:**

**Skolemization**

**(2069 Bhadra)**

⇒ Skolemization in Artificial Intelligence is a procedure used when there is a requirement of the reduction of any first-order formula to its Skolem normal form. This is usually done when there is a need for proving a theorem by using programming. The main objective here is to ensure that the resultant form will not have any quantifiers. All of the quantifiers are flagged and are replaced with entities called Skolem functions. Not every transformation needs to be done when working with Skolemization, only the necessary ones wherein the scope of which will affect the solution. The first-order resolution provided by Skolemization is highly effective in devising AI models that can operate more efficiently.

Eg: $[(\forall x)\ Q(x)] \rightarrow (\forall x)\ (\forall y)\ [\exists z\ P(x, y, z) \rightarrow \forall u\ R(x, y, u, z)]$

Skolemization we have

$\neg\ Q(A) \vee (\forall x, y)\ [\neg\ p\ (x, y, F(x, y)) \vee (\forall u)\ R(x, y, u, F(x, y)]$

⌘⌘⌘

Chapter FIVE

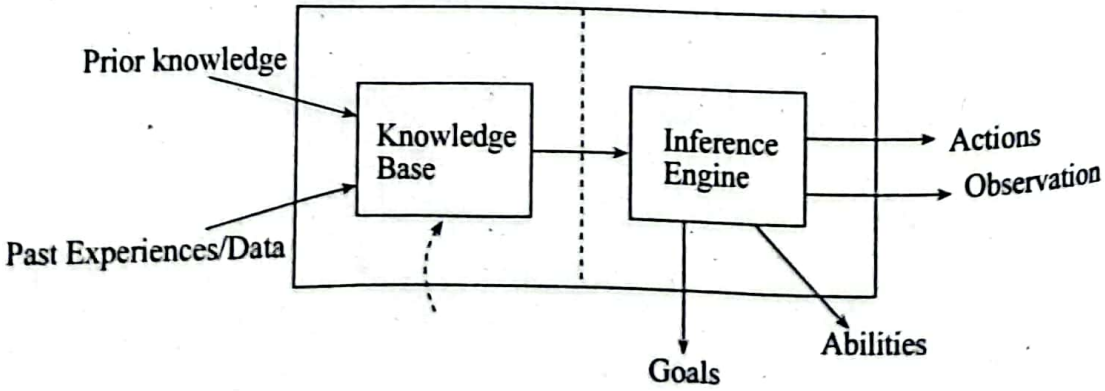
# Structured Knowledge Representation

Humans are best at understanding, reasoning and interpreting knowledge. Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world. But how machines do all these things come under knowledge representation and reasoning. Hence we can describe knowledge representation as following:

- Knowledge representation and reasoning (KR, KK) is the part of artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behaviour of agents.
- It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world condition or communicating with human in natural languages.

**Things To Represent**

- **Object:** All the facts about objects in our world domain e.g. Guitars contain string, trumpets are brass instruments.
- **Event:** Events are the actions which occur in the world.
- **Performance:** It describes behaviour which involves knowledge about how to do things.
- **Meta - Knowledge:** It is knowledge about what we know.
- **Facts:** Facts are the truths about the real world and what we represent.
- **Knowledge Base:** The central component of th knowledge - based agent is the knowledge base. It is represented as KB. The knowledge is a group of the sentences.



## Issues in knowledge Representation

i) **Important Attributes:** Any attributes of objects so basic that they occur n almost every problem domain?

ii) **Relationship Among Attributes:** Any important relationship that exists among object attributes?

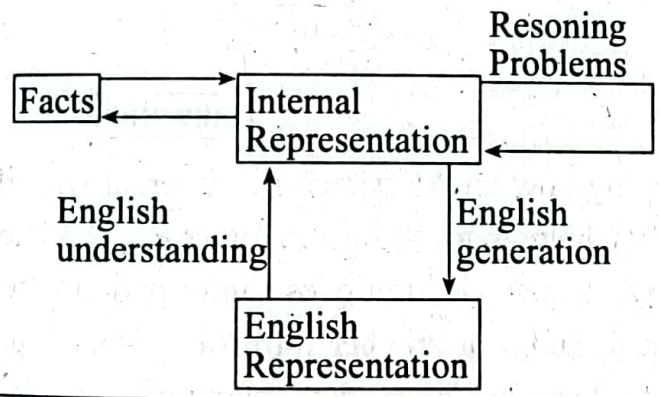
iii) **Choosing Granularity:** At what level of detail should the knowledge be represented?

iv) **Set of objects:** How sets of objects be represented?

v) **Finding Right Structure:** Given a large amount of knowledge stored, how can relevant parts be accessed?

## Mapping between Facts and Representations

- Mapping "Facts to symbols" and "symbols to facts" using forward and backward representation mapping.
- Consider an English sentence:




| Facts | Representation |
|---|---|
| • Spot is a dog | A fact represented in English sentence. |
| • Dog (spot) | Using forward mapping function the above fact is represented in logic. |
| • $\forall x : dog(x) \rightarrow hastail(x)$ | A logical representation of the fact that "all dogs have tails" |

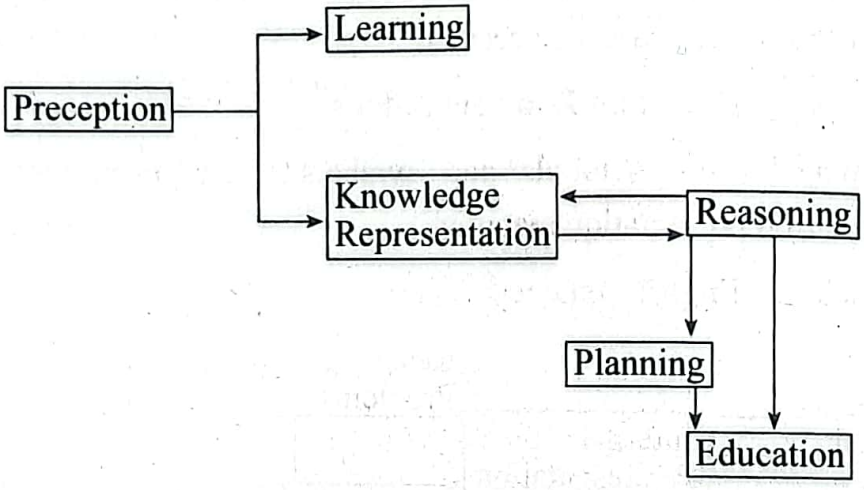
Now using deduction mechanism we can generate a new representation of object.

| | |
|---|---|
| • Hastail (spot) | A new object representation |
| • Spot has a tail | Using backward mapping function to generate english sentence. |

## AI Knowledge Cycle

An Artificial intelligence system has the following components for displaying intelligent behaviour:

- Perception
- Learning
- Knowledge Representation Reasoning
- Planning
- Execution

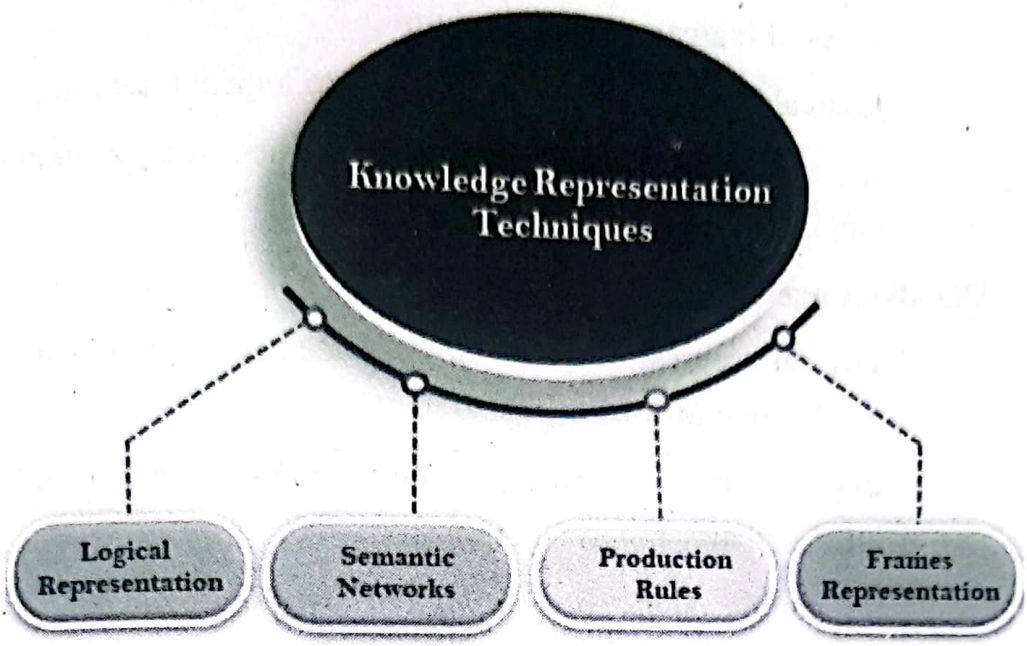



The above diagram is showing how an AI system can interact with the real world and what components help it to show intelligence. AI system has perception component by which it retrieves information from its environment. It can be visual, audio or another form of sensory input. The learning is responsible for learning from data captured by perception components. In the complete cycle, the main components are knowledge representation and reasoning. These two components are involved in showing the intelligence in machine - like humans. These two components are independent with each other but also coupled together. The planning and execution depend on analysis of knowledge representation and reasoning.

### Techniques of knowledge representation

There are mainly four ways of knowledge representation which are given as follows:

1. Logical Representation
2. Semantic Network Representation
3. Frame Representation
4. Production Rules




1. **Logical Representation**

Logical representation is a language with some concrete rules which deals with propositions and has no ambiguity in representation. Logical representation means drawing a conclusion based on various conditions. This representation lays down some important communication rules. It consists of precisely defined syntax and semantics which supports the sound inference. Each sentence can be translated into logics using syntax and semantics.

**Syntax:**

- Syntaxes are the rules which decide how we can construct legal sentences in the logic.
- It determines which symbol we can use in knowledge representation.
- How to write those symbols.

**Semantics:**

- Semantics are the rules by which we can interpret the sentence in the logic.
- Semantic also involves assigning a meaning to each sentence.

Logical representation can be categorised into mainly two logics:

a. Propositional Logics

b. Predicate logics

*Note: We will discuss Prepositional Logics and Predicate logics in later chapters.*

**Advantages of logical representation:**

1. Logical representation enables us to do logical reasoning.
2. Logical representation is the basis for the programming languages.

**Disadvantages of logical Representation:**

1. Logical representations have some restrictions and are challenging to work with.
2. Logical representation technique may not be very natural, and inference may not be so efficient.

*Note: Do not be confused with logical representation and logical reasoning as logical representation is a representation language and reasoning is a process of thinking logically.*

**2. Semantic Network Representation**

Semantic networks are alternative of predicate logic for knowledge representation. In Semantic networks, we can represent our knowledge in the form of graphical networks. This network consists of nodes representing objects and arcs which describe the relationship between those objects. Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.

This representation consist of mainly two types of relations:

a. IS-A relation (Inheritance)

b. Kind-of-relation

**Example:** Following are some statements which we need to represent in the form of nodes and arcs.

**Statements:**

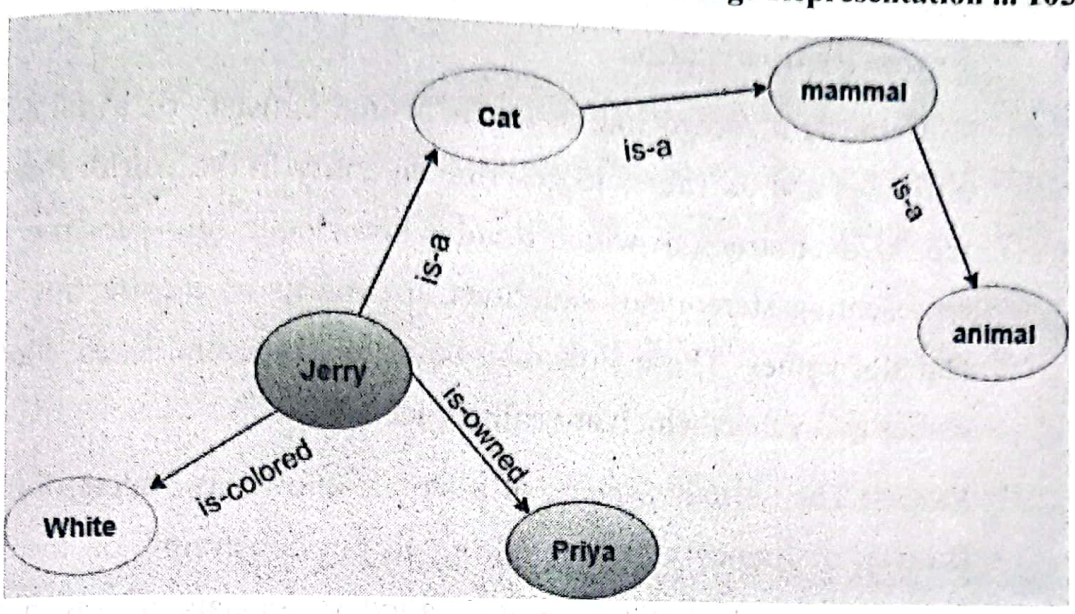
a. Jerry is a cat.

b. Jerry is a mammal

c. Jerry is owned by Priya.

d. Jerry is brown colored.

e. All Mammals are animal.




In the above diagram, we have represented the different type of knowledge in the form of nodes and arcs. Each object is connected with another object by some relation.

**Drawbacks in Semantic representation:**

1. Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
2. Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
3. These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
4. Semantic networks do not have any standard definition for the link names.
5. These networks are not intelligent and depend on the creator of the system.

**Advantages of Semantic network:**

1. Semantic networks are a natural representation of knowledge.
2. Semantic networks convey meaning in a transparent manner.
3. These networks are simple and easily understandable.

## 3. Frame Representation

A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world. Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations. It consists of a collection of slots and slot values. These slots may be of any type and sizes. Slots have names and values which are called facets.

**Facets:** The various aspects of a slot is known as **Facets**. Facets are features of frames which enable us to put constraints on the frames. Example: IF-NEEDED facts are called when data of any particular slot is needed. A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values. A frame is also known as **slot-filter knowledge representation** in artificial intelligence.

Frames are derived from semantic networks and later evolved into our modern-day classes and objects. A single frame is not much useful. Frames system consist of a collection of frames which are connected. In the frame, knowledge about an object or event can be stored together in the knowledge base. The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.

**Example 1:**

**Let's take an example of a frame for a book**

| Slots | Filters |
|---|---|
| Title | Artificial Intelligence |
| Genre | Computer Science |
| Author | Peter Norvig |
| Edition | Third Edition |
| Year | 1996 |
| Page | 1152 |

**Example 2:**

**Let's suppose we are taking an entity, Peter. Peter is an engineer as a profession, and his age is 25, he lives in city London, and the country is England. So following is the frame representation for this:**

| Slots | Filter |
|---|---|
| Name | Peter |
| Profession | Doctor |
| Age | 25 |
| Marital status | Single |
| Weight | 78 |

**Advantages of frame representation:**

1. The frame knowledge representation makes the programming easier by grouping the related data.
2. The frame representation is comparably flexible and used by many applications in AI.
3. It is very easy to add slots for new attribute and relations.
4. It is easy to include default data and to search for missing values.
5. Frame representation is easy to understand and visualize.

**Disadvantages of frame representation:**

1. In frame system inference mechanism is not be easily processed.
2. Inference mechanism cannot be smoothly proceeded by frame representation.
3. Frame representation has a much generalized approach.

4. **Production Rules**

Production rules system consist of (**condition, action**) pairs which mean, "If condition then action". It has mainly three parts:

- The set of production rules
- Working Memory
- The recognize-act-cycle

In production rules agent checks for the condition and if the condition exists then production rule fires and corresponding action is carried out. The condition part of the rule determines which rule may be applied to a problem. And the action part carries out the associated problem-solving steps. This complete process is called a recognize-act cycle.

The working memory contains the description of the current state of problems-solving and rule can write knowledge to the working memory. This knowledge match and may fire other rules.

If there is a new situation (state) generates, then multiple production rules will be fired together, this is called conflict set. In this situation, the agent needs to select a rule from these sets, and it is called a conflict resolution.

**Example:**

- IF (at bus stop AND bus arrives) THEN action (get into the bus)
- IF (on the bus AND paid AND empty seat) THEN action (sit down).
- IF (on bus AND unpaid) THEN action (pay charges).
- IF (bus arrives at destination) THEN action (get down from the bus).

**Advantages of Production rule:**

1. The production rules are expressed in natural language.
2. The production rules are highly modular, so we can easily remove, add or modify an individual rule.

**Disadvantages of Production rule:**

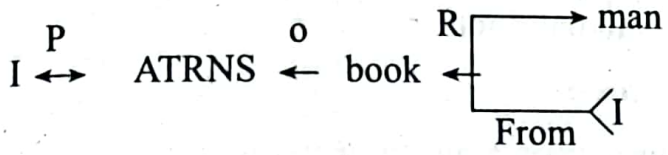
1. Production rule system does not exhibit any learning capabilities, as it does not store the result of the problem for the future uses.
2. During the execution of the program, many rules may be active hence rule-based production systems are inefficient.

## Conceptual Dependency (CD)

This representation is used in natural language processing in order to represent them earning of the sentences in such a way that inference we can be made from the sentences. It is independent of the language in which the sentences were originally stated. CD representations of a sentence is built out of primitives, which are not words belonging to the language but are conceptual, these primitives are combined to form the meaning s of the words. As an example consider the event represented by the sentence.

I gave the man a book.

In CD from the above sentence can be represented with primitives as




**In the above representation the symbols have the following meaning:**

- Arrows indicate direction of dependency
- Double arrow indicates two may link between actor and the action
- P indicates past tense
- ATRANS is one of the primitive acts used by the theory. It indicates transfer of possession
- 0 indicates the object case relation
- R indicates the recipient case relation

Conceptual dependency provides a str5ucture in which knowledge can be represented and also a set of building blocks from which representations can be built. A typical set of primitive actions are

ATRANS - Transfer of an abstract relationship (Eg: give)

PTRANS - Transfer of the physical location of an object (Eg: go)

PROPEL - Application of physical force to an object (Eg: push)

MOVE - Movement of a body part by its owner (Eg : kick)

GRASP - Grasping of an object by an actor (Eg: throw)

INGEST - Ingesting of an object by an animal (Eg: eat)

EXPEL - Expulsion of something from the body of an animal (cry)

MTRANS - Transfer of mental information (Eg: tell)

MBUILD - Building new information out of old (Eg: decide)

SPEAK - Production of sounds (Eg: say)

ATTEND - Focusing of sense organ toward a stimulus (Eg: listen)

## Scripts

A script is a structured representation describing a stereotyped sequence of events in a particular context.

Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should understand. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies eating in a restaurant, shopping in a supermarket, or visiting an ophthalmologist.

Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.

**Scripts are beneficial because:**

- Events tend to occur in known runs or patterns.
- A casual relationship between events exist
- An entry condition exists which allows an event to take place.
- Prerequisites exist upon events taking place.

**Components of a script**

The components of a script include:

**Entry condition:** These are basic condition which must be fulfilled before events in the script can occur.

**Props:** Slots representing objects involved in events.

**Roles:** These are the actions that the individual participants perform.

**Track:** Variations on the script. Different tracks may share components of the same scripts

**Scenes:** The sequence of events that occur Entry condition: These are basic condition which must be fulfilled before events in the script can occur.

- **Results:** Condition that will be true after events in script occurred.
- **Props:** Slots representing objects involved in events
- **Roles:** These are the actions that the individual participants perform.
- **Track:** Variations on the script. Different tracks may share components of the same scripts.
- **Scenes:** The sequence of events that occur.
- **Entry condition:** These are basic condition which must be fulfilled before events in the script can occur.

- **Results:** Condition that will be true after events in script occurred.
- **Props:** Slots representing objects involved in events
- **Roles:** These are the actions that the individual participants perform.
- **Track:** Variations on the script. Different tracks may share components of the same scripts.
- **Scenes:** The sequence of events that occur.

Describing a script, special symbols of actions are used. These are:

| Symbol | Meaning | Example |
|---|---|---|
| ATRANS | transfer a relationship | give |
| PTRANS | transfer physical location of an object | go |
| PROPEL | apply physical force to an object | push |
| MOVE | move body part by owner | kick |
| GRASP | grab an object by an actor | hold |
| INGEST | taking an object by an animal eat | drink |
| EXPEL | expel from animal's body | cry |
| MTRANS | transfer mental information | tell |
| MBUILD | mentally make new information | decide |
| CONC | conceptualize or think about an idea | think |
| SPEAK | produce sound | say |
| ATTEND | focus sense organ | listen |

**Example:**

**Script for going to the bank to withdraw money.**

**SCRIPT:** Withdraw money

**TRACK:** Bank

**PROPS:** Money

Counter

Form

Token

**Roles:** P= Customer

E= Employee

C= Cashier

**Entry conditions:** P has no or less money.

The bank is open.

**Results:** P has more money.

**Scene 1:** Entering

P PTRANS P into the Bank

P ATTEND eyes to E

P MOVE P to E

**Scene 2:** Filling form

P MTRANS signal to E

E ATRANS form to P

P PROPEL form for writing

P ATRANS form to P

E ATRANS form to P

**Scene 3:** Withdrawing money

P ATTEND eyes to counter

P PTRANS P to queue at the counter

P PTRANS token to C

C ATRANS money to P

**Scene 4:** Exiting the bank

P PTRANS P to out of bank

**Advantages of Scripts**

- Ability to predict events.
- A single coherent interpretation maybe builds up from a collection of observations

**Disadvantages of Scripts**

- Less general than frames.
- May not be suitable to represent all kinds of knowledge

## OLD QUESTION SOLUTION

1. **Define a semantic network and frames with an example. List advantages and limitations of both.** **(2076 Baisakh)**

⇒ See on page no 103

2. **What do you mean by knowledge representation? How semantic network is used to represent knowledge?**

⇒ **1st part:** See on page no99

**2nd part:** See on page no 103

3. **Why semantic network and frames and frame are important in AI? Provide example of both with FOPL statements example.**

**(2075 Baisakh)**

⇒ See on page no 103

4. **What is frame? How it is different from semantic net in knowledge representation?**

**(2074 Bhadra)**

**Ans:** **1st part:** See on page no 105

Semantic networks structure the knowledge of a specific part of any information. It uses real-world meanings that are easy to understand. For example, it uses "is a" and "is a part" inheritance hierarchies. Besides this, they can be used to represent events and natural language sentences. In simple words, semantic networks are general in nature but if you're not familiar with the syntax and semantics in each case, then it can be a problem.

The frame is also a data structure to represent similar properties and possibilities for knowledge representation like the semantic networks. It contains the same ideas of inheritance and default values. However, frames are more powerful than the semantic network because their slots contain instructions to be understood well for computing things stored in other slots or in other frames.

Semantic net and frames are widely popular knowledge representation techniques but they are quite different from each other. Semantic representation is easy to understand but when it comes to computing larger data it proves worthless. On the other hand, frames represent the same information in a logical manner but due to excessive slots, it can obstruct the deduction process. However, both the knowledge representation techniques play a vital role in artificial intelligence, yet frames take the upper hand.

5. **Convert given sentences into semantic Network.**

i) **The height of the adult male is 5.10.**

ii) **Baseball player is an adult male.**

iii) **Adult male is a person.**

iv) **Batting average of baseball players is 0.252**

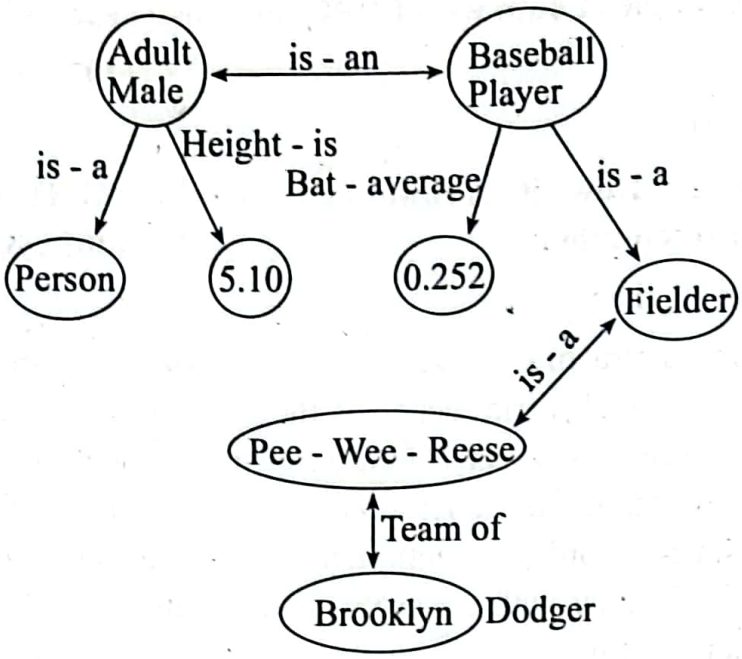
v) **Pee - wee - Reese is a fielder.**

vi) **Fields is Baseball player.**

vii) **Team of Pee - Wee - Reese is Brooklyn Dodger.**

**(2073 Bhadra)**

⇒




**6. What is casual net? How does Bayes Theorem calculate the probability in casual net? Explain with example calculation.**

**(2073 Bhadra)**

⇒ A causal network is an acyclic digraph arising from an evolution of a substitution system, and representing its history.

**2nd part;** See Bayesian Belief Network in page no 81

**7. Explain Frames and semantic Net with examples..List down their advantages and limitations.** **(2073 Magh)**

⇒ See on page no 103

**8. What are semantics nets and frame? How frames are useful in semantic nets?** **(2072 Ashwin)**

⇒ See on page no 103

**9. Convert the given sentences in Semantic Net.**

**i) A person is a mammal.**

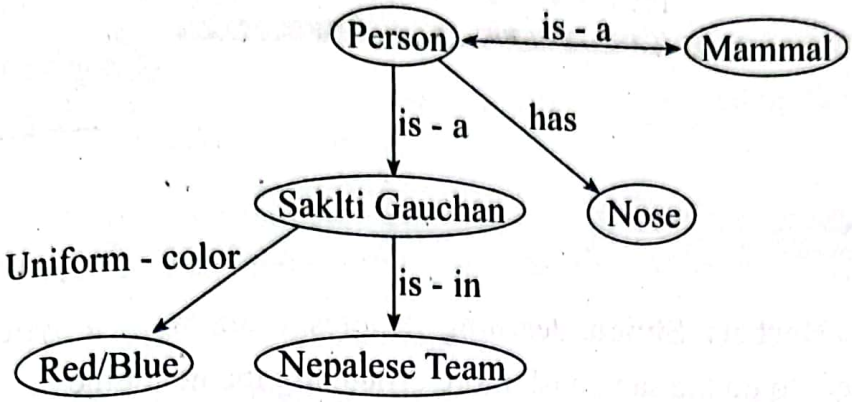
**ii) Sakti Gauchan is a person.**

**iii) Person has nose.**

**iv) Sakti Gauchan is in Nepalese team.**

**v) Uniform color of Sakti Gauchan is Red/Blue.** **(2071 Magh)**

⇒




10. **What is conceptual dependency? Explain with a sample example of a state-space search framework?**

⇒ See on page no 107

11. **What is casual network? Explain reasoning in belief network with suitable example.** **(2071 Magh)**

⇒ **1st part:** See on old question no 6

**2nd part:** See on page no 81

12. **How can knowledge be represented using semantic network? Explain with suitable example.** **(2070 Magh)**

Ans: See on page no103

13. **What is semantic nets? Explain with suitable example.** **(2070 Bhadra)**

⇒ See on page no103

14. **What are different knowledge representation models? Discuss semantic nets with an example.** **(2069 Bhadra)**

⇒ See on page no102

⌘⌘⌘

# Machine Learning

Chapter SIX

## Learning

According to **Herbert Simon**, learning denotes changes in a system that enable a system to do the same task more efficiently the next time..

**Arthur Samuel stated that**, "Machine learning is the subfield of computer science, that gives computers the ability to learn without being explicitly programmed ".

In 1997, **Mitchell** proposed that, "A computer program is said to learn from experience **'E'** with respect to some class of tasks **'T'** and performance measure **'P'**, if its performance at tasks in **'T'**, as measured by **'P'**, improves with experience **E** ".

The main purpose of machine learning is to study and design the algorithms that can be used to produce the predicates from the given dataset.

Besides these, the machine learning includes the agents percepts for acting as well as to improve their future performance.

**The following tasks must be learned by an agent.**

To predict or decide the result state for an action.

To know the values for each state (understand which state has high or low vale).

To keep record of relevant percepts.

**Different forms of learning**

1. **Rote learning**

   Rote learning is possible on the basis of memorization.

   This technique mainly focuses on memorization by avoiding the inner complexities. So, it becomes possible for the learner to recall the stored knowledge.

   **For example:** When a learner learns a poem or song by reciting or repeating it, without knowing the actual meaning of the poem or song.

2. **Induction learning (Learning by example).**

Induction learning is carried out on the basis of supervised learning.

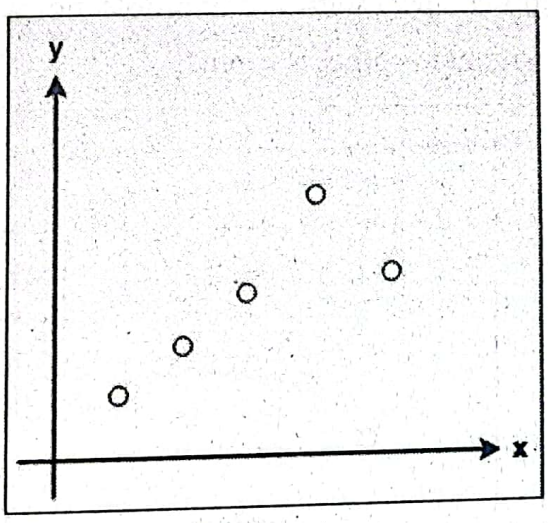
In this learning process, a general rule is induced by the system from a set of observed instance.

However, class definitions can be constructed with the help of a classification method.

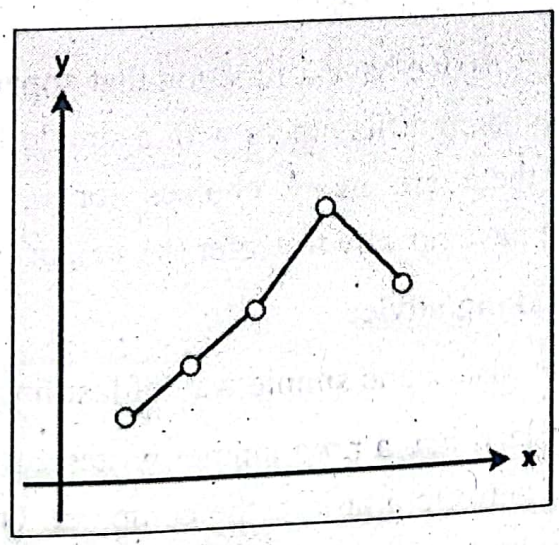
**For Example:** Consider that '$f$' is the target function and example is a pair (x $f$(x)), where 'x' is input and $f$(x) is the output function applied to 'x'.

**Given problem:** Find hypothesis h such as h ≈ $f$

So, in the following fig-a, points (x, y) are given in plane so that y = $f$(x), and the task is to find a function h(x) that fits the point well.

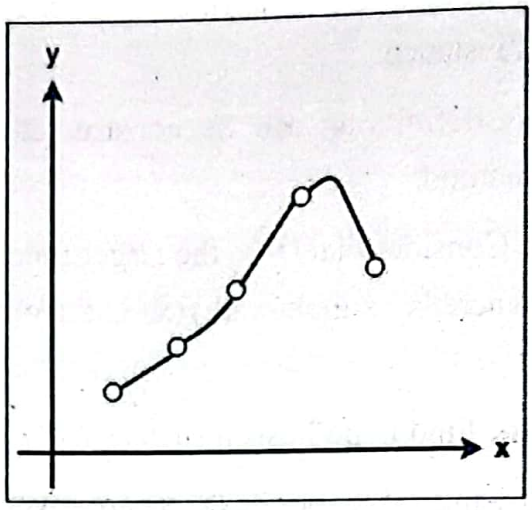



Fig(a)



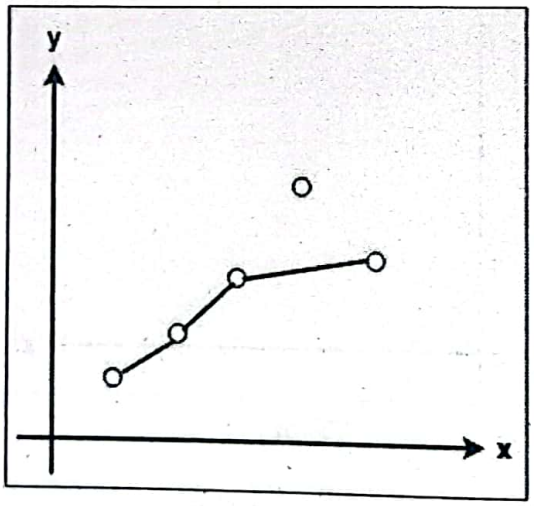

Fig(b)

In fig-b, a piecewise-linear 'h' function is given, while the fig-c shows more complicated 'h' function.




Fig(c)

Both the functions agree with the example points, but differ with the values of 'y' assigned to other x inputs.




Fig(d)

As shown in fig.(d), we have a function that apparently ignores one of the example points, but fits others with a simple function. The true/ is unknown, so there are many choices for h, but without further knowledge, we have no way to prefer (b), (c), or (d).

3. **Learning by taking advice**

This type is the easiest and simple way of learning.

In this type of learning, a programmer writes a program to give some instructions to perform a task to the computer. Once it is learned (i.e. programmed), the system will be able to do new things.

Also, there can be several sources for taking advice such as humans (experts), internet etc.

However, this type of learning has a more necessity of inference than rote learning.

As the stored knowledge in knowledge base gets transformed into an operational form, the reliability of the knowledge source is always taken into consideration.

4. **Explanation based learning**

Explanation-based learning (EBL) deals with an idea of single-example learning.

This type of learning usually requires a substantial number of training instances but there are two difficulties in this:

i. it is difficult to have such a number of training instances

ii. Sometimes, it may help us to learn certain things effectively, especially when we have enough knowledge.

Hence, it is clear that instance-based learning is more data-intensive, data-driven while EBL is more knowledge-intensive, knowledge-driven.

Initially, an EBL system accepts a training example.

On the basis of the given goal concept, an operationality criteria and domain theory, it "generalizes" the training example to describe the goal concept and to satisfy the operationality criteria (which are usually a set of rules that describe relationships between objects and actions in a domain).

Thus, several applications are possible for the knowledge acquisition and engineering aspects.

**Machine Learning**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

**Why do we require machine learning?**

Machine learning plays an important role in improving and understanding the efficiency of human learning.

Machine learning is used to discover a new things not known to many human beings.

**Types of machine learning**

1. **Supervised Learning**

   Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types - **Regression** and **Classification**.

   **Regression** trains on and predicts a continuous-valued response, for example predicting real estate prices.

   **Classification** attempts to find the appropriate class label, such as analyzing positive/negative sentiment, male and female persons, benign and malignant tumors, secure and unsecure loans etc.

   In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called **labeled data**. The learned rule is then used to label new data with unknown outputs.

   Supervised learning involves building a machine learning model that is based on **labeled samples**. For example, if we build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, we first need to create a database and label it. We need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

   Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new examples. There are many **supervised learning algorithms** such as Logistic Regression, Neural networks, Support Vector Machines (SVMs), and Naive Bayes classifiers.

Common **examples** of supervised learning include classifying e-mails into spam and not-spam categories, labeling webpages based on their content, and voice recognition.

2. **Unsupervised Learning**

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called **unlabeled data**.

Suppose that we have a number of data points, and we want to classify them into several groups. We may not exactly know what the criteria of classification would be. So, an unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way.

Unsupervised learning algorithms are extremely powerful tools for analyzing data and for identifying patterns and trends. They are most commonly used for clustering similar input into logical groups. Unsupervised learning algorithms include Kmeans, Random Forests, Hierarchical clustering and so on.

3. **Semi-supervised Learning**

If some learning samples are labeled, but some other are not labeled, then it is semi-supervised learning. It makes use of a large amount of **unlabeled data for training** and a small amount of **labeled data for testing**. Semi-supervised learning is applied in cases where it is expensive to acquire a fully labeled dataset while more practical to label a small subset. For example, it often requires skilled experts to label certain remote sensing images, and lots of field experiments to locate oil at a particular location, while acquiring unlabeled data is relatively easy.

4. **Reinforcement Learning**

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best known instances include self-driving cars and chess master algorithm AlphaGo.

## Genetic Algorithm

Genetic algorithm (GAs) are a class of search algorithms designed on the natural evolution process. Genetic Algorithms are based on the principles of **survival of the fittest**.

A Genetic Algorithm method inspired in the world of Biology, particularly, the Evolution Theory by **Charles Darwin,** is taken as the basis of its working. **John Holland** introduced the Genetic Algorithm in **1975**. Genetic Algorithms are utilized to tackle optimization problems by copying the evolutionary behavior of species. From an initial random population of solutions, this population is advanced through selection, mutation, and crossover operators, inspired in natural evolution. By implementing the given set of operations, the population goes through an iterative procedure in which it reaches various states, and each one is called **generation**. As a result of this procedure, the population is expected to reach a generation in which it contains a decent solution to the problem. In the Genetic Algorithm, the solution of the problem is coded as a **string of bits** or **real numbers**.

They have been shown in practice to be very efficient at functional optimization. It is used in searching for huge and sophisticated spaces. Genetic algorithms (GAs) are algorithms that are used for optimization and machine learning based on various features of biological evolution.

They need the given components:

A process of coding solutions to solve the problem of chromosomes.

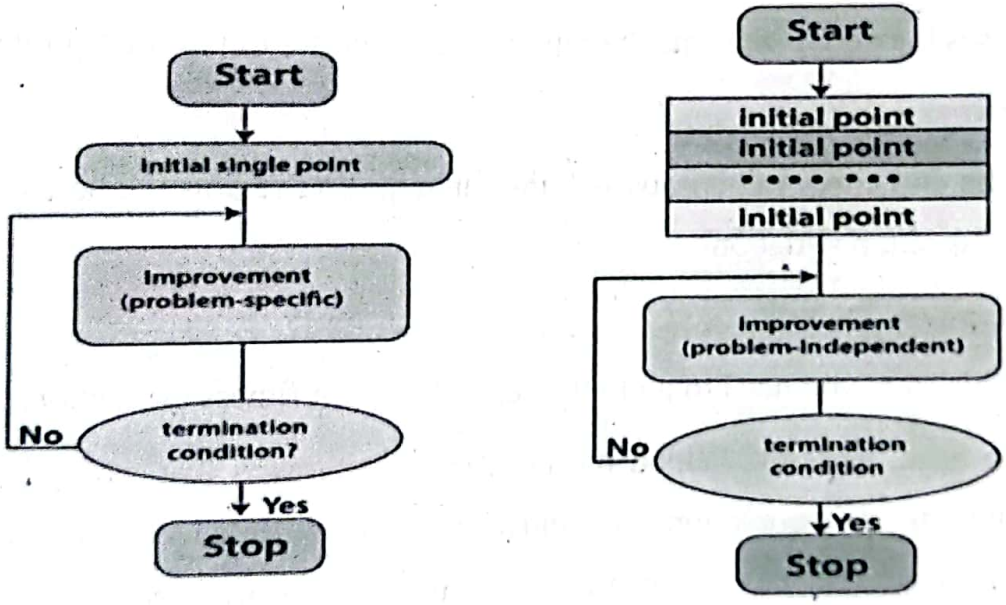
An evaluation function that recovers a rating for each chromosome given to it.

Operators that may be implemented to parents when they reproduce to alter their genetic composition. The standard operators are mutation and crossover.

Operators that may be implemented to parents when they reproduce to modify their genetic composition. The standard operators are mutation and crossover.

**ANNs working principle:**

The working principle of a standard Genetic Algorithm is illustrated in the given figure. The significant steps involved are the generation of a population of the solution, identifying the objective function and fitness function, and the application of genetic operators. These aspects are described with the assistance of a fundamental genetic algorithm as below.




**Start:**

It generates a random population of n chromosomes.

**Fitness:**

It calculates the fitness f(x) of each chromosome x in the population.

**New Population:**

It generates a new population by repeating the following steps until the New population is finished.

**Selection:**

It chooses two parent chromosomes from a population as per their fitness. The better fitness, the higher the probability of getting selected.

**Crossover:**

In crossover probability, cross over the parents to form new offspring (children). If no crossover was performed, the offspring is the exact copy of the parents.

**Mutation:**

In mutation probability, mutate new offspring at each locus.

**Accepting:**

It places new offspring in the new population.

**Replace:**

It uses the newly generated population for a further run of the algorithm.

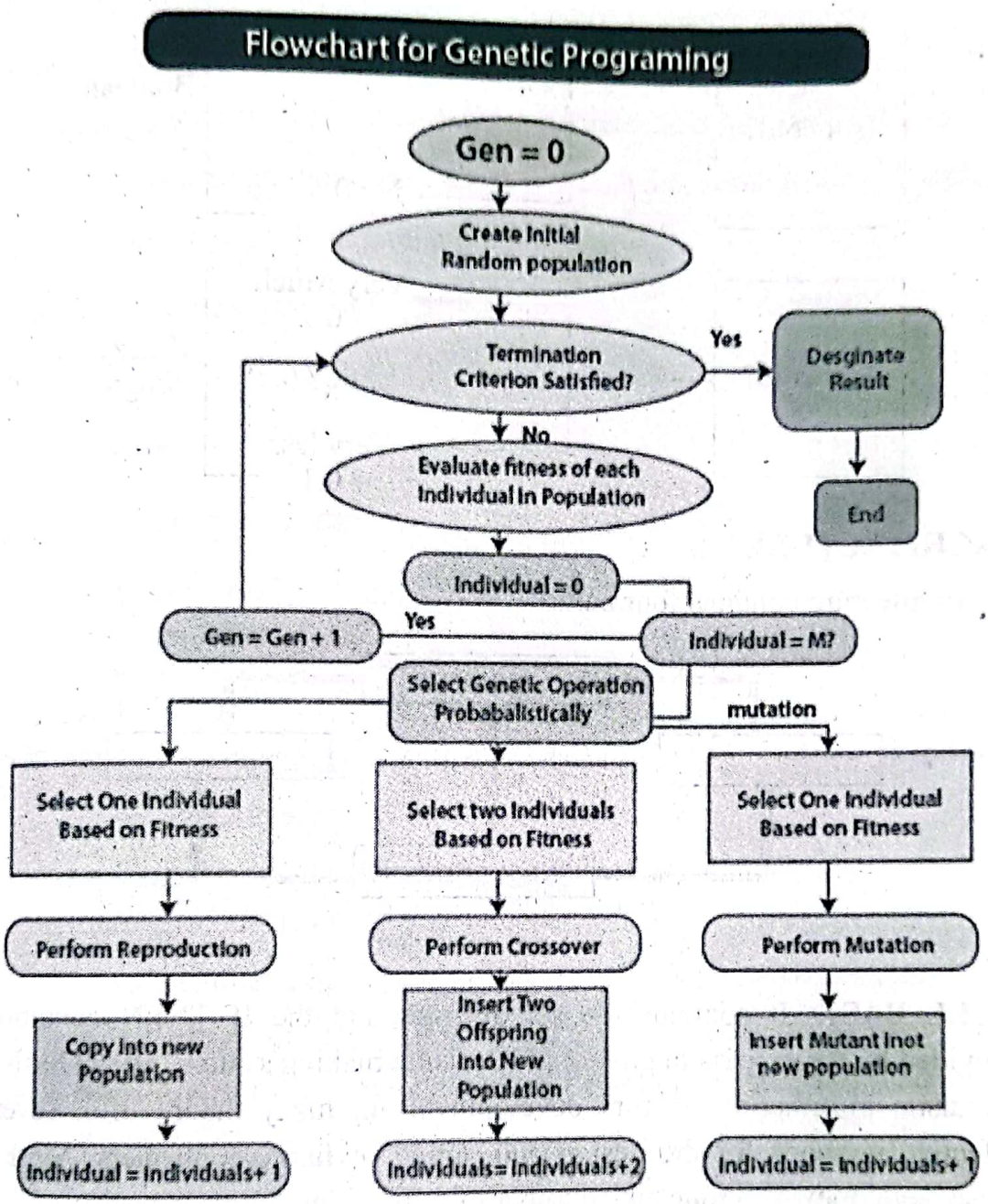
**Test:**

If the end condition is satisfied, then it stops and returns the best solution in the current population.

**Loop:**

In this step, you need to go to the second step for fitness evaluation.

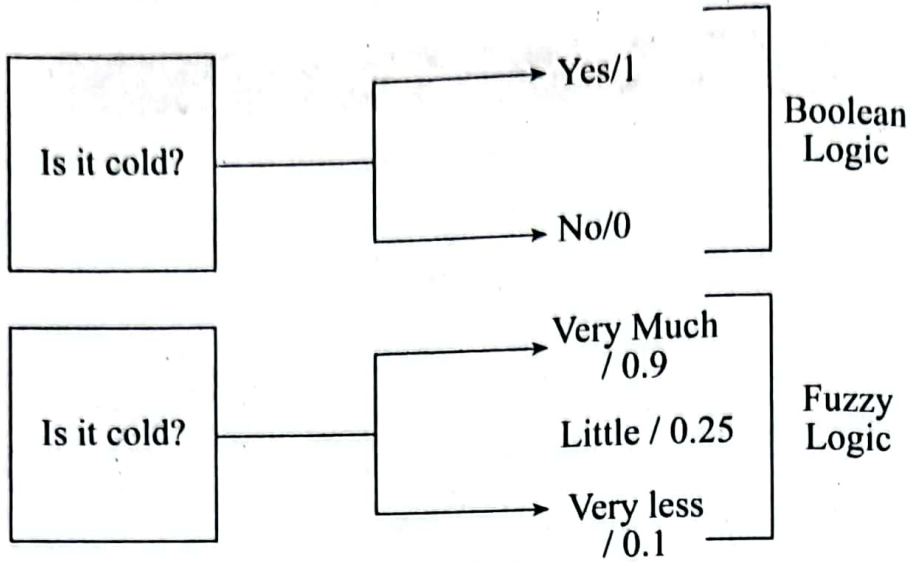
The basic principle behind the genetic algorithms is that they generate and maintain a population of individuals represented by chromosomes. Chromosomes are a character string practically equivalent to the chromosomes appearing in DNA. These chromosomes are usually encoded solutions to a problem. It undergoes a process of evolution as per rules of selection, reproduction, and mutation. Each individual in the environment (represented by chromosome) gets a measure of its fitness in the environment. Reproduction chooses individuals with high fitness values in the population. Through crossover and mutation of such individuals, a new population is determined in which individuals might be an even better fit for their environment. The process of crossover includes two chromosomes swapping chunks of data and is analogous to the process of reproduction. Mutation introduces slight changes into a little extant of the population, and it is representative of an evolutionary step.




## Fuzzy Learning

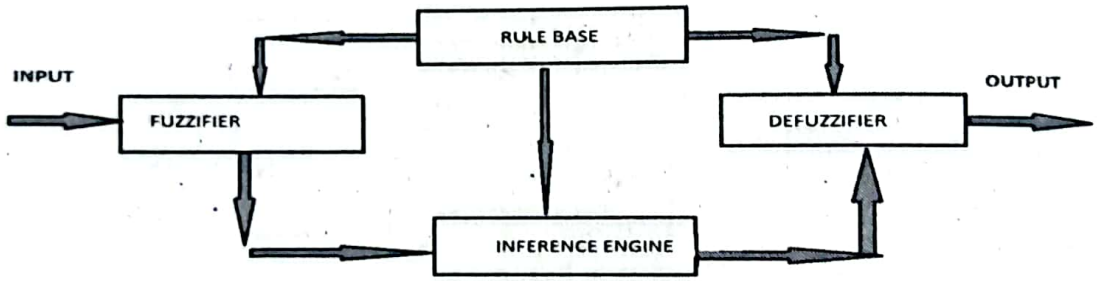
The term **fuzzy** refers to things which are not clear or are vague. In the real world many times we encounter a situation when we can't determine whether the state is true or false, their fuzzy logic provides a very valuable flexibility for reasoning. In this way, we can consider the inaccuracies and uncertainties of any situation.

In boolean system truth value, 1.0 represents absolute truth value and 0.0 represents absolute false value. But in the fuzzy system, there is no logic for absolute truth and absolute false value. But in fuzzy logic, there is intermediate value too present which is partially true and partially false.




## ARCHITECTURE

Its Architecture contains four parts:




FUZZY LOGIC ARCHITECTURE

**RULE BASE:** It contains the set of rules and the IF-THEN conditions provided by the experts to govern the decision making system, on the basis of linguistic information. Recent developments in fuzzy theory offer several effective methods for the design and tuning of fuzzy controllers. Most of these developments reduce the number of fuzzy rules.

**FUZZIFICATION:** It is used to convert inputs i.e. crisp numbers into fuzzy sets. Crisp inputs are basically the exact inputs measured by sensors and passed into the control system for processing, such as temperature, pressure, rpm's, etc.

**INFERENCE ENGINE:** It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules are to be fired according to the input field. Next, the fired rules are combined to form the control actions.

**DEFUZZIFICATION:** It is used to convert the fuzzy sets obtained by inference engine into a crisp value. There are several defuzzification methods available and the best suited one is used with a specific expert system to reduce the error.

**Membership function**

**Definition:** A graph that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred as the universe of discourse or universal set (u), which contain all the possible elements of concern in each particular application.

There are largely three types of fuzzifiers:

Singleton fuzzifier

Gaussian fuzzifier

Trapezoidal or triangular fuzzifier

**What is Fuzzy Control?**

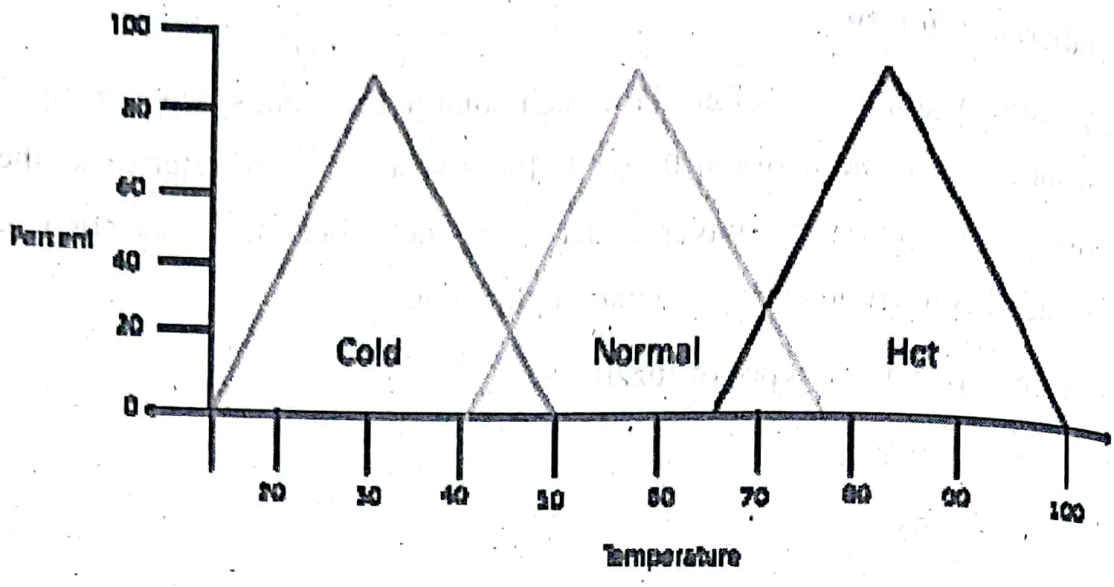
- It is a technique to embody human-like thinking's into a control system.
- It may not be designed to give accurate reasoning but it is designed to give acceptable reasoning.
- It can emulate human deductive thinking, that is, the process people use to infer conclusions from what they know.
- Any uncertainties can be easily dealt with the help of fuzzy logic.

**Fuzzy Logic in AI: Example**

The design of a fuzzy logic system starts with a set of membership functions for each input and a set for each output. A set of rules is then applied to the membership functions to yield a crisp output value. Let's take an example of process control and understand fuzzy logic.

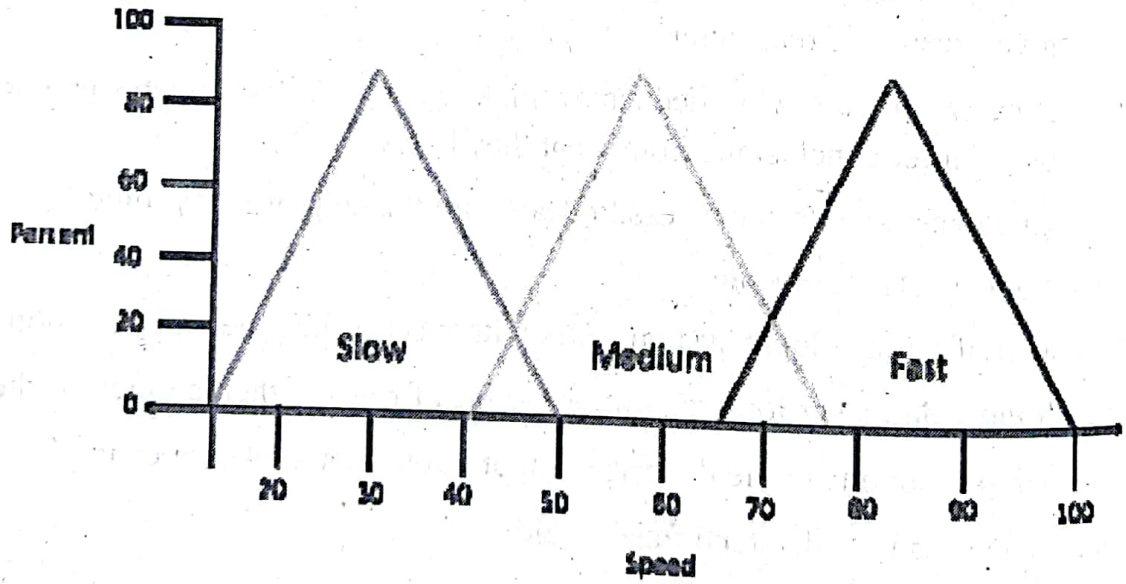
**Step 1**

Here, **Temperature** is the input and **Fan Speed** is the output. You have to create a set of membership functions for each input. A membership function is simply a graphical representation of the fuzzy variable sets. For this example, we will use three fuzzy sets, **Cold, Warm** and **Hot.** We will then create a membership function for each of three sets of temperature:




**Step 2**

In the next step, we will use three fuzzy sets for the output, Slow, Medium and Fast. A set of functions is created for each output set just as for the input sets.



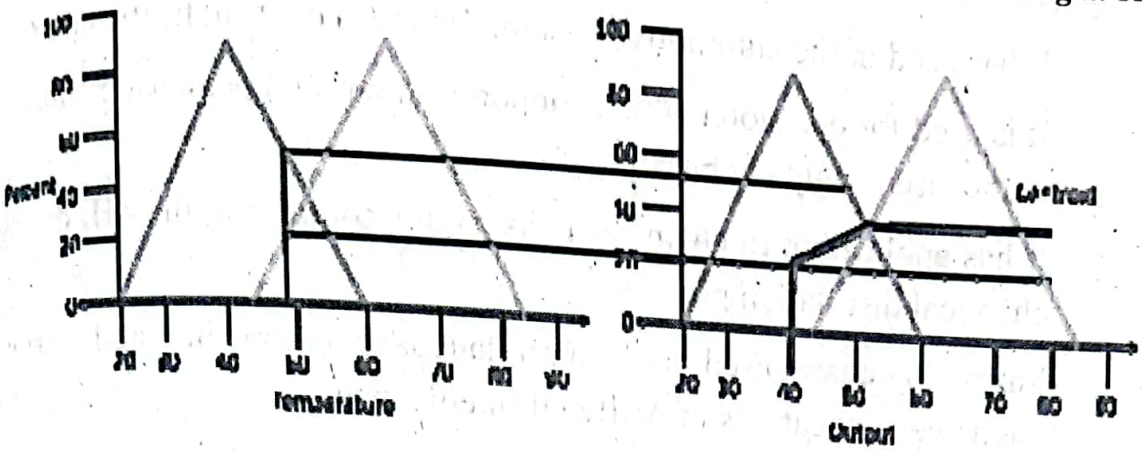

**Step 3**

Now that we have our membership functions defined, we can create the rules that will define how the membership functions will be applied to the final system. We will create three rules for this system.

If Hot then Fast

If Warm then Medium

And, if Cold then Slow




These rules apply to the membership functions to produce the crisp output value to drive the system. Thus, for an input value of 52 degrees, we intersect the membership functions. Here, we are applying two rules as the intersection occurs on both functions. You can extend the intersection points to the output functions to produce an intersecting point. You can then truncate the output functions at the height of the intersecting points.

**Advantages of Fuzzy Logic System**

This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.

The construction of Fuzzy Logic Systems is easy and understandable.

Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.

It provides a very efficient solution to complex problems in all fields of life as it resembles human reasoning and decision making.

The algorithms can be described with little data, so little memory is required.

**Disadvantages of Fuzzy Logic Systems**

Many researchers proposed different ways to solve a given problem through fuzzy logic which lead to ambiguity. There is no systematic approach to solve a given problem through fuzzy logic.

Proof of its characteristics is difficult or impossible in most cases because every time we do not get mathematical description of our approach.

As fuzzy logic works on precise as well as imprecise data so most of the time accuracy is compromised.

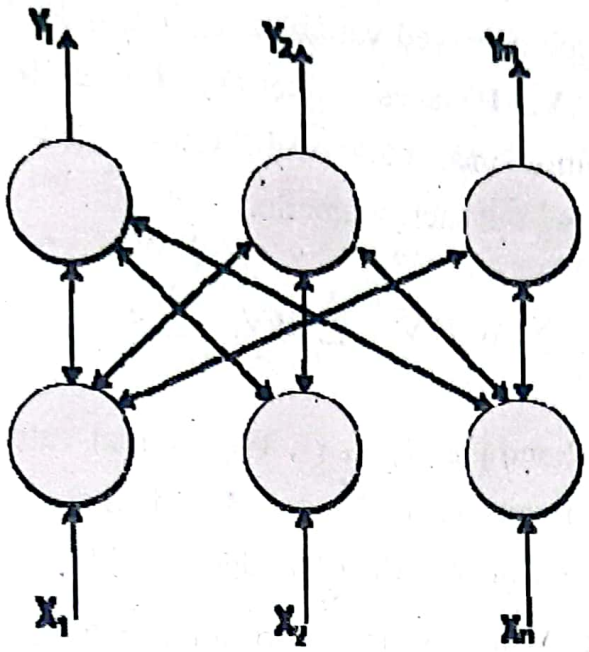
**Application**

- It is used in the aerospace field for altitude control of spacecraft and satellite.

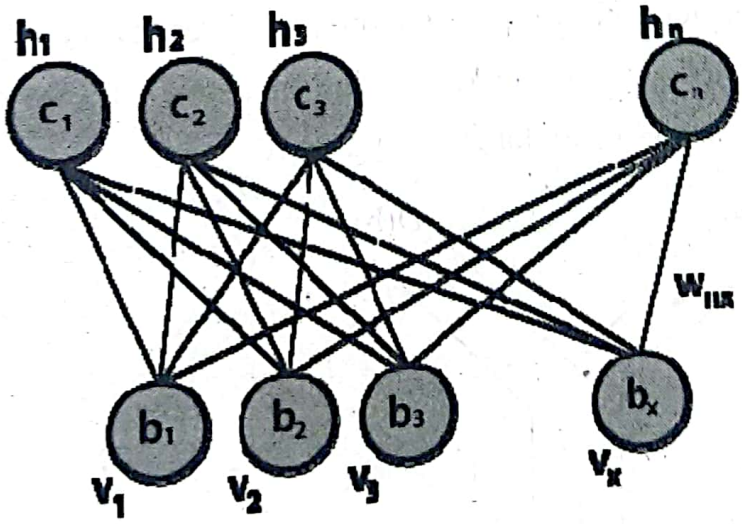
- It has used in the automotive system for speed control, traffic control.
- It is used for decision making support systems and personal evaluation in the large company business.
- It has application in chemical industry for controlling the pH, drying, chemical distillation process.
- Fuzzy logic are used in Natural language processing and various intensive applications in Artificial Intelligence.
- Fuzzy logic are extensively used in modern control systems such as expert systems.
- Fuzzy Logic is used with Neural Networks as it mimics how a person would make decisions, only much faster. It is done by Aggregation of data and changing into more meaningful data by forming partial truths as Fuzzy sets.

**Boltzaman Machine**

Boltzmann machines (BMs) have been introduced as bi-directionally connected networks of **hypothetical** processing units, which can be interpreted as neural network models. Boltzmann machines can be used to learn significant aspects of an unknown probability distribution based on samples from the distribution. Generally, this learning procedure is difficult and tedious. However, the learning issue can be simplified by imposing limitations on the network topology, which leads us to restricted Boltzmann machines. A restricted BM is a generative model representing a probability distribution. Given a few observations, the training data, learning a BM, implies changing the BM parameters to such that the probability distribution represented by the BM fits the training data. Boltzmann machines comprise of two types of units, visible and hidden neurons, which can be arranged in two layers. The visible units establish the primary layer and correspond to the components of observations. For example, one visible unit for each pixel of a digital image. In the hidden units, model dependencies lie between the components of observations. For example, dependencies between pixels in images. They can be viewed as non-linear element indicators.




Boltzmann machines can also be viewed as specific graphical models, more precisely undirected graphical models, also known as Markov random fields. The embedding of BMs into the structure of probabilistic graphical models gives quick access to an abundance of hypothetical outcomes and well-developed algorithms. Computing the probability of an undirected model or its gradient for inference is generally computationally comprehensive. Hence, sampling-based techniques are utilized to estimate the probability and its gradient. Sampling from an undirected graphical model is usually not simple, but form RBMs Markov Chain Monte Carlo (MCMC) techniques are easily applicable in the form of Gibbs sampling.




A restricted Boltzmann machine (RBM) is a Markov random field (MRF) related to a two undirected graph illustrated in the given figure. It includes x visible units **V = (V1,..., Vx)** to show data and n hidden units **H =**

**(H1,..., Hn)** to catch observed variables. In binary RBMs, our focus on the random variables **(V, H)** takes values **(V, H)** $\in \{0,1\}^{x+n}$ and the joint probability distribution under the model given by the Gibbs distribution **P(V, H) = 1/z** $^{e-E(v,h)}$ having the energy function.

$$E(v, h) = \sum_{i=1}^{n} \sum_{j=1}^{x} W_{ij}H_iV_j - \sum_{j=1}^{x} B_jV_j - \sum_{j=1}^{x} C_iH_i$$

For all **i** $\in$ **{1, ..., n}** and **j** $\in$ **{1, ..., x}**, $W_{ij}$ is a real-valued weight connected to the edge between units $V_j$ and $H_i$, and $B_j$ and **Ci** are real-valued bias terms connected to the $j^{th}$ visible and the $i^{th}$ hidden variable, respectively.

The graph of an RBM has an only association between the layer of hidden and visible variables but not between two variables of a similar layer. In terms of probability, it implies that the hidden variables are independent, given the state of the visible variables and vice versa.

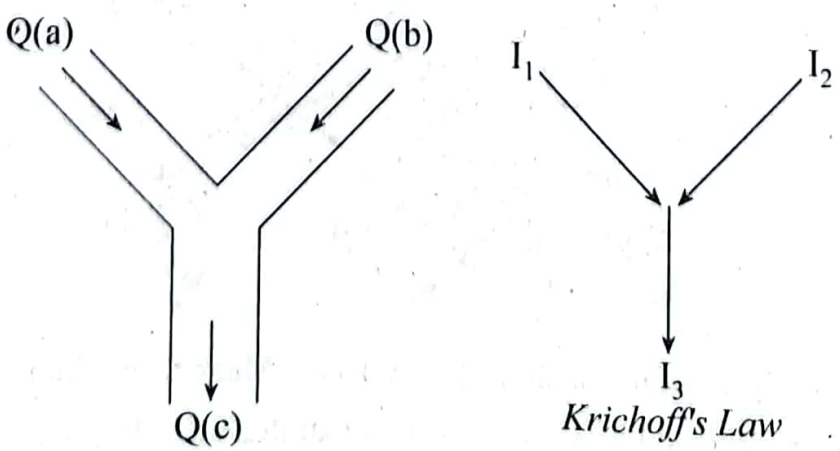
## OLD QUESTION SOLUTION

1. **What is machine learning? Explain learning the analogy with example. (2076 Baisakh)**

⇒ **1st part:** See on page no115

Learning by analogy means acquiring new knowledge about an entity by transferring it from a known similar entity.

E.g: consider the two problem domain. These two problem domain are analogous. Having the knowledge of domain II, we can transfer the some knowledge to domain I and derive.




*Hydraulic Problem* *Krichoff's Law*

2. **What do you understand by swarm intelligence? Suppose chromosomes are of the from x = a, b, c, d, e, f, g, h, with a fixed length of eight genes. Each gene can be any digit between f(x) = (a + b) – (c + d) + (e + f) – (g + h) and let the initial population consist of four individuals with the following chromosomes.**

**$X_1 = 65413532$**

**$X_2 = 87126601$**

**$X_3 = 23921285$**

**$X_4 = 41852094$** **(2075 Bhadra)**

⇒ Swarm intelligence is the discipline that deals with natural and artificial system composed of many individuals that co - ordinate using decentralized control and self - organization.

E.g.: Colonies of ants and termites, schools of fish, flocks of birds, herbs of land animals etc.

a) **Evaluate the fitness of each individuals showing all your working and arrange them in order with the fittest first and the least fit last.**

**Answer:**

$F(X_1) = (6 + 5) - (4 + 1) + (3 + 5) - (3 + 2) = 9$

$F(X_2) = (8 + 7) - (1 + 2) + (6 + 6) - (0 + 1) = 23$

$F(X_3) = (2 + 3) - (9 + 2) + (1 + 2) - (8 + 5) = 16$

$F(X_4) = (4 + 1) - (8 + 5) + (2 + 0) - (9 + 4) = -19$

The order is $x_2$, $x_1$, $x_3$ and $x_4$.

b) **Perform the following crossover operations:**

i) Cross the fittest two individuals using one - point crossover at the middle point.

Answer:

One - point cross over on $x_2$ and $x_1$:

$$\begin{array}{l} x_2 = 8712 \mid 6601 \\ x_1 = 6541 \mid 6532 \end{array} \Rightarrow \begin{array}{l} 0_1 = 87123532 \\ 0_2 = 65416601 \end{array}$$

ii) Cross the second and third fittest individuals using a two - point crossover at the point b and F.

Answer: Two point crossover on $x_1$ and $x_3$.

$$\begin{array}{l} x_1 = 65 \,|\, 4135 \,|\, 32 \\ x_2 = 23 \,|\, 9212 \,|\, 85 \end{array} \Rightarrow \begin{array}{l} o_3 = 65921232 \\ o_4 = 23413585 \end{array}$$

iii) Cross the first and third fittest individuals (ranked 1st and 3rd) using a uniform crossover.

Answer: In the simplest case uniform crossover means just a random exchange of genes between two parents. For e.g: we may swap genes at positions a, d and F of parents $x_2$ and $x_3$.

$$\begin{array}{l} x_2 = 87126601 \\ x_3 = 23921285 \end{array} \Rightarrow \begin{array}{l} Q_5 = 27526201 \\ Q_6 = 83921685 \end{array}$$

**c) Suppose the new population consists of the six offspring individuals received by the crossover operations in above question. Evaluate the fitness of the new population showing all your workings. Has the overall fitness improved?**

**Answer:**

The new population is:

$0_1 = 87123532$

$0_2 = 65416601$

$0_3 = 659211232$

$0_4 = 23413585$

$0_5 = 27126201$

$0_6 = 83921685$

Now apply the fitness function:

$F(2) = (a + b) - (c + d) + (e + f) - (g + h)$

$F(0_1) = (8 + 7) - (1 + 2) + (3 + 5) + (3 + 2) = 15$

$F(0_2) = (6 + 5) - (4+1) + (6 + 6) - (0 + 1) = 17$

$F(0_3) = (6 + 5) - (9 + 2) + (1 +2) - (3 + 2) = -2$

$F(0_4) = (2 + 3) - (4 +1) + (3 + 5) - (8 + 5) = -5$

$F(0_5) = (2 + 7) - (1 + 2) + (6 + 2) - (0 + 1) = 13$

$F(0_6) = (8 + 3) - (9 + 2) + (1 + 6) - (8 + 5) = -6$

The overall fitness has improved.

d) **By looking at the fitness function and considering that genes can only be digits between 0 and 9. Find the chromosome representing the optimal solution (i.e. with the maximum fitness) find the value of the maximum fitness.**

**Answer:**

The optimal solution have a chromosome that gives the maximum of the fitness function.

max F(x) = max [(a + b) - (c + d) + (e + f) - (g + n)]

because genes can only be digits from 0 to 9, the optimal solution should be.

optimal = 99009900

and the maximum fitness is

F(x optimal) = (9 + 9) – (0 + 0) + (9 + 9) – (0 + 0) = 36

3. **What is genetic algorithm? Explain all the steps in genetic algorithm with block diagram and operators? (2075 Baisakh)**

⇒ See on page no 120

4. **Justify that the study of gene is one of the important part in the AI. List down the steps involved in genetic algorithm with an example. (2074 Bhadra)**

⇒ Genetic Algorithms (GAS) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data of direct the search into the region of better performance in solution space. They are commonly used to generate high - quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of fittest" among individual of consecutive generation for solving a problem". Hence, the study of gene is one of the important part in the AI.

**2nd part:** See on page no

5. **Write short notes on:**

**Supervised vs unsupervised learning** **(2074 Bhadra)**

⇒

| Supervised Learning | Unsupervised Learning |
|---|---|
| Supervised learning algorithms are trained using labeled data. | Unsupervised learning algorithms are trained using unlabeled data. |
| Supervised learning model takes direct feedback to check if it is predicting correct output or not. | Unsupervised learning model does not take any feedback. |
| Supervised learning model predicts the output. | Unsupervised learning model finds the hidden patterns in data. |
| In supervised learning, input data is provided to the model along with the output. | In unsupervised learning, only input data is provided to the model. |
| The goal of supervised learning is to train the model so that it can predict the output when it is given new data. | The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset. |
| Supervised learning needs supervision to train the model. | Unsupervised learning does not need any supervision to train the model. |
| Supervised learning can be categorized in **Classification** and **Regression** problems. | Unsupervised Learning can be classified in **Clustering** and **Associations** problems. |

6. **"Learning is an essential characteristic for intelligent agent."List down justification on this statement. Write about the rote learning with suitable example.** **(2073 Bhadra)**

⇒ A learning agent in AI is the type of agent which can learn from its past experiences or it has learning capabilities It starts to act with basic knowledge and then able to act and adapt automatically through learning.

An intelligent agent is a program that can make decision or perform a service based on its environment user input and experiences. The experiences is based on learning. So, learning is based as essential characteristics for intelligent agent.

**2nd part:** See on page no

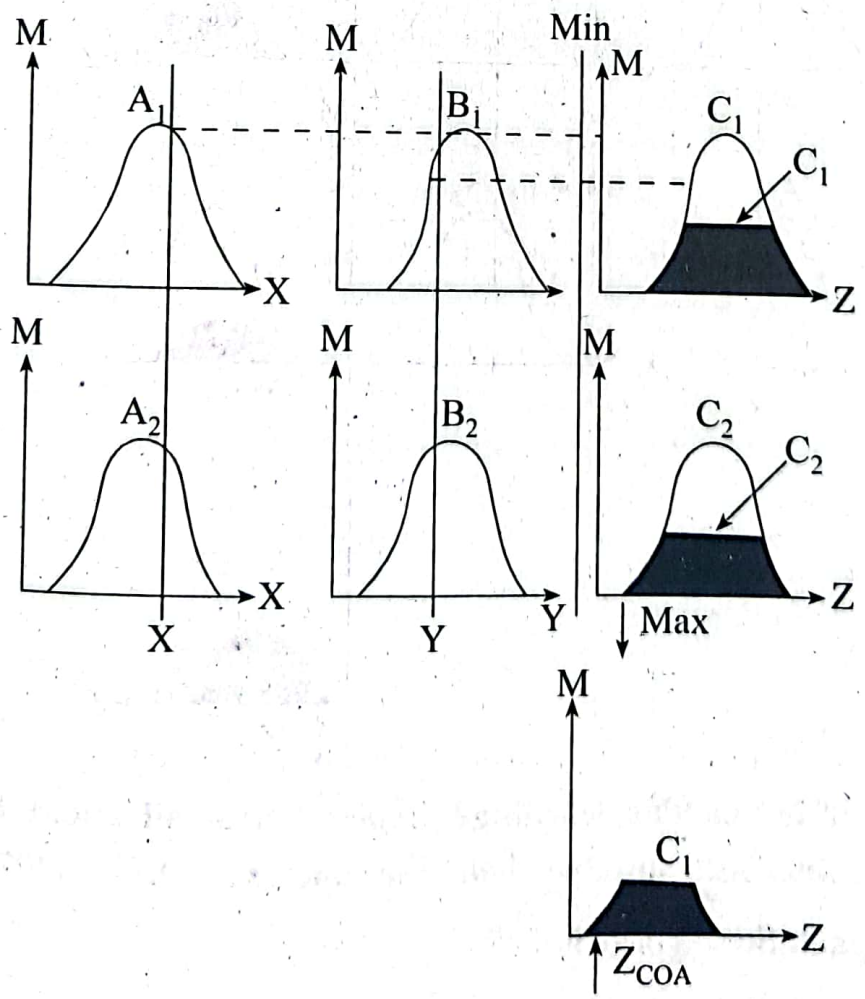
7. **What is fuzzy logic and why is it important? Explain abut Mamdani Fuzzy inference method with example. (2073 Magh)**

⇒ 1st part see on page no

This system was proposed in 1975 by Ebhasim Mamdani. Basically, it was anticipated to control a steam engine and boiler combination by synthesizing a set of Fuzzy rules obtained from people working on the system.

**Steps for computing the output:**

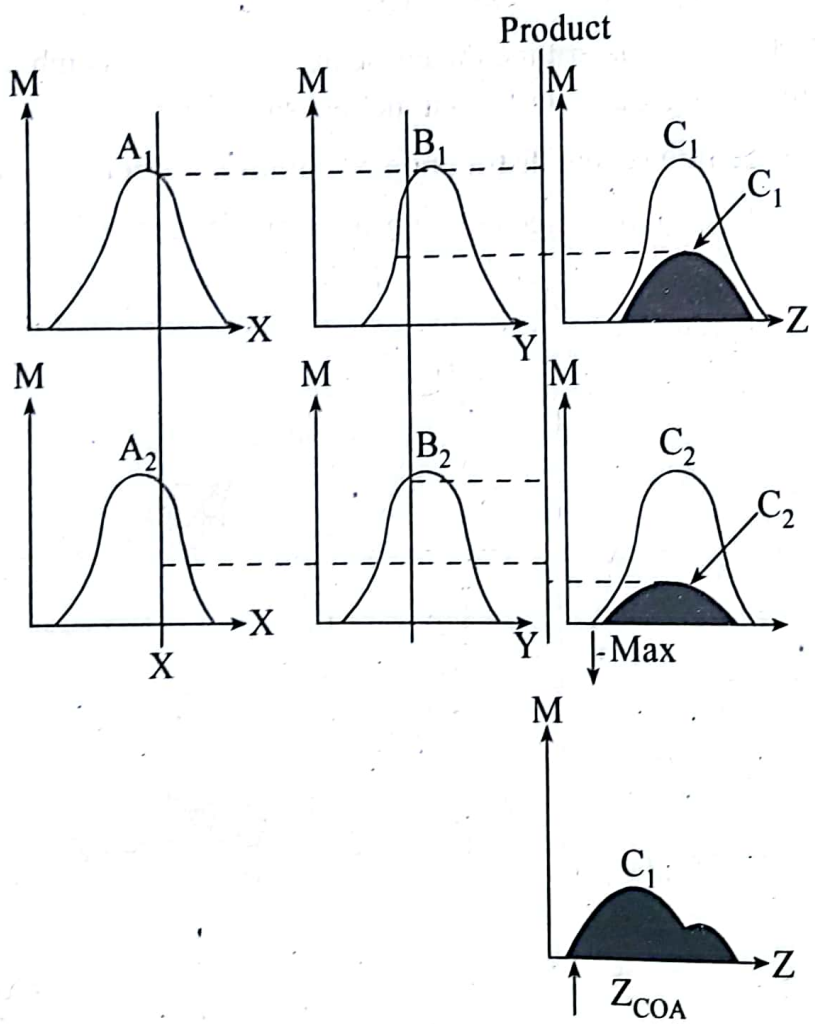
1) Set of fuzzy rules need to be determined in this step.
2) In this step, by using input membership function, the input would be made fuzzy.
3) Now establish the rule strength by combining the Fuzzified inputs according to fuzzy rules.
4) In this step, determine the consequent of rule by combining the rule strength and the output membership function.
5) For getting output distribution combine all the consequences.
6) Finally, a defuzzified output distribution is obtained.




*Fig(i)*

Fig(i) is an illustration of how a two - rule Mamdani fuzzy inference system derives the overall output Z when subjected to two crisp inputs x and y.

If we adopt max and algebric product as our choice for the T – norm and T – conorm operators respectively, and use max - product composition, instead of the original max - min composition, then resulting Fuzzy reasoning is shown in Fig 2, where the inferred output of each rule is a fuzzy set scaled down by its firing strength via algebric product. Although this type of fuzzy reasoning was not employed in Mamdani's original paper, it has often been used in the literature. Other variations are possible if we use different T- norm and T – conorm operators.




8. **What is machine learning? Explain in detail about Boltzmann machines with suitable algorithms and examples.** **(2072 Ashwin)**

⇒ **1st part:** See on page no115

**2nd part:** See on page no128

9. **What is machine learning? Explain genetic algorithm(GA) along with GA operators. List some areas where GA can be applied.** **(2072 Magh)**

⇒ **1st part:** See on page no115

**2nd part:** See on page no120

**Areas that GA can be applied are as follows:**

- Clustering, using **genetic algorithms** to optimize a wide range of different fit-functions.
- Multidimensional systems.
- Multimodal Optimization.
- Multiple criteria production scheduling.
- Multiple population topologies and interchange methodologies.
- Mutation testing.

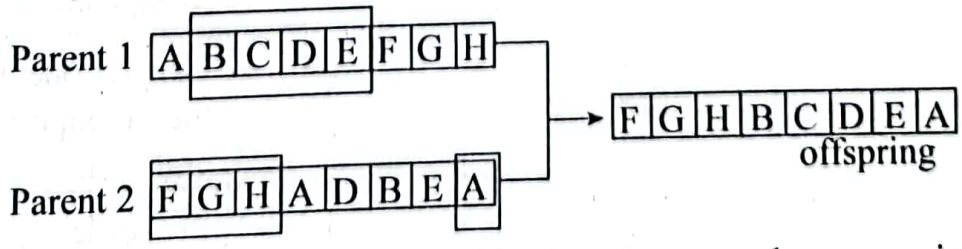
10. **What is machine learning ? Explain genetic algorithm along with different operators of genetic algorithm.** **(2071 Magh)**

⇒ **1st part see on page no115**

The operators involved in genetic algorithm are as follows:

1. **Selection operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to the successive generations.

2. **Crossover Operator:** This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (off spring). For example:




3. **Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For E.g:

Before Mutation | F | G | H | B | C | D | E | A |

After Mutation | F | G | M | B | C | D | E | N |

**11. Compare the following. Genetic algorithm based Versus Fuzzy based learning. (2071 Bhadra)**

⇒

| Fuzzy | GA |
|---|---|
| • While the population of the genetic algorithm undergoes evolution at every generation, the relative 'good' solutions reproduce while the relatively 'bad' solution die. | • A genetic representation for potential solutions to the problem. |
| • To distinguish between solutions an objective (evaluation) function is used. In the simple cases, there is only one criterion for optimization for example, chaximization of profit or minimization of cost. | • Method to create an initial population of potential solutions. |
| • But in many real - world decision making problems, there is a need for simulations optimization of multiple objections. | • An evaluation function to rate solution in terms of the "fitness" |
| | • Selection of individuals for the next generations. |
| | • Genetic operator that after the composition of the children. |
| | • In order to make a successful run of a GA, the values for the parameter of the GA have to be like the population size, parameters for the genetic operator and the terminating condition. |

12. **What is machine learning? What is Fuzzy Logic? Explain the FUZZY inference with suitable example.** **(2070 Magh)**

⇒ **1st part:** See on page no115

**2nd part:** See on page no

13. **What is machine vision? Discuss about the algorithm of Genetic Algorithm** **(2070 Bhadra)**

⇒ **Machine vision** (MV) is the technology and methods used to provide imaging-based automatic inspection and analysis for such applications as automatic inspection, process control, and robot guidance, usually in industry.

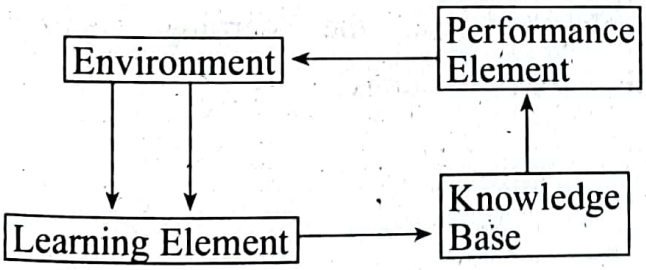
**2nd part:** See on page no 120

14. **What are different operators associated in genetic algorithm.** **(2070 Chaitra)**

**Ans:** See on old question no 10

15. **Explain learning framework with suitable example.** **(2070 Chaitra, 2069 Bhadra)**

⇒




**i) Environment**

- The environment refers the nature and quality of information given to the learning elements.
- The nature of information depends on its level (the degree of generality wrt the performance element)
  - High level information is abstract, it deals with a broad class of problems.
  - Low level information is detailed, it deals with a single problem.

- The quality of information involves
  - Noise free
  - Reliable
  - Ordered

ii) **Learning Elements**

- **Four learning situations**
  - Rote learning
  - Environment provides information at the required level.
- **Learning by being told**
  - Information is too abstract, the learning element must hypothesize missing data.

  **Learning by example**
  - Information is too specific, the learning element must hypothesize more general rules.
- **Learning by analogy**
  - Information provided is relevant only to an analogous task, the learning element must discover the analogy.

iii) **Knowledge Base**

**Expressive**

- The representation contains the relevant knowledge in an easy to get to fashion.

**Modifiable.**

- It must be easy to change the data in the knowledge base.

**Extendibility**

- The knowledge base must contain meta - knowledge (knowledge on how the data base is structured) so the system can change its structure.

iv) **Performance Element**

**Complexity**

- For learning, the simplest task is classification based on a single rule while the most complex task requires the application of multiple rules in sequence.

**Feedback**

- The performance element must send information to the learning system to be used to evaluate the overall performance.

**Transparency**

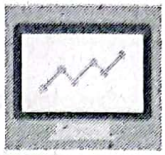
- The learning element should have access to all the internal actions of the performance element.

⌘⌘⌘

# Applications of AI

Chapter SEVEN

## Human Brain Vs Computer

| |  |
|---|---|
| Biological neurons or nerve cells | Silicon transistors |
| 200 billions neurons, 32 trillion interconnections. | 1 billion bytes RAM, trillion of bytes on disk. |
| Neuron size: 10-6 m. | Single transistor size: 10-gm. |
| Energy consumption: 6-10 joules per operation per sec. | Energy consumption: 10-16 joules per operation per second. |
| Learning capability | Programming capability |

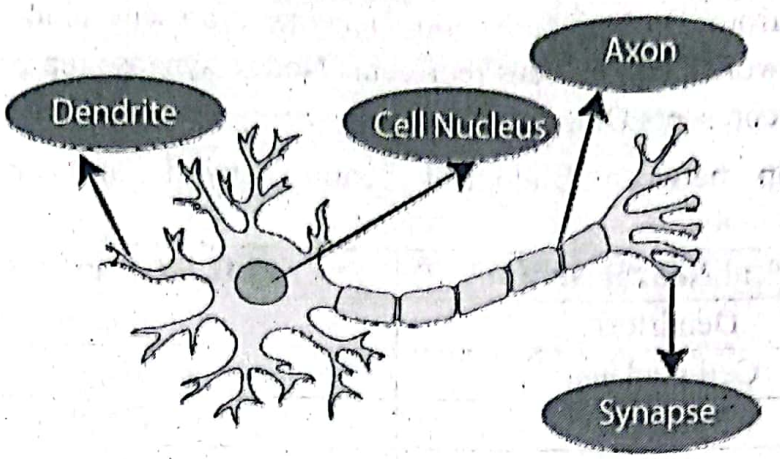
## Neural Network

The term 'Neural' has origin from the human (animal) nervous system's basic functional unit 'neuron' or nerve cells present in the brain and other parts of the human (animal) body. A neural network is a group of algorithms that certify the underlying relationship in a set of data similar to the human brain. The neural network helps to change the input so that the network gives the best result without redesigning the output procedure.

## Parts of Neuron and their Functions

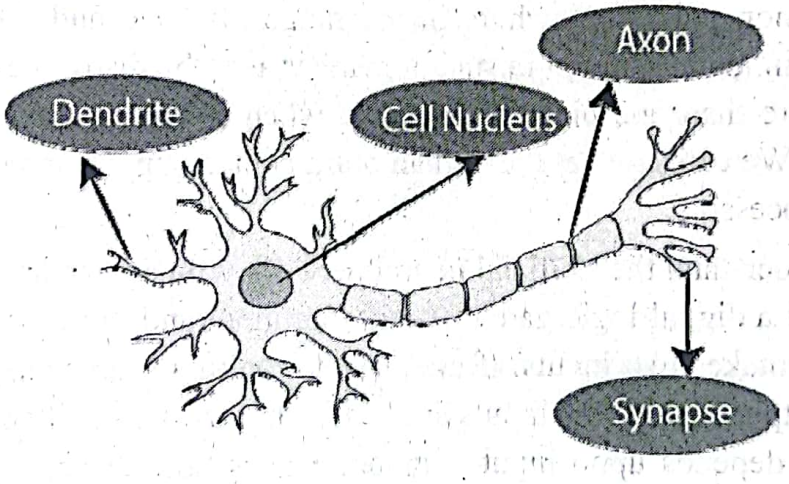
The typical nerve cell of the human brain comprises of four parts –

- **Dendrite:** It receives signals from other neurons.
- **Soma (cell body):** It sums all the incoming signals to generate input.
- **Axon Structure:** When the sum reaches a threshold value, the neuron fires, and the signal travels down the axon to the other neurons.
- **Synapses:** The point of interconnection of one neuron with other neurons. The amount of signal transmitted depends upon the strength (synaptic weights) of the connections.



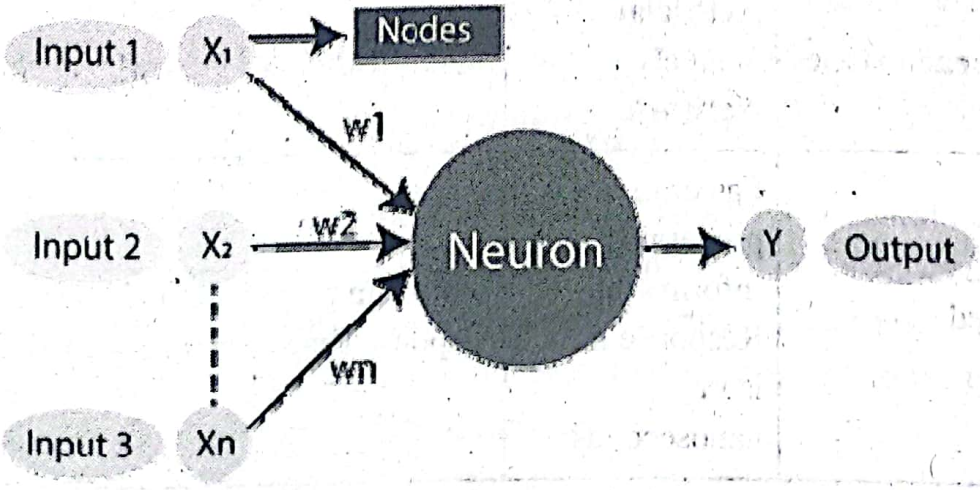

**Artificial Neural Network**

The term "**Artificial Neural Network**" is derived from Biological neural networks that develop the structure of a human brain. Similar to the human brain that has neurons interconnected to one another, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.




The given figure illustrates the typical diagram of Biological Neural Network.

The typical Artificial Neural Network looks something like the given figure.



Dendrites from Biological Neural Network represent inputs in Artificial Neural Networks, cell nucleus represents Nodes, synapse represents Weights, and Axon represents Output.

Relationship between Biological neural network and artificial neural network:

| Biological Neural Network | Artificial Neural Network |
|---|---|
| Dendrites | Inputs |
| Cell nucleus | Nodes |
| Synapse | Weights |
| Axon | Output |

An **Artificial Neural Network** in the field of **Artificial intelligence** where it attempts to mimic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner. The artificial neural network is designed by programming computers to behave simply like interconnected brain cells.

There are around 1000 billion neurons in the human brain. Each neuron has an association point somewhere in the range of 1,000 and 100,000. In the human brain, data is stored in such a manner as to be distributed, and we can extract more than one piece of this data when necessary from our memory parallelly. We can say that the human brain is made up of incredibly amazing parallel processors.

We can understand the artificial neural network with an example, consider an example of a digital logic gate that takes an input and gives an output. "OR" gate, which takes two inputs. If one or both the inputs are "On," then we get "On" in output. If both the inputs are "Off," then we get "Off" in output. Here the output depends upon input. Our brain does not perform the same task. The outputs to inputs relationship keep changing because of the neurons in our brain, which are "learning."

**Artificial Neural Network Vs Biological Neural Network**

| Characteristics | Artificial Neural Network | Biological(Real) Neural Network |
|---|---|---|
| Speed | Faster in processing information. Response time is in nanoseconds. | Slower in processing information. The response time is in milliseconds. |

| | | |
|---|---|---|
| Processing | Serial processing. | Massively parallel processing. |
| Size & Complexity | Less size & complexity. It does not perform complex pattern recognition tasks. | A highly complex and dense network of interconnected neurons containing neurons of the order of 1011 with 1015 of interconnections.<strong |
| Storage | Information storage is replaceable means replacing new data with an old one. | A highly complex and dense network of interconnected neurons containing neurons of the order of 1011 with 1015 of interconnections. |
| Fault tolerance | Fault intolerant. Corrupt information cannot retrieve in case of failure of the system. | Information storage is adaptable means new information is added by adjusting the interconnection strengths without destroying old information. |
| Control Mechanism | There is a control unit for controlling computing activities | No specific control mechanism external to the computing task. |

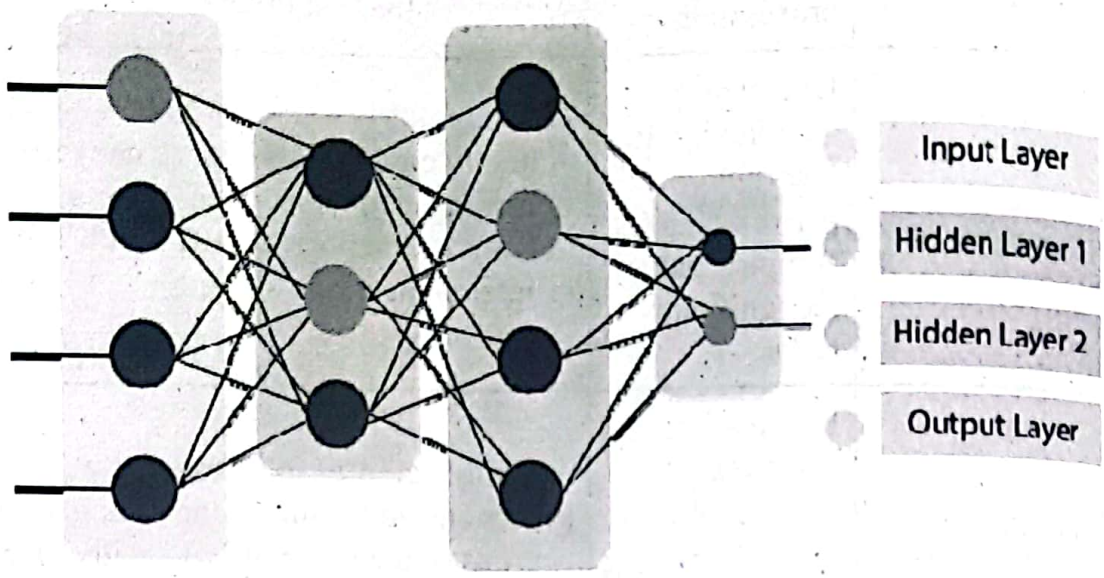
Neural Networks resemble the human brain in the following two ways -

- A neural network acquires knowledge through learning.
- A neural network's knowledge is a store within inter-neuron connection strengths known as synaptic weights.

**Architecture of an artificial neural network:**

To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed units arranged in a sequence of layers. Lets us look at various types of layers available in an artificial neural network.

Artificial Neural Network primarily consists of three layers:




**Input Layer:**

As the name suggests, it accepts inputs in several different formats provided by the programmer.

**Hidden Layer:**

The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

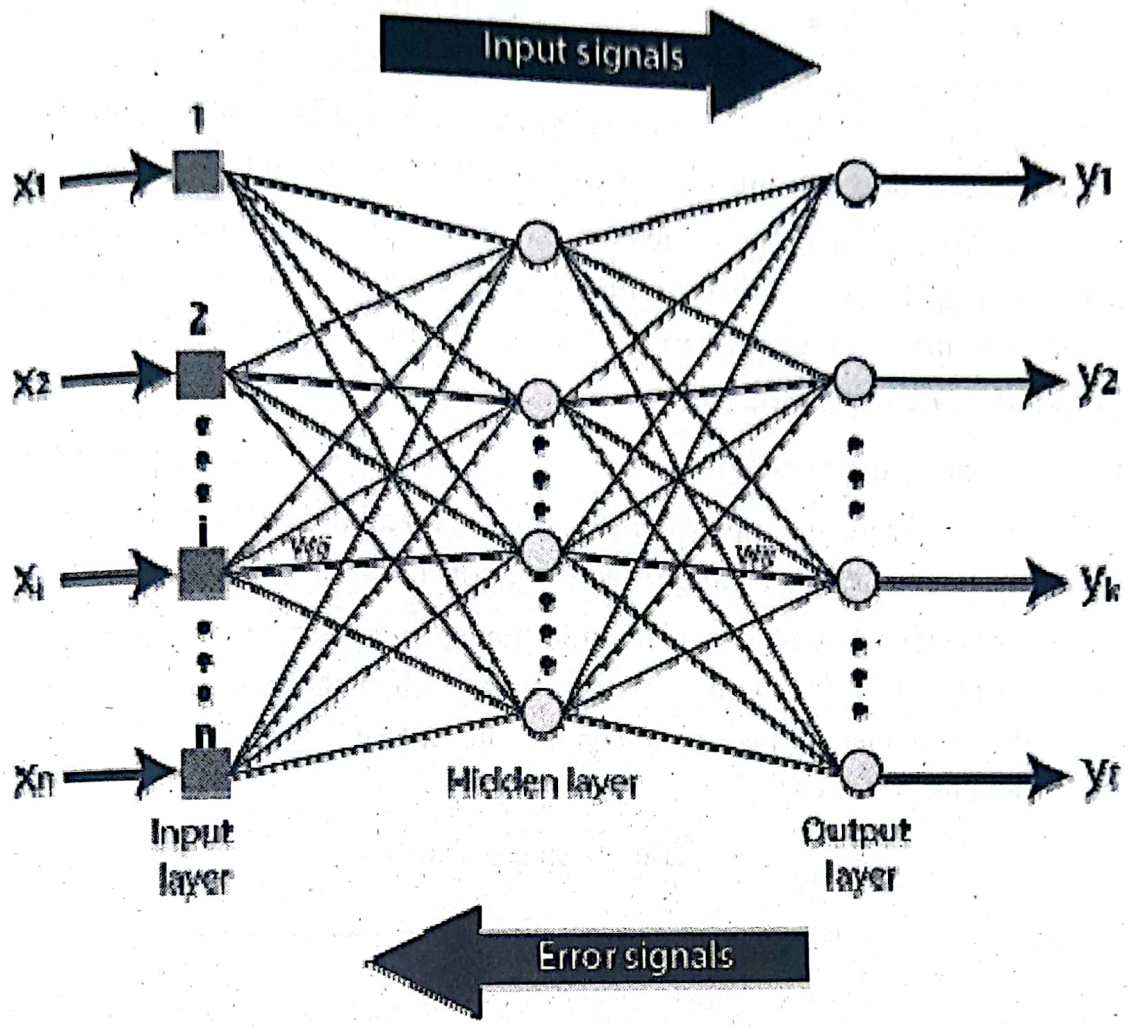
**Output Layer:**

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

The artificial neural network takes input and computes the weighted sum of the inputs and includes a bias. This computation is represented in the form of a transfer function.

**Working of Artificial Neural Networks**

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.




Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions.

## Activation functions

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

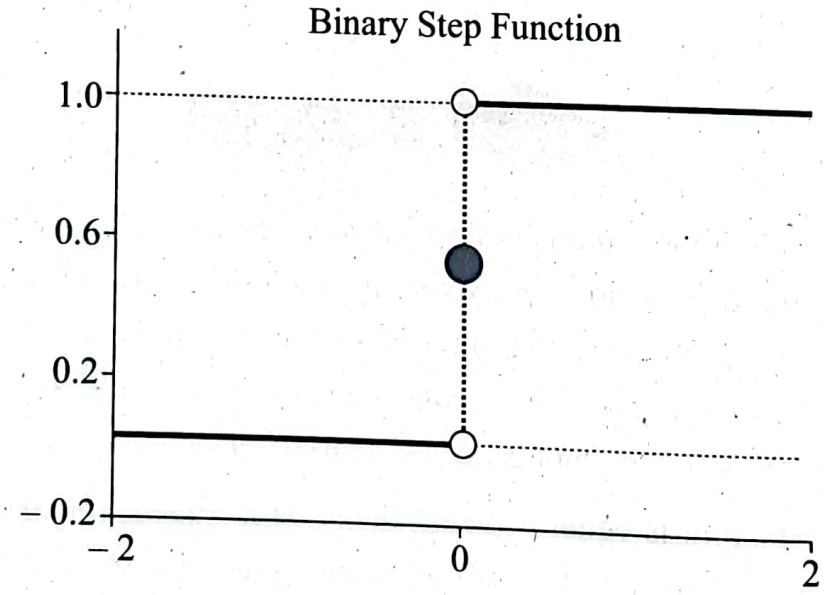
A neural network without an activation function is essentially just a linear regression model. The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

## Types of Activation Functions

1. **Binary Step Function**

   Binary step function depends on a threshold value that decides whether a neuron should be activated or not.

   The input fed to the activation function is compared to a certain threshold; if the input is greater than it, then the neuron is activated, else it is deactivated, meaning that its output is not passed on to the next hidden layer.




Mathematically it can be represented as:

Binary step

$$f(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$

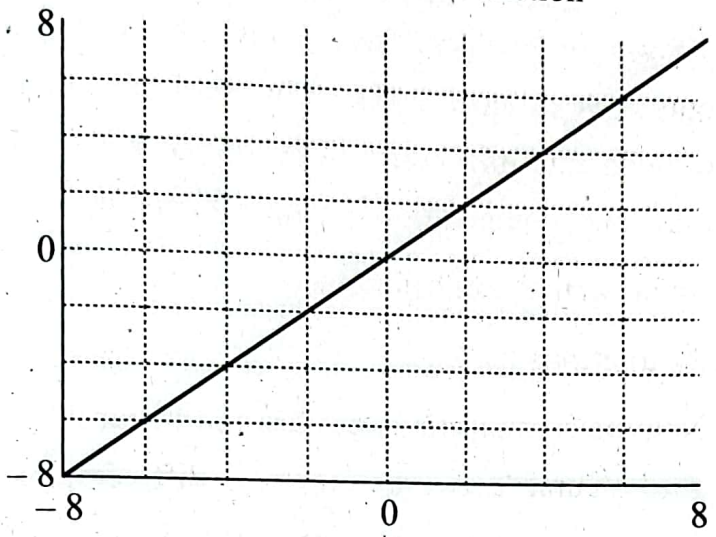
Here are some of the limitations of binary step function:

- It cannot provide multi-value outputs—for example, it cannot be used for multi-class classification problems.
- The gradient of the step function is zero, which causes a hindrance in the backpropagation process.

2. **Linear Activation Function**

The linear activation function is also known as Identity Function where the activation is proportional to the input.

Linear Activation Function




Mathematically it can be represented as:

Linear

$f(x) = x$

However, a linear activation function has two major problems:

- It's not possible to use backpropagation as the derivative of the function is a constant and has no relation to the input x.
- All layers of the neural network will collapse into one if a linear activation function is used. No matter the number of layers in the neural network, the last layer will still be a linear function of the first layer. So, essentially, a linear activation function turns the neural network into just one layer.

3. **Non-Linear Activation Functions**

- The linear activation function shown above is simply a linear regression model.
- Because of its limited power, this does not allow the model to create complex mappings between the network's inputs and outputs.
- Non-linear activation functions solve the following limitations of linear activation functions:

- They allow back propagation because now the derivative function would be related to the input, and it's possible to go back and understand which weights in the input neurons can provide a better prediction.
- They allow the stacking of multiple layers of neurons as the output would now be a non-linear combination of input passed through multiple layers. Any output can be represented as a functional computation in a neural network.

**Types of non-linear activation function:**

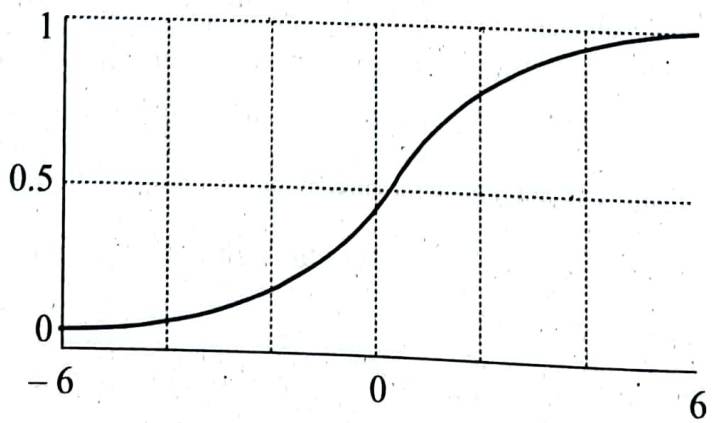
**1. Sigmoid Functions**

In an ANN, the sigmoid function is a non-linear AF used primarily in feedforward neural networks. It is a differentiable real function, defined for real input values, and containing positive derivatives everywhere with a specific degree of smoothness. The sigmoid function appears in the output layer of the deep learning models and is used for predicting probability-based outputs. The sigmoid function is represented as:

Sigmoid/Logistic

$$f(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid / Logistic



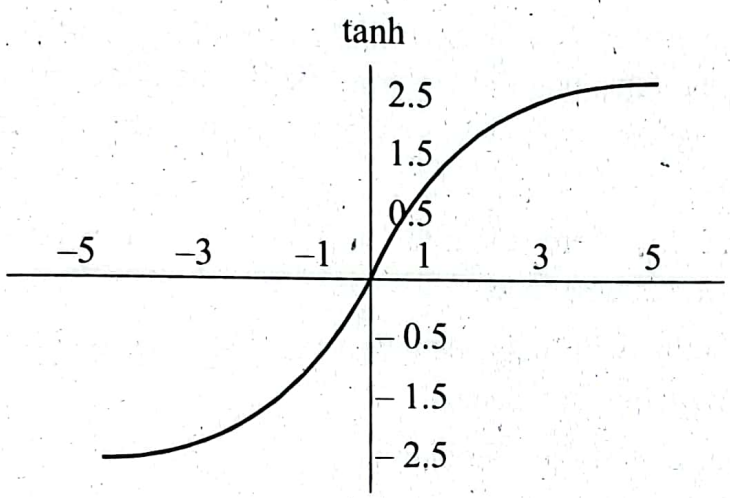

**2. Tanh Function (Hyperbolic Tangent)**

The hyperbolic tangent function, a.k.a., the tanh function, is another type of AF. It is a smoother, zero-centered function having a range between -1 to 1.

The tanh function is much more extensively used than the sigmoid function since it delivers better training performance for multilayer neural networks. The biggest advantage of the tanh function is that it produces a zero-centered output, thereby supporting the backpropagation process. The tanh function has been mostly used in recurrent neural networks for natural language processing and speech recognition tasks. As a result, the output of the tanh function is represented by:

tanh

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$$

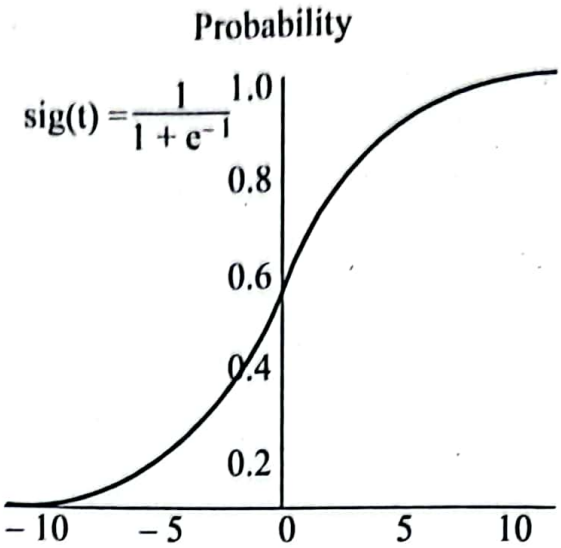



3. **Softmax Function**

The softmax function is another type of AF used in neural networks to compute probability distribution from a vector of real numbers. This function generates an output that ranges between values 0 and 1 and with the sum of the probabilities being equal to 1. This function is mainly used in multi-class models where it returns probabilities of each class, with the target class having the highest probability. It appears in almost all the output layers of the DL architecture where they are used. The primary difference between the sigmoid and softmax AF is that while the former is used in binary classification, the latter is used for multivariate classification. The softmax function is represented as follows:

softmax

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\Sigma_j \exp(z_j)}$$




4. **ReLU Function**

ReLU stands for Rectified Linear Unit. Although it gives an impression of a linear function, ReLU has a derivative function and allows for backpropagation while simultaneously making it computationally efficient.

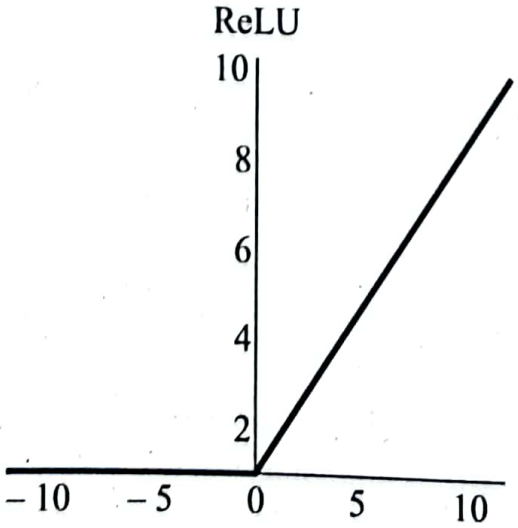
The main catch here is that the ReLU function does not activate all the neurons at the same time.

The neurons will only be deactivated if the output of the linear transformation is less than 0.

Mathematically it can be represented as:

ReLU

$f(x) = \max(0, x)$

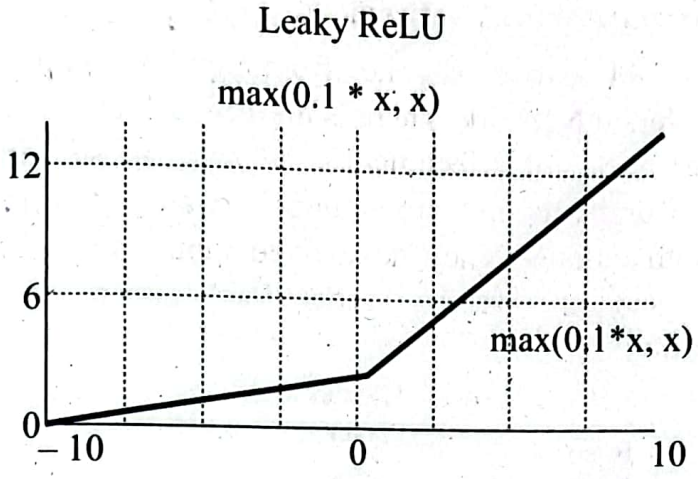


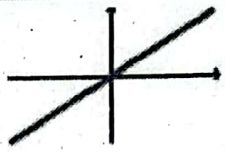
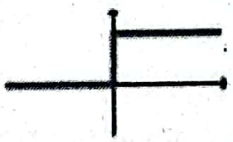
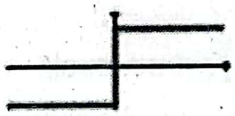
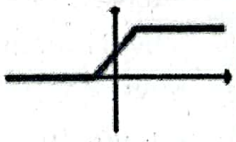
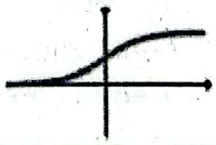
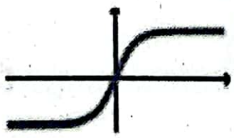
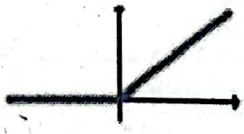

5. **Leaky ReLU Function**

Leaky ReLU is an improved version of ReLU function to solve the Dying ReLU problem as it has a small positive slope in the negative area. Mathematically it can be represented as:

Leaky ReLU

$f(x) = \max(0.1x, x)$



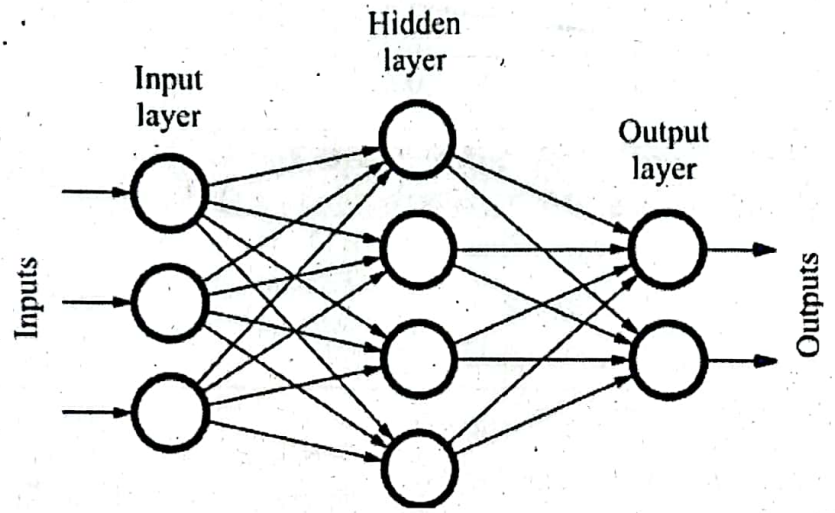

| Activation Function | Equation | Example | 1D Graph |
|---|---|---|---|
| Linear | $\phi(z) = z$ | Adaline, linear regression |  |
| Unit Step (Heaviside Function) | $\phi(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$ | Perceptron variant |  |
| Sign (signum) | $\phi(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$ | Perceptron variant |  |
| Piece-wise Linear | $\phi(z) = \begin{cases} 0 & z \le -\frac{1}{2} \\ z + \frac{1}{2} & -\frac{1}{2} \le z \le \frac{1}{2} \\ 1 & z \ge \frac{1}{2} \end{cases}$ | Support vector machine |  |
| Logistic (sigmoid) | $\phi(z) = \frac{1}{1 + e^{-z}}$ | Logistic regression, Multilayer NN |  |
| Hyperbolic Tangent (tanh) | $\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$ | Multilayer NN, RNNs |  |
| ReLU | $\phi(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$ | Multilayer NN, CNNs |  |

## Types of Artificial Neural Network:

Based on the connection pattern

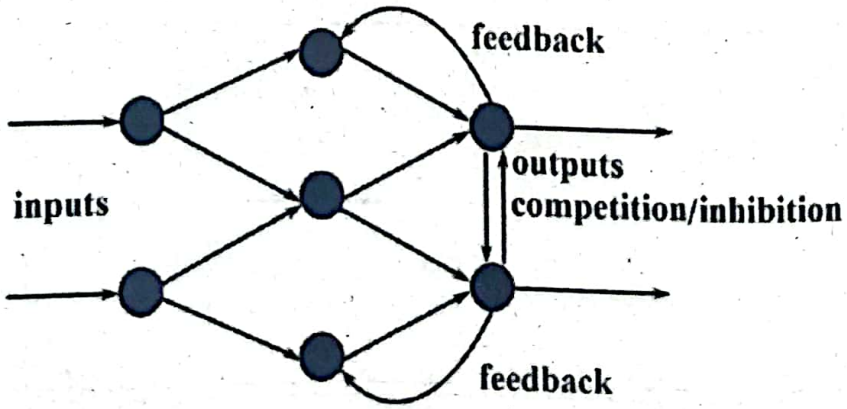
**i) Feed Forward Neural Network**

Signals travel in one way i.e. from input to output only in Feed forward Neural Network. There is no feedback or loops. The output of any layer does not affect that same layer in such networks. Feed forward neural networks are straight forward networks that associate inputs with outputs. They have fixed inputs and outputs. They are mostly used in pattern generation, pattern recognition and classification.




**ii) Feed Back Neural Network**

Signals can travel in both the directions in Feedback neural networks. Feedback neural networks are very powerful and can get very complicated. Feedback neural networks are dynamic. The 'state' in such network keep changing until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback neural network architecture is also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations. Feedback loops are allowed in such networks. They are used in content addressable memories.



Based on the number of hidden layers

(i) **Single Layer** - Having one secret layer. E.g., Single Perceptron

(ii) **Multilayer** - Having multiple secret layers. Multilayer Perceptron

**Models of Artificial Neural Network**

1. **MCCULLOCH-PITTS MODEL**

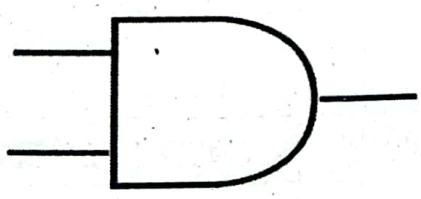
Warren Mcculloch and Walter Pitts have proposed a model in the year 1943 and this model has been considered as the Beginning of the Modern era of ANNs. This model forms a logical calculus of ANN. This neuron is a binary activated neuron in which it allows only the binary states 0 and 1. Here the neuron is associated with a Threshold value and the activation of the neuron depends on the Threshold value. The neuron will activate only when the net input to the neuron is greater than the Threshold value.

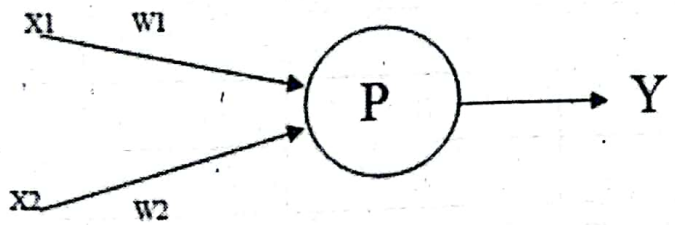
Excitatory connections have positive weights and inhibitory connections have negative weights. There will be same weight for the excitatory connections entering into a particular neuron. The neuron is associated with the Threshold value. The neuron fires if the net input to the neuron is greater than the Threshold. The threshold is set, so that the inhibition is absolute, because the non-zero inhibitory output will prevent the neuron from firing.

The Mc Culloch – Pitts Neuron has an activation function :

$$f(Y) = 1, \text{ if } Y_{in} >= \Theta$$
$$= 0, \text{ if } Y_{in} < \Theta.$$

**AND GATE IMPLEMENTATION:**





**Truth table:**

| INPUT X1 | INPUT X2 | OUTPUT Y = X1 * X2 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Net input P = $\sum$ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

Assuming the weights are excitatory,

W1 = 1 and W2 = 1 $\therefore$ P= (X1*1) +(X2*1) = X1+X2

Here, the Threshold $\Theta$ = 2

Y(P) = 1, if P>= $\Theta$

= 0, if P<$\Theta$.

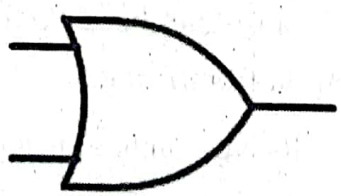
Case 1:X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0 <2, =>Y= 0

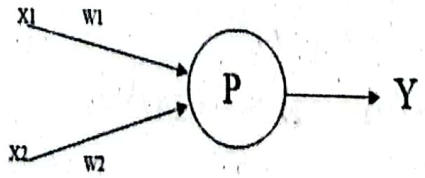
Case 2:X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1<2, => Y= 0

Case 3:X1 = 1, X2 = 0, then P = (1*1)+(0*1) = 1<2,=> Y= 0

Case 4:X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2 >=2, => Y= 1

**OR GATE IMPLEMENTATION**






**Truth table:**

| INPUT X1 | INPUT X2 | OUTPUT Y = X1 + X2 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Net input P = $\sum$ (Inputs * Weights) = [(X1*W1) +(X2*W2)]

Assuming the weights are excitatory,

W1 = 1 and W2 = 1 ∴ P= (X1*1) +(X2*1) = X1+X2

Here the Threshold Θ = 1

Y(P) = 1, if P >= Θ

= 0, if P<Θ.

Case 1:X1 = 0, X2 = 0, then P = (0*1)+(0*1) = 0<1, => Y= 0

Case 2:X1 = 0, X2 = 1, then P = (0*1)+(1*1) = 1>=1, => Y= 1

Case 3:X1 = 1, X2 = 0, then P = (1*1)+(0*1) = 1 >=1, => Y= 1

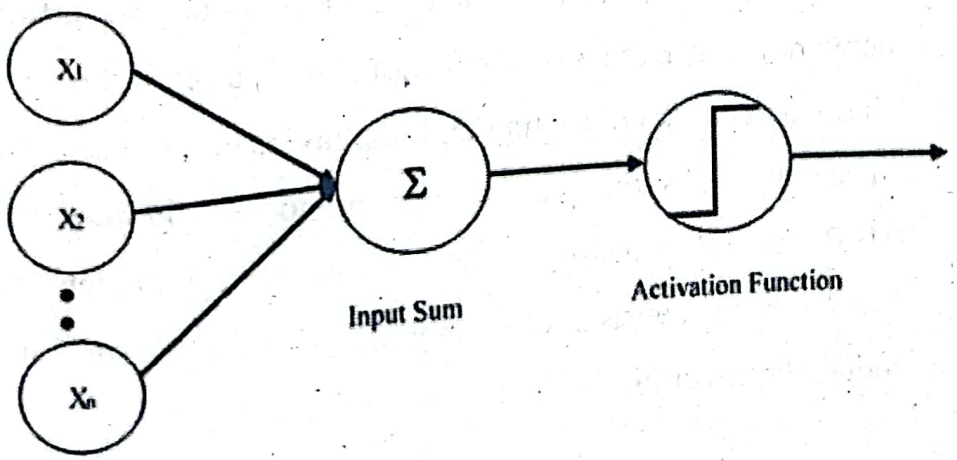
Case 4:X1 = 1, X2 = 1, then P = (1*1)+(1*1) = 2 >=1, => Y= 1

2. **Perceptron**

A Perceptron is a neural network unit that does certain computations to detect features or business intelligence in the input data. It is a function that maps its input "x," which is multiplied by the learned weight coefficient, and generates an output value "f(x).

Perceptron Learning Rule states that the algorithm would automatically learn the optimal weight coefficients. Single layer Perceptrons can learn only linearly separable patterns. Multilayer Perceptron or feedforward neural network with two or more layers have the greater processing power and can process non-linear patterns as well. Perceptrons can implement Logic Gates like AND, OR, or XOR.

Perceptron is a simple Neural Network which contains only one layer. The single layer computation of perceptron is the calculation of sum of input vector with the value multiplied by corresponding vector weight. The displayed output value will be the input of an activation function.

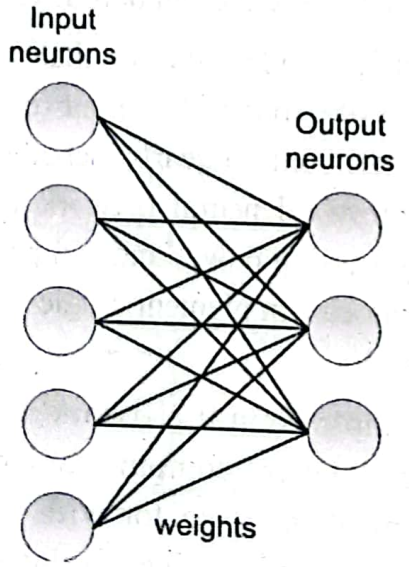


**The perceptron consists of 4 parts.**

1. Input values or One input layer
2. Weights and Bias
3. Net sum
4. Activation Function

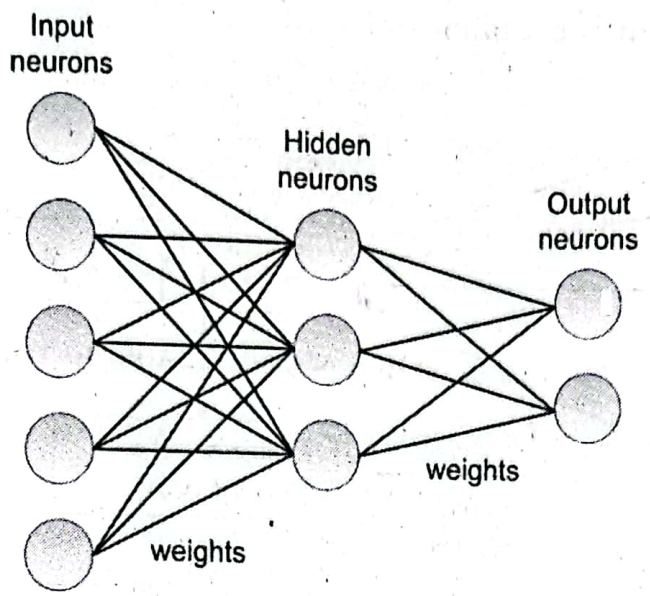
**Single Layer Perceptron**

Single Layer Perceptron has just two layers of input and output. It only has single layer hence the name single layer perceptron. It does not contain Hidden Layers as that of Multilayer perceptron. Input nodes are connected fully to a node or multiple nodes in the next layer. A node in the next layer takes a weighted sum of all its inputs.




**Multi-Layer Perceprton**

A multilayer perceptron is a type of feed-forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is a neural network connecting multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. The MLP network consists of input, output, and hidden layers. Each hidden layer consists of numerous perceptron's which are called hidden layers or hidden unit.




3. **Adaline Network**

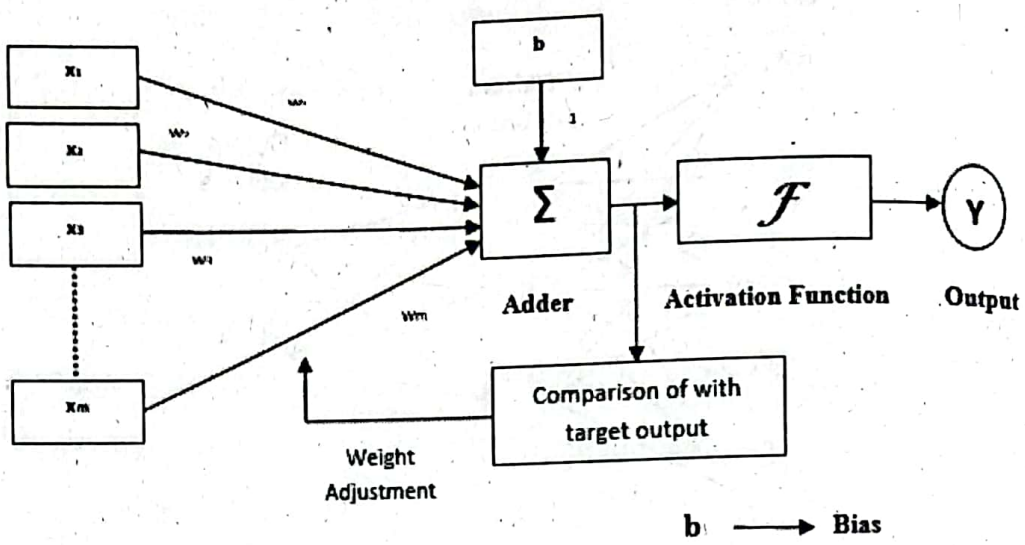
- Stands for Adaptive Linear Element
- It is a simple perceptron-like system that accomplishes classification by modifying weights in such a way as to diminish the MSE (MEAN SQUARE ERROR) at every iteration. This can be accomplished using gradient Adaptive linear elements (Adaline).
- Used in neural nets for
  - Adaptive filtering
  - Pattern recognition

Trained using the Delta Rule which is also known as the least mean squares (LMS) or Widrow-Hoff rule.

The activation function, during training is the identity function.

After training the activation is a threshold function.

**Architecture of Adaline Network.**




**Adaline Algorithm**

**Step 0:** initialize the weights to small random values and select a learning rate, α

**Step 1:** for each input vector s, with target output, t set the inputs to s

**Step 2:** compute the neuron inputs $y_in = b + x_i w_i$

**Step 3:** use the delta rule to update the bias and weights

$$b(new) = b(old) + \alpha(t - y_in)$$

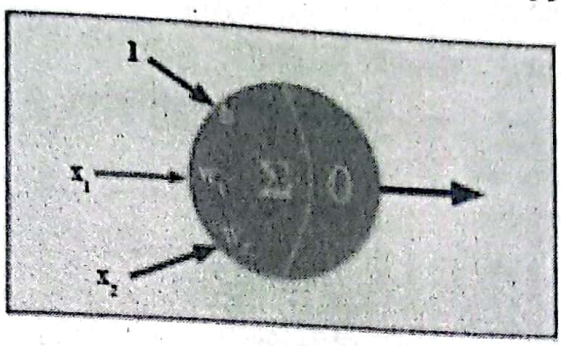
$$wi(new) = wi(old) + \alpha(t - y_in)xi$$

**Step 4:** stop if the largest weight change across all the training samples is less than a specified tolerance, otherwise cycle through the training set again

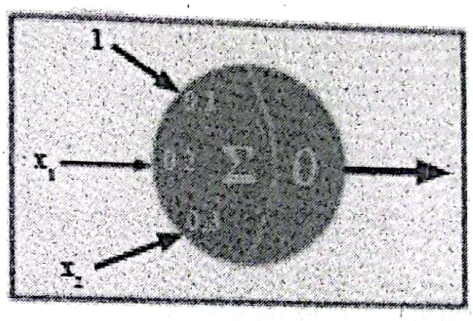
e.g. AND function using Adaline network

Construct an AND function for a ADALINE neuron let α = 0.1

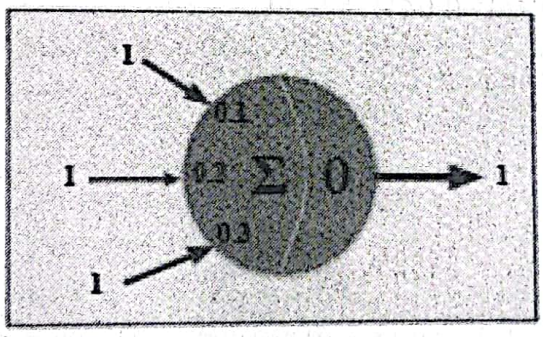
| x1 | x2 | bias | Target |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | – 1 | 1 | – 1 |
| – 1 | 1 | 1 | – 1 |
| – 1 | – 1 | 1 | 1 |




**Initial Conditions: Set the weights to small random values**




**First Training Run**




Apply the input (1, 1) with output 1

The net input is:

y_in = 0.1 + 0.2*1 + 0.3*1 = 0.6

The new weights are:

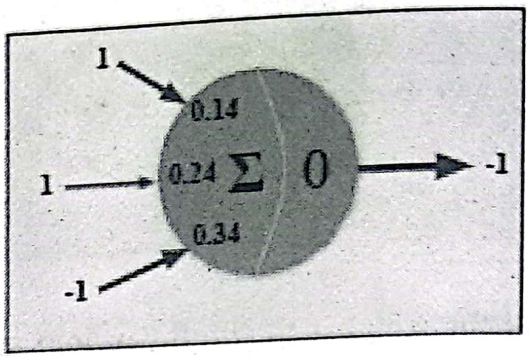
b = 0.1 + 0.1(1-0.6) = 0.14

w1 = 0.2 + 0.1(1-0.6)1 = 0.24

w2 = 0.3 + 0.1(1-0.6)1 = 0.34

The largest weight change is 0.04

**Second Training Run**




Apply the second training set (1 -1) with output -1

The net input is:

$$y_in = 0.14 + 0.24*1 + 0.34*(-1) = 0.04$$

The new weights are:

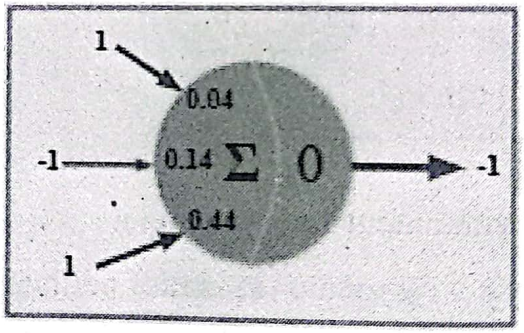
$$b = 0.14 - 0.1(1+0.04) = 0.04$$

$$w1 = 0.24 - 0.1(1+0.04)1 = 0.14$$

$$w2 = 0.34 + 0.1(1+0.04)1 = 0.44$$

The largest weight change is 0.1

Third Training Run




Apply the third training set (-1 1) with output -1

The net input

$$y_in = 0.04 - 0.14*1 + 0.44*1 = 0.34$$

The new weights are:

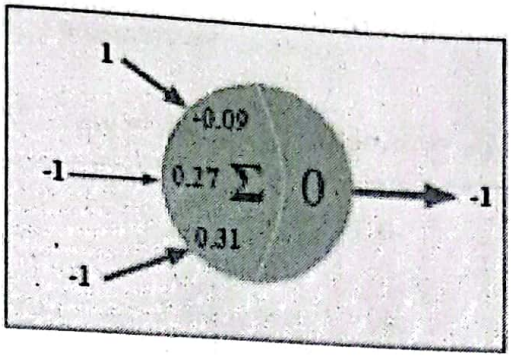
$$b = 0.04 - 0.1(1+0.34) = -0.09$$

$$w1 = 0.14 + 0.1(1+0.34)1 = 0.27$$

$$w2 = 0.44 - 0.1(1+0.34)1 = 0.31$$

The largest weight change is 0.13

**Fourth Training Run**




Apply the fourth training set (-1 -1) with output -1

The net input is:

$$y_in = -0.09 - 0.27*1 - 0.31*1 = -0.67$$

The new weights are:

$$b = -0.09 - 0.1(1+0.67) = -0.27$$

$$w1 = 0.27 + 0.1(1+0.67)1 = 0.43$$

$$w2 = 0.31 + 0.1(1+0.67)1 = 0.47$$

The largest weight change is 0.16

Continue to cycle through the four training inputs until the largest change in the weights over a complete cycle is less than some small number (say 0.01)

In this case, the solution becomes: b = -0.5, w1 = 0.5, w2 = 0.5

## 4. Hopefield Network

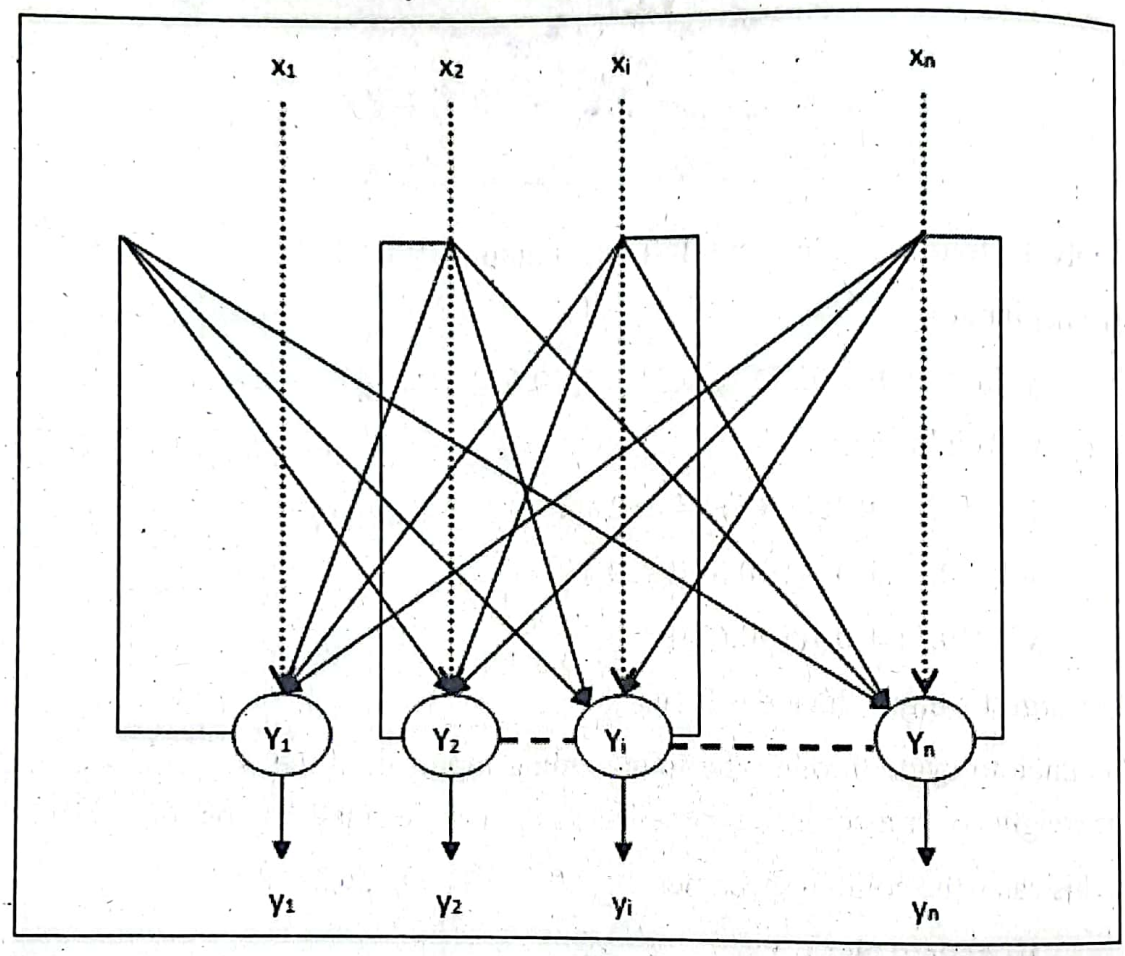
A Hopfield network which operates in a discrete line fashion or in other words, it can be said the input and output patterns are discrete vector, which can be either binary 0,10,1 or bipolar +1,−1+1,−1 in nature. The network has symmetrical weights with no self-connections i.e., $\mathbf{w_{ij} = w_{ji}}$ and $\mathbf{w_{ii} = 0}$.

### Architecture

Following are some important points to keep in mind about discrete Hopfield network –

- This model consists of neurons with one inverting and one non-inverting output.
- The output of each neuron should be the input of other neurons but not the input of self.
- Weight/connection strength is represented by $\mathbf{w_{ij}}$.

- Connections can be excitatory as well as inhibitory. It would be excitatory, if the output of the neuron is same as the input, otherwise inhibitory.
- Weights should be symmetrical, i.e. $w_{ij} = w_{ji}$




The output from $Y_1$ going to $Y_2$, $Y_i$ and $Y_n$ have the weights $w_{12}$, $w_{1i}$ and $w_{1n}$ respectively. Similarly, other arcs have the weights on them.

**Training Algorithm**

During training of discrete Hopfield network, weights will be updated. As we know that we can have the binary input vectors as well as bipolar input vectors. Hence, in both the cases, weight updates can be done with the following relation

**Case 1:** Binary input patterns

For a set of binary patterns s p, p = 1 to P

Here, $s\,p = s_1\,p, s_2\,p, ..., s_i\,p, ..., s_n\,p$

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P} [2s_i(p) - 1]\,[2s_j(p) - 1] \text{ for } i \neq j$$

**Case 2:** Bipolar input patterns

For a set of binary patterns s p, p = 1 to P

Here, $s\,p = s_1\,p, s_2\,p, \ldots, s_i\,p, \ldots, s_n\,p$

Weight matrix is given by

$$w_{ij} \sum_{p=1}^{P} [s_i(p)]\,[s_j(p)] \text{ for } i \neq j$$

**Testing Algorithm**

**Step 1:** Initialize the weights, which are obtained from training algorithm by using Hebbian principle.

**Step 2:** Perform steps 3-9, if the activations of the network is not consolidated.

**Step 3:** For each input vector X, perform steps 4-8.

**Step 4:** Make initial activation of the network equal to the external input vector X as follows:

$y_i = x_i$ for i = 1 to n

**Step 5** For each unit $Y_i$, perform steps 6-9.

**Step 6:** Calculate the net input of the network as follows:

$$y_{ini} = x_i + \sum_j y_j w_{ji}$$

**Step 7:** Apply the activation as follows over the net input to calculate to output:

$$y_i = \begin{cases} 1 & \text{if } y_{ini} > \theta_i \\ y_i & \text{if } y_{ini} = \theta_i \\ O & \text{if } y_{ini} < \theta_i \end{cases}$$

Here, $\theta_i$ is the threshold.

**Step 8:** Broadcast this output $y_i$ to all other units.

**Step 9:** Test the network for conjunction.

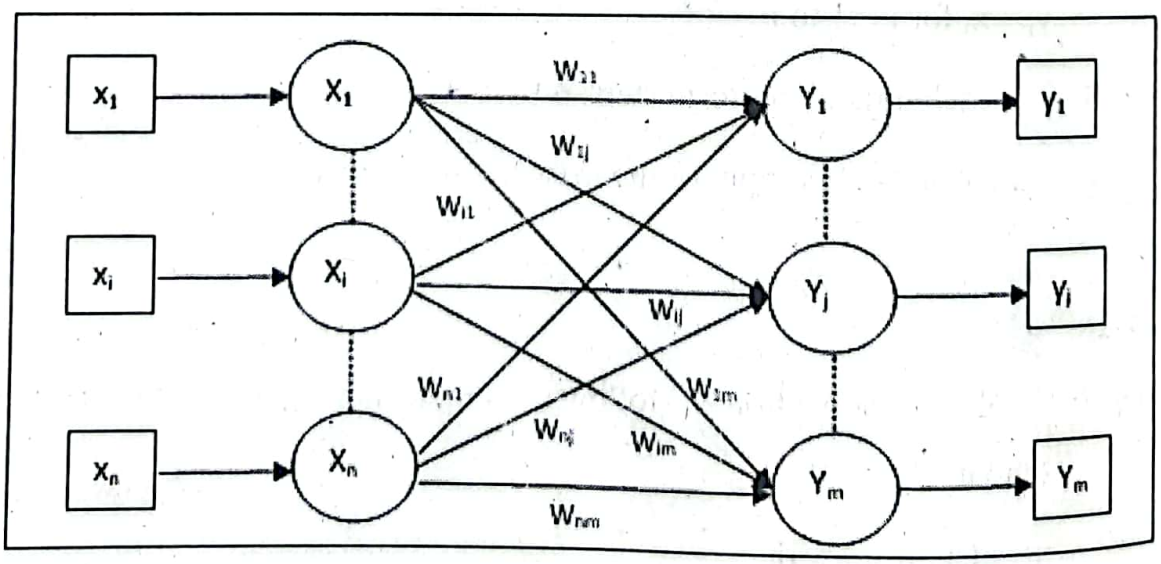
## 5. Kohohen Network

Kohonen Self-Organizing feature map (SOM) refers to a neural network, which is trained using competitive learning. Basic competitive learning implies that the competition process takes place before the cycle of learning. The competition process suggests that some criteria select a winning processing element. After the winning processing element is selected, its weight vector is adjusted according to the used learning law (Hecht Nielsen 1990).

The self-organizing map makes topologically ordered mappings between input data and processing elements of the map. Topological ordered implies that if two inputs are of similar characteristics, the most active processing elements answering to inputs that are located closed to each other on the map. The weight vectors of the processing elements are organized in ascending to descending order. $W_i < W_{i+1}$ for all values of i or $W_{i+1}$ for all values of i (this definition is valid for one-dimensional self-organizing map only).

### Architecture

The architecture of KSOM is similar to that of the competitive network. With the help of neighborhood schemes, discussed earlier, the training can take place over the extended region of the network.




### Algorithm for training

**Step 1:** Initialize the weights, the learning rate α and the neighborhood topological scheme.

**Step 2:** Continue step 3-9, when the stopping condition is not true.

**Step 3:** Continue step 4-6 for every input vector x.

**Step 4:** Calculate square of Euclidean Distance for j = 1 to m

$$D(j) = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_i - w_{ij})^2$$

**Step 5:** Obtain the winning unit J where D j is minimum.

**Step 6:** Calculate the new weight of the winning unit by the following relation:

$$w_{ij}(new) = w_{ij}(old) + a[x_i - w_{ij}(old)]$$

**Step 7:** Update the learning rate $\alpha$ by the following relation:

$$a(t + 1) = 0.5\alpha t$$

**Step 8:** Reduce the radius of topological scheme.

**Step 9:** Check for the stopping condition for the network.

**Learning Rules in Neural Network**

Learning rule or Learning process is a method or a mathematical logic. It improves the Artificial Neural Network's performance and applies this rule over the network. Thus learning rules updates the weights and bias levels of a network when a network simulates in a specific data environment. Applying learning rule is an iterative process. It helps a neural network to learn from the existing conditions and improve its performance. Let us see different learning rules in the Neural network:

- Hebbian learning rule – It identifies, how to modify the weights of nodes of a network.
- Perceptron learning rule – Network starts its learning by assigning a random value to each weight.
- Delta learning rule – Modification in sympatric weight of a node is equal to the multiplication of error and the input.
- Back Propagation learning rule- Desired outputs are compared to achieved system outputs, and then the systems are tuned by adjusting connection weights to narrow the difference between the two as much as possible.

## 1. Hebbian learning rule

The neuropsychologist Donald Hebb postulated in 1949 how biological neurons learn: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place on one or both cells such that A's efficiency as one of the cells firing B, is increased."

In other words, when a weight contributes to firing a neuron, the weight is increased.(If the neuron doesn't fire then it is not.)

**Algorithm:**

**Step 0:** initialize all weights to 0

**Step 1:** Given a training input, s, with its target output, t, set the activations of the input units: $x_i = s_i$

**Step 2:** Set the activation of the output unit to the target value: $y = t$

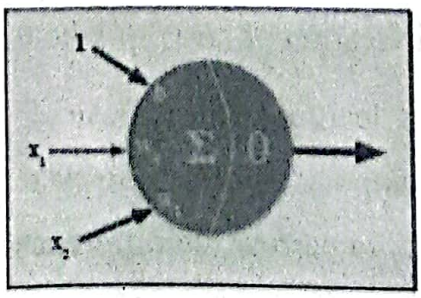
**Step 3:** Adjust the weights: $w_i(new) = w_i(old) + x_i y$

**Step 4:** Adjust the bias (just like the weights): $b(new) = b(old) + y$

E.g. Construct a Hebb Net which performs like an AND function, that is, only when both features are "active" will the data be in the target class.

**TRAINING SET (with the bias input always at 1):**

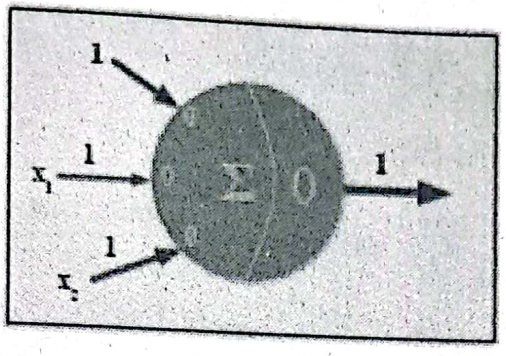
| x1 | x2 | bias | Target |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | – 1 | 1 | – 1 |
| – 1 | 1 | 1 | – 1 |
| – 1 | – 1 | 1 | – 1 |



**Training – First Input**

Initialize the weights to 0

**Present the first input (1 1 1) with a target of 1**

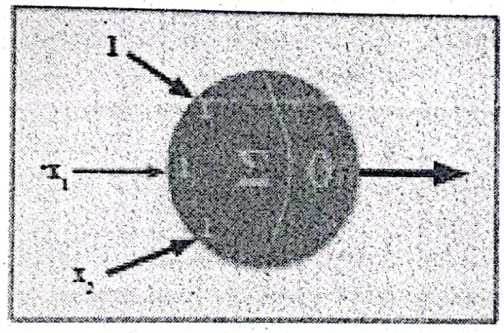



**Update the weights:**

w1(new) = w1(old) + x1t = 0 + 1 = 1

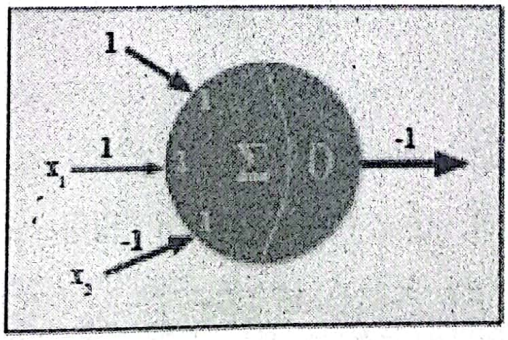
w2(new) = w2(old) + x2t = 0 + 1 = 1

b(new) = b(old) + t = 0 + 1 = 1




**Training – Second Input**

Present the second input: (1 -1 1) with a target of -1

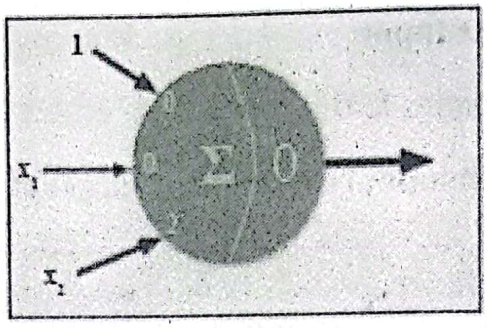



**Update the weights:**

w1(new) = w1(old) + x1t = 1 + 1(-1) = 0

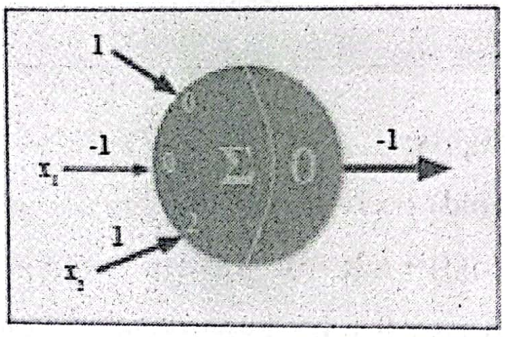
w2(new) = w2(old) + x2t = 1 + (-1)(-1) = 2

b(new) = b(old) + t = 1 + (-1) = 0




**Training – Third Input**

Present the third input: (-1 1 1) with a target of -1

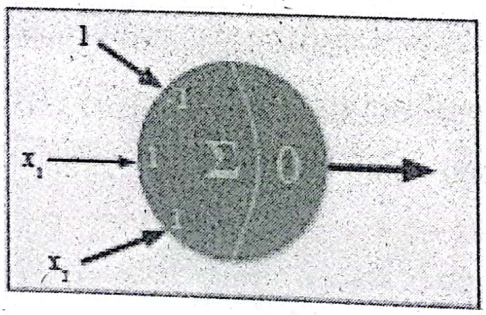



**Update the weights:**

w1(new) = w1(old) + x1t = 0 + (-1)(-1) = 1

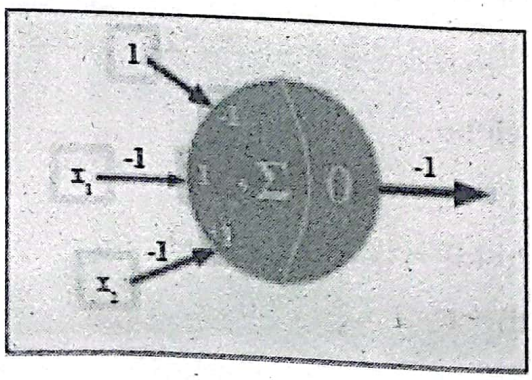
w2(new) = w2(old) + x2t = 2 + 1(-1) = 1

b(new) = b(old) + t = 0 + (-1) = -1




**Training – Fourth Input**

Present the fourth input: (-1 -1 1) with a target of -1

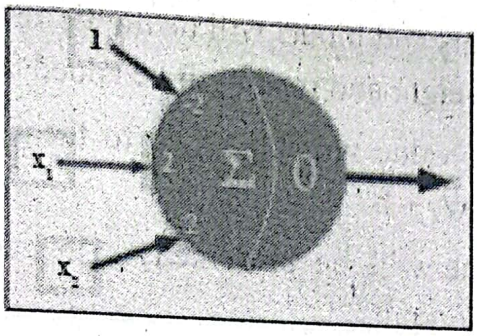


**Update the weights:**

w1(new) = w1(old) + x1t = 1 + (-1)(-1) = 2

w2(new) = w2(old) + x2t = 1 + (-1)(-1) = 1

b(new) = b(old) + t = -1 + (-1) = -2




**Final Neuron**

**This neuron works:**

| x1 | x2 | bias | Target |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | – 1 | 1 | – 1 |
| – 1 | 1 | 1 | – 1 |
| – 1 | – 1 | 1 | – 1 |

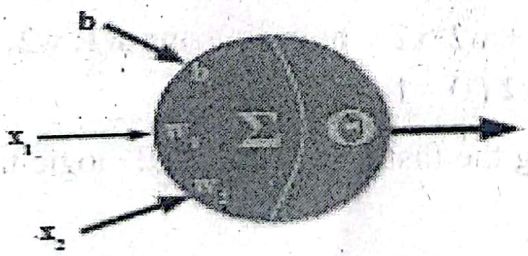
1*2 + 1*2 + 1*(-2) = 2 > 0 = 1

1*2 + (-1)*2 + 1*(-2) = -2 < 0= -1

(-1)*2 + 1*2 + 1*(-2) = -2 < 0 = -1

(-1)*2 + (-1)*2 + 1*(-2) = -6 < 0 = -1

2. **Perception learning algorithm**




In the above example, the perceptron has three inputs x1, x2, and bias b and one output.

Activation function is given by:

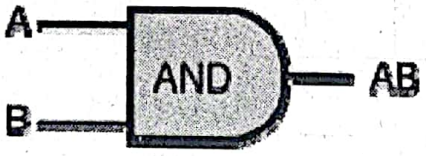
$$f(S) = \begin{cases} 1 & \text{if } s > \theta \\ 0 & \text{if } -\theta \Leftarrow s \Leftarrow \theta \\ -1 & \text{if } s < -\theta \end{cases}$$

1. Assume small weights w1,w2 and b.
2. All weights and inputs will be multiplied – the multiplied result of each weight and input will be added up. i.e. $x_1w_1 + x_2w_2$
3. The Bias value will be added to the summed input. i.e. $\sum = x_1w_1 + x_2w_2 + b$
4. This value will be presented to the activation function and calculate output.
5. Calculate error. (e) = Target output-Actual Output
6. Weights are changed only when an error occurs. The weights are updated using the following:

   $$w_i(new) = w_i(old) + \alpha t x_i$$

   Where t is either + 1 or -1, α is the learning rate.
7. Repeat from step 2

**AND Gate Using Perception**




| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Row 1**

**From w1*x1 + w2*x2 + b, initializing w1, w2, as 1 and b as – 1, we get; x1 (1) + x2 (1) – 1**

- Passing the first row of the AND logic table (x1=0, x2=0), we get;

  0+0–1 = –1

- From the Perceptron rule, if Wx+b≤0, then y`=0. Therefore, this row is correct, so no need to update weight.

**Row 2**

- Passing (x1=0 and x2=1), we get;

  $0 + 1 - 1 = 0$

- From the Perceptron rule, if $Wx + b \leq 0$, then $y = 0$. This row is correct, as the output is 0 for the AND gate.
- From the Perceptron rule, this works (for both row 1, row 2 and 3).

**Row 4**

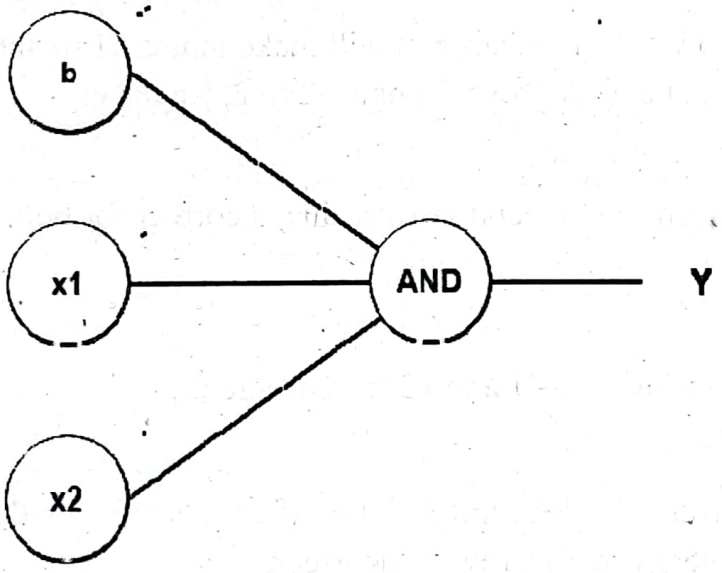
- Passing (x1=1 and x2=1), we get;

  $1 + 1 - 1 = 1$

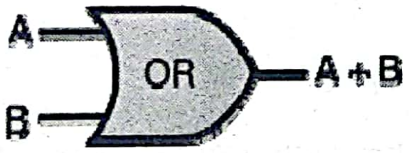
- Again, from the perceptron rule, this is still valid.

Therefore, we can conclude that the model to achieve an AND gate, using the Perceptron algorithm is;

$x1 + x2 - 1$




**OR Gate Using Perception**




| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

From the diagram, the OR gate is 0 only if both inputs are 0.

**Row 1**

- From w1x1+w2x2+b, initializing w1, w2, as 1 and b as –1, we get;

  x1(1) + x2 (1) – 1
- Passing the first row of the OR logic table (x1=0, x2=0), we get;

  0 + 0 – 1 = – 1
- From the Perceptron rule, if Wx+b≤0, then y`=0. Therefore, this row is correct.

**Row 2**

- Passing (x1 = 0 and x2 = 1), we get;

  0 + 1 – 1 = 0
- From the Perceptron rule, if Wx + b < = 0, then y = 0. Therefore, this row is incorrect.
- So we want values that will make inputs x1=0 and x2=1 give y` a value of 1. If we change w2 to 2, we have;

  0 + 2 – 1 = 1
- From the Perceptron rule, this is correct for both the row 1 and 2.

**Row 3**

- Passing (x1 = 1 and x2 = 0), we get;

  1 + 0 – 1 = 0
- From the Perceptron rule, if Wx + b < = 0, then y = 0. Therefore, this row is incorrect.
- Since it is similar to that of row 2, we can just change w1 to 2, we have;

  2 + 0 – 1 = 1
- From the Perceptron rule, this is correct for both the row 1, 2 and 3.

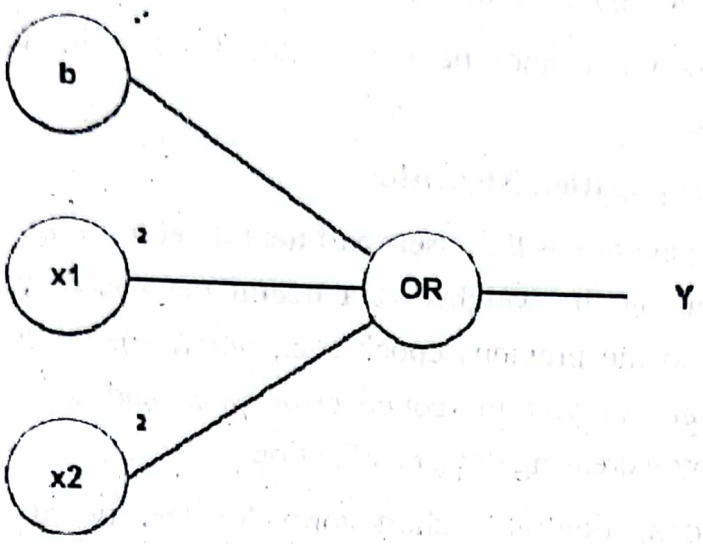
**Row 4**

- Passing (x1=1 and x2=1), we get;

  2 + 2 – 1 = 3
- Again, from the perceptron rule, this is still valid.

Therefore, we can conclude that the model to achieve an OR gate, using the Perceptron algorithm is;

2x 1 + 2x 2 – 1

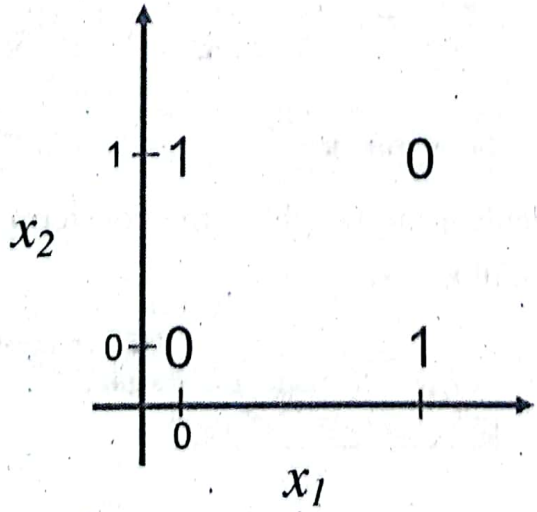



**Ex-OR gate using Perception**

Truth table for ex - or gate

| $X_1$ | $X_2$ | Y |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

The output will be high (1) for only odd number of 1's input.



From the graph, we know that XOR gate is non-linearly separable. So that XOR function cannot be represented by simple and single logic function.

If function is not linearly separable, A single layer network is not sufficient to represent the function.

XOR function cannot be represented by simple and single logic functions.

## 3. Back –Propagation Algorithm

Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Back propagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

**Algorithm**

**Step 0:** Initialize the weights to small random values

**Step 1:** Feed the training sample through the network and determine the final output

**Step 2:** Compute the error for each output unit, for unit k it is:

Actual output

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Required output

Derivative of f

**Step 3:** Calculate the weight correction term for each output unit, for unit k it is:

$$\Delta w_{jk} = \alpha \delta_k z_j$$

Hidden layer signal

A small constant

**Step 4:** Propagate the delta terms (errors) back through the weights of the hidden units where the delta input for the $j^{th}$ hidden unit is:

$$\boxed{\delta'_in_j = \sum_{k=1}^{m} \delta_k w_{jk}}$$

The delta term for the $j^{th}$ hidden unit is:

$$\boxed{\delta_j = \delta_in_j f'(z_in_j)}$$

**Step 5:** Calculate the weight correction term for the hidden units:

$$\boxed{\Delta w_{ij} = \alpha \delta_j x_i}$$

**Step 6:** Update the weights:

$$\boxed{w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}}$$

**Step 7:** Test for stopping (maximum cycles, small changes, etc)

**Note:** There are a number of options in the design of a backprop system:

- Initial weights – best to set the initial weights (and all other free parameters) to random numbers inside a small range of values (say – 0.5 to 0.5)
- Number of cycles – tend to be quite large for backprop systems
- Number of neurons in the hidden layer – as few as possible

## 7.2 Natural Language Processing

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be

- Speech
- Written Text

**Components of NLP**

There are two components of NLP as given –

1. **Natural Language Understanding (NLU)**

   Understanding involves the following tasks –

   - Mapping the given input in natural language into useful representations.
   - Analyzing different aspects of the language.

2. **Natural Language Generation (NLG)**

   It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

   It involves –

   - **Text planning** – It includes retrieving the relevant content from knowledge base.
   - **Sentence planning** – It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
   - **Text Realization** – It is mapping sentence plan into sentence structure.

   The NLU is harder than NLG.

**Difficulties in NLU**

NL has an extremely rich form and structure.

It is very ambiguous. There can be different levels of ambiguity –

- **Lexical ambiguity** – It is at very primitive level such as word-level.
- For example, treating the word "board" as noun or verb?
- **Syntax Level ambiguity** – A sentence can be parsed in different ways.
- For example, "He lifted the beetle with red cap." – Did he use cap to lift the beetle or he lifted a beetle that had red cap?

- **Referential ambiguity** – Referring to something using pronouns. For example, Rima went to Gauri. She said, "I am tired." – Exactly who is tired?
- One input can mean different meanings.
- Many inputs can mean the same thing.

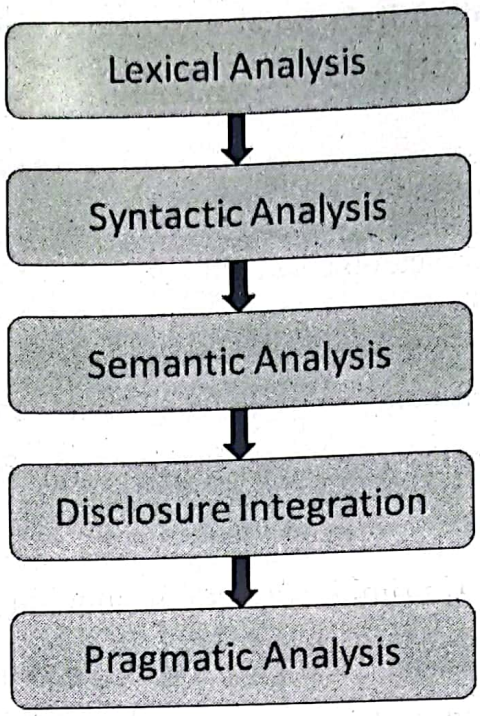
**NLP Terminology**

- **Phonology** – It is study of organizing sound systematically.
- **Morphology** – It is a study of construction of words from primitive meaningful units.
- **Morpheme** – It is primitive unit of meaning in a language.
- **Syntax** – It refers to arranging words to make a sentence. It also involves determining the structural role of words in the sentence and in phrases.
- **Semantics** – It is concerned with the meaning of words and how to combine words into meaningful phrases and sentences.
- **Pragmatics** – It deals with using and understanding sentences in different situations and how the interpretation of the sentence is affected.
- **Discourse** – It deals with how the immediately preceding sentence can affect the interpretation of the next sentence.
- **World Knowledge** – It includes the general knowledge about the world.

**Steps in NLP**

There are general five steps –

- **Lexical Analysis** – It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
- **Syntactic Analysis (Parsing)** – It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as "The school goes to boy" is rejected by English syntactic analyzer.




- **Semantic Analysis** – It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as "hot ice-cream".
- **Discourse Integration** – The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
- **Pragmatic Analysis** – During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

**Implementation Aspects of Syntactic Analysis**

There are a number of algorithms researchers have developed for syntactic analysis, but we consider only the following simple methods –

- Context-Free Grammar
- Top-Down Parser

Let us see them in detail –

**Context-Free Grammar**

It is the grammar that consists rules with a single symbol on the left-hand side of the rewrite rules. Let us create grammar to parse a sentence –

"The bird pecks the grains"

**Articles (DET)** – a | an | the

**Nouns** – bird | birds | grain | grains

**Noun Phrase (NP)** – Article + Noun | Article + Adjective + Noun

= DET N | DET ADJ N

**Verbs** – pecks | pecking | pecked

Verb Phrase (VP) – NP V | V NP

**Adjectives (ADJ)** – beautiful | small | chirping

The parse tree breaks down the sentence into structured parts so that the computer can easily understand and process it. In order for the parsing algorithm to construct this parse tree, a set of rewrite rules, which describe what tree structures are legal, need to be constructed.

These rules say that a certain symbol may be expanded in the tree by a sequence of other symbols. According to first order logic rule, if there are two strings Noun Phrase (NP) and Verb Phrase (VP), then the string combined by NP followed by VP is a sentence. The rewrite rules for the sentence are as follows –

S → NP VP

NP → DET N | DET ADJ N

VP → V NP

Lexocon –

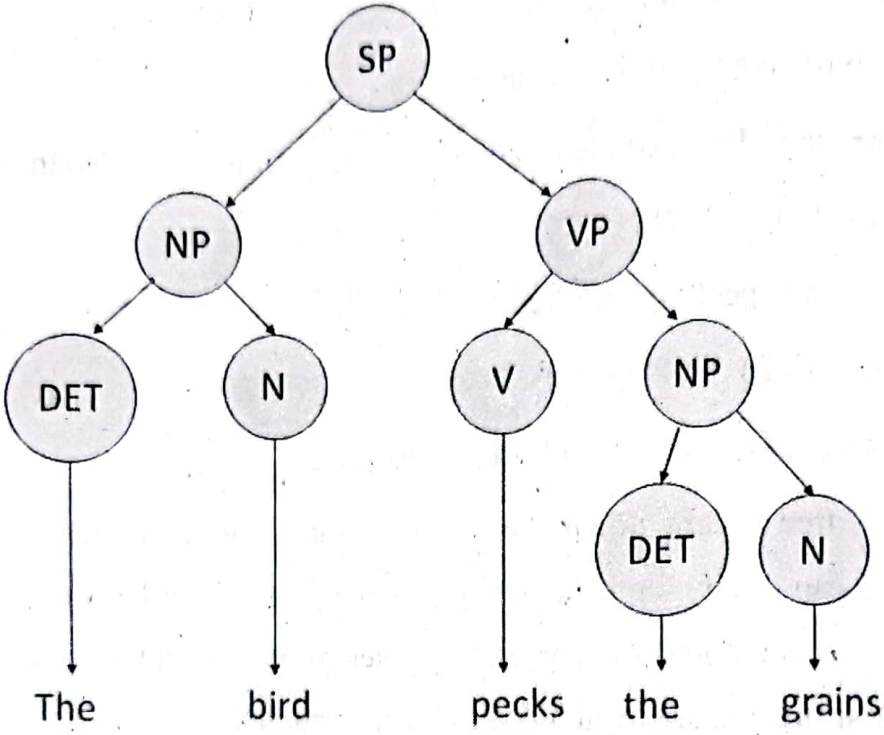
DET → a | the

ADJ → beautiful | perching

N → bird | birds | grain | grains

V → peck | pecks | pecking

The parse tree can be created as shown –




Now consider the above rewrite rules. Since V can be replaced by both, "peck" or "pecks", sentences such as "The bird peck the grains" can be wrongly permitted. i.e. the subject-verb agreement error is approved as correct.

**Merit**– The simplest style of grammar, therefore widely used one.

**Demerits –**

- They are not highly precise. For example, "The grains peck the bird", is a syntactically correct according to parser, but even if it makes no sense, parser takes it as a correct sentence.
- To bring out high precision, multiple sets of grammar need to be prepared. It may require a completely different sets of rules for parsing singular and plural variations, passive sentences, etc., which can lead to creation of huge set of rules that are unmanageable.

**Top-Down Parser**

Here, the parser starts with the S symbol and attempts to rewrite it into a sequence of *terminal symbols* that matches the classes of the words in the input sentence until it consists entirely of terminal symbols.

These are then checked with the input sentence to see if it matched. If not, the process is started over again with a different set of rules. This is repeated until a specific rule is found which describes the structure of the sentence.

**Merit** – It is simple to implement.

**Demerits** –

- It is inefficient, as the search process has to be repeated if an error occurs.
- Slow speed of working.

**Machine vision**

Machine vision is the ability of a computer to see; it employs one or more video cameras, analog-to-digital conversion (ADC) and digital signal processing (DSP). The resulting data goes to a computer or robot controller. Machine vision is similar in complexity to voice recognition.

Two important specifications in any vision system are the sensitivity and the resolution. Sensitivity is the ability of a machine to see in dim light, or to detect weak impulses at invisible wavelengths. Resolution is the extent to which a machine can differentiate between objects. In general, the better the resolution, the more confined the field of vision. Sensitivity and resolution are interdependent. All other factors held constant, increasing the sensitivity reduces the resolution, and improving the resolution reduces the sensitivity.

Human eyes are sensitive to electromagnetic wavelength s ranging from 390 to 770 nanometers (nm). Video cameras can be sensitive to a range of wavelengths much wider than this. Some machine-vision systems function at infrared (IR), ultraviolet (UV), or X-ray wavelengths.

Binocular (stereo) machine vision requires a computer with an advanced processor. In addition, high-resolution cameras, a large amount of random access memory (RAM), and artificial intelligence (AI) programming are required for depth perception.

Machine vision is used in various industrial and medical applications. **Examples include:**

- Electronic component analysis
- Signature identification
- Optical character recognition
- Handwriting recognition
- Object recognition
- Pattern recognition
- Materials inspection
- Currency inspection
- Medical image analysis

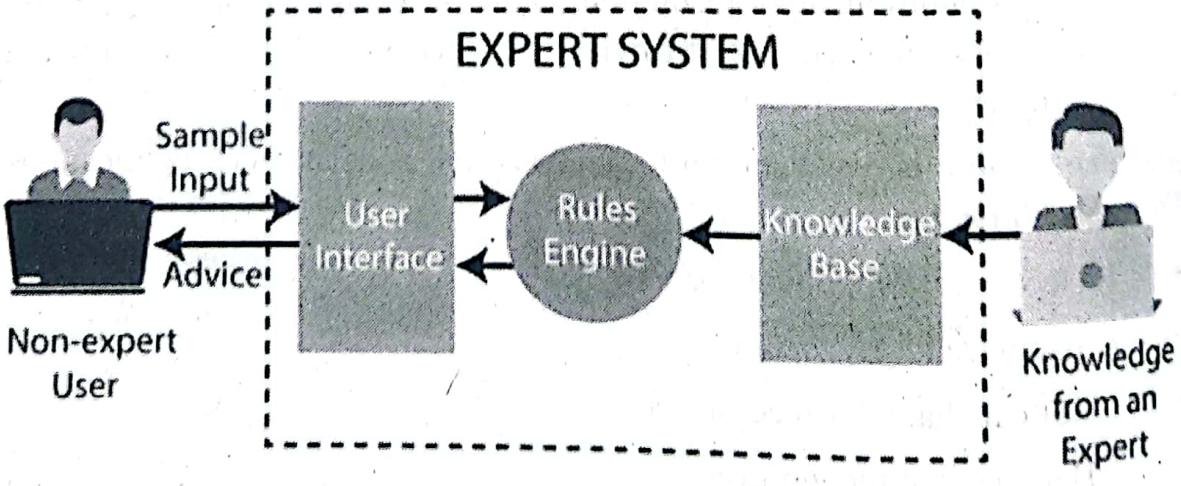
The term *machine vision* is often associated with industrial applications of a computer's ability to see, while the term *computer vision* is often used to describe any type of technology in which a computer is tasked with digitizing an image, processing the data it contains and taking some kind of action.

## 7.3 Expert System

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for compsex problems using **both facts and heuristics like a human expert**. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as **medicine, science,** etc.

Below is the block diagram that represents the working of an expert system:




**Below are some popular examples of the Expert System:**

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.

- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.
- **PXDES:** It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.
- **CaDeT:** The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

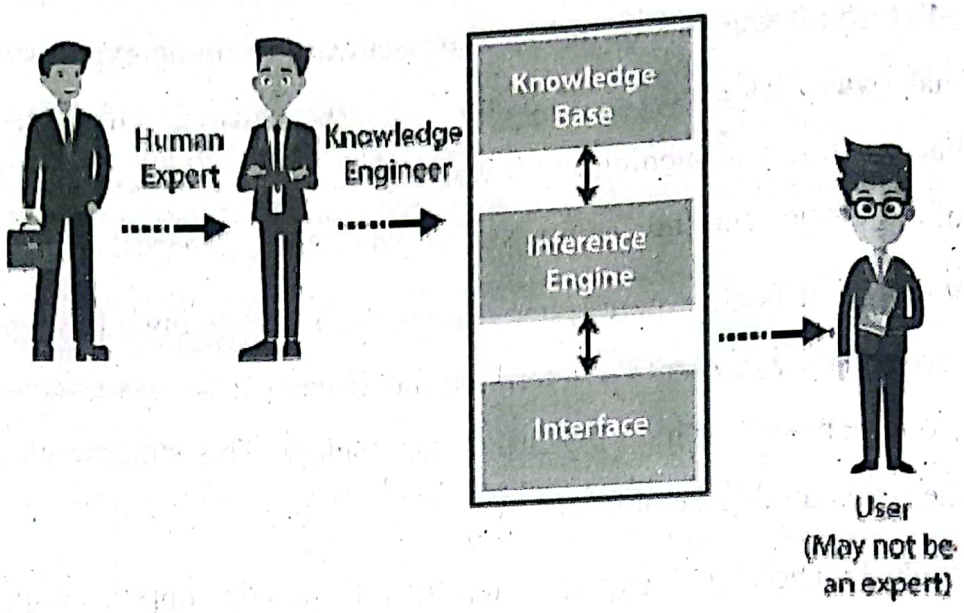
**Characteristics of Expert System**

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

**Components of Expert System**

An expert system mainly consists of three components:

- User Interface
- Inference Engine
- Knowledge Base




1. **User Interface**

With the help of a user interface, the expert system interacts with the user, takes queries as an input in a readable format, and passes it to the inference engine. After getting the response from the inference engine, it displays the output to the user. In other words, **it is an interface that helps a non-expert user to communicate with the expert system to find a solution.**

2. **Inference Engine(Rules of Engine)**

- The inference engine is known as the brain of the expert system as it is the main processing unit of the system. It applies inference rules to the knowledge base to derive a conclusion or deduce new information. It helps in deriving an error-free solution of queries asked by the user.
- With the help of an inference engine, the system extracts the knowledge from the knowledge base.
- There are two types of inference engine:
- **Deterministic Inference engine:** The conclusions drawn from this type of inference engine are assumed to be true. It is based on **facts** and **rules.**
- **Probabilistic Inference engine:** This type of inference engine contains uncertainty in conclusions, and based on the probability.

**Inference engine uses the below modes to derive the solutions:**

- **Forward Chaining:** It starts from the known facts and rules, and applies the inference rules to add their conclusion to the known facts.
- **Backward Chaining:** It is a backward reasoning method that starts from the goal and works backward to prove the known facts.

3. **Knowledge Base**

- The knowledgebase is a type of storage that stores knowledge acquired from the different experts of the particular domain. It is considered as big storage of knowledge. The more the knowledge base, the more precise will be the Expert System.
- It is similar to a database that contains information and rules of a particular domain or subject.
- One can also view the knowledge base as collections of objects and their attributes. Such as a Lion is an object and its attributes are it is a mammal, it is not a domestic animal, etc.

**Components of Knowledge Base**

- **Factual Knowledge:** The knowledge which is based on facts and accepted by knowledge engineers comes under factual knowledge.
- **Heuristic Knowledge:** This knowledge is based on practice, the ability to guess, evaluation, and experiences.

**Knowledge Representation:** It is used to formalize the knowledge stored in the knowledge base using the If-else rules.

**Knowledge Acquisitions:** It is the process of extracting, organizing, and structuring the domain knowledge, specifying the rules to acquire the knowledge from various experts, and store that knowledge into the knowledge base.

**Procedural Knowledge:** Procedural or imperative knowledge clarifies how to perform a certain task. It lays down the steps to perform. For example, - how to copy an array using javascript in a procedural way.

**Example**

```
var a=[1, 2, 3, 4, 5];
var b=[];
for(var i=0;i < a.length;i++) {
    b.push(a[i]);
}
console.log(b);
```

**Output**

```
[1, 2, 3, 4, 5]
```

**Declarative Knowledge**

Declarative or functional knowledge clarifies what to do to perform a certain task. It lays down the function to perform. For example, - how to copy an array using javascript in a declarative way.

**Example**

```
var a=[1, 2, 3, 4, 5];
var b=a.map(function(number){
    return number*1
});
console.log(b);
```

**Output**

```
[1, 2, 3, 4, 5]
```

The following are some of the important differences between Procedural Knowledge and Declarative Knowledge.

| Sr. No. | Key | Procedural Knowledge | Declarative Knowledge |
|---|---|---|---|
| 1 | Name | Procedural knowledge is also termed as imperative knowledge. | Declarative knowledge is also termed as functional knowledge |
| 2 | Basis | Procedural knowledge revolves around **How to** the concept. | Declarative knowledge revolves around **What to** the concept. |
| 3 | Communication | Procedural knowledge is difficult to communicate. | Declarative knowledge is easily communicable. |
| 4 | Orientation | Procedural knowledge is process-oriented. | Declarative knowledge is data-oriented. |
| 5 | Validation | Validation is not very easy in procedural knowledge. | Validation is quite easy in declarative knowledge. |
| 6 | Debugging | Debugging is not very easy in procedural knowledge. | Debugging is quite easy in declarative knowledge. |

**Development of Expert System**

Here, we will explain the working of an expert system by taking an example of MYCIN ES. Below are some steps to build an MYCIN:

- Firstly, ES should be fed with expert knowledge. In the case of MYCIN, human experts specialized in the medical field of bacterial infection, provide information about the causes, symptoms, and other knowledge in that domain.
- The KB of the MYCIN is updated successfully. In order to test it, the doctor provides a new problem to it. The problem is to identify the presence of the bacteria by inputting the details of a patient, including the symptoms, current condition, and medical history.

- The ES will need a questionnaire to be filled by the patient to know the general information about the patient, such as gender, age, etc.
- Now the system has collected all the information, so it will find the solution for the problem by applying if-then rules using the inference engine and using the facts stored within the KB.
- In the end, it will provide a response to the patient by using the user interface.

**Participants in the development of Expert System**

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

**OLD QUESTION SOLUTION**

1. **What is McCulloh/Pitts neuron? Can this neuron trained, to represent Ex - Or gate? If Justify and propose neural network model.** **(2076 Baisakh)**

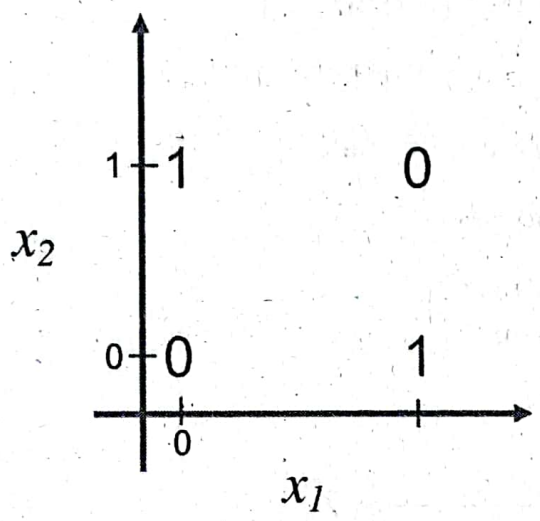
⇒ The fundamental block of deep learning is artificial neuron i.e. takes a weighted aggregate of inputs, applies a function and gives an output. The very first step towards the artificial neuron was taken by Warre McCulloch and Walter Pitts in 1943 inspired by neurobiology created a model known as McCulloh - Pitts Neuron.

**2nd part:**

Truth table for ex - or gate

| $X_1$ | $X_2$ | Y |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

The output will be high (1) for only odd number of 1's input.




From the graph, we know that XOR gate is non-linearly separable. So that XOR function cannot be represented by simple and single logic function.

If function is not linearly separable, A single layer network is not sufficient to represent the function.

XOR function cannot be represented by simple and single logic functions.

**It is represented as:**

$$Y = \overline{X1}\ X2 + X1\ \overline{X2}$$

Now, we have add the weights like

$$yin = \overline{X1}\ X2\ W1 + X1\ \overline{X2}\ W2$$

Assume, W1 = W2 = 1

| $X_1$ | $W_1$ | $X_2$ | $W_2$ | Yin | Y |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |

Yin = 0* 1*1 + 1*0*1 = 0

Yin = 0*0*1 + 1*1*1 = 1

Yin = 1*1*1 + 0*0*1 = 1

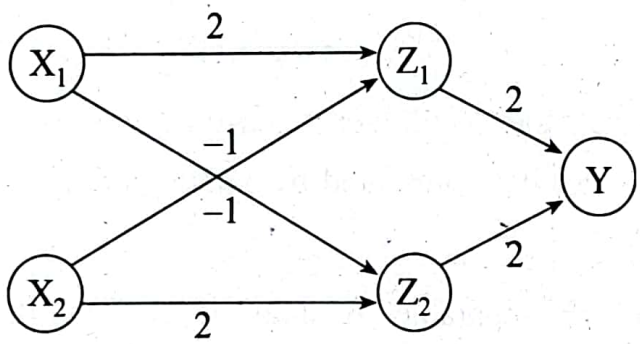
Yin = 1*0*1 + 0*1*1 = 0

Calculating Threshold value

We can simply say

$$\begin{Bmatrix} 1 \text{ if y in} \geq 1 \\ 0 \text{ if y in} \leq 1 \end{Bmatrix}$$

$[\theta \geq 1]$




**2. What is an expert system? Explain the component of an expert system. (2076 Baisakh)**

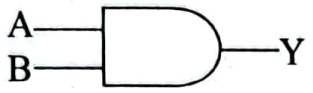
⇒ See on Page No. 162

**3. What is a McCulloh/Pitts neural network? Explain McCulloh/Pitts neural network with reference to AND gate. Justify that it cannot be applied to Exclusive OR gate. (2075 Baisakh)**

⇒ It is a logic gate that implements conjunction whenever both the inputs are high then only output will be high (1) otherwise low (0).

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

*Fig: Truth table of AND gate.*



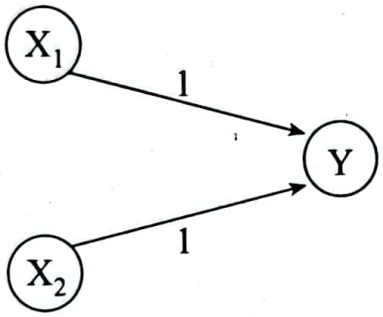




*Fig: Architecture of AND gate*

The threshold value is 2.

Net input is yin = A + B

Output is given by Y = F(yin)

Activation function = $\begin{Bmatrix} 1 \text{ if y in} \geq 2 \\ 0 \text{ if y in} < 2 \end{Bmatrix}$

**It cannot be used for ex-or gate:** See on page no

4. **List the importance of expert system in real life. Draw block diagram of expert system architecture and explain all blocks.**

**(2075 Baisakh)**

⇒ Importance or expert system in real life are as follows:

- Human experts are perishable but an expert system is permanent.
- It helps to distribute the expertise of human.
- It decreases the cost of consulting an expert for various domains such as medical diagnosis.
- They use a knowledge base and in inference engine.

**Second Part:** See on Page No163

**5. Justify that NLP is one of the important part of an AI. Explain the steps involved in the NLP.** **(2075 Bhadra)**

⇒ **First Part:** See on Page No. 156

**Second Part:** See on Page No.157

**6. What is Natural Language Processing (NLP)? Discuss the steps in NLP with suitable examples. Also list down major issues in NLP.** **(2075 Bhadra)**

⇒ **First Part:** See on Page No. 156

**Second Part:** See on Page No. 157

**Issues in NLP**

- Contextual words and phrases and homonyms
- Synonyms
- Irony and sarcasm
- Ambiguity
- Errors in text or speech
- Colloquialisms and slang
- Domain-specific language
- Low-resource languages
- Lack of research and development

**7. How knowledge acquisition is performed in expert system? Explain one real expert system example with architecture.** **(2074 Bhadra)**

⇒ Knowledge acquisition is an activity is of knowledge engineering that is very important for building the fundamental knowledge base, as well as in the application phase of the system for knowledge base updating to the domain knowledge to be initially acquired also belongs, in addition to the textbook knowledge of the domain, the related heuristic and the meta - knowledge if available.

The development phase of knowledge acquisition includes the activities that precede the process of knowledge elicitation from domain experts such as domain identification domain knowledge concept usalization, knowledge formalization and encoding and knowledge refinement and validation.

E.g. of real expert system

**Mycin**

Mycin an early expert system or artificial intelligence (AI) program, For treating blood infection. In 1972 work began on MYCIN at stand Ford university in California. MYCIN would attempt to diagnoses patients based on reported symptoms and medical test results. The program could request further information concerning the patient, as well as suggest additional laboratory tests, to arrive at a probable diagnosis, after which it would recommend a course of treatment. If requested, MYCIN would explain the reasoning that led to its diagnosis and recommendation using about 500 production rules, MYCIN operated at roughly the same level of competence as human specialists in blood infections and rather better than general practitioners.

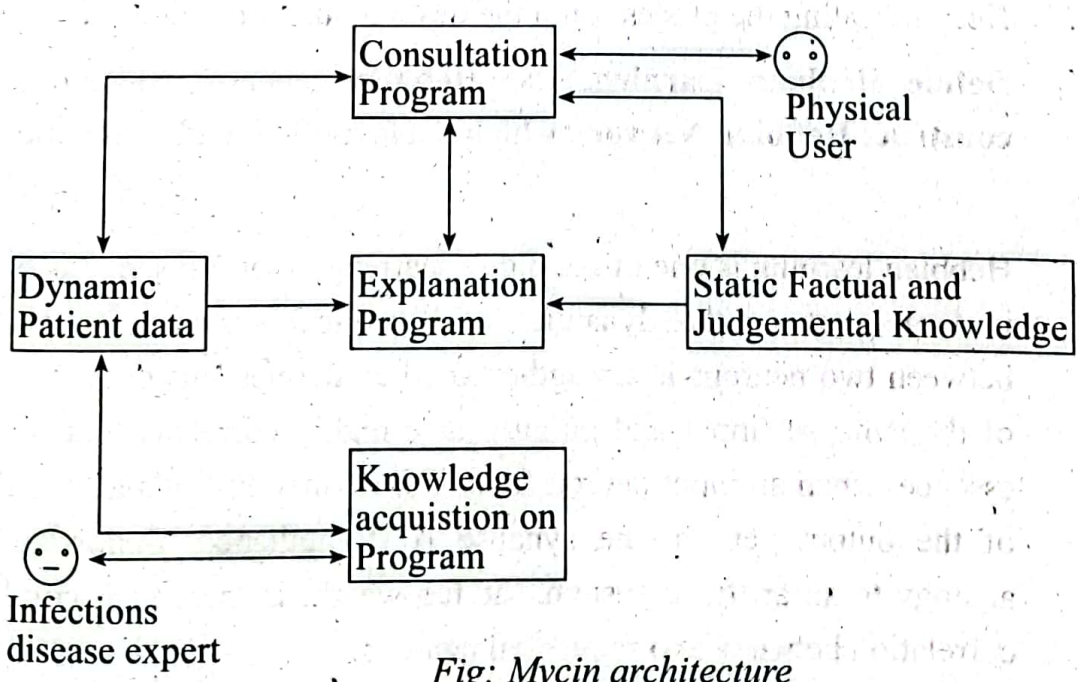



*Fig: Mycin architecture*

**8. Write short notes on:**

**Back Propagation Algorithm** **(2074 Bhadra)**

⇒ Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

How Backpropagation Works

1. Inputs X, arrive through the pre connected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs
   $Error_B$= Actual Output - Desired Output
5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

Keep repeating the process until the desired output is achieved

**9. Define Hebbian learning. Use Hebbian learning algorithm to construct Hebbian Network which perform line And function.**

**(2073 Bhadra)**

⇒ Hebbian learning is one of the oldest learning algorithms, and is based in large part on the dynamics of biological systems. A synapse between two neurons is strengthened when the neurons on either side of the synapse (input and output) have highly correlated outputs. In essence, when an input neuron fires, if it frequently leads to the firing of the output neuron, the synapse is strengthened. Following the analogy to an artificial system, the tap weight is increased with high correlation between two sequential neurons.

**Hebbian network for And function:** See on page no

**10. Define a NLV and NLG. List down the different steps involved in the natural language processing (NLP) with suitable examples.**

**(2073 Bhadra)**

⇒ **NLV and NLG:** See on Page No.156

**Second Part:** See on Page No.157

**11. Differentiate between declarative knowledge and procedural knowledge. Describe expert system with if architecture and practical uses.**

**(2073 Magh)**

⇒ **Difference:** See on Page No. 166

**Architecture of expert system:** See on Page No.163

**Practical uses:** See on Page No. See on internet

**12. What is natural languages processing? Explain the different steps in natural language processing.**

**(2073 Magh)**

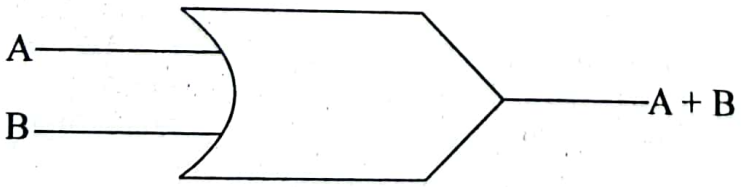
⇒ **First Part:** See on Page No.156

**Second Part:** See on Page No.157.

**13. What do you understand by perception? How can we design a neural networks that acts as an OR gate?**

**(2073 Magh)**

⇒ 1st part see on page no147




| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

From the diagram, the OR gate is 0 only if both inputs are 0.

**Row 1**

- From W1 × 1 + W2 × 2 + b, initializing W1, W2, as I and b as 1, we get;
- Passing the point row of the OR logic table (X1 = 0, X2 = 0), we get

  0 + 0 – 1 = –1
- From the perception rule, if WX + b ≤, then $y^1 = 0$. Therefore, this row is correct.

**Row 2**

- Passing (X1 = 0 and X2 = 1) we get:

  0 + 1 – 1= 0
- From the perception rule, if WX + b < = 0, then $y^1 = 0$. Therefore, this row is incorrect.
- So we want values that will make inputs X1 = 0 and X2 = 1 give $y^1$ a value of 1.

  If we change W2 to 2, we have,

  0 + 2 – 1 = 1
- From the perception rule, this is correct for bath the row 1 and 2.

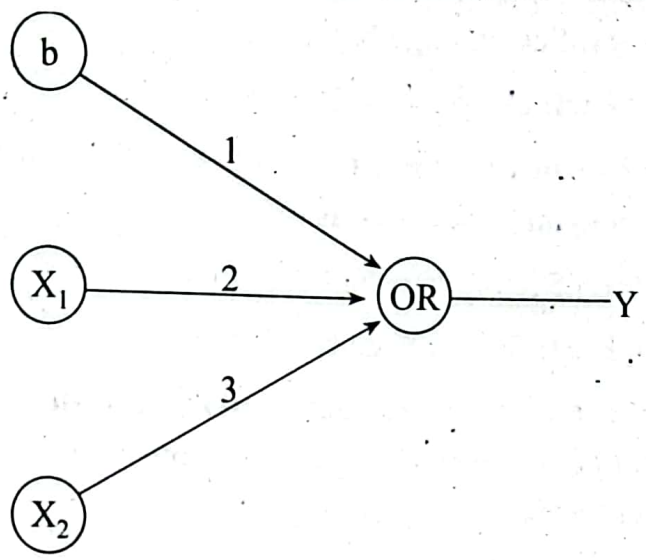
**Row 3**

- Passing (X1 = 1 and X2= 0) we get, 1 + 0 – 1 = 0
- From the perception rule, if WX + b < = 0, then $y^1 = 0$. Therefore, this row is incorrect.
- Since it is similar to that of row 2, we can just change W1 to W2, we have, 2 + 0 = 1
- From the perception rule, this is correct for both the row 1, 2 and 3.

**Row 4**

- Passing (X1 = 1 and X2 = 1) we get, 2 + 2 – 1 = 3
- Again, from the perception rule, this is valid.

Therefore, we can conclude that the model to achieve an OR gate, using the perception algorithm is, 2x1 + 2x2 = .




**14. What is neural network? Explain back propagation algorithm and perception. (2072 Ashwin)**

⇒ **Neural network:** See on Page No. 142

**Back Propagation Algorithm:** See on old question no.8

**Perception:** See on Page No.147

**15. What is an expert system? Explain its advantages and disadvantages. (2072 Ashwin)**

⇒ **First Part:** See on Page No.162

**Advantages of expert system**

1. Providing consistent solutions
2. Provides reasonable explanations.
3. Overcome human limitation
4. Easy to adopt new conditions.

**Disadvantages of using expert system**

1. Lacks common sense
2. High implementation and maintenance cost
3. Difficulty in creating in presence rules.
4. May provide wrong solutions.

**16. What is a natural language processing? Describe natural language processing steps and its application. (2072 Ashwin)**

⇒ **First Part:** See on Page No. 156

**Second Part:** See on Page No. 157.

**17. What do you understand by a perception? How can we design a neural network that acts as on OR gate. (2072 Magh)**

⇒ **First Part:** See on Page No. 175

**Second Part:** See on Page No. 175

**18. Differentiate between declarative knowledge and procedural knowledge. Describe expert system with its architecture and practical uses. (2072 Magh)**

⇒ **First Part:** See on Page No. 166

**Second Part:** See on Page No. 162

**Third Part:** See on Internet

**19. Explain different steps involved in the natural language processing (NLP) with suitable block diagram and examples. (2072 Magh)**

⇒ See on Page No. 157

**20. Compare the following:**

**Pragmatic versus phonetic analysis. (2071 Bhadra)**

⇒

| Phonetic Analysis | Pragmatic Analysis |
|---|---|
| 1. Pronunciation of different speakers. | 1. Understanding the text and English s. |
| 2. Deals with physical building blocks of language sound. | 2. It derives knowledge from external commonsense information. |
| 3. E.g: I ate cake, different k sounds in 'kite', 'coat', that band is banned. If we speak in foreign English (188 cake) similarly K (Hindi क for kite and coat) | 3. Do you know what time it is? It does not mean the speaker asking you the time!!! we should understand what to do? |

**21. What is Artificial Neural Network (ANN)? Compare ANN with human brain with its functional principle. (2071 Bhadra)**

⇒ **First Part:** See on Page No. 142

**Second Part:** See on Page No.143.

**22. Define knowledge acquisition with example. Explain the architecture of expert system. (2071 Bhadra)**

⇒ **First Part:** See on Page No. 171

**Second Part:** See on Page No. 162

**23. What is natural language processing (NLP)? Explain the syntactic, sementic and pregmatic analysis of NLP (2071 Magh)**

⇒ **First Part:** See on Page No. 156

**Second Part:** See on Page No. 157

**24. Define expert system? Discuss about the general architecture of an expert system. (2071 Magh)**

⇒ **First Part:** See on Page No. .162

**Second Part:** See on Page No. .163

**25. How does machine learning help in artificial intelligence? Explain how back propagation algorithm helps in machine learning. (2071 Magh)**

⇒ **First Part:** See in previous chapter.

**Second Part:** See on old question no 8

**26. Compare the following**

**Hopefield Network versus kohonen network (2071 Magh)**

⇒ **Hopefield Network:** See on Page No. 152

**Kohoren Network:** See on Page No. 154

**27. Differentiate declarative knowledge and procedural knowledge. Explain the architecture of expert system. (2070 Magh)**

⇒ **First Part:** See on Page No. 166

**Second Part:** See on Page No. 163

**28. What is the role of perception in neural network? Explain about back propagation algorithm. (2070 Magh)**

⇒ **First Part:** See on Page No. 147

**Second Part:** See on old question no 8.

**29. What is natural language processing (NLP)? Discuss the different issues related with NLP with example. (2070 Magh)**

⇒ **First Part:** See on Page No. 156

**Second Part:** See on old question no.6

**30. What is neural network? Explain back propagation algorithm learning. (2070 Bhadra)**

⇒ **First Part:** See on Page No. 142

**Second Part:** See on old question no 8

**31. What is an expert system? Explain the steps of an expert system development. (2070 Bhadra)**

⇒ **First Part:** See on Page No. 162

**Second Part:** See on Page No. 163.

**32. Define machine transition in NLP. Explain the challenges of machine translation. (2070 Bhadra)**

⇒ Machine transition is the task of automatically converting the natural language into another, preserving the meaning of the input text, and producing fluent text in the output language.

The challenges of machine transition are as follows:

**(i) Domain Mismatch**

As known challenge in transition is that in different domains, words have different transitions and meaning is expressed in different styles. Hence, a crucial step in developing machine translation system targeted at a specific use case in domain adaptation.

**(ii) Amount of Training Data**

A well known property of statistical system is that increasing amount of training data lead to better result.

(iii) **Rare words**

Conventional wisdom states that neural machine translation models perform statistically and particularly on rare words.

(iv) **Long sentences**

A well-known flow of early encoder-decoder NMT models was inability to properly translate long sentences.

(v) **Word Alignment**

There should be attention in neural machine translation on the imposition of an alignment of the output words to the input words.

(vi) **Beam Search**

The task of decoding is to find the full sentence translation with the highest probability. In statistical machine translation, this problem has been addressed with heuristic search techniques that explore a subset of the space or possible translation.

**33. Differentiate between supervised learning and unsupervised learning.** **(2070 Chaitra)**

⇒ See on Page No. 133

**34. Explain the difficult issues of natural language processing (NLP).** **(2070 Chaitra)**

⇒ See on old question no 6

**35. Explain different forms of knowledge modeling techniques.** **(2070 Chaitra)**

⇒ (i) **Diagnostic models**

This type of model is used for diagnosing problem by categorizing and framing problems in order to determine the root as possible cause.

Semantic: Complaint >> possible cause(s)

Example: I have these symptoms. What is the problem?

**(ii) Explorative models**

This type of model is designed to produce possible options for a specific case. This options may be generated using techniques such as Genetic Algorithm or Monte Carlo simulation or retrained from a knowledge and/or case-based system.

**Semantic:** Problem Description >> Possible alternatives.

**Example:** Ok, I know the problem what are my options?

**(iii) Selective models**

This type of model is used mainly for the decision making process in order to access or select different options. Of course, there would be always at least two alternatives, otherwise there is no read for making any decision.

**Semantic:** Alternatives >> Best option

**Example:** Now ± Know the options. Which or is the best for me?

**(iv) Analytic models**

Analytical models are mainly used for analyzing pre-selected options. This type of model has the ability to access suitability, risk or any desire fitness attributes.

**Semantic:** Option >> Fitness

**Example:** I have picked my option. How good and suitable is it for my objective?

**(v) Instructive Models**

Instructive models provide guidance in a bidirectional or interactive process. Among the example are many support selections available in the market.

**Semantic:** Problem statement >> solution instruction

**Example:** How can I achieve that?

**(vi) Constructive models**

A constructive model is able to design or construct the solution, rather than instructing it.

**Semantic:** Problem statement >> Design solution

**Example:** I read a < ... > with these specifications < ... >.

**(vii) Hybrid models**

In many cases more advanced models are constructed by resting or chaining several models together.

**36. What is a hopefield Network? Explain all the steps involved in the Hopefield Network with suitable example. (2069 Bhadra)**

⇒ See on Page No. 152

**37. Explain different steps of expert system development with an example. (2069 Bhadra)**

⇒ See on Page No. 163

**38. What is a natural language processing? Explain it. (2069 Bhadra)**

⇒ See on Page No. 156

**39. Write short notes on**

**Human Brain versus Neural Network (2069 Bhadra)**

⇒ See on page no 143

⌘⌘⌘