

Time: 3 hours

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of total 8 parts (short-answer type questions) covering the entire syllabus and will carry 16 marks. In addition to the compulsory question there will be four units i.e. Unit-I to Unit-IV. Examiner will set two questions from each Unit of the syllabus and each question will carry 16 marks. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit.

#### UNIT - I

**Overview of A.I:** Introduction to AI, Importance of AI, AI and its related field, AI techniques, Criteria for success.

**Problems, problem space and search:** Defining the problem as a state space search, Production system and its characteristics, Issues in the design of the search problem

**Heuristic search techniques :** Generate and test, hill climbing, best first search technique, problem reduction, constraint satisfaction

#### UNIT - II

**Knowledge Representation:** Definition and importance of knowledge, Knowledge representation, Various approaches used in knowledge representation, Issues in knowledge representation.

**Using Predicate Logic :** Representing Simple Facts in logic, Representing instances and is\_a relationship, Computable function and predicate.

#### UNIT - III

**Natural language processing :** Introduction syntactic processing, Semantic processing, Discourse and pragmatic processing.

**Learning:** Introduction learning, Rote learning, Learning by taking advice, Learning in problem solving, Learning from example-induction, Explanation based learning.

#### UNIT - IV

**Expert System:** Introduction, Representing using domain specific knowledge, Expert system shells.

#### Suggested Readings

1. David W. Rolston : Principles of Artificial Intelligence and Expert System Development, McGraw Hill Book Company.
2. Elaine Rich, Kevin Knight : Artificial Intelligence, Tata McGraw Hill.
3. D.W. Patterson, "Introduction to AI and Expert Systems", PHI, 1999 .
4. Nils J Nilsson , "Artificial Intelligence -A new Synthesis" 2nd Edition (2000), Harcourt Asia Ltd.

**Note:** Latest and additional good books may be suggested and added from time to time.

Q2.(c) What do you mean by AI? Write the importance of AI.

### A.15. Artificial Intelligence

Artificial Intelligence is a widely used term which is very complex and difficult to define. As the years have passed there have been many different views and hence many different definitions for the word "Artificial Intelligence".

According to Elaine Rich (1991) "*Artificial Intelligence is the study of how to make computers do things that at the moment people do better*". Although this definition fails to concentrate on the problems that cannot be dealt by either people or computers and that the environment of computer science is always evolving and it never stays the same.

John R. Searle(1979), another AI expert defines AI in two ways .

The first is what he calls '**Weak AI**', where he sees the computer as being a powerful tool for testing and formulating hypotheses in a correct and clearly stated manner. According to the second way which is '**Strong AI**', if a computer is given the right program it can literally understand and have cognitive states and because the programmed computer has cognitive states, the programs are not only tools that permit us to test psychological explanations, rather the programs are themselves the explanation.

*Thus AI is a branch of computer science concerned with the study and creation of computer systems that show signs of some form of intelligence. These are:*

- *Systems that can draw useful conclusions,*
- *Systems that can learn new concepts and tasks,*

- *Systems that can understand a natural language and*
- *Systems that perform other types of feats that require human types of intelligence.*

### Importance of Artificial Intelligence (AI)

AI is one of the important developments of world. It will surely affect the lives of human in the coming days. It is important in the development of computer products, robotics and related fields. In fact, AI is growing in importance everyday. The field of AI is sure to bring in vast opportunities. The importance of AI can be estimated from the following viewpoints:

1. Major companies are investing big in AI
2. Rapid Advancement
3. Vast opportunities
4. Smarter Software
5. Robotics is developing rapidly
6. AI is essential to overcome information Challenges

#### 1. Major companies are investing big in AI

The large hi-tech companies like IBM, Google, Microsoft, Ebay, Yahoo have multiple active developments underway. They seem to be in a race to acquire systems and so hire inventors and creative minds.

DARPA has also been funding the development of ALV (Autonomous Land Vehicle) which is a driverless military vehicle and the development of an expert system to assist pilots.

#### 2. Rapid Advancement

As the costs of hardware are falling rapidly, AI is becoming a leading choice. High speed Graphical

Processing Units (GPU's) can recognize about 88% of the words spoken in normal, human, English language conversation. With such trends, hardware costs will no longer be a barrier to let AI industry accelerate.

### **3. Vast opportunities**

AI offers vast opportunity space. In future AI will power vision systems that would drive your car, a vacuum cleaner that will recognize the furniture and dirt on it and an autonomous lawnmower that will mow the lawn without disturbing the flowers. Just count on a future when AI will be used as a utility and in everyday objects.

### **4. Smarter Software**

AI will increasingly be made a part of other software programs. AI enabled software will serve as virtual assistants, providing hints and previews of processes and help speed up and maximize tasks.

### **5. Robotics is developing rapidly**

Now advanced robots are gaining enhanced senses, skills and intelligence. It is all credited to AI, machine to machine communication and sensors etc. Industrial robots are already there to take up difficult and dangerous tasks such as welding, spray painting etc.

### **6. AI is essential to overcome information Challenges**

The information growth is becoming so challenging and speedy that human need intelligence aided computer systems for data exploration. In today's world where information is exploding at an exponential rate, human expertise becomes limited in certain domains.

Q2.(b) What is Artificial Intelligence? Explain the applications areas of Artificial Intelligence in detail. MDU BCA 2018

Ans. Artificial Intelligence (AI)

*Artificial Intelligence (AI) is the study of what intelligence is and the process of creating systems that are more useful and smart.*

AI is an interdisciplinary field which includes psychology, philosophy, linguistic, electrical engineering and computing science. Other engineering disciplines are also actively involved in some of the branches of AI. For example, mechanical engineering plays an important role in the design of robots.

AI researchers in psychology, philosophy and linguistics are collectively referred to as information processing psychologists or cognitive scientists. The cognitive scientists are interested in understanding what intelligence is, how it is acquired and used. Their central activity involves the modeling of human information processing. The computing science, electrical engineering and other related engineering fields are more interested in building computing systems that perform at high levels of behaviours – building machines that are more useful and smart.

### Definitions of AI

According to Patterson, *“AI is a branch of computer science that deals with the study and the creation of computer systems that exhibit some form of intelligence.”*

‘By intelligence’ we mean:

- *Systems that learn new concepts and tasks,*

- Systems that can understand a natural language and comprehend a visual scene,
- Systems that can reason and draw useful conclusions about the world around us and
- Systems that perform other types of feats that require human types of intelligence.

According to Haugeland, artificial intelligence is, "the exciting new effort to make computers think ... machines with minds, in the full and literal sense."

According to Bellman, artificial intelligence is, "the automation of activities that we associate with human thinking, activities such as decision making, problem solving, learning ..."

### Aim of Artificial Intelligence

The aim of artificial intelligence (AI) is in some way to try to make computers perform tasks that humans tend to be good at.

In a sense it is an investigation into the attempt to make computers behave in a smarter manner. The actual name Artificial Intelligence was coined by John McCarthy in 1956 – he was the designer of the language LISP.

### Application areas of AI

Every branch of science, engineering and technology shares the tools and techniques available in the domain of artificial intelligence. Application areas of AI are classified as:

1. Expert Systems
2. Natural Language Programs (NLP)
3. Speech Recognition

4. Industrial Automation and Manufacturing
5. Robotics
6. Game Playing
7. Image Understanding and Computer Vision
8. Machine Learning

### 1. Expert Systems

*An expert system is a software package that accumulates the knowledge and decision making capabilities of the specialists in a particular field.*

An expert system is a set of programs that manipulate encoded knowledge to solve problems in a specialized domain that normally requires human expertise.

Expert systems are automatic systems. They respond as an intelligent assistant by giving advice and suggesting possible decisions. Expert systems have been built that can diagnose faults in military systems like aircrafts, radars etc., taxonomically classify members of a particular species, advice on possible chemical structures, diagnose diseases etc.

It is also known as Knowledge Based System. It uses a knowledge base for its artificial intelligence. It also uses the decision rules of human specialists to arrive at certain conclusions and to give recommendations.

The primary goal of expert systems research is to make expertise available to decision makers and technicians who need answers quickly. These knowledge-based applications of artificial intelligence have enhanced productivity in business, science, engineering and the military.

The first and firm application of AI was the design and development of the expert system named MYCIN at Stanford University in the mid 1970s.

## 2. Natural Language Programs (NLP)

One of the long standing goals of artificial intelligence is the creation of programs that are capable of understanding and generating human language.

To instruct a computer to do a certain task, we have to use one or the other programming language to enable us to communicate with the computer. Learning the syntax and rules of the programming language is required. This is required for every new language to be mastered. If we could instruct the computer in our own language, it would be very - very convenient. And this is the Natural Language Programming (NLP). Such software allows users to instruct the computer for performing any task through plain English, French, Spanish or any other natural language instructions.

A Natural Language Program interprets the parts of speech, the meaning of a sentence and then converts the sentence/commands in the computer's own language that it can understand and process. Such program transforms sentences occurring as part of a dialog into data structures which convey the intended meaning of the sentences.

## 3. Speech Recognition

Speech recognition provides a means by which speech is broken down into the individual word tokens. Discrete speech recognition is used widely today, presenting a minor inconvenience to the user.

## 4. Industrial Automation and Manufacturing

Industrial Automation is concerned with introduction, incorporation and intervention of automation, starting from raw material handling, planning and production of items and their assembly leading to higher capable

equipment and their inventory. Automation means less human interference, more computing with knowledge enriched environment. Flexible manufacturing system has emerged to tie up with AI for industrial automation in most of the leading and pioneering industries in the world.

## 5. Robotics

Robotics is one of the prime areas of AI applications. AI methodology is applicable to robotics in two ways: One is design and control of robot and the other is application of robots to various fields such as manufacturing, mining, medicine (surgery). Thus, robotics is a field of AI which enhances the ability to move and act in the world, possibly responding to new perceptions.

The application of robotic system is highly useful in places where human movement is restricted as in nuclear plant or mines. Robotic systems work intelligently in such places and atmosphere where hazards prevail for humans with high probability.

## 6. Game Playing

Game playing is one of the oldest areas of attempt in artificial intelligence. A chess-playing computer is a proof of a machine doing something using intelligence. Also, the simplicity of the rules and their application in the program implies that it is easy to represent the game as a search through a space of possible game positions. Thus, game playing is one of the leading domains where AI has been applied with great success.

## 7. Machine Learning

Machine learning is subfield of artificial intelligence that is concerned with the design and development of

Q3.(a) What is AI? Explain its origin and its various techniques. MDU BCA RE 2015

Ans. Artificial Intelligence (AI)

*Artificial Intelligence (AI) is the study of what intelligence is and the process of creating systems that are more useful and smart.*

AI is an interdisciplinary field which includes psychology, philosophy, linguistic, electrical engineering and computing science. Other engineering disciplines are also actively involved in some of the branches of AI. For example, mechanical engineering plays an important role in the design of robots.

AI researchers in psychology, philosophy and linguistics are collectively referred to as information processing psychologists or cognitive scientists. The cognitive scientists are interested in understanding what intelligence is, how it is acquired and used. Their central activity involves the modeling of human information processing. The computing science, electrical engineering and other related engineering fields are more interested in building computing systems that perform at high levels of behaviours – building machines that are more useful and smart.

### Definitions of AI

According to Patterson, *“AI is a branch of computer science that deals with the study and the creation of computer systems that exhibit some form of intelligence.”*

‘By intelligence’ we mean:

- *Systems that learn new concepts and tasks,*

- *Systems that can understand a natural language and comprehend a visual scene,*
- *Systems that can reason and draw useful conclusions about the world around us and*
- *Systems that perform other types of feats that require human types of intelligence.*

According to Haugeland, artificial intelligence is, *"the exciting new effort to make computers think ... machines with minds, in the full and literal sense."*

According to Bellman, artificial intelligence is, *"the automation of activities that we associate with human thinking, activities such as decision making, problem solving, learning ..."*

### Origin of Artificial Intelligence (AI)

The origins of AI can be seen in Alan Turing's work on intelligent machines published in 1950. Turing formulated the first paper models of game playing programs. Some date the origins of AI to McCarthy's invention of the LISP programming language in the 1950s. This was the first language designed for solving symbolic problems rather than numerical ones.

In 1960s, Allen Newell and Herbert Simon developed General Problem Solver or GPS to imitate human problem solving protocols. It was the first computer program designed to solve problems in a human way.

In the 1970s, expert systems were developed which embodied as a set of rules the knowledge of an expert. The most famous of these was MYCIN medical expert system. At the same time, systems were developed to understand language, of which the most famous was Winograd's SHRDLU system. The 1980s saw the

development of neural networks as a method of learning examples.

With the advancements made in the field of AI and increased association of more and more researchers and scientists to the field, it has become possible to understand the problems and their complexities in a better way.

### AI Techniques

AI technique is a method that achieves knowledge. The main AI techniques are:

1. Search
2. Use of Knowledge
3. Abstraction

#### 1. Search

Search provides a way of solving problems for which no more direct approach is available as well as a framework into which any direct techniques that are available can be embedded.

A search program finds a solution for a problem by trying various sequences of actions or operators until a solution is found.

#### Advantages

- *It is the best way so far as no better way has been found to solve the problems.*
- *To solve a problem using search, it is only necessary to code the operators that can be used; the search will find the sequence of actions that will provide the desired result. For example, a program can be written to play chess using search if one knows the*

*rules of chess; it is not necessary to know how to play good chess.*

### Disadvantages

Most problems have search spaces so large that it is impossible to search the whole space. Chess has been estimated to have  $10^{120}$  possible games. The rapid growth of combinations of possible moves is called the combinatoric explosion problem.

### 2. Use of knowledge

Use of knowledge provides a way of solving complicated problems by manipulating the structures of the objects that are concerned.

The ways in which knowledge can be represented for usage in AI techniques:

AI technique is a method that achieves knowledge that should be represented in such a way that:

- *The knowledge captures generalization. This means grouping situations that share important properties rather than representing each situation separately. With such arrangement of knowledge, unreasonable amounts of memory and updating will no longer be required. Anything without this property is called 'data' rather than knowledge.*
- *It should be represented in such a way that it can be understood by the people who must prepare it. For many programs the size of the data can be achieved automatically by taking readings from a number of instruments, but in many AI areas, most of the knowledge a program has must be provided by people in terms that they understand it.*

- *It could easily be adjusted to correct errors and to demonstrate changes in the world.*
- *It can be used to help overcome its own thorough volume by helping to restrict the range of possibilities that must usually be considered or discussed.*
- *It could be used in different situations even though it may not entirely be complete.*

### 3. Abstraction

Abstraction finds a way of separating important features and modifications from the unimportant ones that would otherwise confuse any process.

Q3.(c) Write a short note on Alan Turing's Criteria for Success.

OR

How will you know if the machine you have constructed is intelligent?

**Ans. Alan Turing's Criteria for Success**

To know about the success of any AI research project Alan Turing proposed a method which has become known as Turing Test.

This test was proposed to determine whether the machine constructed is intelligent or to determine whether a machine can think. To conduct Turing Test, it required two people and the machine to be evaluated – all in separate rooms. One person acts as interrogator who asks questions from either the person or the computer by typing questions and receives typed responses. However the interrogator knows them as A and B and tries to determine which is the Person and which is the machine. If the machine is able to fool the interrogator by making him believe that it is the person, then the conclusion is that the machine can think.

The machine is allowed to do whatever it can. For example if asked, "How much is 12,234 times 73,981?" It could wait for several minutes and then show wrong answer in response. To pass the Turing Test, of course the machine would need an appropriate amount of knowledge. To measure a program's performance, if a program is asked to paraphrase a newspaper story whether the program responds in the way a Person does.

The goal in writing such program is to simulate human performance at a task and the measure of success is the extent to which the program responds.

Q4.(a) Explain the following briefly with suitable examples:

1. Problem
2. Problem Characteristics MDU BCA 2016
3. Problem Space MDU BCA RE 2018
4. Search Space
5. Search
6. Issues in design of search problem  
MDU BCA 2016
7. Use/Role of AI in problem solving  
MDU BCA 2016
8. Constraint satisfaction and its uses  
MDU BCA RE 2016

OR

**Constraint satisfaction**

MDU BCA RE 2018, 2016

**Ans. 1. Problem**

*Problem is the question which is to be solved. For solving a problem it needs to be precisely defined. The definition means, defining the start state, goal state, other valid states and transitions.*

So, a problem consists of a set of initial states, a set of goal states and a set of path constraints. The problem is to find a path that satisfies the path constraints and ends at any goal state.

To formulate any problem, four components are defined in a very formal way. These four components are:

- The initial state or starting state.

3. Is the problem universe predictable?	No, since move of other player is not sure.
4. Is a good solution absolute or relative?	Absolute
5. Is the solution a state or a path to state?	Path because winning state describes path to state
6. What is the role of knowledge?	It helps to constrain solution search.
7. Does the task require human interactions?	No, since additional assistance is not required.

### 3. Problem Space

A problem space consists of a set of symbolic structures (the states of the space) and a set of operators over the space. Each operator takes a state as input and produces a state as output, although there may be other inputs and outputs as well. The operators may be partial, i.e. not defined for all states. Sequences of operators define paths that thread their way through sequences of states.

An example of a problem space is that of the game of chess:

- (a) Initial state: chess pieces in starting position
- (b) Operations: legal moves
- (c) Goal states: checkmate, stalemate etc.

A problem space can also be considered to be a search space because in order to solve the problem, we will search the space for a goal state.

#### 4. Search Space

*It is the complete set of states including the start and goal states, where the answer of the problem is to be searched.*

Because of the search space problem, automated provers today can only prove relatively simple theorems in special domains with few axioms.

#### 5. Search

*It is the process of finding the solution in search space.*

The input to search space algorithm is problem and output is solution in the form of action sequence.

#### 6. Issues in design of search problem

To solve problems in AI, search is applied. The solution of problem depends on the search process. Search techniques guide one to reach goal state of the problem.

Because of the cognitive skills possessed, humans can analyze the state space and identify the problem solution. While searching, the state space is traversed. If there is an error, backtrack occurs to find the alternative path. So there is a need of systematic procedure for forward and backward traversing.

The problems in which direction of search is also to be determined, there must be some rules. So, selection of rules is also an important factor.

After the direction comes the path representation.

Hence, the following issues are observed in design of search problems:

Q4.(b) How to define the problem as a state space search? Explain.

Ans. To solve the problem of playing a game, we require the rules of the game and targets for winning as well as representing positions in the game. The opening position can be defined as the initial state and a winning position as a goal state. Moves from initial state to other states leading to the goal state follow legally. However, the rules are far too abundant in most games - especially in chess, where they exceed the number of particles in the universe. Thus, the rules cannot be supplied accurately and computer programs cannot handle easily. The storage also presents another problem but searching can be achieved by hashing.

The number of rules that are used must be minimized and the set can be created by expressing each rule in a form as possible. The representation of games leads to a state space representation and it is common for well-organized games with some structure. This representation allows for the formal definition of a problem that needs the movement from a set of initial positions to one of a set of target positions. It means that the solution involves using known techniques and a systematic search. This is quite a common method in Artificial Intelligence.

### State Space Search

*A state space represents a problem in terms of states and operators that change states.*

A state space consists of:

- A representation of the states the system can be in. For example, in a board game, the board represents the current state of the game.

- A set of operators that can change one state into another state. In a board game, the operators are the legal moves from any given state. Often the operators are represented as programs that change a state representation to represent the new state.
- An initial state.
- A set of final states; some of these may be desirable, others undesirable. This set is often represented implicitly by a program that detects terminal states.

### The Water Jug Problem

In this problem, we use two jugs called **four** and **three**; **four** holds a maximum of four gallons of water and **three** a maximum of three gallons of water. Neither has a measuring marker on it. There is a tap that can be used to fill the jugs with water. How can we get exactly two gallons of water in the **four** jug?

The state space is a set of prearranged pairs giving the number of gallons of water in the pair of jugs at any time, i.e., (**four**, **three**) where **four** = 0, 1, 2, 3 or 4 and **three** = 0, 1, 2 or 3.

The start state is (0, 0) and the goal state is (2, n) where n may be any but it is limited to **three** holding from 0 to 3 gallons of water. Three and four shows the name and numerical number shows the amount of water in jugs for solving the water jug problem. The major production rules for solving this problem are shown below:

Initial condition	Goal comment
1. (four, three) if four < 4	(4, three) fill four from tap
2. (four, three) if three < 3	(four, 3) fill three from tap

- |  |  |
|--|--|
| 3. (four, three) if four > 0           | ((0, three) empty four into drain  |
| 4. (four, three) if three > 0          | ((four, 0) empty three into drain  |
| 5. (four, three) if (four + three) < 4 | ((four + three, 0) empty three into four   |
| 6. (four, three) if (four + three) < 3 | ((0, four + three) empty four into three   |
| 7. (0, three) If three > 0             | ((three, 0) empty three into four  |
| 8. (four, 0) if four > 0               | ((0, four) empty four into three   |
| 9. (0, 2)                              | ((2, 0) empty three into four  |
| 10. (2, 0)                             | ((0, 2) empty four into three  |
| 11. (four, three) if four < 4          | ((4, three-diff) pour diff, 4-four, into four from three   |
| 12. (three, four) if three < 3         | ((four-diff, 3) pour diff, 3-three, into three from four and a solution is given below four three rule |

#### One Solution to the Water Jug Problem is:

Gallons in Four Jug	Gallons in Three Jug	Rules Applied
0	0	-
0	3	2
3	0	7

3	3	2
4	2	11
0	2	3
2	0	10

The problem solved by using the production rules in combination with an appropriate control strategy, moving through the problem space until a path from an initial state to a goal state is found.

In this problem solving process, search is the fundamental concept. For simple problems it is easier to achieve this goal by hand but there will be cases where this is far too difficult.

Q4.(c) Explain production system and its characteristics in detail.

MDU BCA RE 2018, 2018

Ans. Production System

*An artificial intelligence system developed for solution of any problem is called production system.*

Production systems provide search structures that form the core of many intelligent processes. Production systems are applied to problem solving programs that must perform a wide-range of searches. They are useful to structure AI programs in a way that facilitates describing and performing the search process.

A production system consists of:

- A global database
- Production rules
- A control strategy

Transforming a problem statement into these three components of the production system is often called the representation of problem in AI.

- **A global database:** The global database is the system's short-term memory or the central database. These are collections of facts that are to be analyzed. A part of the global database represents the current state of the system's environment. The size of this global database depends on the problem statement. It may be as small as a small matrix or it could be as large as a complex relational database. In a production system whenever a condition in the system is satisfied, the system is allowed to execute or perform a specific action which may be specified under that rule. If the rule is not fulfilled, it may perform another action.

- **Production rules:** A production system consists of a set of rules. These rules are conditional if-then branches. These rules are applied on the global database. Each rule consists of a left hand and a right hand side. Left hand side or pattern determines the applicability of the rule and right side describes the operation to be performed in case the rule is applied i.e. each rule has a precondition. If a precondition of the rule is satisfied only then the corresponding rule could be applied to the respective state.
- **A control strategy:** A control strategy performs the task of selecting rules. It specifies the order in which the rules will be compared to the database. There are two kinds of control strategies:
  - **Irrecoverable:** In this when a rule is selected and applied; the state of the problem cannot be recovered.
  - **Tentative:** In this when a rule is selected and applied, provision is made to return later to this point in the computation.

The control strategy also resolves the conflict that arises when several rules match at once. This is the major issue to be sorted out in a production system.

### Different conflict resolution strategies

Some of conflict resolution strategies are:

- **Perform the first:** In this method, the system chooses the first rule that matches.
- **Sequencing Techniques:** Adopt the rules in the sequence they are.

Q6.(b) Explain best first search technique in detail.

MDU BCA 2018

OR

Write an algorithm for best first techniques.

MDU BCA 2015

Ans. Best First Search

*Best first search is a combination of Depth First Search (DFS) and Breadth First Search (BFS). In best first search method, a single path is followed at a time, but paths are switched when more-promising paths than the current one are found.*

The breadth first search starts at the start node and maintains the set of candidate paths. It chooses for expansion of a shortest candidate path. It refines this principle by computing a heuristic estimate for each candidate and chooses for expansion the best candidate according to this estimate.

The heuristic function used in best first search is called an evaluation function. An evaluation function indicates how far the node is from the goal node. Goal nodes have an evaluation function value as zero.

**Example:** Consider the following example:

Q6.(a) Explain problem reduction technique and its merits with examples. MDU BCA RE 2016

OR

Discuss problem reduction technique with an example. MDU BCA 2016

OR

What is problem reduction? How is it useful and used? Explain. MDU BCA RE 2015

**Ans. Problem Reduction**

*The process of decomposing a complex problem into a set of sub-problems, finding the sub-solutions for these sub-problems and then integrating all these sub-solutions to get the solution of the given complex problem is known as problem reduction.*

Thus, the problem reduction idea involves replacing a problem goal by a set of subgoals such that if the subgoals are solved, the main goal is also solved. Problem reduction is better if it is easy to decompose a problem into independent sub-problems. It is a basic problem-solving technique of AI.

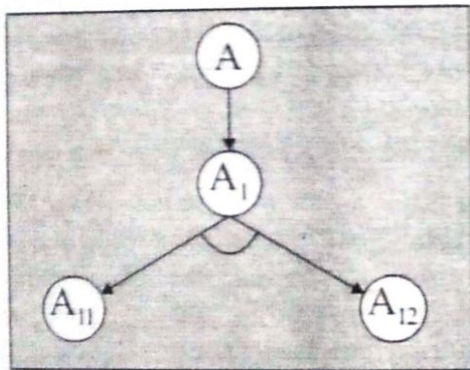
**How Problem Reduction is used and useful?**

The problem reduction technique is used to divide large problems into more readily solvable sub-problems. The structure of sub-problems can be represented in terms of an AND/OR graph. A path in such a graph can denote the solution to an overall problem.

Thus in the problem reduction approach, a user's problem is reduced to sub-problems. Each sub-problem is then treated as a problem and reduced into further sub-problems and so on. The reduction process continues

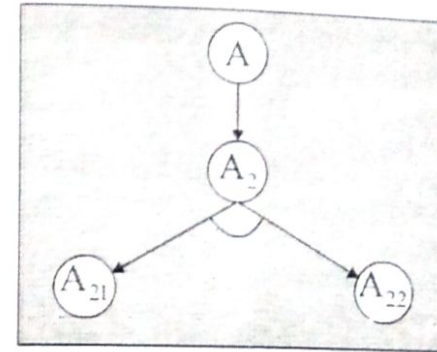
until all problems have been reduced to primitive sub-problems. The primitive sub-problems are those that can be solved readily. An AND/OR graph can be used to depict the various ways a problem can be reduced.

The start node of the graph depicts the original problem. Each directed arc from the node represents the application of a reduction operator to the problem. Reduction operators can be classified into two categories: AND reductions and OR reductions. The applications of an AND reduction operator as shown in the figure results in two sub-problems  $A_{11}$  and  $A_{12}$ .



If both  $A_{11}$  and  $A_{12}$  can be solved then problem  $A$  can be solved. The node  $A_1$  is called a synthesis node and represents the notion that the results of the sub-problems have to be synthesized to obtain a solution for  $A$ . Because the solution depends on solutions existing for both  $A_{11}$  and  $A_{12}$ , these nodes are called AND nodes and are represented in the graph by incoming arcs which are cut by semicircles.

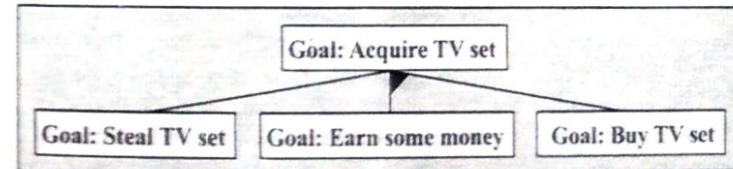
The application of an OR reduction operator is shown below:



Application of the operator results in two problems  $A_{21}$  and  $A_{22}$ . Because these are not AND nodes, solving either one of the problems would allow the solution of the problem  $A$ .  $A_{21}$  and  $A_{22}$  are therefore called OR nodes.

The node  $A_2$  is used to denote a point where heuristic information would be valuable in deciding which sub-program would be most profitable to explore first.

**Example:** How to acquire TV set?



(AND - OR Graphs)

Q5.(a) Explain Hill Climbing in detail.

MDU BCA RE 2018

OR

Write the algorithm for hill climbing search.  
What are the problems in hill climbing?  
Explain using examples.

MDU BCA 2015

**Ans.** Imagine you are climbing a mountain and you are in a very thick fog. You can only see a distance equal to one step length. To try to climb you take the step in the direction that is steepest to get to the highest point of all the locations you can currently see.

In other words, hill-climbing search simply evaluates the objective function for all states that are neighbours to the current state, and takes the neighbour state with the best objective function value as the new current state. If there are more than one next best state, one is picked randomly.

The Hill-Climbing search always moves towards the goal. Using heuristics it finds which direction will take it closest to the goal.

The name 'hill climbing' comes from an analogy:

A hiker is lost halfway up/down a mountain at night. His camp is at the top of the mountain. Even though it is dark, the hiker knows that every step he takes up the mountain is a step towards his goal. So a hill-climbing search always goes to the node closest to the goal.

Hill climbing is actually a combination of Generate-and-Test plus direction to move. In this the heuristic function is to estimate how close a given state is to a goal state.

### Algorithm for Hill Climbing

**Step 1:** Place the initial node on a list, START-LIST.

**Step 2:** If (START-LIST is empty) or (START-LIST is GOAL) then terminate search.

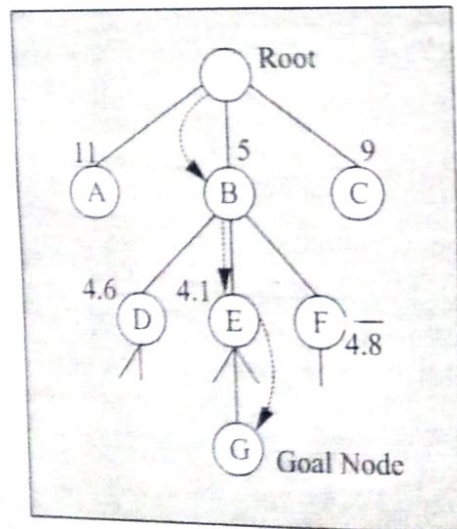
**Step 3:** Remove the first-node from the START-LIST. Call this node, f.

**Step 4:** If (f = GOAL) then terminate search with SUCCESS.

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

**Step 6:** Goto Step 2.

**Example:** Consider the following tree:



- Here, let the initial node i.e. root be on a START-LIST. Since START-LIST is not empty nor the Goal, remove this node (f) from the START-LIST and generate its successors. Sort this START-LIST on the

basis of the distances from the goal and keep repeating this process till a goal is found.

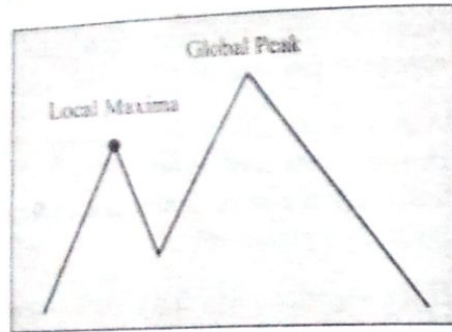
- A new state has to be better, meaning that if one considers the cost function then it means a lower value and if one considers the objective function then it means a higher value.
- If the algorithm is constantly following the direction which gives the fastest rate of increase then it is called as the **steepest ascent hill climbing**. But if the lowest cost is best then it is a **down hill algorithm**.
- This algorithm is called as a steepest rise with backtracking search method because it is prepared to back up and try a different path if the steepest one leads to a dead end.
- Hill climbing is sometimes called as a greedy local search because it grabs a good neighbour state without thinking ahead about where to go next.

### Problems of Hill Climbing Techniques

Hill climbing can produce lot of saving over blind searches when an informative, reliable function is available to guide the search to a global goal. It suffers from some serious drawbacks when this is not the case. A major problem of hill climbing strategies is their tendency to become stuck at foothills, a plateau or a ridge. If the algorithm reaches any of the above mentioned states, then the algorithm fails to find a solution. These drawbacks are:

#### 1. Local Maxima

Local maxima is a state that is better than all its neighbours but farther away from the actual goal i.e. the global peak.

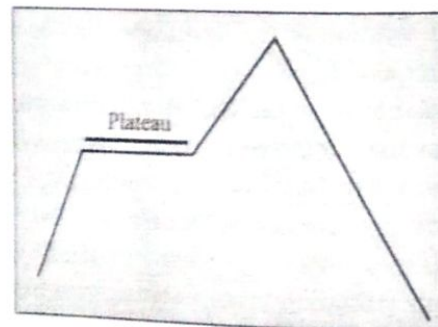


#### Solution to this problem:

- One possible solution is backtracking. Backtracking helps in undoing what has been done so far and permits to try a totally different path to attain the global peak.
- Another solution can be a list of promising plan.

### 2. Plateau

It is a flat area of the search space in which all neighbouring states have the same value.



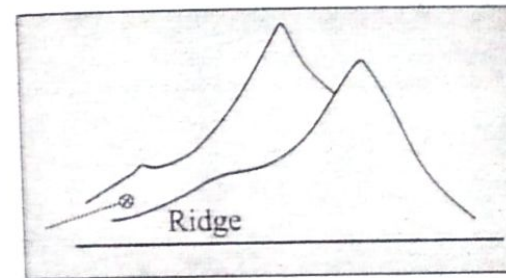
#### Solution to this problem:

- One possible solution is a big jump to escape from the plateau. A huge jump is recommended because in a plateau all neighbouring points have the same value.

- Another solution is to apply small steps several times in the same direction. This depends on the rules available.

### 3. Ridge

Ridge is a special kind of local maximum. It is an area of the search space which is higher than the surrounding areas and that itself has a slope as shown in the figure:



One can not travel the ridge by single moves as the orientation of the high region compared to the set of available moves makes it impossible.

#### Solution to this problem:

- Trying different paths at the same time is the solution of this problem. One can apply two or more rules before doing the test. This corresponds to moving in several directions at once.

Q7.(a) What is knowledge? Describe the importance of knowledge.

Ans. Knowledge

*Knowledge can be defined as the body of facts and the principles accumulated by human act, fact or state of knowing.*

*Knowledge is the ability to understand information and then to form judgments, opinions, make predictions and decisions based on that understanding.*

Knowledge also means a familiarity with language, concepts, procedure, rules, ideas, places, facts, associations, coupled with an ability to use these notions effectively in modeling different aspects of world. Without this ability, the facts and concepts are meaningless.

By the Oxford dictionary definition, "*Knowledge is defined as expertise and skills acquired by a person through experience or education, the theoretical or practical understanding of a subject.*"

Knowledge is human understanding of a subject matter that has been acquired through proper study and experience. Knowledge is usually built by learning, thinking and proper understanding of the problem area. It can be considered as the integration of human perceptive process that helps to draw meaningful conclusions.

In order to perform an intelligent action (like solving a given problem, answering a question, playing a specific game, speaking a natural language), one must possess the relevant knowledge in the form of specific facts and the inter-relations amongst them.

Knowledge is a progression that starts with data that is of limited utility. By organizing or analysing the data, we understand what the data means, and this becomes information. The interpretation or evaluation of information yield knowledge.

Knowledge simply means *"the ideas or understandings which an entity possesses that are used to take effective action to achieve the entity's goal(s). This knowledge is specific to the entity which created it."*

Knowledge is viewed by the AI experts as a collection of facts about the world to be formally represented in a computer. It forms the knowledge base of an AI system on which the problem solving software should work.

### Importance of Knowledge

- *Intelligence requires knowledge. That is, to exhibit intelligence, knowledge is required. Knowledge plays a major role in building intelligent systems.*
- *To solve real world problems easily, effectively, efficiently and economically, expert systems are developed using knowledge.*
- *Knowledge serves as a background for articulating possible courses of action, for judging whether courses of action will yield the intended result and for using this judgment in selecting among them, for deciding how actions should be implemented and for actually implementing actions.*
- *Knowledge plays an important role in building AI systems. It is the core aspect of Artificial Intelligence.*

Q7.(b) What is knowledge representation? What are the different types of knowledge representation scheme? Discuss significance of each.

MDU BCA RE 2015

Ans. Knowledge Representation

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

Knowledge representation is a study of ways of how knowledge is actually picturized and how effectively it resembles the representation of knowledge in human brain.

Knowledge representation plays an important role in AI. In order to solve a complex problem in artificial intelligence, one needs both large amount of knowledge and some mechanism for manipulating the knowledge to create solution to new problems. A variety of ways to represent knowledge have been exploited in artificial intelligence programs.

Following are the desirable characteristics of a knowledge representation scheme:

- *This representation scheme should have a set of well defined syntax and semantics. This will help in representing various kinds of knowledge.*
- *The representation must be efficient i.e. it should use only limited resources without compromising on the expressive power.*
- *This representation scheme should have a good expressive capacity. A good expressive capability will*

*catalyze the inferencing mechanism in its reasoning process.*

### Types of knowledge representation schemes

In the early days, knowledge to be represented was divided into two classes, namely, declarative knowledge and procedural knowledge. Therefore, the knowledge representation schemes have been simply classified as:

1. Declarative representation schemes
2. Procedural representation schemes

#### 1. Declarative representation scheme

Declarative representation scheme emphasizes the representation of facts and includes guidance on how the knowledge is to be used. It declares every piece of knowledge and permits the reasoning system to use the rules of inference to come out with new pieces of information.

Declarative knowledge representation can be understood from the example:

$$\forall x \text{ PERSON}(x) \supset \text{MORTAL}(x)$$

$$\forall x \text{ ANIMAL}(x) \supset \text{MORTAL}(x)$$

PERSON (RAVI)

ANIMAL (BIRD)

that represents some simple facts that every person and every animal is mortal and that RAVI is a person and bird an animal. Using modus ponens, the system can automatically conclude that Ravi is mortal and so is every bird.

The typical means used for declarative knowledge representation are:

- *Formal Logic*
- *Semantic Networks*
- *Frames*
- *Scripts*

### Significance of declarative knowledge representation scheme

The significance of declarative knowledge representation scheme are:

- *Transparency and unambiguity of represented knowledge.*
- *Flexibility of knowledge storage.*
- *Each piece of knowledge is an independent chunk on its own. Hence modularity is higher.*
- *Storage efficiency.*
- *Direct inference*

#### 2. Procedural representation scheme

A procedural representation scheme represents knowledge as procedures and the inferencing mechanisms manipulate these procedures to arrive at the result.

If a procedural knowledge representation is used for the same purpose, separate procedures have to be written for each person, animal etc., as well as for mortal:

```
procedure PERSON (x)
  if (x = Ravi) then return true
  else return false
```

```
procedure MORTAL (x)
  if PERSON (x) then return true
```

Q10.(a) Explain issues in knowledge representation in detail. MDU BCA RE 2018

OR

Describe the role of various issues in knowledge representation with examples.

MDU BCA RE 2016

OR

Explain various issues in knowledge representation with examples. MDU BCA 2016

Ans. The key to problem solving is a large amount of knowledge and some mechanism for manipulating that knowledge. The basic goal of knowledge representation is to help derive conclusions from the knowledge. When many knowledge based techniques are used into a single problem many issues may arise. Some of the issues that arise while using knowledge representation technique are the following:

1. Important attributes
2. Relationships among attributes
3. Choosing Granularity
4. Set of objects
5. Finding Right Structure

#### 1. Important attributes

There are two attributes that are of very general significance. These are "instance" and "is a". These attributes are significant because they support property inheritance.

#### 2. Relationships among attributes

The attributes that are used to describe objects are entities that we represent. They have certain properties

independent of the specific knowledge they encode. The relationship between the attributes of an object may hold the following four properties:

- (a) Inverses
- (b) An 'is a' hierarchy of attribute
- (c) Techniques for Reasoning about values
- (d) Single Valued Attributes

#### (a) Inverses

Entities are related to each other in a number of different ways. These relationships are described as attributes. This may begin at the object being described and terminate at the object representing the value of the specified attribute. But the focus can be on the object representing the value. Although different but there is still a relationship between the two entities.

#### (b) An 'is a' hierarchy of attribute

It is generalization vs specialization. For example, the height of a person is a specialization of general attribute physical size. These generalization-specialization relationships are important for attributes because they support inheritance.

#### (c) Techniques for Reasoning about values

Sometimes values attributes are specified explicitly when a knowledge base is created. The reasoning system must reason about values which have not been given explicitly. Several kinds of information play a role in this reasoning. For example, the value of height must be a number measured in a unit of length. Another example, the age of a person can not be more than the age of either of his parents etc.

### (d) Single Valued Attributes

A specific but very useful kind of attribute is one that is guaranteed to take a unique value. For example, President of India. Another example is that a player can at a time be a member of only one team.

The knowledge representation system takes several different approaches to provide support for single-valued attributes.

### 3. Choosing Granularity

Granularity means what level should the knowledge be represented and what are the primitives.

- Should there be a small number or should there be a large number of low-level primitives or High-level facts.
- High level facts may not be adequate for inference while Low-level primitive may require a lot of storage.

Granularity of representation is an important aspect in knowledge representation. It deals with depth of detail. The Granularity decides the size of knowledge base. As more and more information is stored, the size of knowledge base grows. The more detail will mean more accurate information but it also increases the size of knowledge base. If fewer details are stored the decisions requiring more depth can not be generated.

### 4. Set of objects

There are certain properties of objects that are true as a member of a set but not as individual. It is important to represent sets of objects if a property is true for all or most elements of a set then it is more efficient to associate it with the set rather than to associate it

explicitly with every element of the set. This can be done in the following manner:

- A *logical representation through the use of universal quantifier and*
- A *hierarchical structure where nodes represent sets and inheritance propagate set level assertion down to individual.*

### 5. Finding Right Structure

For describing a particular situation right structure is selected. This requires selecting an initial structure and then revising or reviewing the choice. For this, it is important to know the following:-

- *How to make an initial selection of the most appropriate structure?*
- *How to fill in details from the current situations?*
- *How to find a better structure in case the one chosen initially is not appropriate?*
- *What to do if none of the available structure is appropriate?*
- *When to create a new structure?*

**Q10.(d)** Write and explain the various approaches to knowledge representation.

**Ans.** A knowledge base requires knowledge and that knowledge is to be stored systemically and represented in proper form in knowledge base. Hence a technique of knowledge representation is to be adopted for properly representing the knowledge. A good knowledge representation system should allow fast and accurate access to knowledge.

To organize and represent the knowledge there is no single technique. A number of approaches for knowledge representation are the following:

1. Simple Relational Knowledge
2. Inheritable Knowledge
3. Inferential Knowledge
4. Procedural Knowledge

### **1. Simple Relational Knowledge**

This approach represents declarative facts as a set of relations of the same sort of data in the database system. It is the simplest way. It uses tabular form to store facts. The facts about a set of objects are put systematically in columns. It is made up of objects consisting of attributes and their corresponding values. Though such representation provides weak inferential capabilities yet may provide support for other powerful inferential engines.

Consider the following example:

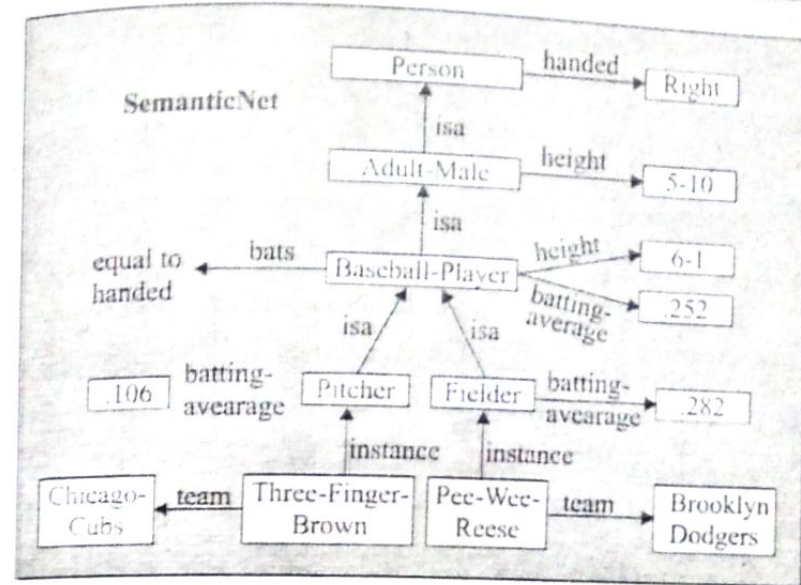
Player's name	Height	Weight
Anshul	6'4"	78 Kg
Ajay	5'4"	87 Kg
Ravi	6'2"	67 Kg
Hari	5'6"	46 Kg

From the given facts it is not possible to answer a simple question "Who is the tallest player". But if a procedure for finding tallest player is provided, then these facts can compute an answer.

## 2. Inheritable Knowledge

Inheritable knowledge centers on relationships and shared attributes between kinds or classes of objects. One of the most useful inferencing is Property Inheritance in which elements of specific classes inherit attributes and values from more general classes (i.e. their parents). To support property inheritance, objects are organized into classes and classes are organized in a generalization hierarchy. Inheritance is a powerful form of inference, but not adequate. This knowledge representation in hierarchical fashion is shown below and is called slot-and-filler structure or semantic network or collection of frames.

Consider the semantic network for Base ball knowledge:



In a semantic network

- Lines represent attributes.
- Boxed nodes represent objects and values of attributes of objects.
- The arrows on the lines point from an object to its value along the attribute line.
- Two attributes provide basis for property inheritance as an inference technique i.e. 'isa' – used to represent class inclusion. 'instance' – used to represent class membership.

This technique helps the knowledge base to retrieve both.

- Facts that have been explicitly stored.
- Facts that can be derived from those that are explicitly stored.

### 3. Inferential Knowledge

It generates new information from the given information. This new information does not require further data gathering from source, but does require analysis of the given information to generate new knowledge.

Given a set of relations and values, one may infer other values or relations. In addition to algebraic relations, a predicate logic is used to infer from a set of attributes.

### 4. Procedural Knowledge

Procedural knowledge is, in general terms, the type of knowledge someone has and demonstrates through the procedure of doing something i.e. Coding actions to be performed when a condition is satisfied.

Example:

IF

Student has deposited fees

Student has attended 90% classes

Student has passed the examination

THEN

Declare the student pass

Q11.(b) Explain "is a" and "instance" relationships.

**Ans.** Two attributes "is a" and "instance" play an important role in many aspects of knowledge representation. The reason for this is that they support property inheritance.

Knowledge can be represented as classes, objects, attributes and Super class and sub class relationships. In property inheritance, elements of specific classes inherit attributes and values from general classes in which they are included. For example, if we know that birds can fly and that canaries are birds, then we can infer that canaries too can fly.

Attribute "is a" is used to represent the relationship "Class inclusion" (super class, sub class relationship)

For example: is a (mega\_star, rich)

Attribute "instance" is used to represent the relationship "Class membership" (element of the class).

For example instance (prince, mega\_star)

**Example:** Represent the following facts:

1. Shreekant was a man.
2. Shreekant is Hindu.
3. All Hindus are Indians.
4. Shivaji is ruler.
5. All Indians were loyal to Shivaji.

**Representing these facts in predicate calculus:**

1. man (Shreekant)
2. Hindu (Shreekant)
3.  $\forall x : \text{Hindu}(x) \rightarrow \text{Indian}(x)$

Q11.(a) Explain computable function in detail.

MDU BCA 2018

OR

What is computable function? How is it useful and used? Explain with an example.

MDU BCA RE 2016

Ans. Computable functions are used to represent simple facts like greater than or less than. For example,

gt(12, 3)                      lt(2, 12)

gt(5, 2)                      lt(3, 7)

greater than              less than

It is useful to have computable functions as well as computable predicates. This helps to evaluate the truth of gt(7+3, 2).

To do this, the value of the plus function is computed first giving the arguments 7 and 3 and then these arguments 10 and 2 are sent to the gt function.

Consider the following set of facts:

1. Chaminda was a witch.  
witch (Chaminda)
2. Chaminda was a Ravansenapati.  
Ravansenapati (Chaminda)
3. Chaminda was born in 20 A.D.  
born (Chaminda, 20)
4. All witches are mortal.  
 $\forall x : \text{witch}(x) \rightarrow \text{mortal}(x)$

**Q10.(c) How Propositional Logic and Predicate Logic represent simple facts in logic? Explain.**

**Ans. Propositional Logic**

Propositional Logic is a simple language that is useful for showing key ideas and definitions. User defines a set of Propositional symbols, like P and Q. User defines the semantics of each of these symbols.

For example,

P means "It is hot"

Q means "It is humid"

R means "It is raining"

**The alphabet of propositional logic contains the following symbols:**

1. The letters of the English alphabet; that is, A, B, C, ..., Z and each of these letters with an index (e.g  $A_3$ ).
2. The logical values True and False.
3. These special symbols:
  - $\neg$  (NOT)
  - $\wedge$  (AND)
  - $\vee$  (OR)
  - $\Rightarrow$  (IF - THEN)
  - $\Leftrightarrow$  (IF AND ONLY IF)
  - ( ) (GROUPING)

The symbol  $\neg$  is called a unary connective and the symbols  $\wedge$ ,  $\vee$ ,  $\Rightarrow$  and  $\Leftrightarrow$  are called binary connectives.

The rules for creating propositions are as follows:

1. All letters, all indexed letters and the logical values True and False are propositions. They are called atomic propositions.
2. If P and Q are propositions, then so are  $\neg P$ ,  $P \wedge Q$ ,  $P \vee Q$ ,  $P \Rightarrow Q$ ,  $P \Leftrightarrow Q$  and (P). They are called compound propositions.

The compound proposition  $\neg P$  is called the negation of P,  $P \wedge Q$  is called the conjunction of P and Q, and  $P \vee Q$  is called the disjunction of P and Q.

**Example:** Suppose the propositions P and Q stand for these statement about the world:

P : It is raining outside.

Q : The street is wet.

Then the following compound propositions stand for these statements about the world:

$\neg P$  : It is not raining outside.

$P \wedge Q$  : It is raining outside and the street is wet.

$P \vee Q$  : It is raining outside or the street is wet.

$P \Rightarrow Q$  : It is raining outside, then the street is wet.

$P \Leftrightarrow Q$  : It is raining outside if and only if the street is wet.

### Predicate Logic or First Order Predicate Logic

Predicate logic or first order predicate logic (FOPL) is one of the oldest and most important knowledge representation schemes used in Artificial Intelligence.

Its components are:

- **Predicate symbols:** These are used to represent a relation in a domain. For example, to represent a sentence like "Suman writes books", the predicate symbol Write is used. The simple formula will be:

Write(Suman, Books)

- **Constant symbols:** A constant symbol is used to represent objects or entities in a domain. These objects or entities may be physical objects, people, concepts or anything that is to be named. For example, in the above formula Suman and Books are constant symbols.
- **Variable symbols:** Variable symbols like x and y are also terms. They allow the users to be indefinite about which entity is to be referred to. For example, Write (x, y)
- **Function symbols:** They denote functions in the domain of discourse. For example, Ravi's mother is married to Ravi's father, the atomic formula would be: MARRIED[mother (Ravi), father (Ravi)] where mother and father are function symbols.

### Quantifiers

A quantifier is a symbol that permits one to declare or identify the range or scope of the variable in a logical expression.

There are two basic quantifiers used in logic. They are Universal quantifier ( $\forall$ ) and Existential quantifier ( $\exists$ ).

- **Universal quantifiers:** It means something is true for all possible values of a variable. Using this, the sentence "All parrots are green" can be expressed as:  
 $\forall x [\text{parrot}(x) \Rightarrow \text{colour}(x, \text{green})]$

- **Existential quantifier:** It means the formula consisting of an existential quantifier has a value true for at least one assignment of  $x$  to an entity in the domain. For example, "There is a person who drives the car" could be represented as:  

$$\exists x \text{ drives}(x, \text{car})$$

### Representation of Simple Facts in Logic

Consider the following statement:

P: All employees earning greater than Rs. 25,000 per month pay tax.

Q: Some employees are on leave today.

R: NO employee earns more than the director.

To represent these statements in predicate logic, abbreviations for the predicates and functions have to be defined.

#### Example:

EMP( $x$ ) for  $x$  is an employee

DIR( $x$ ) for  $x$  is director

Inc( $x$ ) for income of  $x$

GE( $u, v$ ) for  $u$  is greater than  $v$

ON\_LEAVE( $x$ ) for  $x$  is on leave today

TAX( $x$ ) for  $x$  pays tax

Using these abbreviations P, Q and R are:

P :  $\forall x ((\text{EMP}(x) \wedge \text{GE}(\text{inc}(x), 25000)) \rightarrow \text{TAX}(x))$

Q :  $\exists x ((\text{EMP}(x) \rightarrow \text{ON\_LEAVE}(x)))$

R :  $\forall xy (\text{EMP}(x) \wedge \text{DIR}(y)) \rightarrow \neg \text{GE}(\text{inc}(x), \text{inc}(y))$

Q12.(a) What is Natural Language Processing?

MDU BCA RE 2018

OR

Explain NLP(Natural Language Processing) in detail.

MDU BCA 2018

OR

What are various processing schemes in NLP?

MDU BCA RE 2017, 2017

OR

What is NLP? Discuss various Natural Language Processing techniques.

MDU BCA RE 2015, 2015

Ans. Natural Language Processing (NLP)

*Natural language processing is a field of artificial intelligence and linguistics. It focuses on developing systems that allow computers to communicate with people using everyday language.*

OR

*Natural language processing (NLP) can be defined as the automatic (or semi-automatic) processing of human language.*

OR

*Natural language processing is the engineering of systems that processes or analyzes written or spoken natural language.*

Natural language processing is a significant area of AI because a computer would be considered intelligent if it can understand the commands given in natural language.

The main task of natural language processing is to deal with the interactions between computers and human languages. This in itself is an entire separate area of

computer science, which is called human-computer interaction.

The main problems that face people working on natural language processing are summed up in the task of natural language understanding, which means enabling computers to understand in a certain way what human language input is meant to convey.

The most recent natural language processing ideas involve machine learning, and more specifically statistical machine learning. Machine learning is also a branch of artificial intelligence and it concerns the construction and study of systems that can learn from data.

Some of the most common topics of research in natural language processing include: machine translation, which involves translating the input text from one human language to another. This is one of the hardest problems and requires a very wide range of knowledge types in order for it to be solved.

There is also automatic summarization, which involves producing a readable summary of a passage or a text. This is an often used application of natural language processing. There is discourse analysis, and it includes a lot of tasks such as identifying the discourse structure of text, and recognizing and classifying speech acts in a chunk of text. Other commonly known applications are: conference resolution, name entity recognition, natural language generation, natural language understanding, optical character recognition, question answering, speech recognition and sentiment analysis.

This is just a small portion of what natural language processing involves. And, the span of what natural language processing can become in the future is very

wide. One of the most difficult problems still facing professionals in the field is human-level natural language processing which if solved is equivalent to solving the central artificial intelligence problem that is making computers as intelligent as people.

The future of natural language processing is therefore tied closely to the development of artificial intelligence. As natural language understanding improves, future computers will have the ability to obtain data and learn online and apply that in the real world. Combined with natural language generation, computers will soon be more capable taking in and giving out instructions.

### Goal of NLP

The goal of NLP is to accomplish human-like language processing i.e. the goal of NLP is to design and build a computer system that will analyze, understand, and generate natural human-languages.

A complete NLP system would be able to:

- *Translate of one human-language text to another.*
- *Generate human-language text such as fiction, manuals, and general descriptions.*
- *Interface to other systems such as databases and robotic systems thus enabling the use of human-language type commands and queries.*
- *Answer questions about the contents of the text.*
- *Understand human-language text to provide a summary or to draw conclusions.*

There are more practical goals for NLP, many related to the particular application for which it is being utilized.

## Natural Language Processing techniques

Following are the various techniques for natural language processing are:

1. Pattern Matching
2. Syntactically-driven Parsing
3. Semantic Grammars
4. Case frame instantiation
5. Robust Parsing

### 1. Pattern Matching

This technique involves interpreting input utterances as a whole, rather than building up their interpretation by combining the structure and meaning of words or other lower-level constituents. The approach is thus wholistic rather than constructive. With this approach, the interpretations are obtained by matching patterns of words against the input utterance. Associated with each pattern is an interpretation, so that the derived interpretation is the one attached to the pattern that matched.

### 2. Syntactically-driven Parsing

This technique deals with the ways that words can fit together to form higher level units such as phrases, clauses and sentences. Syntactically driven parsing means interpretation of larger groups of words are built up out of the interpretation of their syntactic constituent words or phrases. In a way this is the opposite of pattern matching as here the interpretation of the input is done as a whole.

### 3. Semantic Grammars

Natural language analysis based on semantic grammar is similar to syntactically driven parsing except that in

semantic grammar the categories used are defined semantically and syntactically. Thus, here semantic grammar is also involved. However, this technique only works properly in restricted domains. Thus, it is a technique useful only for applied natural language processing, not for general NLP.

### 4. Case frame instantiation

Case frame instantiation is one of the major parsing techniques under active research today. It has some very useful computational properties such as its recursive nature and its ability to combine bottom-up recognition of key constituents with top-down instantiation of less structured constituents.

### 5. Robust Parsing

Any natural language interface which is used in a practical application with a multitude of users must be able to handle input that is outside its grammar or expectations in various ways. Methods of robust parsing are under active investigation at the moment with the chief outstanding problem being the coordination of multiple, independent, construction-specific parsing strategies on the same input.

Q13.(a) Explain syntactic processing in detail.

MDU BCA RE 2018, 2018

OR

Define syntactic processing. How is it used and useful? Explain with examples.

MDU BCA RE 2016

OR

How is syntactic processing done? Explain the two basic top-down and bottom-up parsing techniques used for syntactic processing.

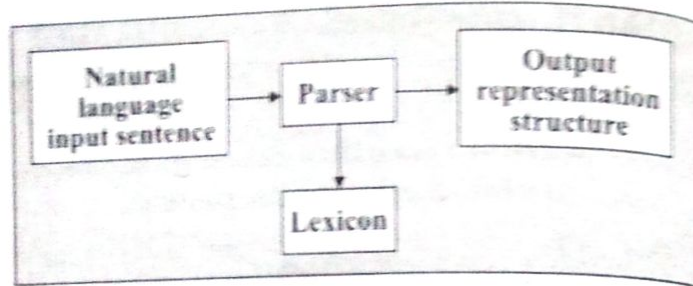
Ans. Syntax provides rules to put together words to form components of sentence and to put together these components to form sentences. Parsing is a method to perform syntactic analysis of a sentence.

To perform the syntactic processing, the knowledge of grammar and parsing technique is required. The grammar is formal specification of rules allowable in the language and parsing is a method of analyzing a sentence to determine its structure according to grammar.

Syntactic processing is done using parsing. The parser generates basic syntactic structure of the sentence. This stage uses grammatical information to perform some structural preprocessing on the input. It performs the task of applying grammar rules and computes syntactic representation of the meaning. This preprocessing stage is called parsing.

The parsing performs grouping and labeling of parts of a sentence in a way that displays their relationships to each other in a useful way. It checks the validity of a sentence. Hence a sentence, if grammatically correct and semantically incorrect will be considered as correct after first phase.

The basic parsing technique is shown below:



The parser is a computer program, which accepts the natural language sentence as input and generates an output structure suitable for analysis. The lexicon is a dictionary of words, where each word contains some syntactic, some semantic and possibly some pragmatic information.

Two basic parsing techniques are: top-down parsing and bottom-up parsing.

**Top-down parsing**

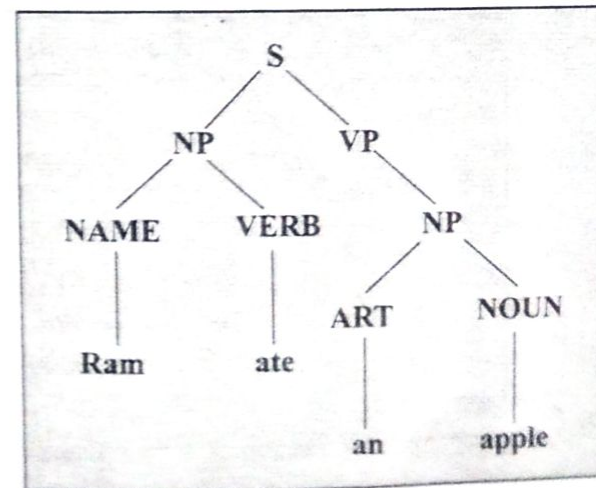
A simple parsing algorithm used is the top-down parsing algorithm. In top-down parsing, grammar rules are applied to a starting symbol (i.e. sentence) until it generates terminal symbols, and the terminal symbols are found in the object sentence. Since all non-terminal symbols are functional blocks, any non-terminal symbol may be viewed as a starting symbol. When all of the terminal symbols are found, and no unknown terminal symbols remain in the sentence, then the top-down parsing succeeds.

Thus, in top-down parsing, words of sentence are replaced by their categories like noun phrase (NP), verb phrase (VP) etc. These symbols like (NP, VP) are rewritten as per grammar rules. Finally, terminal symbols are replaced by language words.

To illustrate the concept, consider the following example: "Ram ate an apple"

This consists of a noun phrase 'Ram' and a verb phrase 'ate an apple'. The noun phrase is just a proper noun, while the verb phrase consists of a verb 'ate' and another noun phrase 'an apple'. This noun phrase consists of a determiner 'an' and a noun 'apple'. The complete top-down parse and parsing tree for this is given below:

- S → NP VP (Replace S by its linguistic constituents NPVP)
- Name VP (Replace NP by Name)
- Ram VP (Replace constituent category name by terminal value)
- Ram Verb NP (Replace VP by constituent category)
- Ram ate NP (Replace verb by terminal value)
- Ram ate ART noun (Replace NP by category)
- Ram ate an noun (Replace ART by terminal value)
- Ram ate an apple (Replace noun by terminal value)

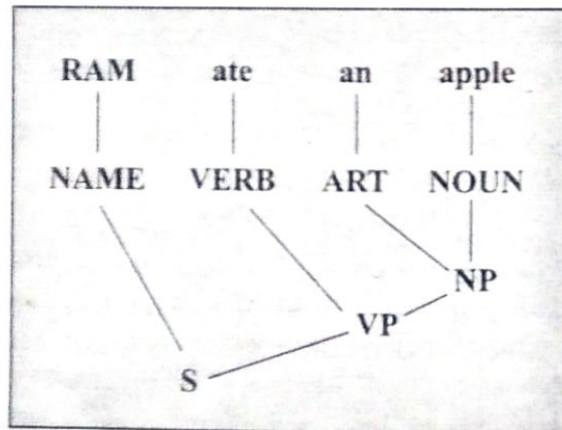


**Bottom-up parsing**

Bottom-up parsing is more general than top-down parsing. In Bottom-up parsing we start with the sentence and try to apply the production rules in reverse, in order to finish up with the start symbol of the grammar. This corresponds to starting at the leaves of the parse tree, and working back to the root. The bottom up parse for sentence "Ram ate an apple" would be as follows:

- Ram ate an apple.
- Name ate an apple. (replace Ram)
- Name verb an apple. (replace ate)
- Name verb art apple. (replace an)
- NP verb art apple. (replace Name)
- NP verb NP (replace Noun phrase)
- NP VP (replace verb NP)
- S (replace NP VP)

The parse tree of the above sentence is shown below:



The incorrect sentences will be excluded by the grammar. For example, 'Apple Ram ate' starts with two nouns, which is not allowed in the grammar. However, some odd sentences will be allowed, such as 'an apple ate Ram'. This sentence is syntactically acceptable, just semantically odd, so should still be parsed. Note that sentences, which are syntactically sound but make no sense, are acceptable by the parser. Thus, the next stage (semantic processing) would be needed to filter out these nonsensical parses.

Q12.(e) Explain pragmatic processing.  
MDU BCA RE 2018, 2018, RE 2016, 2016

Ans. Pragmatic Processing

*Pragmatic Processing is the last stage of Natural Language Processing. It refers to the contextual interpretation.*

The pragmatic component explains how utterances relate to the world. In this process, meaning of words is elaborated based on the contextual and world knowledge and a final meaning representation is obtained.

This is high-level knowledge which relates the use of sentences in different contexts and how the context affects the meaning of the sentence.

To understand the concept, consider the following sentences which might appear at the start of an AI textbook:

- This book is about artificial intelligence.
- The green frogs sleep soundly.
- Colorless green ideas sleep furiously.
- Furiously sleep ideas green colorless.

The first sentence would be quite appropriate at the start of such a book; it is syntactically, semantically, and pragmatically well formed.

The second sentence is syntactically and semantically well formed, but it would appear very strange at the start of an AI book; it is thus not pragmatically well formed for that context.

The third sentence is syntactically well formed, but it is semantically non-sensical.

The fourth sentence is syntactically ill formed; it does not make any sense - syntactically, semantically, or pragmatically.

**Q13.(b) What is Semantic Processing? What are the approaches to semantic analysis? Explain. Also differentiate between Syntactic, Semantic and Pragmatic processing.**

**Ans. Semantic Processing**

*Semantics is the study of the meaning of words and semantic analysis is the analysis used to extract meaning from utterances.*

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

For example: the sentence "There are colourless red ideas" would be rejected because colourless and red make no sense although the sentence is correct.

**Approaches to semantic analysis**

Following are the approaches to semantic analysis:

1. Syntax driven semantic analysis approach
2. Lexical semantic approach
3. Compositional semantic approach

**1. Syntax driven semantic approach**

This is the elementary approach of semantic analysis though its scope is limited. This approach assigns meaning representation to input, based on the knowledge from the lexicon (dictionary) and the grammar. The basic idea behind this approach is that the meaning of a

sentence can be composed from the meaning of its parts. This idea is not very effective because according to this principle, to find out the meaning of a sentence, the basic word meaning will play prime role but it does not include the ordering of words and relations between them in the sentence.

**2. Lexical semantic approach**

This approach makes the use of semantic grammar. Using this approach, input sentences are transformed through the domain dependent semantic rules.

Thus, it uses conceptual dependency theory. Conceptual dependency structures provide a form of linked knowledge that can be used in larger structures.

**3. Compositional semantic approach**

In compositional semantic approach, the meaning of an expression is derived from the meaning of the parts of that expression. Here the target knowledge structures constructed are typically logical expressions like the formulas of FOPL.

**Following are the differences between Syntactic, Semantic and Pragmatic processing**

Syntactic Processing	Semantic Processing	Pragmatic Processing
1. In syntactic processing, sentences are checked according to the grammar.	1. In semantic processing, sentences are checked whether they are meaningful or not.	1. In pragmatic processing, sentences are checked whether they are true or false.

Q14.(a) What is Learning? Why is learning considered as important area in AI? Explain.

Ans. Learning

*Learning is defined as the process that acquires knowledge to the system.*

A definition by Simon (1983) is one of the best:

*“Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task (or tasks drawn from a population of similar tasks) more effectively the next time.”*

One can easily extend this definition easily to the AI systems:

*“Machine learning denotes automated changes in an AI system that are adaptive in the sense that they enable the system to do the same task (or tasks drawn from a population of similar tasks) more effectively the next time.”*

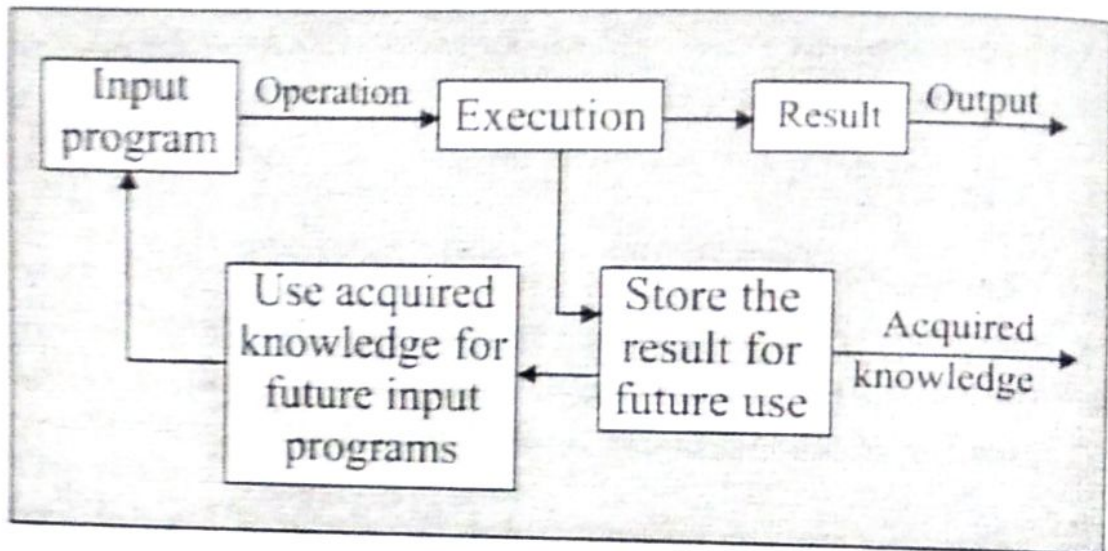
Learning provides power to handle new problems on the basis of similar problems tackled some other time.

Learning is acquiring new knowledge, behaviours, skills, values, preferences or understanding, and may involve synthesizing different types of information.

Learning is considered as an important area in AI. Machines can not be called intelligent until they are able to learn to do new things and to adapt to new situations, rather than simply doing as they are told to do. This requires knowledge acquisition, inference, updating of knowledge base, applying faster searches etc. Knowledge acquisition refers to the situation in which computer executes one program and remembers the process for

future use. Learning is important for practical applications of Artificial Intelligence.

The general model of learning is shown below:



Here the input program is any general program executed for solution of a problem. The results are reported as output, but results are also stored back to acquire knowledge for future use. It means, in case similar problem is required to be solved next time, the results can automatically be taken from previously acquired knowledge.

*In brief one can say that "Learning is an area of AI that focuses on processes of self-improvement". Information processes that improve their performance or enlarge their knowledge bases are said to have learning.*

Q14.(b) What is Rote Learning? What are its features, advantages and disadvantages? Also write the challenges / issues in the design of rote learning systems.

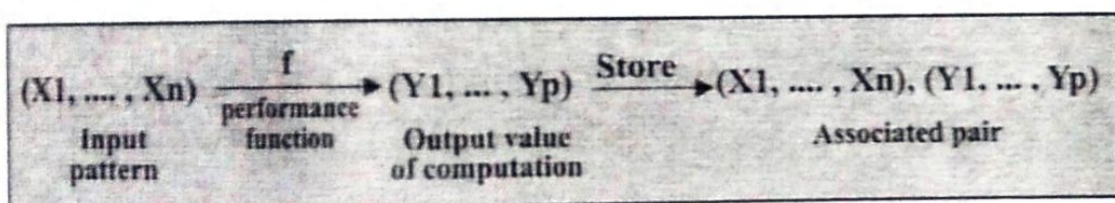
Ans. Rote Learning

*The simple memorizing of individual items like solutions to problems, words of vocabulary etc. is known as rote learning.*

The major practice involved in rote learning techniques is learning by repetition, based on the idea that one will be able to quickly recall the meaning of the material the more it is repeated.

Rote learning is the basic learning activity. It is also called memorization. Rote learning is relatively easy to implement on a computer.

*Rote learning consists of memorizing the solutions of the solved problems so that the system needs not to solve them again:*



*During subsequent computations of  $f(X_1, \dots, X_n)$ , the performance element can simply retrieve  $(Y_1, \dots, Y_p)$  from memory rather than recomputing it.*

This concept says that knowledge can be saved so it can be used again. But once knowledge is stored, how can it be retrieved. There are varieties of ways of retrieval.

A simple example of rote learning is *caching*.

- *Caching is the process which stores computed values or in other words it stores large amount of data.*
- *This information is recalled when required by computation.*
- *Significant time savings can be achieved.*
- *Many AI programs have used caching very effectively.*

One good example of this is Samuel's checkers program. This program employed two kinds of learning: rote learning and parameter adjustment. It was designed to store the chess moves played by its creator and thus used to learn the game. It learned to play checkers to such a good extent that ultimately it was able to beat its own creator. Main features of Samuel's checkers program are:

- *Samuel's program used the minimax search procedure to explore the game tree.*
- *Time constraints do not permit complete searches i.e. only few levels in the tree are permitted to be searched.*
- *It records board positions and scores at search ends.*
- *Now if the same board position arises later in the game the stored value can be recalled and the end effect is that deeper search has occurred.*

### Features of Rote learning

Rote learning has the following features:

- *The knowledge, without any modification is simply copied into the knowledge base. So, saved knowledge can be used again.*

- *This technique can save significant amount of time as computed values are stored in this learning. Therefore, no repeated computations, inference or query is necessary.*

### Advantages of Rote learning

Rote learning has the following advantages:

- *It enables machine to store organized information.*
- *Saved knowledge can be used again.*
- *It is faster to reuse information stored than to collect it again.*
- *One can get generalized information stored in the database using rote learning.*
- *No repeated computations, inference or query is necessary. Thus, it saves a lot of time.*

### Disadvantages of Rote learning

Rote learning has the following disadvantages:

- *The information and data stored would become very large, complex and hence unmanageable.*
- *It is not the best method as it is elementary in form and hence can not be used in sophisticated applications.*

### Challenges / issues in the design of rote learning systems

Rote learning is basically a simple process. However it does illustrate some issues that are relevant to more complex learning issues. These are:

**1. Generalization**

The number of potentially stored objects can be very large. Therefore, there is need to generalize some information to make the problem manageable.

**2. Stability of the Environment**

Stability of the environment means the information stored at one time should still be valid later. If the environment rapidly changes then rote learning is not very effective.

The exact change in the environment is to be detected and handled properly.

**3. Store-versus-compute trade-off**

The cost of storing and retrieving the memorized information should be smaller than the cost of recomputing it.

**4. Memory organization**

For the faster access of the stored data it would be better to recompute it but there must be a way to access the appropriate stored value quickly. Methods such as hashing, indexing and sorting can be employed to enable this.

Thus, rote learning requires useful organization of the memory so that the retrieval of the desired information will be very fast.

**Conclusion**

*Rote learning:*

- *Method of study is based on learning facts etc by heart without considering their meaning.*
- *Most trivial form of learning.*

- *Simple storing of computed information (Everything is memorized) e.g. computer simply stores a piece of data in knowledge base.*
- *Many computer programs e.g. database systems can be said to learn. The act of storage allows the program to perform better in future.*
- *There may be situations where computation is more expensive than recalling the previously done job. Thus remembering the previous would certainly help perform better. This form of learning is called rote learning.*

Q15.(a) Explain learning by taking advice with examples. MDU BCA RE 2016

OR

What is Learning by taking advice in AI?  
What are the approaches to advice taking?  
Explain.

**Ans. Learning by taking advice**

It involves receiving direct instructions on how to respond to certain situations. In a machine, this amounts to straight-forward procedural programming. In situations where the instructions do not correspond to direct procedures ( For example, "Take control of the centre of the board" in Chess), an interpreter is required to translate the instructions to concrete execution steps.

*Thus, this kind of learning takes place when a computer runs a particular program by taking advice from its programmer (or creator of that program).*

A computer can do very little without a program for it to run. When a programmer writes a series of instructions into a computer, a rudimentary kind of learning is taking place: The programmer is a sort of teacher, and the computer is a sort of student. After being programmed, the computer is now able to do something it previously could not. Executing the program may not be such a simple matter.

Suppose the program is written in a high-level language like LISP. Some interpreter or compiler must intervene to change the teacher's instructions into code that the machine can execute directly.

The idea of advice taking in AI based learning was proposed as early as 1958. However, very few attempts

were made in creating such systems until the late 1970s. Learning in the case of problems like expert systems can be done basically by advice taking. Expert systems are thus providing a major encouragement in this area.

In some instances, it is almost as difficult to give advice to a computer program as it is to human being. The major problem with the computer taking advice is its ability of understanding or interpretation. It is usually desirable to input the advice information in the form of the human's natural language – the normal language, such as English, that the human speaks and understands. However, for a computer, human natural language understanding is extremely difficult.

### Approaches to advice taking

There are two basic approaches to advice taking:

1. Rules that can guide performance elements of the system are constructed by taking high level, abstract advice. All aspects of advice taking are automated by:

#### (a) Request

This can be simple question asking about general advice or more complicated by identifying shortcomings in the knowledge base and asking for a remedy.

#### (b) Operationalise

Translated advice may still not be usable so this stage seeks to provide a representation that can be used by the performance element.

#### (c) Interpret

Translate the advice into an internal representation.

Q15.(b) Explain learning in problem solving.

MDU BCA RE 2018

OR

Explain role of learning in problem solving with examples.

MDU BCA RE 2016

OR

Explain Learning in Problem Solving with examples.

MDU BCA 2016

**Ans.** A problem can be solved by experience. Efficiency of solving a similar process increases by experience. This does not usually involve gathering new knowledge but may involve reorganization of data or remembering how to achieve to solution.

Thus, learning in problem solving involves:

- *Learning ways of problem solving from own experience – without an instructor/advisor.*
- *Does not involve an increase in knowledge, just the methods in using the knowledge.*
- *Learnt rules that may prove good in directing the problem solving process, but it incurs a cost (utility) because the problem solver needs to store and consult those rules.*

### Types of Learning in Problem Solving

There are three basic methods in which a system can learn from its own experience:

1. Learning by Parameter Adjustment
2. Learning by Macro Operators
3. Learning by Chunking

### 1. Learning by Parameter Adjustment

In many programs the state of the search process depends on the evaluation procedure used. A very common example of this is game playing. Many have static evaluation function. The evaluation function is represented as a polynomial of the form:

$$c_1t_1 + c_2t_2 + c_3t_3 + \dots$$

The  $t$  terms are values of features and the  $c$  terms are weights.

For example, the factors such as demand and production capacity may be combined into a single score to indicate the chance for increase of production. But it is difficult to know a priori how much weight should be attached to each factor.

The correct weight can be found by taking some estimate of the correct settings and then allowing the program to modify its settings based on its experience. This type of learning system is useful when little knowledge is available.

### 2. Learning by Macro-operators

The basic idea here is similar to Rote learning. It avoids expensive recomputation. Macro-operators can be used to group a whole series of actions into one.

In this learning:

- *Rote learning as a sequence of operations is found to be useful during problem solving.*
- *Sequence of actions that can be treated as a whole are called macro-operations.*
- *Once a problem is solved, the learning component takes the computed plan and stores it as a macro-*

*operator. The preconditions are the initial conditions of the problem just solved and its post conditions correspond to the goal just achieved.*

- *The problem solver efficiently uses the knowledge base it gained from its previous experiences.*
- *By generalizing macro-operators the problem solver can even solve different problems.*

### 3. Learning by Chunking

Chunking is a process where a sequence of actions are put together and treated as a whole to perform a particular task.

Chunking is similar to learning with macro-operators. Generally, it is used by problem solver systems that make use of production systems.

A production system consists of a set of rules that are in If - THEN form. SOAR is a system that uses production rules to represent its knowledge. It also employs chunking to learn from experience. SOAR solves problems. It fires productions. These productions are stored in long term memory. When SOAR detects a useful sequence of firing, it creates chunks. Chunks may be generalized before storing.

Q16.(a) Explain learning from example – Induction.

Ans. Induction Learning

*Inductive learning means, generalization of knowledge gathered from real world examples and use of the same for solving other similar problems. In simple words, learning by induction or inductive learning is a method which generalizes the rules of past.*

Inductive learning involves the process of learning by example, where a system tries to induce a general rule from a set of observed instances i.e. Induction involves drawing general conclusions from specific observations. Learning from example and learning by discovery are instances of inductive learning. Most of the knowledge that the human has acquired about its environment is through the process of induction.

For example, observing the specific facts that Sun rises in the east on day 1, day 2, ... we infer a general statement, namely, Sun always rises in the east.

*Inductive learning methods can be defined as those methods that systematically produce general descriptions or knowledge from the specific knowledge provided by domain examples.*

The learning process which belongs to supervised learning, does classification and constructs class definitions is called induction or concept learning.

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

1. Winston's Learning Program
2. Version Spaces
3. Decision Trees

### 1. Winston's Learning Program

It is a structural concept of learning program. This program was operated in a simple block world domain. Its goal was to construct representations of the definitions of concepts in the block domain.

**Example: Concepts such as "house".**

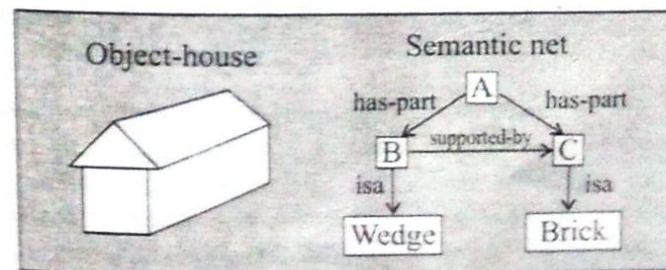
Start with input, a line drawing of a blocks world structure. It learned concepts House, Tent, Arch as:

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

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

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

An example of such a structure for the house is shown below:



Node A represents entire structure, which is composed of two parts: Node B, a Wedge and node C, a Brick.

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

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

1. Select one known instance of the concept. Call this the concept definition.
2. Examine definition of other known instance of the concept. Generalize the definition to include them.
3. Examine description of near misses. Restrict the definition to exclude these.

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

## 2. Version Spaces

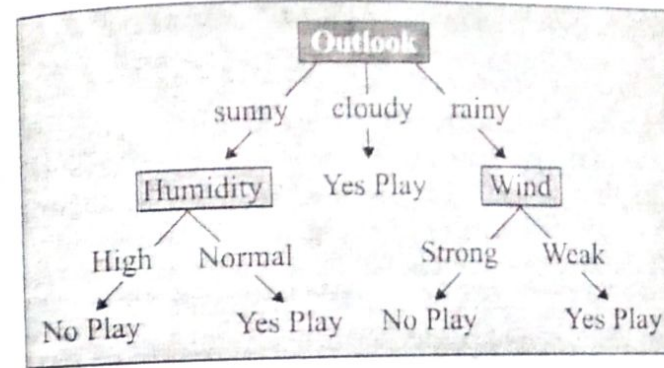
A version space is a representation that is used to get relevant information from a set of learning examples. It is a hierarchical representation of knowledge that keeps track of all the useful information supplied by a sequence of learning examples without remembering any of the examples.

## 3. Decision Trees

Another approach to concept learning is the induction of decision tree as exemplified by the Quinlan in his ID3 system introduced the idea of decision trees.

ID3 is a program that can build trees automatically from given positive and negative instances. Basically each leaf of a decision tree asserts a positive or negative concept. To classify a particular input one starts at the top and follows assertions down until answer is reached.

The classification tree built using ID3 algorithm is shown below. It tells if the weather was amenable to play?



Decision trees are useful when the concept space is very large, decision tree learning algorithm runs more quickly than version spaces. On the other side, large, complex decision trees can be difficult for human to understand and so a decision tree system may find it hard to explain its classification.

Q16.(c) What is learning? Discuss different types of learning with example.

MDU BCA RE 2017, 2017

OR

Explain various learning methods.

MDU BCA 2015

Ans. Learning

*Learning is defined as the process that acquires knowledge to the system.*

Learning is done by viewing, listening, interactions, studying and by experience. Learning provides us the power to reason, ability to handle new situations and enables to act in an intelligent way. Human beings are intelligent as they possess knowledge of world. Similarly, making a machine intelligent means it should have the power of learning. So, machine learning is a challenge now. An intelligent machine should be able to learn new things and to adapt to new situation rather than simply doing steps as they are made to do so.

A definition by Simon (1983) is one of the best:

*“Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task (or tasks drawn from a population of similar tasks) more effectively the next time.”*

One can easily extend this definition easily to the AI systems:

*“Machine learning denotes automated changes in an AI system that are adaptive in the sense that they enable the system to do the same task (or tasks drawn from a population of similar tasks) more effectively the next time.”*

Learning provides power to handle new problems on the basis of similar problems tackled some other time. Learning is acquiring new knowledge, behaviours, skills, values, preferences or understanding, and may involve synthesizing different types of information.

Learning is considered as an important area in AI. Machines cannot be called intelligent until they are able to learn to do new things and to adapt to new situations, rather than simply doing as they are told to do. This requires knowledge acquisition, inference, updating of knowledge base, applying faster searches etc. Knowledge acquisition refers to the situation in which computer executes one program and remembers the process for future use. Learning is important for practical applications of Artificial Intelligence.

### Types of learning

There are five main methods of learning. These are:

1. Rote Learning
2. Learning by taking advice
3. Learning by induction
4. Learning in Problem Solving
5. Explanation Based Learning

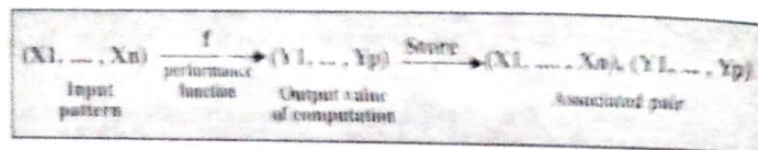
#### 1. Rote Learning

*The simple memorizing of individual items like solutions to problems, words of vocabulary etc. is known as rote learning.*

The major practice involved in rote learning techniques is learning by repetition, based on the idea that one will be able to quickly recall the meaning of the material the more it is repeated.

Rote learning is the basic learning activity. It is also called memorization. Rote learning is relatively easy to implement on a computer.

Rote learning consists of memorizing the solutions of the solved problems so that the system needs not to solve them again:



During subsequent computations of  $f(X_1, \dots, X_n)$ , the performance element can simply retrieve  $(Y_1, \dots, Y_p)$  from memory rather than recomputing it.

This concept says that knowledge can be saved so it can be used again. But once knowledge is stored, how can it be retrieved. There are varieties of ways of retrieval.

A simple example of rote learning is *caching*.

- *Caching is the process which stores computed values or in other words it stores large amount of data.*
- *This information is recalled when required by computation.*
- *Significant time savings can be achieved.*
- *Many AI programs have used caching very effectively.*

### Features of Rote learning:

Rote learning has the following features:-

- *The knowledge, without any modification is simply copied into the knowledge base. So, saved knowledge can be used again.*

- *This technique can save significant amount of time as computed values are stored in this learning. Therefore, no repeated computations, inference or query is necessary.*

### Advantages of Rote learning

Rote learning has the following advantages:

- *It enables machine to store organized information.*
- *Saved knowledge can be used again.*
- *It is faster to reuse information stored than to collect it again.*
- *One can get generalized information stored in the database using rote learning.*
- *No repeated computations, inference or query is necessary. Thus, it saves a lot of time.*

### Disadvantages of Rote learning:

Rote learning has the following disadvantages:-

- *The information and data stored would become very large, complex and hence unmanageable.*
- *It is not the best method as it is elementary in form and hence can not be used in sophisticated applications.*

### 2. Learning by taking advice

It involves receiving direct instructions on how to respond to certain situations. In a machine, this amounts to straight-forward procedural programming. In situations where the instructions do not correspond to direct procedures (For example, "Take control of the centre of the board" in Chess), an interpreter is required to translate the instructions to concrete execution.

*Thus, this kind of learning takes place when a computer runs a particular program by taking advice from its programmer (or creator of that program).*

### 3. Learning by Induction

*Inductive learning means, generalization of knowledge gathered from real world examples and use of the same for solving other similar problems. In simple words, learning by induction or inductive learning is a method which generalizes the rules of past.*

Inductive learning involves the process of learning by example, where a system tries to induce a general rule from a set of observed instances i.e. Induction involves drawing general conclusions from specific observations. Learning from example and learning by discovery are instances of inductive learning. Most of the knowledge that the human has acquired about its environment is through the process of induction.

For example, observing the specific facts that Sun rises in the east on day 1, day 2, ... we infer a general statement, namely, Sun always rises in the east.

*Inductive learning methods can be defined as those methods that systematically produce general descriptions or knowledge from the specific knowledge provided by domain examples.*

The learning process which belongs to supervised learning, does classification and constructs class definitions is called induction or concept learning.

### 4. Learning in Problem Solving

A problem can be solved by experience. Efficiency of solving a similar process increases by experience. This does not usually involve gathering new knowledge but

may involve reorganization of data or remembering how to achieve to solution.

Thus, learning in problem solving involves:

- *Learning ways of problem solving from own experience – without an instructor/advisor.*
- *Does not involve an increase in knowledge, just the methods in using the knowledge.*
- *Learnt rules that may prove good in directing the problem solving process, but it incurs a cost (utility) because the problem solver needs to store and consult those rules.*

There are three basic methods in which a system can learn from its own experience:

- Learning by Parameter Adjustment
- Learning by Macro Operators
- Learning by Chunking

### 5. Explanation Based Learning

An Explanation Based Learning (EBL) system accepts an example (i.e. a training example) and explains what it learns from the example. The EBL system takes only the relevant aspects of the training. This explanation is translated into particular form that a problem solving program can understand. The explanation is generalized so that it can be used to solve other problems.

Explanation-Based Learning (EBL) techniques are knowledge-based and require a great deal of knowledge about the domain of the concept to be learned. Instead of providing several instances of a concept, only a single example is required for a new concept to be learned.

Q17.(a) What is an Expert System (ES)? What are the different applications of expert system?

MDU BCA RE 2018, 2018, 2015

Ans. Expert System

*Expert Systems are computer programs that are derived from a branch of computer science research called Artificial Intelligence (AI). Expert System is an information system that is capable of mimicking human thinking and making considerations during the process of decision-making.*

AI programs that achieve expert-level competence in solving problems in task areas by bringing to bear a body of knowledge about specific tasks are called Expert Systems.

An expert system refers to a computer system which exhibits the human expert's intelligence. An expert system handles real-world problems requiring the expert's involvement, uses a computer model of expert knowledge and expert reasoning.

Prof. Edward from Stanford University, a famous researcher on ES defines ES as:

*"... an intelligent computer programme that uses knowledge and reasoning procedures to solve different problems that need certain expertise to solve the problems."*

Expert system is developed to model the ability of an expertise in solving problems. Expert system must be able to provide users with the services and facilities that an expert can usually provide.

Thus expert systems are computer programs which store the factual and inferential knowledge of human experts in

a narrow domain. An expert system is an AI program which uses knowledge to solve problems which would normally require a human expert. It attempts to reproduce the performance of one or more human experts.

### Application of Expert Systems

The range of applications of expert systems is very vast. It includes areas such as:

1. Medical Diagnosis
2. Prediction
3. Accounting and Finance
4. Designing
5. Planning
6. Debugging
7. Scheduling
8. Assessment
9. Marketing
10. Manufacturing
11. Monitoring and Control
12. Training

#### 1. Medical Diagnosis

Different types of medical diagnostic packages are used for different types of medical diagnoses. These determine the cause of malfunctions in complex situations based on observable symptoms.

e.g. MYCIN – used to diagnose infectious blood diseases.

HELP – used to diagnose heart problems.

PUFF – measures lung functions.

## 2. Prediction

Another application of expert system is to predict probable consequences of a given situation. For example, forecasting crop damage etc.

## 3. Accounting and Finance

In the field of accounting and finance the expert system helps to provide tax advice, to select forecasting models, to credit authorization decisions etc.

## 4. Designing

Expert system proves to be very useful for determining the configuration of system components to meet the given specification.

e.g. The design of VLSI system.

Thus the system helps to prepare design of the plan and even recalculate before actually manufacturing the object.

## 5. Planning

Expert systems are:

- *Helpful in a number of applications related to space planning and explorations.*
- *Helps in planning curricula for students.*
- *Useful in planning educational experiments in biology, chemistry and molecular genetics etc.*

## 6. Debugging

Debugging is another beneficial application of expert system. It involves prescribing and implementing remedies for malfunctions.

For example, location of faults in computer or communication system.

## 7. Scheduling

Expert system aids in scheduling of:

- *Customer orders.*
- *Computer resources for operating systems.*
- *Production operations.*
- *Various manufacturing tasks.*

## 8. Assessment

Assessment is very important for effective decisions. The assessment of geological structures can be made with the usage of expert system.

On the basis of an area's geological formation an expert system gives possible location of minerals etc.

e.g. Geology packages like PROSPECTOR.

Also applicable in numerous military applications like battlefield assessment etc.

## 9. Marketing

Expert system is used in the field of marketing in:

- *Establishing sales quotas.*
- *Responding to customer inquiries.*
- *Determining rebate policies.*

## 10. Manufacturing

Expert system is effectively used in manufacturing.

It helps in:

Q17.(c) What do you mean by Expert System? Write and explain the components of Expert System.

Ans. Expert System

*An expert system is a software package that accumulates the knowledge and decision making capabilities of the specialists in a particular field.*

OR

*An expert system is a set of programs that manipulate encoded knowledge to solve problems in a specialized domain that normally requires human expertise.*

OR

*An expert system is a rule-based artificial intelligence application program that performs a task that requires expert knowledge.*

The expert system responds as an intelligent assistant by giving advice and suggesting possible decisions. It is designed to act as an expert in a particular domain.

Thus an expert system is an information system that captures and stores the knowledge of human experts.

These expert systems represent the expertise knowledge as data or rules within the computer. These rules and data can be called upon when needed to solve problems. Expert systems are also called as "Knowledge based system".

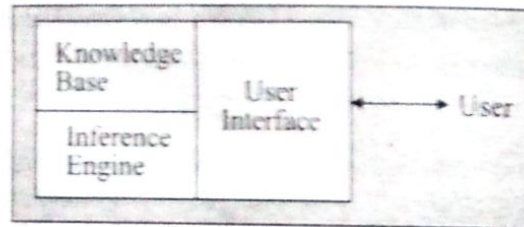
Expert systems thus use the decision rules of human specialists to arrive at certain conclusions and to give recommendations. Expert systems are computer programs which store the factual and inferential knowledge of human experts in a narrow domain. An

expert system is an AI program which uses knowledge to solve problems which would normally require a human expert. It attempts to reproduce the performance of one or more human experts.

### Components of an Expert System

Components of an expert system can be classified as:

1. The User Interface
2. The Knowledge Base
3. The Inference Engine



#### 1. The User Interface

This component connects the user to the expert system. The user interface is the means of communication between a user and the expert system's problem-solving processes. It allows the user to enter instructions and information into expert system and to receive information from it.

A good expert system is not very useful unless it has an effective interface. It has to be able to accept the queries or instructions in a form that the user enters and translate them into working instructions for the rest of the system.

It also has to be able to translate the answers, produced by the system, into a form that the user can understand. Careful attention should be given to the screen design in

order to make the expert system appear 'friendly' to the user.

#### 2. The Knowledge Base

It is the most important component of expert system. It is because of this that expert system is called knowledge based system.

The knowledge base stores all the facts and rules about a particular problem domain. It makes these available to the inference engine in a form that it can use.

The facts may be in the form of background information built into the system or facts that are input by the user during a consultation. The rules include both the production rules that apply to the domain of the expert system and the heuristics or rules-of-thumb that are provided by the domain expert in order to make the system find solutions more efficiently by taking short cuts.

Knowledge bases are large systems of "if then" statements or may contain only associative relationships among different concepts, or simply large databases of facts that can be compared to one another based on simple conventions intrinsic to the expert system.

Thus, knowledge base contains the domain knowledge which is used by the inference engine to draw conclusions. The inference engine is the generic control mechanism that applies the axiomatic knowledge to the task-specific data to arrive at some conclusion. When a user supplies facts or relevant information of query to the expert system he receives advice or expertise in response. That is given the facts it uses the inference engine which in turn uses the knowledge base to infer the solution.

### 3. The Inference Engine

The inference engine tells how the rules are to be applied in knowledge base to solve the problem. The inference engine is the program that locates the appropriate knowledge in the knowledge base and infers new knowledge by applying logical processing and problem-solving strategies.

Thus the major task of inference engine is to trace its way through a large number of rules to arrive at a solution. Inference engine may be termed as the brain of expert systems.

Q18.(a) Explain the advantages of an expert system.  
MDU BCA RE 2016

OR

What are the advantages of Expert Systems (ES)? Also write and explain the disadvantages and weaknesses of Expert System.

#### Ans. Advantages of Expert System

Expert systems have many attractive features. Some of the advantages of using expert system are the following:

1. Consistency
2. Training Tool for Trainees
3. Increased availability
4. Reduced Danger
5. Combination of Knowledge and Expertise from Various Sources
6. More efficient and cost effective
7. Permanence
8. Documentation
9. Explanation
10. Fast response

#### 1. Consistency

One of the advantages of an ES is that the results given are consistent. This might be due to the fact that:

- *Similar transactions are handled in the same way.*
- *The system makes comparable recommendations for like situations.*

- *There are no elements such as exhaustion and emotions as experienced by humans.*

## 2. Training Tool for Trainees

An expert system can be used for training future human experts. An ES can be used by trainees to learn about the knowledge-based system. Trainee who uses an ES would be able to observe how an expert solves a problem.

## 3. Increased availability

Expert systems are available easily due to mass production of expertise. One expert system can be duplicated, at very little cost, to yield as many copies as are required.

## 4. Reduced Danger

Expert systems can be used in environments that might be harmful for humans. They can be used in any risky environments where humans cannot work with. Through an ES, we can avoid exposing ourselves to a toxic or radioactive environment. An ES can take over the place of an expert to handle problems in a high-risk area such as a nuclear power plant.

## 5. Combination of Knowledge and Expertise from Various Sources

One of the important components in an ES is the knowledge base. This component contains the accumulated knowledge and acquired or transferred expertise from many experts. Thus, an ES is sometimes more superior than an expert because its knowledge and expertise have come from many sources. The combined level of expertise of several experts may exceed that of a single human expert.

## 6. More efficient and cost effective

Expert systems could be more efficient and cost effective than human systems, and will become increasingly so as wages of human professionals rise.

## 7. Permanence

The expertise of the expert system is permanent. Human expertise can quickly fade. An expert must constantly practice and rehearse to maintain proficiency in some problem area. Once expertise is acquired in an expert system, it is around forever. Unlike human experts, who may retire, quit or die, an expert system's knowledge will last indefinitely.

## 8. Documentation

An expert system is much easier to document. Documenting human expertise is extremely difficult and time consuming. Documenting an expert system is a straight forward mapping between the ways in that the expertise is represented in the system and the natural language description of that representation. Thus, an expert system can provide permanent documentation of the decision process.

## 9. Explanation

Expert systems can explicitly explain the reasoning that led to a conclusion. Thus, one can see how the expert system used rules from its knowledge base to convert the input into advice. The ability to reason increases one's confidence that the correct decision was made.

## 10. Fast response

Fast or real-time response may be necessary for some applications. Depending on the software and hardware

used, an expert system may respond faster and be more available than a human expert. Some emergencies may require faster responses than a human expert's, and so a real-time expert system is a good choice.

### Disadvantages of Expert System

Some of the disadvantages of using expert system are the following:

1. Not Widely used
2. Common Sense
3. Difficult to use
4. Learning
5. Probable Decision Error
6. Difficult to Maintain
7. Creativity
8. Degradation

#### 1. Not Widely used

ES is not widely used in business firms or organizations. Due to limited usage, firms are still in doubt about the capability and, most definitely, the high cost involved in an ES.

#### 2. Common Sense

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

#### 3. Difficult to use

Using an ES is very difficult and learning and mastering it requires a long time. Moreover, developing an ES that is user-friendly is the biggest challenge for ES developers.

#### 4. Learning

Human experts automatically adapt to changing environments, expert systems must be explicitly updated.

#### 5. Probable Decision Error

The main source of the knowledge is experts. Humans make mistakes. If the experts input wrong information into the expert system, this will have a negative impact on the results produced.

#### 6. Difficult to Maintain

The information in ES must be constantly updated to solve new problems. Every new problem that occurs needs new knowledge and expertise. This means that there must be an on-going relationship between the domain experts and the ES developer. This situation requires the domain experts to update the source of knowledge and expertise to suit the current situation.

#### 7. Creativity

Human experts can respond creatively to unusual situations, expert systems cannot.

#### 8. Degradation

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

Q21.(a) Explain in detail the expert system shells.

MDU BCA RE 2018, 2018

OR

What is an expert system shell? Explain why the use of an expert system shell can dramatically reduce the development time of an expert system.

Ans. Expert System Shell

*An expert system shell is a skeleton expert system with the knowledge removed. To build a new expert system application, all the user has to do is to add the knowledge in the form of rules and provide relevant data.*

Shells are tools for building expert systems that provide knowledge representation facilities and inferencing mechanisms i.e. Expert system shell provides all the tools necessary, including an inference engine, to manage a knowledge base. In addition, it also provides a module where the user can create the knowledge base.

Expert system shell provides user friendly software environment to the knowledge engineer for building an expert system. It contains all the generic expert system logic required to build an expert system.

A shell may be thought of as an expert system with all domain specific knowledge removed and a facility for entering a new knowledge base provided.

An expert system shell can greatly facilitate development, but this is often at the expense of flexibility. The developer is limited to using the knowledge representation mechanisms, inference techniques, ways to handle uncertainty and user interfaces that the shell

provides. This can result in trying to fit the problem to the shell, rather than customizing the software to fit the problem. A plethora of expert system shells exists in the market today.

*Thus, an expert system shell is a development system for designing an expert system. It provides an efficient and user-friendly software environment to the knowledge engineer for building an expert system.*

### Functions of Expert System Shell

An expert system shell is a programme used to develop an Expert System. The Expert System Shell executes three main functions:

- *Helps the programmer build a knowledge database by permitting the developer to input knowledge in the knowledge representation structure.*
- *Provides the procedures for inference or reasoning deductions based on the information stored in the information database and new facts input by the user.*
- *Provides the interface to let the user prepare reasoning tasks and questions to be queried to the system on strategic reasoning.*

Expert system shell provides a wide range of facilities and advantages, that's why expert system shell offers a dramatic reduction in the development time of expert systems.

Expert system shell provides the following facilities:

- *A knowledge representation language.*
- *Knowledge base editor.*
- *Tracing and debugging facilities.*

- *User interface facilities.*
- *Facilities for uncertain reasoning.*
- *Rule induction facilities.*
- *Good debugging and value-checking aids.*
- *A mechanism for handling uncertainty both from the developer and from the end user.*

Expert system shell provides the following advantages:

- *The programming efforts that go into building the user interface and inference engine is reused.*
- *The level of programming skills needed to produce the finished system is much lower compared to developing a system from scratch using conventional programming methods.*
- *Expert system development project can be completed faster, cheaper and in more efficient way.*