Biyani's Think Tank

Concept based notes

AI

MCA

Ms Sonia Bhargawa
Deptt. of IT
Biyani Girls College, Jaipur

Biyani's
Group of Girls' Colleges

Published by :
Think Tanks
Biyani Group of Colleges

Sector-3, Vidhyadhar Nagar,
Jaipur-302 023 (Rajasthan)
Ph : 0141-2338371, 2338591-95 • Fax : 0141-2338007
E-mail : acad@biyanicolleges.org
Website :www.gurukpo.com; www.biyanicolleges.org

Edition : 2012

Leaser Type Setted by :
Biyani College Printing Department

Preface

I am glad to present this book, especially designed to serve the needs of the students. The book has been written keeping in mind the general weakness in understanding the fundamental concepts of the topics. The book is self-explanatory and adopts the "Teach Yourself" style. It is based on question-answer pattern. The language of book is quite easy and understandable based on scientific approach.

Any further improvement in the contents of the book by making corrections, omission and inclusion is keen to be achieved based on suggestions from the readers for which the author shall be obliged.

I acknowledge special thanks to Mr. Rajeev Biyani, *Chairman* & Dr. Sanjay Biyani, *Director* (*Acad.*) Biyani Group of Colleges, who are the backbones and main concept provider and also have been constant source of motivation throughout this endeavour. They played an active role in coordinating the various stages of this endeavour and spearheaded the publishing work.

I look forward to receiving valuable suggestions from professors of various educational institutions, other faculty members and students for improvement of the quality of the book. The reader may feel free to send in their comments and suggestions to the under mentioned address.

Author

MCA-404 Artificial Intelligence

Chapter 1 :Problem solving & Search techniques

Concept of intelligence, Artificial intelligence, definition turning test, areas of application.Search techniques, state space, Production rules, problem characteristics, production systemcharacteristic, depth first, breadth first search methods and their analysis, Heuristic searchmethod, generate and test, hill climbing, best first method, graph search, AND OR search methods, constraint satisfaction, backtracking.

Chapter2: Knowledge representation

Introduction to list and string processing and dynamic databases concept of knowledge,characteristics and representation schemes, Logic, prepositional and predicate calculus,resolution, semantics nets, frames, conceptual dependency, scripts Monotonic reasoning, logical reasoning induction, natural deduction.

Chapter 3:Nonmonotonic reasoning

Nonmonotonic reasoning – default reasoning minimalist reasoning, statistical reasoning –Baye's theorem, certainty factors, dempster shafer theory, Fuzzy logic.Concept of learning, inductive and deductive. Knowledge acquisition, rote learning, discovery analogy.

Chapter4:Expert System

Concept of expert system, need for an expert system, Component and categories of an expertsystem, need for an expert system, Stages in the development of an expert system.

Chapter 1
Problem solving & Search techniques

Q.1 What is artificial intelligence?

Ans It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence.

``The exciting new effort to make computers think ... *machines with minds*, in the full and literal sense'' (Haugeland, 1985)

``The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...'' (Bellman, 1978)

``The study of mental faculties through the use of computational models'' (Charniak and McDermott, 1985)

``The study of the computations that make it possible to perceive, reason, and act'' (Winston, 1992)

``The art of creating machines that perform functions that require intelligence when performed by people'' (Kurzweil, 1990)

``The study of how to make computers do things at which, at the moment, people are better'' (Rich and Knight, 1991)

``A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes'' (Schalkoff, 1990)

``The branch of computer science that is concerned with the automation of intelligent behavior'' (Luger and Stubblefield, 1993

Q.2 What are the applications of AI?

Ans. The applications of AI are:

game playing

You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and

known reliable heuristics requires being able to look at 200 million positions per second.

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

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

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

Expert systems
A ``knowledge engineer'' interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better than medical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death,

recovery, and events occurring in time. Its interactions depended on a single patient being considered. Since the experts consulted by the knowledge engineers knew about patients, doctors, death, recovery, etc., it is clear that the knowledge engineers forced what the experts told them into a predetermined framework. In the present state of AI, this has to be true. The usefulness of current expert systems depends on their users having common sense.

Heuristic classification

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

Q.3 What is the Turing test?

Ans Alan Turing's 1950 article *Computing Machinery and Intelligence* discussed conditions for considering a machine to be intelligent. He argued that if the machine could successfully pretend to be human to a knowledgeable observer then you certainly should consider it intelligent. This test would satisfy most people but not all philosophers. The observer could interact with the machine and a human by teletype (to avoid requiring that the machine imitate the appearance or voice of the person), and the human would try to persuade the observer that it was human and the machine would try to fool the observer.

The Turing test is a one-sided test. A machine that passes the test should certainly be considered intelligent, but a machine could still be considered intelligent without knowing enough about humans to imitate a human.

Q 4 What are the steps we need to take in solving a problem?

Ans Search in Artificial Intelligence

Search plays a major role in solving many Artificial Intelligence (AI) problems. Search is a universal problem-solving mechanism in AI. In

many problems, sequence of steps required to solve is not known in advance but must be determined by systematic trial-and-error exploration of alternatives.

There are four things that are followed in order to solve a problem:

1) Define a problem: A precise definition that must include precise specifications of what the initial situation will be as what final constitute acceptable solutions to the problem.
2) Problem must be analysed: Some of the important features land up having an immense impact on the appropriateness of various possible techniques for solving the problem.
3) Isolate & represent the task knowledge that is necessary to solve a problem.
4) Amongst the available ones choose the best problem solving techniques & apply the same to the particular problem.

Q.5 What are different search algorithms?

Ans . Brute-Force Approach

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

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

Generate-And-Test

Breadth-First Search

Generate-And-Test Algorithm

Generate-and-test search algorithm is a very simple algorithm that guarantees to find a solution if done systematically and there exists a solution.

Algorithm: Generate-And-Test

1.Generate a possible solution.

2.Test to see if this is the expected solution.

3.If the solution has been found quit else go to step 1.

Potential solutions that need to be generated vary depending on the kinds of problems. For some problems the possible solutions may be particular points in the problem space and for some problems, paths from the start state.

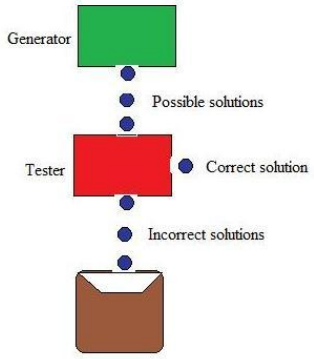


Figure: Generate And Test

Generate-and-test, like depth-first search, requires that complete solutions be generated for testing. In its most systematic form, it is only an exhaustive search of the problem space. Solutions can also be generated randomly but solution is not guaranteed. This approach is what is known as British Museum algorithm: finding an object in the British Museum by wandering randomly.

Q.6 Explain breadth first search & depth first search.

Ans **Breadth First Search**

Breadth First Search (BFS) searches breadth-wise in the problem space. Breadth-First search is like traversing a tree where each node is a state which may a be a potential candidate for solution. Breadth first search expands nodes from the root of the tree and then generates one level of the tree at a time until a solution is found. It is very easily implemented by maintaining a queue of nodes. Initially the queue contains just the root. In each iteration, node at the head of the queue is removed and then expanded. The generated child nodes are then added to the tail of the queue.

Algorithm: Breadth-First Search

Create a variable called NODE-LIST and set it to the initial state.
Loop until the goal state is found or NODE-LIST is empty.
Remove the first element, say E, from the NODE-LIST. If NODE-LIST was empty then quit.
b) For each way that each rule can match the state described in E do:
i) Apply the rule to generate a new state.
ii) If the new state is the goal state, quit and return this state.
Iii) Otherwise add this state to the end of NODE-LIST

Since it never generates a node in the tree until all the nodes at shallower levels have been generated, breadth-first search always finds a shortest path to a goal. Since each node can be generated in constant time, the amount of time used by Breadth first search is proportional to the number of nodes generated, which is a function of the branching factor b and the solution d. Since the number of nodes at level d is b^d, the total number of nodes generated in the worst case is $b + b^2 + b^3 + \ldots + b^d$ i.e. $O(b^d)$, the asymptotic time complexity of breadth first search.

Advantages of Breadth-First Search

Breadth first search will never get trapped exploring the useless path forever.
If there is a solution, BFS will definitely find it out.
If there is more than one solution then BFS can find the minimal one that requires less number of steps.

Disadvantages of Breadth-First Search

The main drawback of Breadth first search is its memory requirement. Since each level of the tree must be saved in order to generate the next level, and the amount of memory is proportional to the number of nodes stored, the space complexity of BFS is $O(b^d)$. As a result, BFS is severely space-bound in practice so will exhaust the memory available on typical computers in a matter of minutes.
If the solution is farther away from the root, breath first search will consume lot of time.

Breadth First Search (BFS) searches breadth-wise in the problem space. Breadth-First search is like traversing a tree where each node is a state which may a be a potential candidate for solution. Breadth first search expands nodes from the root of the tree and then generates one level of the tree at a time until a solution is found. It is very easily implemented by maintaining a queue of nodes. Initially the queue contains just the root. In each iteration, node at the head of the queue is removed and then expanded. The generated child nodes are then added to the tail of the queue.

Algorithm: Breadth-First Search

Create a variable called NODE-LIST and set it to the initial state.
Loop until the goal state is found or NODE-LIST is empty.
Remove the first element, say E, from the NODE-LIST. If NODE-LIST was empty then quit.
b) For each way that each rule can match the state described in E do:
i) Apply the rule to generate a new state.
ii) If the new state is the goal state, quit and return this state.
Iii) Otherwise add this state to the end of NODE-LIST
Since it never generates a node in the tree until all the nodes at shallower levels have been generated, breadth-first search always finds a shortest path to a goal. Since each node can be generated in constant time, the amount of time used by Breadth first search is proportional to the number of nodes generated, which is a function of the branching factor b and the solution d. Since the number of nodes at level d is b^d, the total number of nodes generated in the worst case is $b + b^2 + b^3 + \ldots + b^d$ i.e. $O(b^d)$, the asymptotic time complexity of breadth first search.

Advantages of Breadth-First Search

Breadth first search will never get trapped exploring the useless path forever.
If there is a solution, BFS will definitely find it out.
If there is more than one solution then BFS can find the minimal one that requires less number of steps.

Disadvantages of Breadth-First Search

The main drawback of Breadth first search is its memory requirement. Since each level of the tree must be saved in order to generate the next level, and the amount of memory is proportional to the number of nodes stored, the space complexity of BFS is $O(b^d)$. As a result, BFS is severely space-bound in practice so will exhaust the memory available on typical computers in a matter of minutes.

If the solution is farther away from the root, breath first search will consume lot of time.

Q .7 What is Heuristic Search?

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

Q .8 Explain A* Algorithm

Ans: The A* algorithm combines features of uniform-cost search and pure heuristic search to efficiently compute optimal solutions. A* algorithm is a best-first search algorithm in which the cost associated with a node is f(n) = g(n) + h(n), where g(n) is the cost of the path from the initial state to node n and h(n) is the heuristic estimate or the cost or a path from node n to a goal. Thus, f(n) estimates the lowest total cost of any solution path going through node n. At each point a node with lowest f value is chosen for expansion. Ties among nodes of equal f value should be broken in favour of nodes with lower h values. The algorithm terminates when a goal is chosen for expansion.

A* algorithm guides an optimal path to a goal if the heuristic function h(n) is admissible, meaning it never overestimates actual cost. For example,

since airline distance never overestimates actual highway distance, and manhatten distance never overestimates actual moves in the gliding tile.
For Puzzle, A* algorithm, using these evaluation functions, can find optimal solutions to these problems. In addition, A* makes the most efficient use of the given heuristic function in the following sense: among all shortest-path algorithms using the given heuristic function h(n). A* algorithm expands the fewest number of nodes.
The main drawback of A* algorithm and indeed of any best-first search is its memory requirement. Since at least the entire open list must be saved, A* algorithm is severely space-limited in practice, and is no more practical than breadth-first search on current machines. For example, while it can be run successfully on the eight puzzle, it exhausts available memory in a matter of minutes on the fifteen puzzle.
A Production System is best suited AI Technique to solve the State Space Problems like Water Jug, and Missionaries and Cannibals. these problems consists of State, State Space and Goal State.
State = Initial state of the Problem. State space = Intermediate States between Initial State and Goal State. Goal State = Final state of the problem. - to solve state space problems Production System is best suited AI Technique.
A production system consists of rules and factors. Knowledge is encoded in a declarative from which comprises of a set of rules of the form and Control strategy. PRODUCTION SYSTEM SITUATION that implies ACTION. Example:- IF the initial state is a goal state THEN quit. The major components of an AI production system are i. A global database ii. A set of production rules and iii. A control system The goal database is the central data structure used by an AI production system.

Q.9 What are production system ? What are it's components?

Ans: The production rules operate on the global database. Each rule has a precondition that is either satisfied or not by the database. If the precondition is satisfied, the rule can be applied. Application of the rule changes the database. The control system chooses which applicable rule should be applied and ceases computation when a termination condition on the database is satisfied. If several rules are to fire at the same time, the control system resolves the conflicts.
Four classes of production systems:-

1. A monotonic production system
2. A non monotonic production system
3. A partially commutative production system
4. A commutative production system.

Advantages of production systems:-

1. Production systems provide an excellent tool for structuring AI programs.
2. Production Systems are highly modular because the individual rules can be added, removed or modified independently.
3. The production rules are expressed in a natural form, so the statements contained in the knowledge base should the a recording of an expert thinking out loud.

Disadvantages of Production Systems:-One important disadvantage is the fact that it may be very difficult analyse the flow of control within a production system because the individual rules don't call each other. Production systems describe the operations that can be performed in a search for a solution to the problem. They can be classified as follows. Monotonic production system :- A system in which the application of a rule never prevents the later application of another rule, that could have also beenapplied at the time the first rule was selected. Knowledge Based and Expert Systems

Q.10 What are the different control strategies?

Ans: 1) The first requirement of a control strategy is that it must cause motion.

2) The second requirement of a control strategy is that issue must be systematic.

We will explain these two with respect to water jug problem. If we have implemented choosing the first operator and then the one which matches the first one, then we would not have solved the problem. If we follow any strategy which can cause some motion then will lead to a solution. But if it is not followed systematically , and then got the solution. One day to follow a systematic control strategy is to construct a tree with the initial state as its root. By applying all possible combinations from the first level leaf nodes. Continue the process until some rule produces a goal state. For the water jug problem a tree can be constructed as given in following diagram.

The control strategy for the search process is called breadth first search. Other systematical control strategies are also available . for example, we can select one single branch of a tree until it yields a solution or until some pre specified depth has been reached. If not we go back and explore to other branches . this is called depth – first – search. The water jug problems will lead to an answer by adoption any control strategy because the problem is simple. This is not always the case.

Q.11 What do you mean by backtracking?

Ans: The order in which variables are instantiated can have a large effect on the size of the search tree. The idea of variable ordering is to order the variables form most constrained to least constrained. For example, if a variable has only a single value remaining that is consistent with the previously instantiated variable, it should be assigned that value immediately. In general, the variables should be instantiated in increasing order of the size of their remaining domains. This can either be done statically at the beginning of the search or dynamically, reordering the remaining variables each time a variable is assigned a new value.

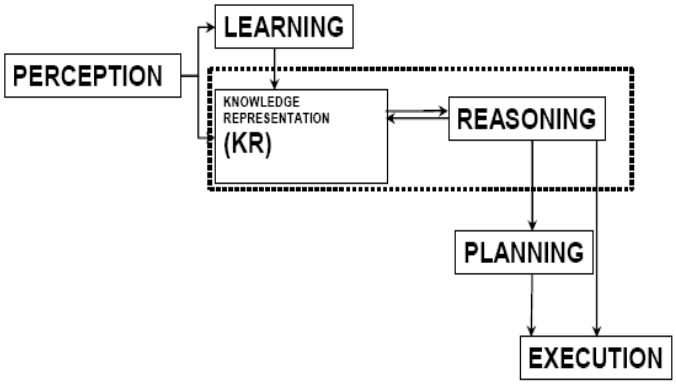
Chapter2
Knowledge representation

Q.1 What is KR? Explain its types also.

Ans: Durkin refers to it as the "Understanding of a subject area". A well-focused subject area is referred to as a knowledge domain, for example, medical domain, engineering domain, business domain, etc..

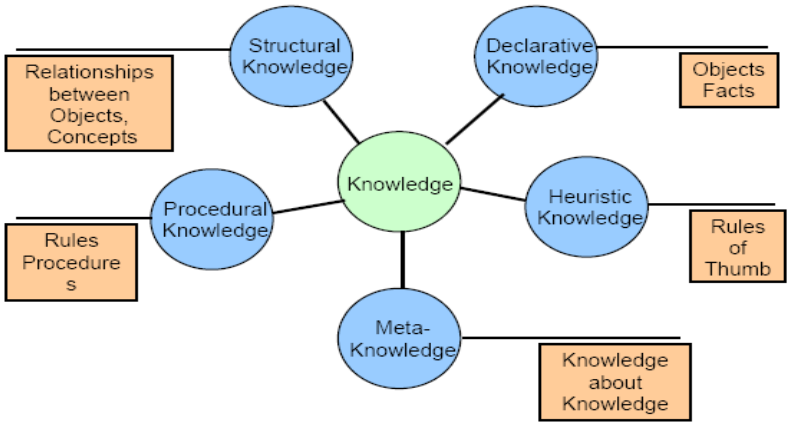
A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.

- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world.



Knowledge and its types

- Procedural knowledge: Describes how to do things, provides a set of directions of how to perform certain tasks, e.g., how to drive a car.
- Declarative knowledge: It describes objects, rather than processes. What is known about a situation, e.g. it is sunny today, and cherries are red.
- Meta knowledge: Knowledge about knowledge, e.g., the knowledge that blood pressure is more important for diagnosing a medical condition than eye color.
- Heuristic knowledge: Rule-of-thumb, e.g. if I start seeing shops, I am close to the market.
    - Heuristic knowledge is sometimes called shallow knowledge.
    - Heuristic knowledge is empirical as opposed to deterministic
- Structural knowledge: Describes structures and their relationships. e.g.
    - how the various parts of the car fit together to make a car, or knowledge structures in terms of concepts, sub concepts, and objects.



Q .2 What are different issues that arise in KR?

Ans: Issue in Knowledge Representation

Below are listed issues that should be raised when using a knowledge representation technique:

Important Attributes

-- Are there any attributes that occur in many different types of problem?
There are two *instance* and *isa* and each is important because each supports property inheritance.

Relationships
-- What about the relationship between the attributes of an object, such as, inverses, existence, techniques for reasoning about values and single valued attributes. We can consider an example of an inverse in
band(John Zorn,Naked City)
This can be treated as John Zorn plays in the band *Naked City* or John Zorn's band is *Naked City*.
Another representation is *band = Naked City*
band-members = John Zorn, Bill Frissell, Fred Frith, Joey Barron, . . .

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

If *Tom feeds a dog* then it could become:
feeds(tom, dog)
If *Tom gives the dog a bone* like:
gives(tom, dog,bone) Are these the same?
In any sense does giving an object food constitute feeding?
If *give*(*x*, *food*) →*feed*(*x*) then we are making progress.
But we need to add certain inferential rules.
In the famous program on relationships *Louise is Bill's cousin* How do we represent this? *louise = daughter (brother or sister (father or mother(bill)))*
Suppose it is *Chris* then we do not know if it is *Chris* as a male or female and then *son* applies as well.
Clearly the separate levels of understanding require different levels of primitives and these need many rules to link together apparently similar primitives.

Obviously there is a potential storage problem and the underlying question must be what level of comprehension is needed.

Q.3 What is the difference between predicate & prepositional logics?

Ans 1) In propositional logic a complete sentence can be presented s an atomic proposition. and complex sentences can be created using AND, OR, and other operators.....these propositions has only true of false values and we can use truth tables to define them... like book is on the table....this is a single proposition... In predicate logic there are objects, properties, functions (relations) are involved.

2) In propositional logic, we use letters to symbolize entire propositions. Propositions are statements of the form "x is y" where x is a subject and y is a predicate. For example, "Socrates is a Man" is a proposition and might be represented in propositional logic as "S".

3) In predicate logic, we symbolize subject and predicate separately. Logicians often use lowercase letters to symbolize subjects (or objects) and uppercase letter to symbolize predicates. For example, Socrates is a subject and might be represented in predicate logic as "s" while "man" is a predicate and might be represented as "M". If so, "Socrates is a man" would be represented "Ms".

4) The important difference is that you can use predicate logic to say something about a set of objects. By introducing the universal quantifier ("∀"), the existential quantifier ("∃") and variables ("x", "y" or "z"), we can use predicate logic to represent thing like "Everything is green" as "∀Gx" or "Something is blue" as "∃Bx

Q.4 Explain resolution.

Ans: Robinson in 1965 introduced the resolution principle, which can be directly applied to any set of clauses. The principal is "Given any two clauses A and B, if there is a literal P1 in A which has a complementary literal P2 in B, delete P1 & P2 from A and B and construct a disjunction of the remaining clauses. The clause so constructed is called resolvent of A and B."

For example, consider the following clauses
A: P V Q V R

B: `p V Q V R

C: `Q V R

Clause A has the literal P which is complementary to `P in B. Hence both of them deleted and a resolvent (disjunction of A and B after the complementary clauses are removed) is generated. That resolvent has again a literal Q whose negation is available in C. Hence resolving those two, one has the final resolvent.

A: P V Q V R (given in the problem)

B: `p V Q V (given in the problem)

D: Q V R (resolvent of A and B)

C: `Q V R (given in the problem)

E: R (resolvent of C and D)

Q.5 What are the Knowledge representation techniques ?

Ans: All of these, in different ways, involve hierarchical representation of data.

- Lists - linked lists are used to represent hierarchical knowledge
- Trees - graphs which represent hierarchical knowledge. LISP, the main programming language of AI, was developed to process lists and trees.
- Semantic networks - nodes and links - stored as propositions. Examples in Stillings et. al. pp. 146-147
- Schemas - used to represent commonsense or stereotyped knowledge.
    - Frames (Minsky) - Describe objects. Consist of a cluster of nodes and links manipulated as a whole. Knowledge is organised in slots. Frames are hierarchically organised. Example on p. 151 of Stillings

- Scripts (Schank and Abelson) - Describe event rather than objects. Consist of stereotypically ordered causal or temporal chain of events. Example on p. 156 of Stillings

- Rule-based representations (Newell and Simon) - used in specific problem-solving contexts. Involve production rules containing *if-then* or *situation-action* pairs. Specific example: problem space representations. Contain:
  - Initial state
  - Goal state
  - Legal operators, i.e. things you are allowed to do
  - Operator restrictions, i.e. factors which constrain the application of operators

(More on Problem-space representations and strategies in Semester 2 - Problem solving - expert-novice studies)

- Logic-based representations - may use deductive or inductive reasoning. Contain:
  - Facts and premises
  - Rules of propositional logic (Boolean - dealing with complete statements
  - Rules of predicate calculus (allows use of additional information about objects in the proposition, use of variables and functions of variables
  - Measures of certainty - may involve Certainty Factors (eg. If symptom then (CF) diagnosis) which could be derived from expert estimation or from statistical data; Bayesian probability; or Fuzzy logic (in which the concepts or information itself has some associated certainty value).

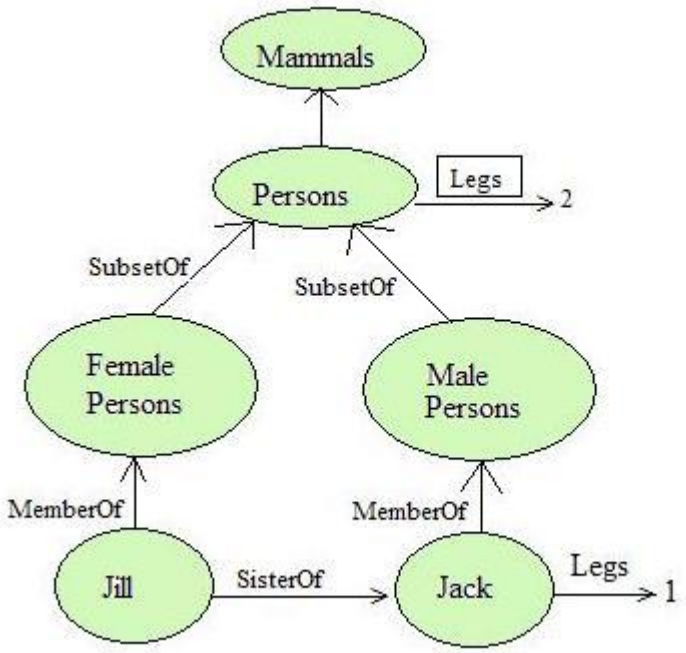
Q.6 Explain following representation schemes-

a) Semantic net:

Ans

- A semantic net (or semantic network) is a knowledge representation technique used for propositional information. So it is also called a propositional net. Semantic nets convey meaning. Semantic nets are two dimensional representations of knowledge. Mathematically a semantic net can be defined as a labeled directed graph.

- Semantic nets consist of nodes, links (edges) and link labels. In the semantic network diagram, nodes appear as circles or ellipses or rectangles to represent objects such as physical objects, concepts or situations. Links appear as arrows to express the relationships between objects, and link labels specify particular relations. Relationships provide the basic structure for organizing knowledge. The objects and relations involved need not be so concrete. As nodes are associated with other nodes semantic nets are also referred to as associative nets.



Advantages of Semantic Nets

Semantic nets have the ability to represent default values for categories. In the above figure Jack has one leg while he is a person and all persons have two legs. So persons have two legs has only default status which can be overridden by a specific value.

- Semantic nets convey some meaning in a transparent manner.
- Semantic nets are simple and easy to understand.
- Semantic nets are easy to translate into PROLOG.

Disadvantage of Semantic Nets

One of the drawbacks of semantic network is that the links between the objects represent only binary relations. For example, the sentence Run(ChennaiExpress, Chennai,Bangalore,Today) cannot be asserted directly.

There is no standard definition of link names.

a) Frames:

- Natural language understanding requires inference i.e., assumptions about what is typically true of the objects or situations under consideration. Such information can be coded in structures known as frames.
- **Need of frames**
- Frame is a type of schema used in many AI applications including vision and natural language processing. Frames provide a convenient structure for representing objects that are typical to a stereotypical situations. The situations to represent may be visual scenes, structure of complex physical objects, etc. Frames are also useful for representing commonsense knowledge. As frames allow nodes to have structures they can be regarded as three-dimensional representations of knowledge.
- A frame is similar to a record structure and corresponding to the fields and values are slots and slot fillers. Basically it is a group of slots and fillers that defines a stereotypical object. A single frame is not much useful. Frame systems usually have collection of frames connected to each other. Value of an attribute of one frame may be another frame.
    - A frame for a book is given below.

Slots	Fillers
publisher	Thomson
title	Expert Systems
author	Giarratano
edition	Third
year	1998
pages	600

- The above example is simple one but most of the frames are complex. Moreover with filler slots and inheritance provided by frames powerful knowledge representation systems can be built.
- Frames can represent either generic or frame. Following is the example for generic frame.

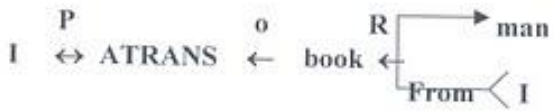
Slot	Fillers
Name	computer
specialization_of	a_kind_of machine
Types	(desktop, laptop,mainframe,super) if-added: Procedure ADD_COMPUTER
Speed	default: faster if-needed: Procedure FIND_SPEED
Location	(home,office,mobile)
under_warranty	(yes, no)

- The fillers may values such as computer in the name slot or a range of values as in types slot. The procedures attached to the slots are called procedural attachments. There are mainly three types of procedural attachments: if-needed, default and if-added. As the name implies if-needed types of procedures will be executed when a filler value is neeeded. Default value is taken if no other value exists. Defaults are used to represent *commonsense knowledge*. Commonsense is generally used when no more situation specific knowledge is available.
- The if-added type is required if any value is to be added to a slot. In the above example, if a new type of computer is invented ADD_COMPUTER procedure should be executed to add that information. An if-removed type is used to remove a value from the slot.

a) Conceptual Dependency:This representation is used in natural language processing in order to represent them earning of the sentences in such a way that inference we can be made from the sentences. It is independent of the language in which the sentences were originally stated. CD representations of a sentence is built out of primitives , which are not words belonging to the language but are conceptual , these primitives are combined to form the meaning

s of the words. As an example consider the event represented by the sentence.

I gave the man a book.
In CD from the above sentence can be represented with primitives as



In the above representation the symbols have the following meaning:

- Arrows indicate direction of dependency
- Double arrow indicates two may link between actor and the action
- P indicates past tense
- ATRANS is one of the primitive acts used by the theory . it indicates transfer of possession
- 0 indicates the object case relation
- R indicates the recipient case relation
- Conceptual dependency provides a str5ucture in which knowledge can be represented and also a set of building blocks from which representations can be built. A typical set of primitive actions are
- ATRANS - Transfer of an abstract relationship(Eg: give)
- PTRANS - Transfer of the physical location of an object(Eg: go)
- PROPEL - Application of physical force to an object (Eg: push)
- MOVE - Movement of a body part by its owner (eg : kick)

- GRASP - Grasping of an object by an actor(Eg: throw)
- INGEST - Ingesting of an object by an animal (Eg: eat)
- EXPEL - Expulsion of something from the body of an animal (cry)
- MTRANS - Transfer of mental information(Eg: tell)
- MBUILD - Building new information out of old(Eg: decide)
- SPEAK - Production of sounds(Eg: say)
- ATTEND - Focusing of sense organ toward a stimulus (Eg: listen)
- A second set of building block is the set of allowable dependencies among the conceptualization describe in a sentence.

Scripts: **Scripts** were developed in the early AI work by Roger Schank, Robert P. Abelson and their research group, and are a method of representing procedural knowledge. They are very much like frames, except the values that fill the slots must be ordered.
The classic example of a script involves the typical sequence of events that occur when a person dines in a restaurant: *finding a seat, reading the menu, ordering drinks from the waitstaff...* In the script form, these would be decomposed into conceptual transitions, such as **MTRANS** and **PTRANS**, which refer to *mental transitions [of information]* and *physical transitions [of things]*.
Scripts can be inflexible. To deal with inflexibility, smaller modules called memory organization packets (MOP) can be combined in a way that is appropriate for the situation

Q.7 What do you mean by monotonic reasoning ?

Ans: Traditional systems based on predicate logic are monotonic . Here number of statements known to be true increases with time. New statements are added and new theorems are proved, but the previously known statements never become invalid.

In monotonic systems there is no need to check for in consistencies between new statements and the old knowledge. When a proof is made , the basis of the proof need not be remembered, since the old statements never disappear. But monotonic systems are not good in real problem domains where the information is incomplete, situations change and new assumptions are generated while solving new problems.

we visit a friend's home, we buy biscuits for the children . because we believe that most children like biscuits. In this case we do not have information to the contrary. A computational description of default reasoning must relate the lack of information on X to conclude on Y.

Default reasoning (or most probabilistic choice) is defined as follows:

Definition 1 : If X is not known, then conclude Y.

Definition 2 : If X can not be proved, then conclude Y.

Definition 3: If X can not be proved in some allocated amount of time then conclude Y.

It is to be noted that the above reasoning process lies outside the realm of logic. It conclude on Y if X can not be proved, but never bothers to find whether X can be proved or not. Hence the default reasoning systems can not be characterized formally. Even if one succeeds in gaining complete information at the moment, the validity of the information may not be for ever, since it is a changing world. What appears to be true now, may be so at a later time (in a non monotonic system).

One way to solve the problem of a changing world to delete statements when they are no longer accurate, and replace them by more accurate statements. This leads to a non monotonic system in which statements can be deleted as well as added to the knowledge base. When a statement is deleted, other related statements may also have to be deleted. Non monotonic reasoning systems may be necessary due to any of the following reasons.

- The presence of incomplete inforamtio0n requires default reasoning.

- A changing world must be described by a changing database.

- Generating a complete solution to a problem may require temporary assumptions about partial solutions.

Q.8 Explain natural deduction.

Ans: Natural deduction is a formal inference system which is said naturally to mirror the way in which humans reason. A natural deduction system consists of rules of inference eliminating and introducing each of the connectives and quantifiers of the Predicate Calculus. There are twelve rules which may be used to infer conclusions. Two examples of such rules are given below:

A & & $A \rightarrow B$		
& B		

$\overline{A}$
|
B
$A \rightarrow B$

The rule on the left is that to eliminate the $\rightarrow$ connective and that on the right to introduce it. Proofs are trees, the leaves representing the assumptions, and the root, the conclusion that has been deduced from those assumptions. In natural deduction we can assume temporarily certain formulae, and then `discharge' them later in the proof, using certain of the deduction rules. The $\rightarrow$ introduction rule above is an example of a rule discharging an assumption (A in this case). Natural deduction is sound and complete.

Chapter 3
Nonmonotonic reasoning

Q.1 What do you mean by non-monotonic reasoning?

Ans: Non monotonic system are harder to deal with than monotonic systems. This is because when a statement is deleted as " no more valid", other related statements have to be backtracked and they should be either deleted or new proofs have to be found for them. This is called dependency directed backtracking (DDB). In order to propagate the current changes into the database, the statements on which a particular proof depends, should also be stored along with the proof. Thus non – monotonic systems require more storage space as well as more processing time than monotonic systems.

Q.2 Explain Bay's theorem.

Ans: The essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence. In other words, it allows scientists to combine new data with their existing knowledge or expertise. The canonical example is to imagine that a precocious newborn observes his first sunset, and wonders whether the sun will rise again or not. He assigns equal prior probabilities to both possible outcomes, and represents this by placing one white and one black marble into a bag. The following day, when the sun rises, the child places another white marble in the bag. The probability that a marble plucked randomly from the bag will be white (ie, the child's degree of belief in future sunrises) has thus gone from a half to two-thirds. After sunrise the next day, the child adds another white marble, and the probability (and thus the degree of belief) goes from two-thirds to three-quarters. And so on. Gradually, the initial belief that the sun is just as likely as not to rise each morning is modified to become a near-certainty that the sun will always rise.

Bayes' Theorem

- **Bayes' theorem lets us calculate a conditional probability:**

$$P(B \mid A) = \frac{P(A \mid B) \cdot P(B)}{P(A)}$$

- **P(B) is the prior probability of B.**
- **P(B | A) is the posterior probability of B**

Bayes' Theorem Deduction

$$P(B \mid A) = \frac{P(B \wedge A)}{P(A)}$$

Q.3 Explain certainty factor.

Ans: One of the unique aspects of MYCIN was its ability to handle "uncertainty". Unlike a game playing situation or a planning system, diagnosis is fraught with uncertainty. In order to model that uncertainty, MYCIN introduced a concept called *Certainty Factors* (CF). Based on backward chaining, MYCIN's algorithm works as follows:

- Add "diagnose-and-treat" to working memory along with all reported patient data.
- Repeat
  - Identify all rules that can provide the conclusion currently sought
  - Match right hand sides (that is, search for rules whose right hand sides match anything in working memory)
  - Use conflict resolution to identify a single rule
  - Fire (execute) that rule
    - Find and remove a piece of knowledge which is no longer needed
    - Find and modify a piece of knowledge now that more specific information is known
      - Add a new subgoal (left-hand side conditions that need to be proved)
- Until the action "done" is added to working memory

Based on the rules, MYCIN would go through several distinct phases, identify the illness(es), order tests as needed to refine the identification(s), generate a treatment. MYCIN also had the capability of answering

questions during the diagnosis. For instance, when the user is asked a question or to perform a test, the user can ask "why?" in which case MYCIN would explain why. Similarly, when a diagnostic conclusion and/or treatment is reported, the user can ask why MYCIN decided that. The explanation is essentially the rules that are in the current chain of logic. For instance, given rule 52, if MYCIN is attempting to determine if pseudomonas is a proper identification of the illness and MYCIN asks the user if the patient has serious burns and the user asks "why?", MYCIN can report by saying "if this is true, and given that we already know that the patient's blood culture sample has been identified as having a morphology of rod and is gram negative, then this would allow us to conclude the possibility of pseudomonas.

Q.4 Explain Dempster-Shafer theory.

Ans. Dempster–Shafer theory is a generalization of the Bayesian theory of subjective probability; whereas the latter requires probabilities for each question of interest, belief functions base degrees of belief (or confidence, or trust) for one question on the probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ depends on how closely the two questions are related.[5] Put another way, it is a way of representing epistemic plausibilities but it can yield answers that contradict those arrived at using probability theory.

Often used as a method of sensor fusion, Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule[6] for combining such degrees of belief when they are based on independent items of evidence. In essence, the degree of belief in a proposition depends primarily upon the number of answers (to the related questions) containing the proposition, and the subjective probability of each answer. Also contributing are the rules of combination that reflect general assumptions about the data.

Q .5 What is FUZZY LOGIC?

Ans In this context, FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or

workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

Q .6 Why use FL?

Ans: FL offers several unique features that make it a particularly good choice for many control problems.

1) It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.
2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.
3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.
4) Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rulebase quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.
5) FL can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

Q.7 How is FL Used?

Ans : 1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?

2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

3) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.

5) Create the necessary pre- and post-processing FL routines if implementing in S/W, otherwise program the rules into the FL H/W engine.

6) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

Q.8 Explain learning.What are different types of learning?

Ans: Learning denotes changes in a system that enable the system to do the same task more efficiently next time or we can say generallize the experience in a way that allows to improve your performance on the task.

Types of learning:

- **Supervised learning** generates a function that maps inputs to desired outputs (also called **labels**, because they are often provided by human experts labeling the training examples). For example, in a classification problem, the learner approximates a function mapping a vector into classes by looking at input-output examples of the function.
- **Unsupervised learning** models a set of inputs, like clustering. See also data mining and knowledge discovery.

- **Semi-supervised learning** combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- **Reinforcement learning** learns how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback in the form of rewards that guides the learning algorithm.
- **Transduction**, or *transductive inference*, tries to predict new outputs on specific and fixed (test) cases from observed, specific (training) cases.
- **Learning to learn** learns its own inductive bias based on previous experience.

Q.9 Why do we require machine learning?

Ans: 1) Understand and improve efficiency of human learning.
2) Discover new things & structure that is unknown to human.
3) Fill in incomplete specification about a domain.

Q.10 Explain inductive & deductive learning?

Ans: Two very distinct and opposing instructional approaches are inductive and deductive. Both approaches can offer certain advantages, but the biggest difference is the role of the teacher. In a deductive classroom, the teacher conducts lessons by introducing and explaining concepts to students, and then expecting students to complete tasks to practice the concepts; this approach is very teacher-centred. Conversely, inductive instruction is a much more student-centred approach and makes use of a strategy known as 'noticing'.

In contrast with the deductive method, inductive instruction makes use of student "noticing". Instead of explaining a given concept and following this explanation with examples, the teacher presents students with many examples showing how the concept is used. The intent is for students to "notice", by way of the examples, how the concept works.

Deductive reasoning arrives at a specific conclusion based on generalizations. Inductive reasoning takes events and makes generalizations

Deductive reasoning can be described as reasoning of the form if A then B. Deduction is in some sense the direct application of knowledge in the production of new knowledge.

Chapter4
Expert System

Q.1 What is an Expert System ?Explain with examples.

Ans: Expert system is an artificial intelligence program that has expert-level knowledge about a particular domain and knows how to use its knowledge to respond properly. Domain refers to the area within which the task is being performed. Ideally the expert systems should substitute a human expert. *Edward Feigenbaum* of Stanford University has defined expert system as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions." It is a branch of artificial intelligence introduced by researchers in the Stanford Heuristic Programming Project.

The *expert systems* is a branch of AI designed to work within a particular domain. As an expert is a person who can solve a problem with the domain knowledge in hands it should be able to solve problems at the level of a human expert. The source of knowledge may come come from a human expert and/or from books, magazines and internet. As knowledge play a key role in the functioning of expert systems they are also known as knowledge-based systems and knowledge-based expert systems. The expert's knowledge about solving the given specific problems is called knowledge domain of the expert.

Components of Expert Systems

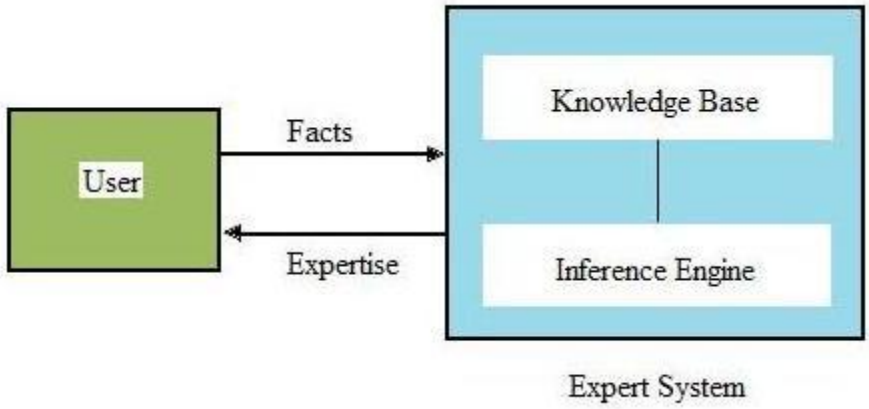


Figure: Expert System

Examples of Expert Systems and Knowledge Bases

- American Express uses an expert system to automate the process of checking for fraud and misuses of its no-limit credit card. Credit checks must be completed within 90 sec. while the customer waits, and the cost of an error can be high.
- XCON, one of the most successful expert systems in commercial use today, has been configuring complex computer systems since it was developed at Digital Equipment Corp

Q.2 What are the components of an expert system?

Ans The expert system consists of two major components: knowledge base and inference engine.

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.

Q3 What are the different characteristics of Expert System?

Ans Characteristics of Expert Systems

High performance: They should perform at the level of a human expert.

Adequate response time: They should have the ability to respond in a reasonable amount of time. Time is crucial especially for real time systems.

Reliability: They must be reliable and should not crash.

Explanation: They are capable of explaining in detail the reasoning that led to a conclusion.

Multiple expertise: It can be designed to have knowledge of many experts.

Understandable: They should not be a black box instead it should be able explain the steps of the reasoning process. It should justify its conclusions in the same way a human expert explains why he arrived at particular conclusion.

Q.4 What are prons & cons of an expert system?

Ans: **Pros/Cons of Expert Systems**

Pros	Cons
• Aid experts by providing automated data analysis and informed second options	• Today, they are difficult to build
• Support non-experts by providing advice based on judgments of one or more experts.	• Expert system shells are sold to simplify the process but they do not include the part that is most difficult to build—the knowledge base

• Some new expert systems can grow their own knowledge bases while observing human decision makers doing their jobs.	• Poor at planning strategies
• Many modern expert systems are based on *Fuzzy Logic*, which allows conclusions to be stated as probabilities.	• Lack of flexibility makes them less creative than human thinkers
• Includes a human interface, which enables the user to interact with the system, and an inference engine, which puts the user input together with the knowledge base, applies logical principles, and produces the requested expert advice.	• Powerless outside of their narrow, deep domains of knowledge

Q.5 How expert systems work?

Ans:

- They apply information gleaned from human experts to new problems.
- Their suggested solutions to these problems can help users determine the best course of action.

- They function within narrow, carefully defined domains.

Q.6 What do you mean by expert shell?

Ans: A shell is a special purpose tool designed based on the requirements of particular applications. User should supply the knowledge base to the shell. Example for the shell is EMYCIN (Empty MYCIN) shell. Shell manages the input and output. It processes the information given by the user, relates it to the concepts contained in the knowledge base, and provides solution for a particular problem.

Multiple Choice Questions

1. Artificial Intelligence is associated with which generation?
   a. First Generation
   b. Second Generation
   c. Fifth Generation
   d. Sixth Generation

2. Which operation is not performed by computer?
   a. Inputting
   b. Processing
   c. Controlling
   d. **Understanding**

3. Fifth generation computer is also known as:
   a. Knowledge information processing system
   b. Very large scale integration (VLSI)

c. Both of above
d. None of above

4. What is the term used for describing the judgmental or commonsense part of problem solving?
   a) Heuristic
   b) Critical
   c) Value based
   d) Analytical

5. What was originally called the "imitation game" by its creator?
   a) The Turing Test
   b) LISP
   c) The Logic Theorist
   d) Cybernetics

6. Decision support programs are designed to help managers make:
   a) budget projections
   b) visual presentations
   c) business decisions
   d) vacation schedules

7. PROLOG is an AI programming language which solves problems with a form of symbolic logic known as predicate calculus. It was developed in 1972 at the University of Marseilles by a team of specialists. Can you name the person who headed this team?
   a) Alain Colmerauer
   b) Nicklaus Wirth
   c) Seymour Papert
   d) John McCarthy

8. Programming a robot by physically moving it through the trajectory you want it to follow is called:
   a) contact sensing control

b) continuous-path control
c) robot vision control
d) pick-and-place control

9. To invoke the LISP system, you must enter
a) AI
b) LISP
c) CL (Common Lisp)
d) both b and c

10. Seymour Papert of the MIT AI lab created a programming environment for children called:
a) BASIC
b) LOGO
c) MYCIN
d) FORTRAN

11. The area of AI that investigates methods of facilitating communication between people and computers is:
a) natural language processing
b) symbolic processing
c) decision support
d) robotics

12. For speech understanding systems to gain widespread acceptance in office automation, they must feature:
a) speaker independence
b) speaker dependence
c) isolated word recognition
d) All of the above

13. Output segments of AI programming contain(s)
a) printed language and synthesized speech
b) Manipulation of physical object

c) Locomotion
d) All of the above

14. The characteristics of the computer system capable of thinking, reasoning and learning is known is
a) machine intelligence
b) human intelligence
c) artificial intelligence
d) virtual intelligence

15.The primary interactive method of communication used by humans is:
a) reading
b) writing
c) speaking
d) All of the above

15. The original LISP machines produced by both LMI and Symbolics were based on research performed at:
a) CMU
b) MIT
c) Stanford University
d) RAMD

16. In LISP, the addition 3 + 2 is entered as
a) 3 + 2
b) 3 add 2
c) 3 + 2 =
d) (+ 3 2)

17. Weak AI is
a) the embodiment of human intellectual capabilities within a computer.
b) a set of computer programs that produce output that would be

considered to reflect intelligence if it were generated by humans.

c) **the study of mental faculties through the use of mental models implemented on a computer**.

d) All of the above

18. In LISP, the function assigns the symbol x to y is

a) (setq y x)

b) (set y = 'x')

c) (setq y = 'x')

d) (setq y 'x')

19. In a rule-based system, procedural domain knowledge is in the form of:

a) production rules

b) rule interpreters

c) meta-rules

d) control rules

20. If a robot can alter its own trajectory in response to external conditions, it is considered to be:

a) intelligent

b) mobile

c) open loop

d) non-servo

21. One of the leading American robotics centers is the Robotics Institute located at:

a) **CMU**

b) MIT

c) RAND

d) SRI

22. A computer program that contains expertise in a particular domain is called an:

a) intelligent planner

b) automatic processor
c) expert system
d) operational symbolizer

23. Nils Nilsson headed a team at SRI that created a mobile robot named:
a) Robitics
b) Dedalus
c) Shakey
d) Vax

22. An AI technique that allows computers to understand associations and relationships between objects and events is called:
a) heuristic processing
b) cognitive science
c) relative symbolism
d) pattern matching

23. The field that investigates the mechanics of human intelligence is:
a) history
b) cognitive science
c) psychology
d) sociology

24. What is the name of the computer program that simulates the thought processes of human beings?
a) Human logic
b) Expert reason
c) Expert system
d) Personal information

25. What is the name of the computer program that contains the distilled knowledge of an expert?
a) Data base management system
b) Management information System
c) Expert system

d) Artificial intelligence

26. Claude Shannon described the operation of electronic switching circuits with a system of mathematical logic called:
a) LISP
b) XLISP
c) Boolean algebra
d) neural networking

27. Ambiguity may be caused by:
a) syntactic ambiguity
b) multiple word meanings
c) unclear antecedents
d) All of the above

28. Which company offers the LISP machine considered to be "the most powerful symbolic processor available"?
a) LMI
b) Symbolics
c) Xerox
d) Texas Instruments

29. Natural language processing is divided into the two subfields of:
a) symbolic and numeric
b) time and motion
c) algorithmic and heuristic
d) understanding and generation

30. High-resolution, bit-mapped displays are useful for displaying:
a) clearer characters
b) graphics
c) more characters
d) All of the above

31. Which of the following have people traditionally done better than computers?
    a) recognizing relative importance
    b) finding similarities
    c) resolving ambiguity
    d) All of the above

32. Research scientists all over the world are taking steps towards building computers with circuits patterned after the complex inter connections existing among the human brain's nerve cells. What name is given to such type of computers?
    a) Intelligent computers
    b) Supercomputers
    c) Neural network computers
    d) Smart computers

33. The hardware features of LISP machines generally include:
    a) large memory and a high-speed processor
    b) letter-quality printers and 8-inch disk drives
    c) a mouse and a specialized keyboard
    d) both (a) and (c)

34. The explanation facility of an expert system may be used to:
    a) construct a diagnostic model
    b) expedite the debugging process
    c) explain the system's reasoning process
    d) both (b) and (c)

35. A process that is repeated, evaluated, and refined is called:
    a) diagnostic
    b) descriptive
    c) interpretive
    d) iterative

36. A natural language generation program must decide:
a) what to say
b) when to say something
c) why it is being used
d) both (a) and (b)

37. Who is considered to be the "father" of artificial intelligence?
a) Fisher Ada
b) John McCarthy
c) Allen Newell
d) Alan Turning

38. A network with named nodes and labeled arcs that can be used to represent certain natural language grammars to facilitate parsing.
a. Tree Network
b. Star Network
c. Transition Network
d. Complete Network

39. A Personal Consultant knowledge base contain information in the form of:
a) parameters
b) contexts
c) production rules
d) All of the above

40. Which approach to speech recognition avoids the problem caused by the variation in speech patterns among different speakers?
a) Continuous speech recognition
b) Isolated word recognition
c) Connected word recognition
d) Speaker-dependent recognition

41. Which of the following, is a component of an expert system?

b) inference engine
d) knowledge base
f) user interface
h) All of the above

42. DARPA, the agency that has funded a great deal of American AI research, is part of the Department of:
a) Defense
b) Energy,
c) Education
d) Justice

43. LISP machines also are known as:
a) AI workstations
b) time-sharing terminals
c) super mini computers
d) All of the above

44. Which of the following have computers traditionally done better than people?
a) storing information
b) responding flexibly
c) computing numerically
d) both (a) and (c)

45. LISP was created by:
a) John McCarthy
b) Marvin Minsky
c) Alan Turing
d) Allen Newell and Herbert Simon

46. n which of the following situations might a blind search be acceptable?
a) real-life situation
b) complex game

c) small search space
d) All of the above

47. In AI programming, a list may contain:
a) cells
b) fields
c) pointers
d) All of the above

The first AI programming language was called:
a) BASIC
b) FORTRAN
c) IPL
d) LISP

The CAI (Computer-Assisted Instruction) technique based on programmed instruction is:
a) frame-based CAI
b) generative CAI
c) problem-solving CAI
d) intelligent CAI

One definition of AI focuses on problem-solving methods that process:
a) smell
b) symbols
c) touch
d) algorithms

AI and Expert Systems Key Terms

artificial intelligence (AI)	The science of creating machines capable of performing activities that require intelligence when they are done by people.
availability heuristic	A prediction about the probability of an event based on the ease of recalling or imagining similar events.
avoidant coping	Coping with a problem by trying one's best to ignore it.
classical model	Model stating that all instances of a concept share defining properties.
Cognition	The way in which information is processed and manipulated in remembering, thinking, and knowing.
cognitive appraisal	Individuals' interpretation of the events in their lives as harmful, threatening, or challenging and their determination of whether they have the resources to cope effectively with the events.

Common-sense Knowledge: The wealth of knowledge and understanding about the world that people share.

Concepts	Mental categories that are used to group objects, events, and characteristics.
confirmation bias	The tendency to search for and use information that supports, rather than refutes, our ideas.
convergent thinking	Thinking that produces one correct answer; characteristic of the type of thinking required on traditional intelligence tests.
Coping	Managing taxing circumstances, expending effort to

	solve life's problems, and seeking to master or reduce stress.
Creativity	The ability to think about something in novel and unusual ways and come up with unconventional solutions to problems.
culture-fair tests	Intelligence tests that are intended to be culturally unbiased.
decision making	Evaluating alternatives and making choices among them.
deductive reasoning	Reasoning from the general to the specific.
divergent thinking	Thinking that produces many answers to the same question; characteristic of creativity. **Domain**: Area in which task is being performed.
emotion-focused coping	Responding to the emotional aspects of stress rather than focusing on the problem causing the stress.
Expertise	The quality of having a particular talent—that "something special"—for the things that one does in a particular domain.

Expert System: Information system or software programs designed to replicate the decision-making process of a human expert.

Fixation	Using a prior problem-solving strategy and failing to look at a problem from a fresh, new perspective.
functional fixedness	A type of fixation in which individuals fail to solve a problem because they are fixated on a thing's usual functions.

Fuzzy Logic: A type of logic that allows conclusions to

be stated as possibilities rather than certainties.

Gifted	Descriptive of individuals who have an IQ of 130 or higher and/or superior talent in a particular area.
Heritability	The proportion of the IQ differences in a population that is attributed to genetic differences.
Heuristics	Shortcut strategies or guidelines that suggest, but do not guarantee, a solution to a problem.
hindsight bias	The tendency to report falsely, after the fact, that we accurately predicted an outcome.
inductive reasoning	Reasoning from the specific to the general or from the bottom-up.
infinite generativity	The ability to produce an infinite number of sentences using a relatively limited set rules.
Intelligence	Problem-solving skills and the ability to adapt to and learn from life's everyday experiences.
intelligence quotient (IQ)	An individual's mental age divided by chronological age multiplied by 100.

Knowledge Base: A database that contains both facts and a system of rules for determining and changing the relationship among those facts.

Knowledge Engineer: Computer systems expert responsible for interpreting the presented information and relaying it to computer programmers who code the information in to systems databases to be accessed by end-users. *Knowledge engineers* are used primarily in the construction process of computer systems

Language	A form of communication, whether spoken, written, or signed, that is based on a system of symbols. Machine Learning: Artificial intelligence techniques that make it possible for machine performance to improve based on feedback from past performance.
mental age (MA)	An individual's level of mental development relative to that of others.
mental retardation	A condition of limited mental ability in which the individual has a low IQ, usually below 70, has difficulty adapting to everyday life, and has an onset of these characteristics in the so-called developmental period.
Mindfulness	Being alert and mentally present for one's everyday activities.
Morphology	A language's rules for word formation. **Natural Language**: Language that people speak and write every day.
normal distribution	A symmetrical, bell-shaped curve with a majority of the scores falling in the middle of the possible range and few scores appearing toward the extremes of the range.
open-mindedness	Being receptive to the possibility of other ways of looking at things. **Parallel Processing:** Using multiple processors to

	divide jobs into pieces and work simultaneously on the pieces. **Pattern Recognition:** Identifying recurring patterns in input data with the goal of understanding or categorizing that input.
phonics approach	An approach to learning to read that emphasizes basic rules for translating written symbols into sounds.
Phonology	A language's sound system.
problem solving	An attempt to find an appropriate way of attaining a goal when the goal is not readily available.
problem-focused coping	The cognitive strategy of squarely facing one's troubles and trying to solve them.
prototype model	Model emphasizing that when people evaluate whether a given item reflects a certain concept, they compare the item with the most typical item(s) in that category and look for a "family resemblance."
Reasoning	The mental activity of transforming information to reach conclusions.
Reliability	The extent to which a test yields a consistent, reproducible measure of performance.
Semantics	The meaning of words and sentences in a particular language.
Standardization	Developing uniform procedures for administering and scoring a test, as well as creating norms for the test.
Subgoaling	Setting intermediate goals or defining intermediate problems in order to be in a better position to reach the final goal or solution.
syntax	A language's rules for the way words are combined to form acceptable phrases and sentences.

thinking	Manipulating information mentally, as when we form concepts, solve problems, make decisions, and reflect in a creative or critical manner.
triarchic theory of intelligence	Sternberg's theory that there are three main types of intelligence: analytical, creative, and practical. **Turing Test:** A way to test machine intelligence.
whole-language approach	An approach to learning to read that stresses that reading instruction should parallel a child's natural language learning; so reading materials should be whole and meaningful.

References of Artificial Intelligence

- George Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Fourth Edition Addison-Wesley, 2002 -- a well-respected introduction to artificial intelligence, as witnessed by its being in its fourth edition.
- Peter Norvig, AI on the Web, http://aima.cs.berkeley.edu/ai.html -- a list of over 800 links on various aspects of artificial intelligence.
- Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers, 1998 -- another fine introductory textbook on artificial intelligence.
- Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Second Edition, Prentice-Hall, 2003 -- the leading introductory textbook in the field.
- G.Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving
- Poole, Mackworth & Goebel: Computational Intelligence - A Logical Approach. By David Poole, Alan Mackworth and Randy Goebel. 1998. Oxford University Press, New York
- S. Russell & P. Norvig, Artificial Intelligence: A Modern Approach