DEPARTMENT OF INFORMATION TECHNOLOGY

IT8601 COMPUTATIONAL INTELLIGENCE

QUESTION BANK

PART A

UNIT I

INTRODUCTION

1. **Define Artificial Intelligence (AI).**

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

- Systems that think like humans
- Systems that act like humans
- Systems that think rationally
- Systems that act rationally.

2. **Define Artificial Intelligence formulated by Haugeland.**

The exciting new effort to make computers think machines with minds in the full and literal sense.

3. **Define Artificial Intelligence in terms of human performance.**

The art of creating machines that performs functions that require intelligence when performed by people.

4. **Define Artificial Intelligence in terms of rational acting.**

A field of study that seeks to explain and emulate intelligent behaviors in terms of computational processes-Schalkoff. The branch of computer science that is concerned with the automation of intelligent behavior-Luger & Stubblefield.

5. **Define Artificial in terms of rational thinking.**

The study of mental faculties through the use of computational models-Charniak & McDermott.The study of the computations that make it possible to perceive, reason and act-Winston.

6. **What is meant by Turing test?**

To conduct this test we need two people and one machine. One person will be an interrogator(i.e.) questioner, will be asking questions to one person and one machine.

Three of them will bein a separate room.Interrogator knows them just as A and B. so it has to identify which is the person and machine.The goal of the machine is to make Interrogator believe that it is the person's answer. If machine succeeds by fooling

Interrogator, the machine acts like a human. Programming a computer to pass Turing test is very difficult.

7. **What is called materialism?**

An alternative to dualism is materialism, which holds that the entire world operate according to physical law. Mental process and consciousness are therefore part of physical world, but inherently unknowable they are beyond rational understanding.

8. **What are the capabilities, computer should posses to pass Turing test?**

- Natural Language Processing Knowledge representation
- Automated Reasoning Machine Learning.

9. **Define Total Turing Test?**

The test which includes a video signals so that the interrogator can test the perceptual abilities of the machine.

10. **What are the capabilities computers needs to pass total Turing test?**

- Computer Vision
- Robotics

11. Define Rational Agent.

It is one that acts, so as to achieve the best outcome (or) when there is uncertainty, the best expected outcome.

12. Define Agent.

An Agent is anything that can be viewed as perceiving (i.e.) understanding its environment through sensors and acting upon that environment through actuators.

13. Define an Omniscient agent.

An omniscient agent knows the actual outcome of its action and can act accordingly; but omniscience is impossible in reality.

14. What are the factors that a rational agent should depend on at any given time?

1. The performance measure that defines degree of success.
2. Ever thing that the agent has perceived so far. We will call this complete perceptual history the percept sequence.
3. When the agent knows about the environment.
4. The action that the agent can perform.

15. Define Architecture.

The action program will run on some sort of computing device which is called as Architecture

16. List the various type of agent program. Simple reflex agent program.

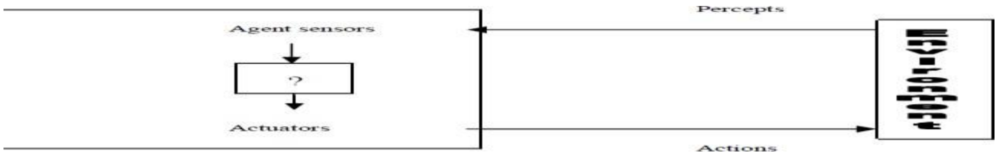
Agent that keep track of the world.
Goal based agent program.

Utility based agent program

17. Give the structure of agent in an environment?

Agent interacts with environment through sensors and actuators.

An Agent is anything that can be viewed as perceiving (i.e.) understanding its environment through sensors and acting upon that environment through actuators.



18. Define Percept Sequence. An agent’s choice of action at any given instant can depend on the entire percept sequence observed to elate.

19. Define Agent Function. It is a mathematical description which deals with the agent’s behavior that maps the given percept sequence into an action.

20. Define Agent Program.

Agent function for an agent will be implemented by agent program.

21. How agent should act?

Agent should act as a rational agent. Rational agent is one that does the right thing, (i.e.) right actions will cause the agent to be most successful in the environment.

22. How to measure the performance of an agent?

Performance measure of an agent is got by analyzing two tasks. They are How and When actions.

23. Define performance measures.

Performance measure embodies the criterion for success of an agent’s behavior.

24. Define Ideal Rational Agent.

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built in knowledge the agent has.

25. Define Omniscience.

An Omniscience agent knows the actual outcome of its actions and can act accordingly.

26. Define Information Gathering.

Doing actions in order to modify future percepts sometimes called information gathering.

27. What is autonomy?

A rational agent should be autonomous. It should learn what it can do to compensate for partial (or) in correct prior knowledge.

28. What is important for task environment?
PEAS → P- Performance measure

E - Environment

A- Actuators

S – Sensors
Example

Interactive English tutor performance measure maximize student's score on test.

Environment

Set of students testing Agency

Actuators

Display exercises suggestions, corrections.

Sensors

Keyboard entry

29. What is the structure of intelligent Agent? Intelligent Agent = Architecture + Agent Program **33. Define problem solving agent.**

Problem solving agent is one kind of goal based agent, where the agent Should select one action from sequence of actions which lead to desirable states.

34. List the steps involved in simple problem solving technique.

i. Goal formulation ii. Problem formulation
ii. Search iv. Solution
iii. Execution phase

35. What are the different types of problem?

Single state problem, multiple state problems, Contingency problem, Exploration problem

36. What are the components of a problem?

There are four components. They are

- initial state
- Successor function
- Goal test
- Path cost
- Operator
- state space
- path

37. Define State Space.

The set of all possible states reachable from the initial state by any sequence of action is called state space.

38. Define Path.

A path in the state space is a sequence of state connected by sequence of actions.

39. Define Path Cost.

A function that assigns a numeric cost to each path, which is the sum of the cost of the each action along the path.

40. **Give example problems for Artificial Intelligence**.

i. Toy problems
ii. ii. Real world problems

41. Give example for real world end toy problems.

- Real world problem examples:
  - i. Airline travel problem.
- Touring problem.
  - Traveling salesman problem.
- VLSI Layout problem
- Robot navigation
- Automatic Assembly

- Internet searching Toy problem Examples:
- Vacuum world problem.
- 8 – Queen problem 8 – Puzzle problem

42. Define search tree.

The tree which is constructed for the search process over the state space is called search tree.

43. Define search node.

The root of the search tree that is the initial state of the problem is called search node.

44. Define fringe.

The collection of nodes that have been generated but not yet expanded, this collection is called fringe or frontier.

45. List the performance measures of search strategies.

i. Completeness ii. Optimality iii. Time complexity iv. Space complexity

46. Define branching factor (b).

The number of nodes which is connected to each of the node in search tree is called Branching factor.

47. Differentiate Blind Search and

Heuristic Search.

i) No information about the path cost from the current state to goal state.

ii) Problem is solved with the information we which we know. iii) Example

a) Breadth first search

b) Uniform cost search

c) Depth first Search

d) Depth limited search

e) Iterative deepening search

f) Bi – Directional Search Backtracking search.

48. Define Uniform cost search.

i) We have some information like minimum path caused to move ii) Problem can be solved by the information which is already given. iii) Example

a) Best first search

b) Greedy search

c) A* search

Uniform cost search expands the node ‘n’ with the lowest path cost instead of expanding the Shallowest node.

49. Define Depth first search.

It expands the deepest node in the current fringe of the search tree.

50. Define depth limited search.

The problem of unbounded tress can be avoided by supplying depth limit 1(i.e.) nodes at depth 1 are treated as if they have no successors. This is called Depth Limited search.

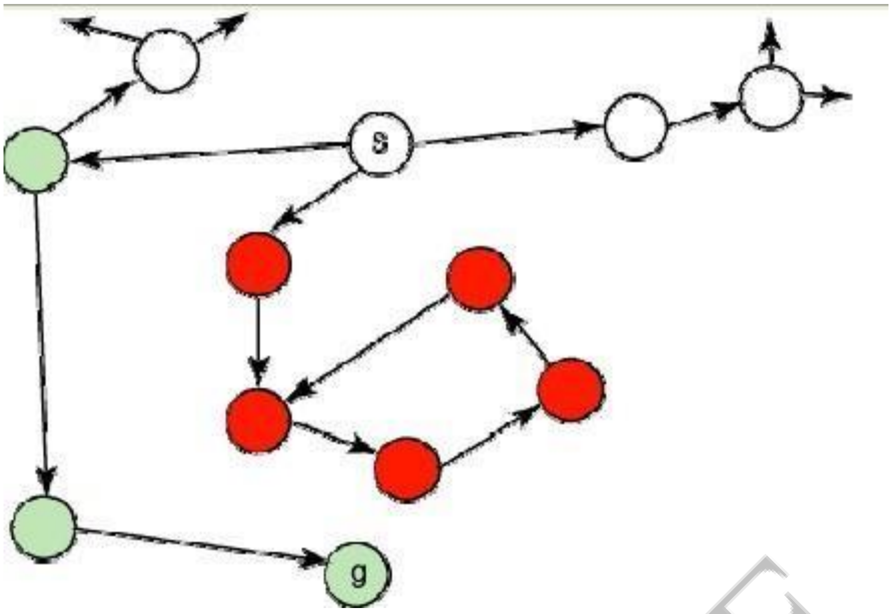
PART – B

1. **Explain in detail about Uninformed Search and Informed Search Strategies.** Uninformed Search Strategies have no additional information about states beyond that provided in the problem definition. Strategies that know whether one non goal state is ―more promising than another are called informed search or heuristic search strategies. There are five uninformed search str ategies as given below. o Breadth-first search o Uniform-cost search o Depth-first search o Depth-limited search o Iterative deepening search

2. Explain in detail about Depth first search Algorithm?

The *h* function can be extended to be applicable to (non-empty) paths. The heuristic value of a path is the heuristic value of the node at the end of the path. That is:
h(⟨no,...,nk⟩)=h(nk)

A simple use of a heuristic function is to order the neighbors that are added to the stack representing the frontier in depth-first search. The neighbors can be added to the frontier so that the



best neighbor is selected first. This is known as heuristic depth first search. This search chooses the locally best path, but it explores all paths from the selected path before it selects another path. Although it is often used, it suffers from the problems of depth-fist search.

Another way to use a heuristic function is to always select a path on the frontier with the lowest heuristic value. This is called best-first search. It usually does not work very well; it can follow paths that look promising because they are close to the goal, but the costs of the paths may keep increasing.

4. Discuss in detail about Constraints Satisfaction Problem.

Constraint satisfaction is the process of finding a solution to a set of constraints that impose conditions that the variables must satisfy. A solution is therefore a set of values for the variables that satisfies all constraints—that is, a point in the feasible region.

The techniques used in constraint satisfaction depend on the kind of constraints being considered. Often used are constraints on a finite domain, to the point that constraint satisfaction problems are typically identified with problems based on constraints on a finite domain. Such problems are usually solved via search, in particular a form of backtracking or local search. Constraint propagation are other methods used on such problems; most of them are incomplete in general, that is, they may solve the problem or prove it unsatisfiable, but not always.Constraint propagation methods are also used in conjunction with search to make a given problem simpler to solve. Other considered kinds of constraints are on real or rational numbers; solving problems on these constraints is done via variable elimination or the simplex algorithm.

Complexity

Solving a constraint satisfaction problem on a finite domain is an NP complete problem with respect to the domain size. Research has shown a number of tractable subcases, some limiting the allowed constraint relations, some requiring the scopes of constraints to form a tree, possibly in a reformulated version of the problem. Research has also established relationship of the constraint satisfaction problem with problems in other areas such as finite model theory.

5. Explain in details about Production System?

Types of Production Systems.

A Knowledge representation formalism consists of collections of condition-action rules(Production Rules or Operators), a database which is modified in accordance with the rules, and a Production System Interpreter which controls the operation of the rules i.e

The 'control mechanism' of a Production System, determining the order in which Production Rules are fired. A system that uses this form of knowledge representation is called a production system.

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 Situation ------------ Action

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. The production system. 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.

EXTRAS (FOR PRACTICE)

1. Define Heuristic function.

2. Name the elements of an agent and list down the characteristics of intelligent agent.

3. How would you rank Production system?

4.How would you quote PEAS description?

5.Apply problem solving algorithm to measure performance.

6.Explain the theme of Backtracking search for CSP.

7.Illustrate the categories of production systems.

8.List the types of constraints

9. Point out some of the uninformed search techniques.

10.How would you formulate Constraint Satisfaction Problem?11.Express what is ridge?

12.What do you infer from hill-climbing search algorithm?

13.Generalize your opinion about admissible heuristic.

14.Define problem solving agents and list its algorithms.

15.Why problem formulation must follow goal Formulation?

16.Summarize the factors that make up rationality.

17.What do you infer from the word Agent?

18.How much knowledge would be required by a perfect program for the problem of playing chess? Assume that unlimited computing power is available.

19.Will you state or interpret in your own words PEAS description for a Vacuum cleaner?

20.Show what would happen if problem is decomposed.

UNIT II KNOWLEDGE REPRESENTATION AND REASONING

1. What is informed search?

One that uses problem – specific knowledge beyond the definition of the problem itself and it can find solutions more efficiently than an uninformed strategy.

2. What is the use of QUEUING_FN?

QUEUING_FN inserts asset of elements into the queue. Different varieties of queuing fn produce different varieties of the search algorithm.

3. Mention the criteria for the evaluation of search strategy.

There are 4 criteria: Completeness, time complexity, space complexity, optimality

4. List the various search strategies.

a. BFS

b. Uniform cost search

c. DFS

d. Depth limited search

e. Iterative deepening search

f. Bidirectional search

5. List the various informed search strategy.

Best first search –greedy search ,A* search

Memory bounded search-Iterative deepening A*search -simplified memory bounded A*search -Iterative improvement search –hill climbing -simulated annealing **6. What is Best First Search?**

Best First Search is an instance of the general TREE SEARCH or GRAPH SEARCH algorithm in which a node is selected for expansion based on an evaluation function, f(n).

7. Define Evaluation function, f(n).

A node with the lowest evaluation is selected for expansion, because evaluation measures distance to the goal.

8. **Define Heuristic function, h (n).** h (n) is defined as the estimated cost of the cheapest path from node n to a goal node.

9. **Define Greedy Best First Search.**

It expands the node that is closest to the goal (i.e.) to reach solution in a quicker way. It is done by using the heuristic function: f(n) = h(n).

10. Define A* search.

A* search evaluates nodes by combining g(n), the cost to reach the node and h(n), the cost to get from the node to the goal. f(n) = g(n) + h(n)

11. Define Admissible heuristic h (n).

In A* search, if it is optimal then, h(n) is an admissible heuristic which means h(n) never overestimates the cost to reach the goal.

12. What is triangle inequality?

It states that each side of a triangle cannot be longer than the sum of the other two slides of the triangle.

13. What are the 2 types of memory bounded heuristic algorithms?

i. Recursive Best First Search(RBFS)
ii. Memory bounded A*(MA*)

14. Differentiate BFS & DFS.

BFS means breath wise search. Space complexity is more. Do not give optimal solution Queuing fn is same as that of queue operator

DFS means depth wise search. Space complexity is less Gives optimal solution Queuing fn is somewhat different from queue operator.

15. What is RBFS?

It keeps track of the f-value of the best alternative path available from any ancestor of the current node. RBFS remembers the f-value of the best leaf in the forgotten sub tree and therefore decide whether its worth re expanding the sub tree sometimes later.

16. Define iterative deepening search.

Iterative deepening is a strategy that sidesteps the issue of choosing the best depth limit by trying all possible depth limits: first depth 0, then depth 1,then depth 2& so on.

17. What are the 2 ways to use all available memory?

i. Memory bounded A*(MA*) ii. Simplified Memory bounded A*(SMA*) **18. What is SMA* search?**

SMA* expands the best leaf until memory is full and it drops the oldest worst leaf node and expands the newest best leaf node.

19. What is called as bidirectional search?

The idea behind bidirectional search is to simultaneously search both forward from the initial state & backward from the goal & stop when the two searches meet in the middle.

20. What is metalevel state space?

Each state in a metalevel state space captures the internal state of a program that is searching in an object level state space.

21. What is Manhattan distance, h2?

The sum of the horizontal and vertical distances of the tiles from their goal positions in a 15 puzzle problem is called Manhattan distance (or) city block distance.

22. Give the drawback of DFS.

The drawback of DFS is that it can get stuck going down the wrong path. Many problems have very deep or even infinite search tree. So dfs will never be able to recover from an unlucky choice at one of the nodes near the top of the tree. So DFS should be avoided for search trees with large or infinite maximum depths **23. Define Branching factor b*.**

Uniform tree of depth d would have to be in order to contain N+1 nodes is called branching factor.

24. Write the time & space complexity associated with depth limited search. Time complexity =O (bd) , b-branching factor, d-depth of tree

Space complexity=o (bl)

25. What is Released problems?

A problem with fewer restrictions on the actions is called a relaxed problem.

26. What is a pattern database?

This database is the storage of exact solution costs for every possible sub problem instance.

27. What is a disjoint pattern database?

The sum of the two costs is still a lower bound on the cost of solving the entire problem is

called a disjoint pattern database.

28. What is local search?

It operates using a single current state rather than multiple paths and generally moves only to neighbors of that state.

29. Define Optimization Problems.

The aim of this problem is to find the best state according to an objective function.

30. What are the 2 parts of Landscape?

i. Location defined by the state.

ii. Elevation defined by the value of the heuristic cost function (or) objective function.

31. Define Global minimum.

If elevation corresponds to cost, then the aim is to find the lowest valley is called global minimum.

32. Define Global Maximum.

If elevation corresponds to an objective function, then the aim is to find the highest peak is called global maximum. **33. Define Hill Climbing search.**

It is a loop that continually moves in a increasing value direction (i.e.) up hill and terminates when it reaches a “peak” where no neighbor has a higher value.

34. List some drawbacks of hill climbing process.

Local maxima: A local maxima as opposed to a goal maximum is a peak that is lower that the highest peak in the state space. Once a local maxima is reached the algorithm will halt even though the solution may be far from satisfactory.

Plateaux: A plateaux is an area of the state space where the evaluation fn is essentially flat. The search will conduct a random walk.

35. What is the meaning for greedy local search?

It goals (picks) a good neighbor state without thinking ahead about where to go next.

36. Define Local maxima.

A local maximum is a peak that is higher than each of its neighboring states, but lower than the global maximum.

37. What are the variants of hill climbing?

i. Stochastic hill climbing ii. First choice hill climbing iii. Simulated annealing search iv. Local beam search

v. Stochastic beam search

38. Define annealing.

Annealing is the process used to harden metals (or) glass by heating them to a high temperature and then gradually cooling them, thus allowing the material to coalesce into a low energy crystalline state.

39. Define simulated annealing.

This algorithm, instead of picking the best move, it picks a random move. If the move improves the situation, it is always accepted.

40. What is the advantage of memory bounded search techniques?

We can reduce space requirements of A* with memory bounded algorithm such as IDA* & SMA*.

41. Give the procedure of IDA* search.

Minimize f(n)=g(n)+h(n) combines the advantage of uniform cost search + greedy search A* is Complete optimal. Its space complexity is still prohibitive.

Iterative improvement algorithms keep only a single state in memory, but can get stuck on local maxima. In this algorithm each iteration is a dfs just as in regular iterative deepening. The depth first search is modified to use an f-cost limit rather than a depth limit. Thus each iteration expands all nodes inside the contour for the current f-cost.

42. List some properties of SMA* search.

* It will utilize whatever memory is made available to it.

* It avoids repeated states as for as its memory allow.

* It is complete if the available memory is sufficient to store the shallowest path.

* It is optimal if enough memory is available to store the shallowest optimal solution path.

Otherwise it returns the best solution that can be reached with the available memory.

*When enough memory is available for entire search tree, the search is optimally efficient. *Hill climbing.

*Simulated annealing.

43. What is Genetic Algorithms?

Genetic Algorithm is a variant of stochastic beam search in which successor states are generated by combining two parent states, rather than by modifying a single state.

44. Define Online Search agent.

Agent operates by interleaving computation and action (i.e.) first it takes an action, and then it observes the environment and computes the next action.

45. What are the things that agent knows in online search problems? a. Actions(s)

b. Step cost function C(s, a, s’)

c. Goal TEST(s)

46. Define CSP.

Constraint Satisfaction problem (CSP) is defined by a set of variables X1,X2,…Xn and set of constraints C1,C2,…Cm.

47. Define Successor function.

A value can be assigned to any unassigned variable, provided that does not conflict with previously assigned variables.

48. What are the types of constraints?

There are 5 types,

a. Unary constraints relates one variable.

b. A binary constraint relates two variables.

c. Higher order constraints relate more than two variables.

d. Absolute constraints.

e. Preference constraints.

49. Define MRV.

Minimum remaining values heuristic chooses the variable with the fewest "legal" values.

50. Define LCV.

Least constraining value heuristic prefers the value that rules out the fewest choices for the Neighboring variables in the constraint graph.

51. Define Conflict directed back jumping.

A back jumping algorithm that uses conflict sets defined in this way is called Conflict

directed back jumping.

52. Define constraint propagation.

It is the general term for propagating (i.e.) spreading the implications of constraints on the variable on to other variable.

53. Define Cycle cut set.

The process of choosing a subset S from variables [CSP] such that the constraint graph becomes a tree after removal of S. S is called a cycle cut set.

54. Define Tree decomposition.

The constraint graph is divided into a set of connected sub problems. Each sub problem is solved independently and the resulting solutions are then combined. This process is called tree decomposition.

55. Define Alpha beta pruning.

Alpha beta pruning eliminates away branches that cannot possibly influence the final decision

PART – B

1. Describe the Issues in knowledge representation?

Typically, a problem to solve or a task to carry out, as well as what constitutes a solution, is only given informally, such as "deliver parcels promptly when they arrive" or "fix whatever is wrong with the electrical system of the house.

To solve a problem, the designer of a system must flesh out the task and determine what constitutes a solution; represent the problem in a language with which a computer can reason; use the computer to compute an output, which is an answer presented to a user or a sequence of actions to be carried out in the environment; and interpret the output as a solution to the problem. Knowledge is the information about a domain that can be used to solve problems in that domain. To solve many problems requires much knowledge, and this knowledge must be represented in the computer. As part of designing a program to solve problems, we must define how the knowledge will be represented. A representation scheme is the form of the knowledge that is used in an agent. A representation of some piece of knowledge is the internal representation of the knowledge. A representation scheme specifies the form of the knowledge.

A knowledge base is the representation of all of the knowledge that is stored by an agent.

2. Explain in details about first-order logic?

Whereas propositional logic assumes the world contains facts,

- first-order logic (like natural language) assumes the world contains
- Objects: people, houses, numbers, colors, baseball games, wars, … • Relations: red, round, prime, brother of, bigger than, part of, comes between, … Atomic sentences

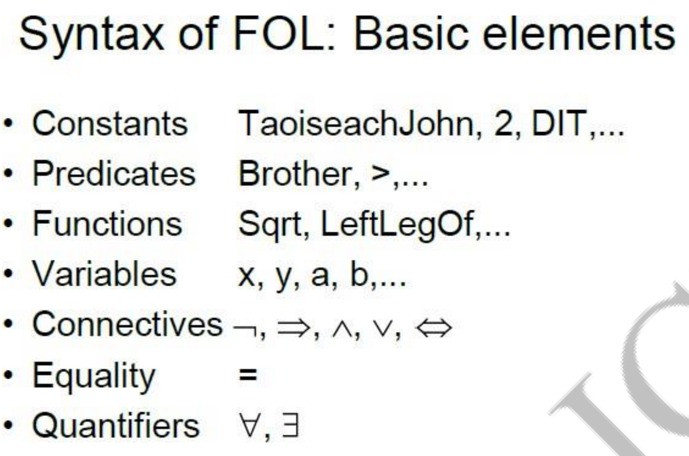
Atomic sentence = *predicate* (*term1,...,termn*) or *term1 = term2*

Term = *function* (*term1,...,termn*) or *constant* or *variable*

• E.g., *Brother(TaoiseachJohn,RichardTheLionheart) >*

(Length(LeftLegOf(Richard)),

Length(LeftLegOf(TaoiseachJohn)))

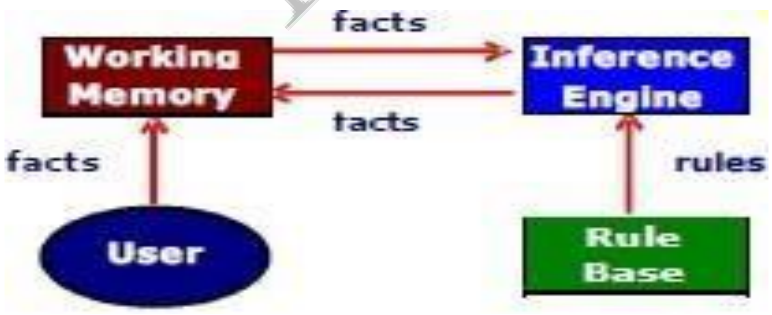


Syntax of FOL: Basic elements

• Constants	TaoiseachJohn, 2, DIT,...
• Predicates	Brother, >,...
• Functions	Sqrt, LeftLegOf,...
• Variables	x, y, a, b,...
• Connectives	¬, ⇒, ∧, ∨, ⇔
• Equality	=
• Quantifiers	∀, ∃

3. Illustrate in detail about forward and backward chaining with suitable example.

Forward Chaining: The Forward chaining system, properties , algorithms, and conflict resolution strategy are illustrated.



‡ facts are held in a working memory

‡ condition-action rules represent actions to be taken when specified facts occur in working memory.

‡ typically, actions involve adding or deleting facts from the working memory.

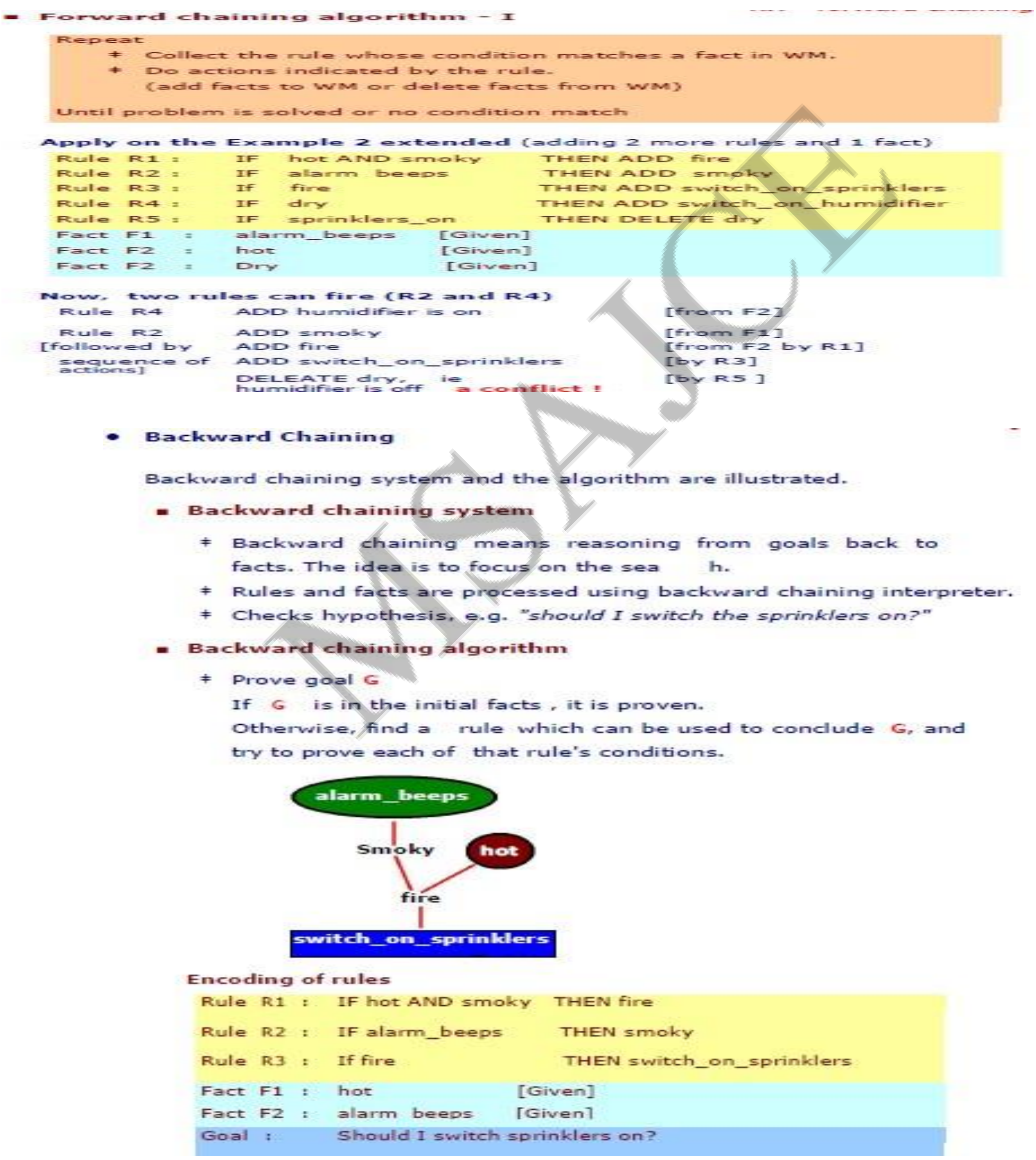
■ Properties of Forward Chaining

‡ all rules which can fire do fire.

‡ can be inefficient - lead to spurious rules firing, unfocused problem solving

‡ set of rules that can fire known as conflict set.

‡ decision about which rule to fire is conflict resolution.



■ **Forward chaining algorithm - I**

```
Repeat
    + Collect the rule whose condition matches a fact in WM.
    + Do actions indicated by the rule.
      (add facts to WM or delete facts from WM)
Until problem is solved or no condition match
```

Apply on the Example 2 extended (adding 2 more rules and 1 fact)

Rule R1 :	IF	hot AND smoky	THEN ADD fire
Rule R2 :	IF	alarm beeps	THEN ADD smoky
Rule R3 :	If	fire	THEN ADD switch_on_sprinklers
Rule R4 :	IF	dry	THEN ADD switch_on_humidifier
Rule R5 :	IF	sprinklers_on	THEN DELETE dry
Fact F1 :		alarm_beeps	[Given]
Fact F2 :		hot	[Given]
Fact F2 :		Dry	[Given]

Now, two rules can fire (R2 and R4)

Rule R4	ADD humidifier is on	[from F2]
Rule R2	ADD smoky	[from F1]
[followed by	ADD fire	[from F2 by R1]
sequence of actions]	ADD switch_on_sprinklers	[by R3]
	DELEATE dry, ie humidifier is off a conflict !	[by R5]

- **Backward Chaining**

Backward chaining system and the algorithm are illustrated.

■ **Backward chaining system**

‡ Backward chaining means reasoning from goals back to facts. The idea is to focus on the sea h.

‡ Rules and facts are processed using backward chaining interpreter.

‡ Checks hypothesis, e.g. *"should I switch the sprinklers on?"*

■ **Backward chaining algorithm**

‡ Prove goal G

If G is in the initial facts , it is proven.

Otherwise, find a rule which can be used to conclude G, and try to prove each of that rule's conditions.

Encoding of rules

Rule R1 :	IF hot AND smoky	THEN fire
Rule R2 :	IF alarm_beeps	THEN smoky
Rule R3 :	If fire	THEN switch_on_sprinklers
Fact F1 :	hot	[Given]
Fact F2 :	alarm beeps	[Given]
Goal :	Should I switch sprinklers on?	

4. Knowledge engineering in FOL

1. Identify the task

2. Assemble the relevant knowledge

3. Decide on a vocabulary of predicates, functions, and constants

4. Encode general knowledge about the domain

5. Encode a description of the specific problem instance

6. Pose queries to the inference procedure and get answers

7. Debug the knowledge base

5. Explain in details about Predicate Calculus?

- An interpretation over D is an assignment of the entities of D to each of the constant, variable, predicate and function symbols of a predicate calculus expression such that:
- 1: Each constant is assigned an element of D
- 2: Each variable is assigned a non-empty subset of D;(these are the allowable substitutions for that variable)
- 3: Each predicate of arity n is defined on n arguments from D and defines a mapping from Dn into {T,F}
- 4: Each function of arity n is defined on n arguments from D and defines a mapping from Dn into D

Truth Value of Predicate Calculus expressions

- Assume an expression E and an interpretation I for E over a non empty domain D. The truth value for E is determined by:
- The value of a constant is the element of D assigned to by I
- The value of a variable is the set of elements assigned to it by I

Similarity with Propositional logic truth values

- The value of the negation of a sentence is F if the value of the sentence is T and F otherwise

UNIT III

UNCERTAINTY

1. Define FOL.

FOL is a first order logic. It is a representational language of knowledge which is powerful than propositional logic (i.e.) Boolean Logic. It is an expressive, declarative, compositional language.

2. Define a knowledge Base:

Knowledge base is the central component of knowledge base agent and it is described as a set of representations of facts about the world.

3. With an example, show objects, properties functions and relations. Example

“EVIL KING JOHN BROTHER OF RICHARD RULED ENGLAND IN 1200”

Objects : John, Richard, England, 1200

Relation : Ruled

Properties : Evil, King

Functions : BROTHER
OF **4. Define a Sentence?**

Each individual representation of facts is called a sentence. The sentences are expressed in a language called as knowledge representation language.

5. Define an inference procedure

An inference procedure reports whether or not a sentence is entiled by knowledge base

provided a knowledge base and a sentence .An inference procedure ‘i’ can be described by the sentences that it can derive. If i can derive from knowledge base, we can write. KB -- Alpha is derived from KB or i derives alpha from KB.

6. Define Ontological commitment.

The difference between propositional and first order logic is in the ontological commitment. It assumes about the nature of reality.

7. Define Epistemological commitment.

The logic that allows the possible states of knowledge with respect to each fact.

8. Define domain and domain elements.

The set of objects is called domain, sometimes these objects are referred as domain elements.

9. What are the three levels in describing knowledge based agent?

Logical level

Implementation level

Knowledge level or epistemological level

10. Define Syntax?

Syntax is the arrangement of words. Syntax of a knowledge describes the possible configurations that can constitute sentences. Syntax of the language describes how to make sentences.

11. Define Semantics

The semantics of the language defines the truth of each sentence with respect to each possible world. With this semantics, when a particular configuration exists with in an agent, the agent believes the corresponding sentence.

12. Define Logic

Logic is one which consist of i. A formal system for describing states of affairs, consisting of a) Syntax b)Semantics.

ii. Proof Theory – a set of rules for deducing the entailment of a set sentences.

13. What are the 3 types of symbol which is used to indicate objects, relations and functions?

i) Constant symbols for objects ii) Predicate symbols for relations iii) Function symbols for functions **14. Define terms.**

A term is a logical expression that refers to an object. We use 3 symbols to build a term.

15. Define Atomic sentence.

Atomic sentence is formed by both objects and relations.

Example

Brother (William, Richard)

i. Logical Constants (True, False)

16 .Define Modus Ponen's rule in Propositional logic?

The standard patterns of inference that can be applied to derive chains of conclusions that lead to the desired goal is said to be Modus Ponen's rule. **17 .Define AND –Elimination rule in propositional logic**

AND elimination rule states that from a given conjunction it is possible to inference any of the conjuncts. OR-Introduction rule states that from, a sentence, we can infer its disjunction with anything.

18. Define Unification.

Lifted Inference rule require finding substitutions that make different logical expressions look identical (same). This is called Unification.

19. Define Occur check.

When matching a variable in 2 expressions against a complex term, one must check whether the variable itself occurs inside the term, If it does the match fails. This is called occur check.

20. Define pattern matching.

The inner loop of an algorithm involves finding all the possible unifiers with facts in the KB. This is called pattern matching.

21. Explain the function of Rete Algorithm?

This algorithm preprocess the set of rules in KB to constant a sort of data flow network in which each node is a literals from rule a premise.

22. Define magic set.

To rewrite the rule set, using information from the goal, so that only relevant variable bindings called magic set.

23. Define backward chaining.

This algorithm works backward from the goal, chaining through rules to find known facts that support the proof.

24. Define Prolog program.

It is a set of definite clauses written in a notation somewhat different from standard FOL.

25. What are the divisions of knowledge in OTTER theorem?

i. Set of Support (SOS)
ii. Usable axioms

iii. Rewrites (or) Demodulators iv.
A set of parameters and sentences

26. What are the 2 types of frame problem?

i. Representational Frame Problem ii. Inferential Frame Problem

27. What are the 2 types of processes?

i. Discrete events – it have definite structure ii.
Liquid events - Categories of events with process.

28. Define fluent calculus.

Discard Situation Calculus and invent a new formalism for writing axioms is Called Fluent Calculus.

29. What is important for agent?

Time (i.e.) intervals is important for agent to take an action. There are 2 kinds; i. Moments ii. Extended Intervals **30. Define runtime variables.**

Plans to gather and use information are represented using short hand Notation called runtime variables (n).

Example [Look up (Agent, “Phone number (Divya)”.N), Dial (n)]

PART – B

1. Describe about types of Knowledge representation?
There are four types of Knowledge representation :

Relational, Inheritable, Inferential, and Declarative/Procedural.

◊ **Relational Knowledge :** provides a framework to compare two objects based on equivalent attributes. any instance in which two different objects are compared is a relational type of knowledge.

◊ **Inheritable Knowledge**

− is obtained from associated objects.

− it prescribes a structure in which new objects are created which may inherit all or a subset of attributes from existing objects.

◊ **Inferential Knowledge**

− is inferred from objects through relations among objects. − e.g., a word alone is a simple syntax, but with the help of other words in phrase the reader may infer more from a word; this inference within linguistic is called semantics.

◊ **Declarative Knowledge**

− a statement in which knowledge is specified, but the use to which that knowledge is to be put is not given.

− e.g. laws, people's name; these are facts which can stand alone, not dependent on other knowledge;

Procedural Knowledge

− a representation in which the control information, to use the knowledge, is embedded in the knowledge itself.

− e.g. computer programs, directions, and recipes; these indicate specific use or implementation. − a representation in which the control information, to use the knowledge, is embedded in the knowledge itself.

− e.g. computer programs, directions, and recipes; these indicate specific use or implementation.

2. Discuss in details about inferential knowledge?

This knowledge generates new information from the given information.

This new information does not require further data gathering form soue, 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.

− a predicate logic (a mathematical deduction) is used to infer from a set of attributes.

− inference through predicate logic uses a set of logical operations to relate individual data.

− the symbols used for the logic operations are :

" → " (implication), " ¬ " (not), " V " (or), " Λ " (and),
" ∀ " (for all), " ∃ " (there exists).

Examples of predicate logic statements :

1. *"Wonder"* is a name of a dog : **dog (wonder)**

2. All dogs belong to the class of animals : ∀ **x : dog (x) → animal(x)** 3.All animals either live on land or in ∀ **x : animal(x) → live (x,** water : **land) V live (x, water)**

From these three statements we can infer that
: *"* ***Wonder*** **lives either on land or on**
water.*"* **3.** ***Discuss about*** **Predicate Logic ?**

Predicate Logic

The propositional logic, is not powerful enough for all types of assertions;
Example : The assertion **"x > 1"**, where **x** is a variable, is not a proposition
because it is neither true nor false unless value of **x** is defined.

For **x > 1** to be a proposition ,

− either we substitute a specific number for **x** ;

− or change it to something like

"There is a number x for which x > 1 holds"; − or **"For every number x, x > 1 holds"**. Consider example :

**" All men are mortal.
Socrates is a man.**

Then Socrates is mortal" ,

These cannot be expressed in propositional logic as a finite and logically valid argument (formula).

We need languages : that allow us to describe properties (*predicates*) of objects, or a relationship among objects represented by the variables .

Predicate logic satisfies the requirements of a language.

− *Predicate logic* is powerful enough for expression and reasoning.
− *Predicate logic* is built upon the ideas of *propositional logic.*

Predicate :

Every complete "sentence" contains two parts : a "subject" and a "predicate".

The *subject* is what (or whom) the sentence is about.

The *predicate* tells something about the subject;
Example :

A *sentence* **"Judy {runs}".**

The subject is **Judy** and the predicate is **runs** .

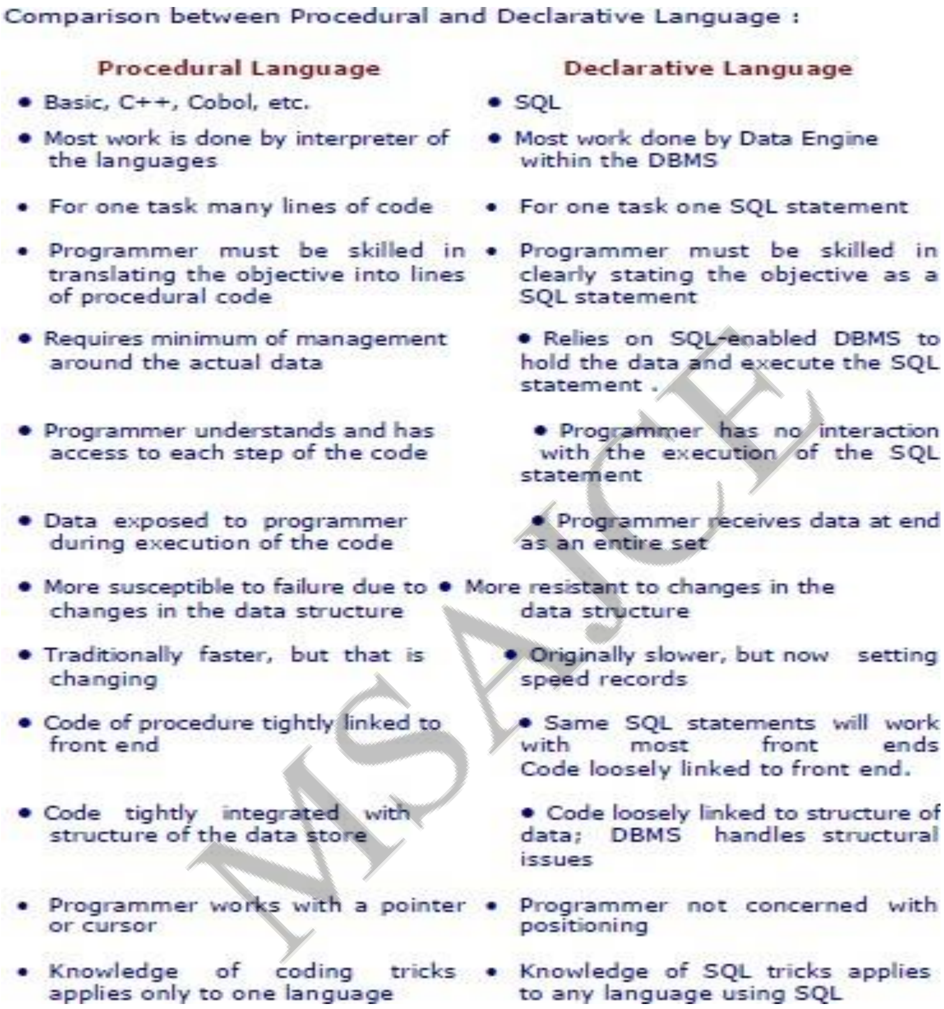
Predicate, always includes verb, tells something about the subject.

Predicate is a verb phrase template that describes a property of objects, or a relation among objects represented by the variables. Example:

"The car Tom is driving *is blue*"; "The sky *is blue*" ;"The cover of this book *is blue*" Predicate is **"is blue"** , describes property.

Predicates are given names; Let **„B‟** is name for predicate **"is_blue"**. Sentence is represented as **"B(x)"** , read as **"x is blue"**; Symbol **“x”** represents an arbitrary Object .

4. Comparison between procedural and declarative language?



Comparison between Procedural and Declarative Language :

Procedural Language	Declarative Language
• Basic, C++, Cobol, etc.	• SQL
• Most work is done by interpreter of the languages	• Most work done by Data Engine within the DBMS
• For one task many lines of code	• For one task one SQL statement
• Programmer must be skilled in translating the objective into lines of procedural code	• Programmer must be skilled in clearly stating the objective as a SQL statement
• Requires minimum of management around the actual data	• Relies on SQL-enabled DBMS to hold the data and execute the SQL statement .
• Programmer understands and has access to each step of the code	• Programmer has no interaction with the execution of the SQL statement
• Data exposed to programmer during execution of the code	• Programmer receives data at end as an entire set
• More susceptible to failure due to changes in the data structure	• More resistant to changes in the data structure
• Traditionally faster, but that is changing	• Originally slower, but now setting speed records
• Code of procedure tightly linked to front end	• Same SQL statements will work with most front ends Code loosely linked to front end.
• Code tightly integrated with structure of the data store	• Code loosely linked to structure of data; DBMS handles structural issues
• Programmer works with a pointer or cursor	• Programmer not concerned with positioning
• Knowledge of coding tricks applies only to one language	• Knowledge of SQL tricks applies to any language using SQL

UNIT IV LEARNING

1. Why does uncertainty arise ?

Agents almost never have access to the whole truth about their environment.

Agents cannot find a caterorial answer.

Uncertainty can also arise because of incompleteness, incorrectness in agents understanding of properties of environment.

2. **State the reason why first order, logic fails to cope with that the mind like medical diagnosis.**

Three reasons

a.laziness: o it is hard to lift complete set of antecedents of consequence, needed to

ensure and exceptionless rule.

b. Theoritical Ignorance: o medical science has no complete theory for the domain. Practical ignorance: even if we know all the rules, we may be uncertain about a particular item needed. **3. Define the term utility?**

The term utility is used in the sense of "the quality of being useful .", utility of a state is relative to the agents, whose preferences the utility function is supposed to represent.

4. **What is the need for probability theory in uncertainty ?**

Probability provides the way of summarizing the uncertainty that comes from our laziness and ignorance . Probability statements do not have quite the same kind of semantics known as evidences.

5. **What is the need for utility theory in uncertainty?**

Utility theory says that every state has a degree of usefulness, or utility to in agent, and that the agent will prefer states with higher utility. The use utility theory to represent and reason with preferences.

6. **What is called as principle of maximum expected utility ?**

The basic idea is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over all the possible outcomes of the action. This is known as

MEU.

7. **What Is Called As Decision Theory ?**

Preferences As Expressed by Utilities Are Combined with Probabilities in the General Theory of Rational Decisions Called Decision Theory.

Decision Theory = Probability Theory + Utility Theory.

8. **Define Prior Probability?** p(a) for the Unconditional or Prior Probability Is That the Proposition A is True. It is important to remember that p(a) can only be used when there is no other information.

9. **Define conditional probability?**

Once the agents has obtained some evidence concerning the previously unknown propositions making up the domain conditional or posterior probabilities with the notation p(A/B) is used. This is important that p(A/B) can only be used when all be is known.

10. Define probability distribution:

Eg. P(weather) = (0.7,0.2,0.08,0.02). This type of notations simplifies many equations.

11. What is an atomic event?

An atomic event is an assignment of particular values to all variables, in other words, the complete specifications of the state of domain.

12. Define joint probability distribution

This completely specifies an agent's probability assignments to all propositions in the domain. The joint probability distribution p(x1,x2,--------xn) assigns probabilities to all possible atomic events; where

X1,X2 ----- Xn 10 =variables.

13. Give the Baye's rule equation

W.K.T P(A ^ B) = P(A/B) P(B)------------------------- 1

P(A ^ B) = P(B/A) P(A) ------------------------- 2

DIVIDING BY P(A) ;

WE GET

P(B/A) = P(A/B) P(B) -------------------- P(A)

14. What is meant by belief network?

A belief network is a graph in which the following holds

A set of random variables

A set of directive links or arrows connects pairs of
nodes. The conditional probability table for each node
The graph has no directed cycles.

15. **What are the ways in which one can understand the semantics of a belief network?** There are two ways to see the network as a representation of the joint probability distribution to view it as an encoding of collection of conditional independence statements.

16. **What is the basic task of a probabilistic inference?**

The basic task is to reason in terms of prior probabilities of conjunctions, but for the most part, we will use conditional probabilities as a vehicle for probabilistic inference.

17. **What are called as Poly trees?**

The algorithm that works only on singly connected networks known as Poly trees. Here at most one undirected path between any two nodes is present.

18. **Define casual support**

E+X is the casual support for X- the evidence variables "above" X that are connected to X through its parent.

19. **Define evidential support**

E-X is the evidential support for X- the evidence variables "below" X that are connected to X through its children.

20. **What is called as multiple connected graph?**

A multiple connected graph is one in which two nodes are connected by more than one path.

21. What is the purpose of learning?

The idea behind learning is that percepts should be used not only for acting but also for improving the agent’s ability to act in the future.

22. What are issues in learning element?

i. Component
ii. Feedback
iii.
Representation

23. What are the types of machine learning?

i. Supervised Learning

ii. Unsupervised Learning iii.
Reinforcement Learning

24. Differentiate Supervised and	**Unsupervised Learning**
It involves learning a function from examples of its inputs And outputs	It involves learning patterns in the input when no specific output values are supplied.
Example: Applying Brake on the wet road, we can even skip on the road is a result.	Example: Day by day agent will learn about “Good traffic days” and “Bad traffic days” without any advice.

25. Define Reinforcement Learning.

This Learning is rather than being told what to do by teacher, a reinforcement learning agent must learn from occasional rewards.

Example

If taxi driver does not get a tip at the end of journey, it gives him a indication that his behavior is undesirable.

26. Define Inductive Learning.

An algorithm for supervised learning is given as input the correct value of the unknown function for particular inputs and it must try to recover the unknown function.

27. Define Classification Learning.

Learning a discrete valued function is called is called classification learning.

28. Define Regression learning.

Learning a continuous valued function is called regression learning.

29. What is parity and majority function?

Parity Function : It Returns 1 if and only if an even number of inputs are 1. Majority function : It Returns 1 if more than half of its inputs are 1.

30. What is training set?

The complete set of examples is called the training set.

Example

Restaurant problem
Goal predicate "will wait"

31. Define Information gain.

Information gain from the attribute test is the difference between the original information requirement and the new requirement.

Gain (A) = I(p/(p+n)), n/ (p+n)) – Remainder(A)

32. What is test set?

Prediction is good if it turns out to be true, so can assess quality of hypotheses by

Checking its predictions against the correct classification once we know it. We do this on a set of examples is known as Test Set.

33. What is over fitting?

Whenever there is a large set of possible hypotheses, one has to be careful not to use the resulting freedom to find meaningless "regularity" in the data. This problem is called over fitting.

34. What is the purpose of cross validation?

It reduces over fitting. It can be applied to any learning algorithm, not just decision tree learning. The basic idea is to estimate how well each hypotheses will predict unseen data.

35. Mention the exercises which broaden the applications of decision trees.

i. Missing data ii. Multivalued attributes iii. Continuous and integer valued input attributes
iv. Continuous valued output attributes.

36. What is ensemble learning?

The idea of this learning is to select a whole collection or ensemble, of hypotheses from the hypotheses space and combine their predictions.

37. Define PAC – Learning Algorithm.

An learning algorithm that return hypotheses that are approximately correct is called PAC learning algorithm. **38. Define Decision list.**

It is a logical expression of a restricted form, It consists of a series of tests, each of which conjunction of literals. If test succeeds, value is returned. If test fails, processing continues with the next test in the list.

39. What is the purpose of current best hypotheses search? This search is to maintain a single hypothes

40. Differentiate generalization and specialization.

The extension of the hypotheses must be increased to include it. This is called generalization. The extension of the hypotheses must be decreased to exclude the example. This is called specialization.

41. Define Boundary set.

Each boundary will not be a point but rather a set of hypotheses called a Boundary set.

42. What are the two boundary sets?

i. G Set : a most general boundary set.
ii. S Set : a most specific boundary set.

43. Show the relationship of an entailment constraint. Hypothesis ^ Descriptions |= classifications **44. Define EBL.**

Explanation based learning, from the prior knowledge (or) information; we can infer a general rule. This kind of generalization process called explanation based learning (or) EBL.

45. What is the entailment constraints satisfied by EBL?

Hypothesis ^ Description |= classification Background |= Hypothesis **46. Define RBL.**

Relevance based Learning; the prior knowledge background concerns the relevance of a set of features to the goal predicate. This knowledge together with the observations, Allows the agent to infer a new, general rule that explains the observations.

Hypothesis ^ Description |= classifications,

Background ^ Description ^ classifications |= Hypothesis.

47. Define knowledge based Inductive learning.

KBIL algorithm finds inductive hypotheses that explain sets of observations with the help of background knowledge.

48. Define Inductive Logic Programming (ILP).

ILP techniques perform KBIL on knowledge that is expressed in first order logic. ILP methods can learn relational knowledge that is not expressible in attribute based systems.

49. What is the purpose of memorization?

Memorization used in computer science to speed up programs by saving the results of computation. The basic idea of memo function is to accumulate a database of input and output pairs, when the function is called; it first checks the database to see whether it can avoid solving the problem.

50. What is the basic EBL process step?

i. Construct a proof using the available background knowledge. ii. Construct a generalized proof tree for the variabilized goal using the same inference steps as in the original proof.

iii. Construct a new rule where LHS consists of the leaves of the proof tree and R.H.S is the variabilized goal. iv. Drop any conditions that are true.

51. Define constructive induction algorithm.

Algorithms that can generate new predicates are called constructive induction algorithms.

52. What are the two main subroutines used for generating literals?

i. NEW – LITERALS
ii. CHOOSE - LITERALS

53. What are the 3 kinds of literals that can be added?

i. Literals using Predicate ii. Equality and inequality literals iii. Arithmetic comparisons **54. Define Bayesian Learning.**

It calculates the probability of each hypotheses, given the data and makes predictions on that basis,

(i.e.) predictions are made by using all the hypotheses, weighted by their probabilities rather than by using just single "best" hypotheses.

55. Define MAP.

Maximum A Posteriori. A very common approximation is to make predictions based on single most probable hypotheses. This is MAP.

56. Define MDL.

The MDL (Maximum Description Length), is a learning method which attempts to minimize the size of the hypotheses and data encodings rather than work with probabilities.

57. What is Maximum – Likelihood hypotheses?

ML – it is reasonable approach when there is no reason to prefer one hypotheses over another a prior.

58. What are the methods for maximum likelihood parameter learning?

i. Write down an expression for the likelihood of the data as a function of the parameter.

ii. Write down the derivative of the log likelihood with respect to each parameter.

iii. Find the parameter values such that the derivatives are zero.

59. Define Naïve Bayes model.

In this model, the "class" variable C is the root and the "attribute" variable XI are the leaves. This model assumes that the attributes are conditionally independent of each other, given the class.

60. **Define sum of squared errors.** The difference between the actual value yj and the predicated value (θ1 xj + θ2) so E is the sum of squared errors.

61. Define EM.

Expectation Maximization: the idea of EM is to pretend that we know the parameters of the model and then to infer the probability that each data point belongs to each component. After that we refit the components to the data, where each component is fitted to the entire data set with each point weighted by the probability.

62. What are the 2 steps in mixture model parameters?

i. E –
Step ii. M
– Step

63. Define Neural Networks.

It consists of nodes or units connected by directed links. A link propagates the activation. Each link has a numeric weight which determines the strength and sign of the connection.

64. Give Activation function. n ai = g(ini) = g(Σ Wj,i aj) j=0

65. What are the two functions in Neural network's Activation functions? i. Threshold function ii. Sigmoid function

66. What are the categories of neural network structures?

i. Acyclic (or) Feed – forward networks
ii. Cyclic (or) Recurrent Networks

67. What is single layer feed forward neural network?

A network with all the inputs connected directly to the outputs is called a single layer neural network or a perceptron networks.

68. What is multilayer feed forward neural networks?

It consists of many hidden units. Each hidden unit act as a perceptron that represents a soft threshold functions in the input space. Output unit act as a soft threshold linear combination of several such functions. **69. Define Passive learning.** The agent's policy is fixed and the task is to learn the utilities of states, this could also involve learning a model of the environment.

70. Define Active Learning.

The agent must learn what to do. An agent must experience as much as possible of its environment in order to learn how to behave in it.

71. Define TD.

Temporal Difference learning: The key of TD is to use the observed transitions to adjust the values of the observed states so that they agree with the constraint equations. **Part - B**

1. What are the components of planning system and explain in detail.

Reasoning is the act of deriving a conclusion from certain premises using a given methodology. • Reasoning is a process of thinking; reasoning is logically arguing; reasoning is drawing inference.

- When a system is required to do something, that it has not been explicitly told how to do, it must reason. It must figure out what it needs to know from what it already knows.

- Many types of Reasoning have long been identified and recognized, but many questions regarding their logical and computational properties still remain controversial.

- The popular methods of Reasoning include abduction, induction, model based, explanation and confirmation. All of them are intimately related to problems of belief revision and theory development, knowledge assimilation, discovery and learning.

List the Machine learning algorithms and explain in detail

A formal language may be viewed as being analogous to of words or a collection of sentences.

AI – Reasoning: a collection

In computer science, a formal language is defined by precise mathematical or machine process able formulas.

‡ A formal langu;age **L** is characterized as a set **F** of finite-length sequences of elements drawn from a specified finite set **A** of symbols.

‡ Mathematically, it is an unordered pair **L = { A, F }**

‡ If **A** is words then the set **A** is called alphabet of **L**, and the elements of **F** are called words.
‡ If **A** is sentence then the set **A** is called the lexicon or vocabulary of **F**, and the elements of **F** are then called sentences.

‡ The mathematical theory that treats formal languages in general is known as **formal language theory**.

2. List out the different Methods of Reasoning?

Mostly three kinds of logical reasoning: Deduction, Induction, Abduction.

■ **Deduction**

- Example: "When it rains, the grass gets wet. It rains. Thus, the grass is wet."
- This means in determining the conclusion; it is using rule and its precondition to make a conclusion.
- Applying a general principle to a special case.
- Using theory to make predictions
- Usage: Inference engines, Theorem provers, Planning.

■ **Induction**

- Example: "The grass has been wet every time it has rained. Thus, when it rains, the grass gets wet."
- This means in determining the rule; it is learning the rule after numerous examples of conclusion following the precondition.
- Deriving a general principle from special cases
- From observations to generalizations to knowledge
- Usage: Neural nets, Bayesian nets, Pattern recognition

AI - Reasoning

Abduction

‡ Example: "When it rains, the grass gets wet. The grass is wet, it must have rained."
Means determining the precondition; it is using the conclusion and the rule to support that the precondition could explain the conclusion.

‡ Guessing that some general principle can relate a given pattern of cases

‡ Extract hypotheses to form a tentative theory

‡ Usage: Knowledge discovery, Statistical methods, Data mining.

■ **Analogy**

‡ Example: "An atom, with its nucleus and electrons, is like the sola system, with its sun and planets."

Means analogous; it is illustration of an idea by means of a more familiar idea that is similar to it in some significant features. And thus said to be analogous to it.

‡ finding a common pattern in different cases

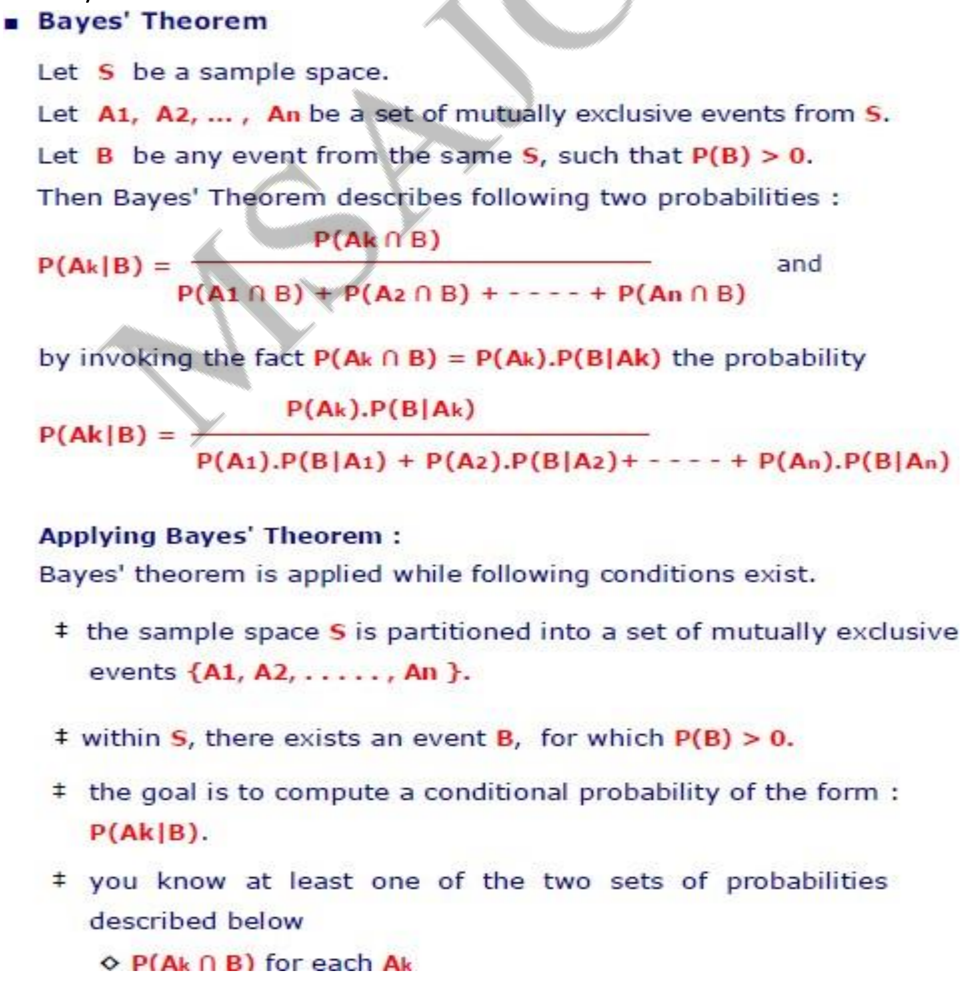
‡ usage: Matching labels, Matching sub-graphs, Matching transformations.

Note: Deductive reasoning and Inductive reasoning are the two most commonly used explicit methods of reasoning to reach a conclusion.

3. Describe Bayes' Theorem?

It is a measure of the plausibility of an event given incomplete knowledge.

- Bayes' theorem is also known as Bayes' rule or Bayes' law, or called Bayesian reasoning.
- The probability of an event A conditional on another event B ie P(A|B) is generally different from probability of B conditional on A ie P(B|A).
- There is a definite relationship between the two, P(A|B) and P(B|A), and Bayes' theorem is the statement of that relationship.
- Bayes theorem is a way to calculate P(A|B) from a knowledge of P(B|A).
- Bayes' Theorem is a result that allows new information to be used to update the conditional probability of an event.



■ **Bayes' Theorem**

Let S be a sample space.

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

Let B be any event from the same S, such that $P(B) > 0$.

Then Bayes' Theorem describes following two probabilities :

$$P(A_k|B) = \frac{P(A_k \cap B)}{P(A_1 \cap B) + P(A_2 \cap B) + \dots + P(A_n \cap B)}$$ and

by invoking the fact $P(A_k \cap B) = P(A_k).P(B|A_k)$ the probability

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

Applying Bayes' Theorem :

Bayes' theorem is applied while following conditions exist.

‡ the sample space S is partitioned into a set of mutually exclusive events $\{A1, A2, \dots, An\}$.

‡ within S, there exists an event B, for which $P(B) > 0$.

‡ the goal is to compute a conditional probability of the form : $P(Ak|B)$.

‡ you know at least one of the two sets of probabilities described below

- ◇ $P(A_k \cap B)$ for each A_k

4. Discuss in details about Rule Based Systems

A rule is an expression of the form **"if A then B"** where

A is an assertion and **B** can be either an action or another assertion.

Example : Trouble shooting of water pumps

- If pump failure then the pressure is low
- If pump failure then check oil level
- If power failure then pump failure
- Rule based system consists of a library of such rules.
- Rules reflect essential relationships within the domain.
- Rules reflect ways to reason about the domain.
- Rules draw conclusions and points to actions, when specific information about the domain comes in. This is called inference.
- The inference is a kind of chain reaction like : If there is a power failure then (see rules 1, 2,
- mentioned above)
- Rule 3 states that there is a pump failure, and Rule 1 tells that the pressure is low, and Rule 2 gives a (useless) recommendation

It is very difficult to control such a mixture of inference back and forth in the same session and resolve such uncertainties.

5. Explain in details about Bayesian Networks and Certainty Factors

A Bayesian network (or a belief network) is a probabilistic graphical model that represents a set of variables and their probabilistic independencies.

For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Bayesian Networks are also called : Bayes nets, Bayesian Belief Networks (BBNs) or simply Belief Networks. Causal Probabilistic Networks (CPNs).

A Bayesian network consists of : a set of nodes and a set of directed edges between nodes. the edges reflect cause-effect relations within the domain. The effects are not completely deterministic (e.g. disease -> symptom) the strength of an effect is modeled as a probability.

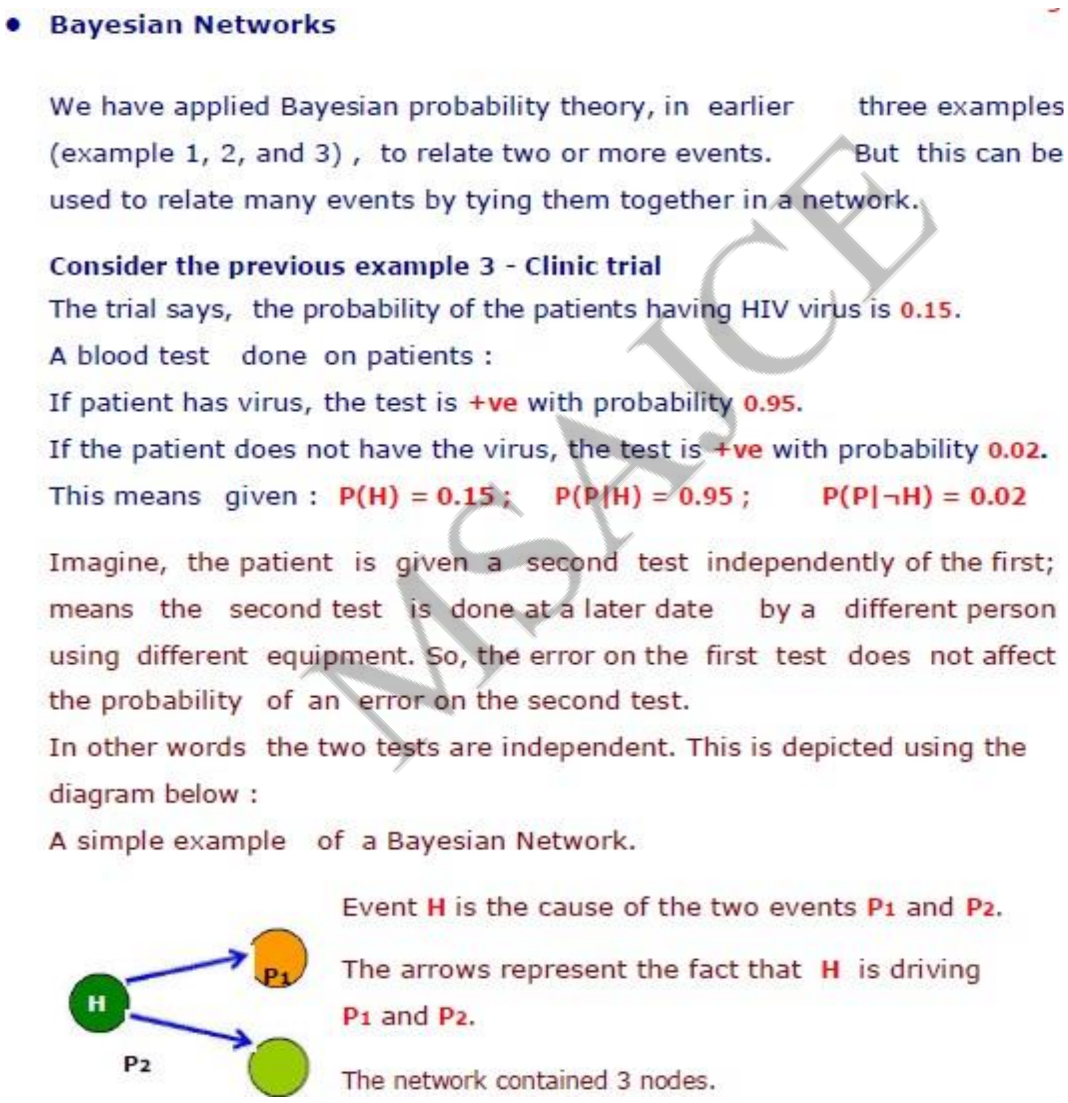
More Complicated Bayesian Networks: The previous network was simple contained three nodes. Let us look at a slightly more complicated one in the context of heart disease.

Given the following facts about heart disease.

- Either smoking or bad diet or both can make heart disease more likely.
- Heart disease can produce either or both of the following two symptoms:
  - high blood pressure
  - an abnormal electrocardiogram

■ Here smoking and bad diet are regarded as causes of heart disease.

The heart disease in turn is a cause of high blood pressure and an abnormal electrocardiogram.



- **Bayesian Networks**

We have applied Bayesian probability theory, in earlier three examples (example 1, 2, and 3) , to relate two or more events. But this can be used to relate many events by tying them together in a network.

Consider the previous example 3 - Clinic trial

The trial says, the probability of the patients having HIV virus is 0.15.

A blood test done on patients :

If patient has virus, the test is +ve with probability 0.95.

If the patient does not have the virus, the test is +ve with probability 0.02.

This means given : $P(H) = 0.15$; $P(P|H) = 0.95$; $P(P|\neg H) = 0.02$

Imagine, the patient is given a second test independently of the first; means the second test is done at a later date by a different person using different equipment. So, the error on the first test does not affect the probability of an error on the second test.

In other words the two tests are independent. This is depicted using the diagram below :

A simple example of a Bayesian Network.

Event **H** is the cause of the two events P_1 and P_2.

The arrows represent the fact that **H** is driving P_1 and P_2.

The network contained 3 nodes.

6. Describe in details about Fuzzy Logic

We have discussed only binary valued logic and classical set theory like :

A person belongs to a set of all human beings, and if given a specific subset, say all males, then one can say whether or not the particular person belongs to this set.

This is ok since it is the way human reason. e.g.,

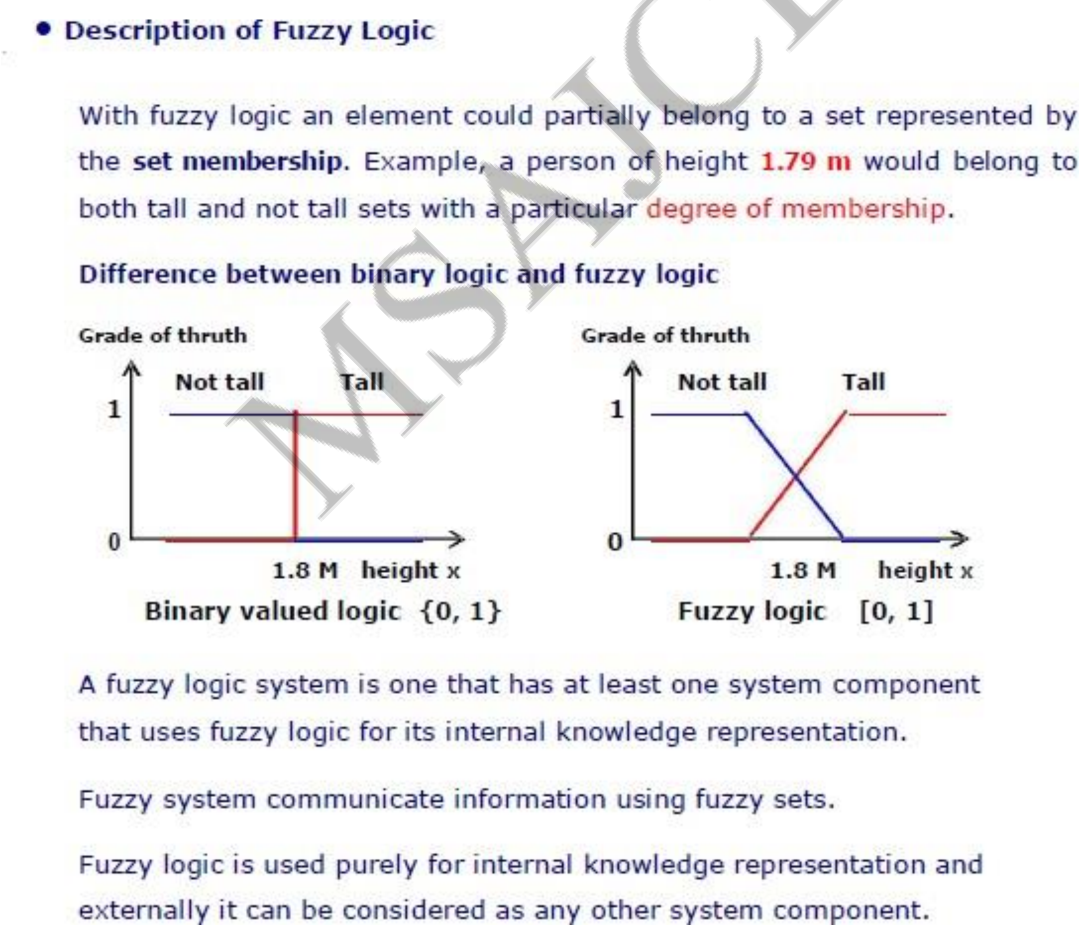
IF person is male AND a parent THEN person is a father. The rules are formed using operators.

Here, it is intersection operator "AND" which manipulates the sets.

However, not everything can be described using binary valued sets.

The grouping of persons into "male" or "female" is easy, but as "tall" or "not tall" is problematic. A set of "tall" people is difficult to define, because there is no distinct cut-off point at which tall begins.

Fuzzy logic was suggested by Zadeh as a method for mimicking the ability of human reasoning using a small number of rules and still producing a smooth output via a process of interpolation.



- **Description of Fuzzy Logic**

With fuzzy logic an element could partially belong to a set represented by the **set membership**. Example, a person of height **1.79 m** would belong to both tall and not tall sets with a particular degree of membership.

Difference between binary logic and fuzzy logic

Binary valued logic {0, 1} **Fuzzy logic [0, 1]**

A fuzzy logic system is one that has at least one system component that uses fuzzy logic for its internal knowledge representation.

Fuzzy system communicate information using fuzzy sets.

Fuzzy logic is used purely for internal knowledge representation and externally it can be considered as any other system component.

UNIT V INTELLIGENCE AND APPLICATIONS

1. Define planning.

Planning can be viewed as a type of problem solving in which the agent uses beliefs about actions and their consequences to search for a solution.

2. What are the features of an ideal planner?

i. The planner should be able to represent the states, goals and actions.

ii. The planner should be able to add new actions at any time. iii. The planner should be able to use Divide and Conquer method for solving very big problems.

3. What are the components that are needed for representing an action?

The components that are needed for representing an action are: Action description. Precondition. Effect.

4. What are the components that are needed for representing a plan?

The components that are needed for representing a plan are: A set of plans steps. A set of ordering constraints.

5. What are the different types of planning?

The different types of planning are as follows:

i. Situation space planning. ii. Progressive planning. iii. Regressive planning.
iv. Partial order planning. v. Fully instantiated planning.

6. What are the ways in which incomplete and incorrect information’s can be handled in planning?

They can be handled with the help of two planning agents namely,
i. Conditional planning agent.

ii. Replanning agent. **7. Define a solution.**

A solution is defined as a plan that an agent can execute and that guarantees the achievement of goal.

8. Define a complete plan.

A complete plan is one in which every precondition of every step is achieved by some other step.

9. Define a consistent plan.

A consistent plan is one in which there are no contradictions in the ordering or binding constraints.

10. Define conditional planning.

Conditional planning is a way in which the incompleteness of information is incorporated in terms of adding a conditional step, which involves if – then rules.

11. Give the classification of learning process.

The learning process can be classified as:

Process which is based on coupling new information to previously acquired knowledge
a. Learning by analyzing differences.

b. Learning by managing models.

c. Learning by correcting mistakes.

d. Learning by explaining experience.

Process which is based on digging useful regularity out of data, usually called as Data base mining:

a. Learning by recording cases.

b. Learning by building identification trees.

c. Learning by training neural networks.

Descriptions from positive and negative examples.

12. What are the different types of induction heuristics? There are two different types of induction heuristics. They are: i. Require-link heuristics. ii. Forbid-link heuristics.

13. What are the principles that are followed by any learning procedure?

i. The wait and see principle.
ii. The no altering principle.

iii. Martin’s law.

14. State the wait and see principle.

The law states that, “When there is doubt about what to do, do nothing”

15. State the no altering principle.

The law states that, “ When an object or situation known to be an example, fails to match a general model, create a special case exception model”.

16. State Martin’s law. The law states that, “ You cannot learn anything unless you almost know it already”.

17. Define Similarity nets.

Similarity net is an approach for arranging models. Similarity net is a representation in which nodes denotes models, links connect similar models and links are tied to different descriptions.

18. Define Reification.

The process of treating something abstract and difficult to talk about as though it were concrete and easy to talk about is called as reification.

19. What is reified link?

The elevation of a link to the status of a describable node is a kind of reification. When a link is so elevated then it is said to be a reified link.

20. Define Communication.

Communication is the international exchange of information brought about by the production and perception of signs drawn from a shared system of conventional signs.

21. Define Language.

Language enables us to communicate most of what we have observed about the environment.

22. Define Formal Language.

A formal language is defined as a set of strings of terminal symbols. It is called as words.

23. What are the processes in communication?

i. Intention ii. Generation iii. Synthesis iv. Perception v. Analysis vi Disambiguation vii. Incorporation **24. What are the parts in analyze?**

i. Syntactic Interpretation ii. Semantic Interpretation iii. Pragmatic Interpretation

25. Define Semantic Interpretation.

The process of extracting the meaning of an utterance an expression in some representation language.

26. What are the processes of Relative Clause.

i. Generating Good English sentences

ii. Over generation iii. Under generation **27. Define Parsing.**

Parsing is the process of finding a parse tree for a given input string. It is also known as syntactic analysis.

28. What are the types of parsing?

i. Top down parsing ii. Bottom up parsing

29. Define Top down parsing.

It starts with root node S and search for a tree that has the words as it leaves.

30. Define Bottom up parsing.

We start from the leaf nodes (i.e.) with the words and search for a tree with root S.

31. What are the algorithms to have efficient parsing?

i. Left to right parsing algorithm
ii. Chart Parsing algorithm.

iii. Left corner parsing

32. Define Augmentation.

The process of adding the existing rules of a grammar instead of introducing new rules. It is called Augmentation.

33. Define DCG.

The method of rewriting the existing rules in the grammar by the method of augmentation is called as DCG (Define Clause Grammar).

34. Define Sub categorization.

E 2 eliminates VP by mentioning which phrases can allow which verbs which are known as sub categorization.

35. Define Ambiguity.

The sentence that does not provide exact meaning are called ambiguous sentence.

36. What are the types of Ambiguity?

i. Lexical Ambiguity ii. Syntactic Ambiguity iii. Semantic Ambiguity **37. Define Disambiguation.** The speaker's aim is to communicate some words in utterance and hearer work is to get back the meaning of the world from the knowledge of situation.

38. Define Discourse understanding.

A discourage is any string of language usually one that is more than one sentence long.

39. What are the two sub problems in discourse understanding? The structure of coherent discourse.

40. What are the tasks in probabilistic language model?

i. Information retrieval ii. Information Extraction iii. Machine Translation

41. What are the types of smoothing?

i. Add one smoothing. ii.
Linear Interpolation
Smoothing

42. Define Segmentation?

The process of finding the words boundaries in a text with no spaces.

43. Define Information Retrieval (IR).

IR is the task of finding documents that are relevant to user’s need for information.

44. What are the characteristics of IR?

i. A document collection.

ii. A query posed in a query language.

iii. A result set iv. A presentation of the result set.

45. What are the types of evaluation of IR systems?

i. Recall
ii.
Precisio
n

46. What are the methods to do IR Refinements?

i. Case folding ii. Stemming iii. Recognize synonyms iv. Spelling correction
v. Meta data

47. What are the 3 mechanism to achieve performance improvement?

i. Relevance feedback
ii. Document
classification iii.
Document clustering

48. What are the types of clustering technique?

i. Agglomerative clustering ii. K-means clustering

49. What are the two data structures that make IR systems efficient? i. Lexicon ii. Inverted Index

50. Define Information Extraction.

It is a process of creating database entries by skimming a text and looking for occurrences of a particular class of object.

51. What are the types of information extraction systems?

i. Attribute based system ii. Relational based system

52. What are the stages in Relational based systems?

i. Tokenization ii. Complex word handling iii. Basic groups iv. Complex phrases v. Merges structures

i. Reference Resolution

53. What are the types of machine translation?

i. Rough translation ii. Restricted source translation iii. Pre edited translation iv. Literacy translation

54. How to use the parameters for machine translation?

i. Segment into sentences ii. Estimate the French language model

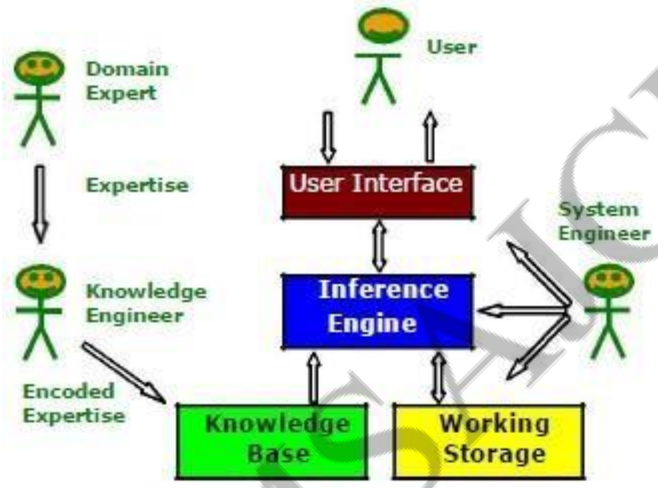
iii. Align sentences iv. Estimate the initial fertility model ,v. Estimate the initial word choice model vi. Estimate the initial offset model ,vii. Improve all the estimates.

PART - B

1. What is an expert system shell

Expert systems which embody some non-algorithmic expertise for solving certain types of problems.

Expert systems have a number of major system components and interface with individuals who interact with the system in various roles. These are illustrated below.



Components of Expert System

Components and Interfaces

‡ Knowledge base : A declarative representation of the expertise; often in IF THEN rules
; ‡ Working storage : The data which is specific to a problem being solved;

‡ Inference engine : The code at the core of the system which derives recommendations from the knowledge base and problem-specific data in working storage;

‡ User interface : The code that controls the dialog between the user and the system.

■ Roles of Individuals who interact with the system

‡ Domain expert : The individuals who currently are experts in solving the problems; here the system is intended to solve; ‡

Knowledge engineer : The individual who encodes the expert's
knowledge in a declarative form that can be used by the expert
system;

‡ User : The individual who will be consulting with the system to get advice which would have been provided by the expert.

2. Explain in detail about Expert system shells?

Expert System Shells Many expert systems are built with products called expert system shells. A shell is a piece of software which contains the user interface, a format for declarative knowledge in the knowledge base, and an inference engine. The knowledge and system engineers uses these shells in making expert systems.

‡ Knowledge engineer : uses the shell to build a system for a particular problem domain.
‡ System engineer : builds the user interface, designs the declarative format of the knowledge base, and implements the inference engine.

Depending on the size of the system, the knowledge engineer and the system engineer might be the same person.

3. Explain in details about Expert System Features

The features which commonly exist in expert systems are :

■ Goal Driven Reasoning or Backward Chaining

An inference technique which uses IF-THEN rules to repetitively break a goal into smaller sub-goals which are easier to prove;

■ Coping with Uncertainty

The ability of the system to reason with rules and data which are not precisely known;

■ Data Driven Reasoning or Forward Chaining

An inference technique which uses IF-THEN rules to deduce a problem solution from initial data;

■ Data Representation

The way in which the problem specific data in the system is stored and accessed;

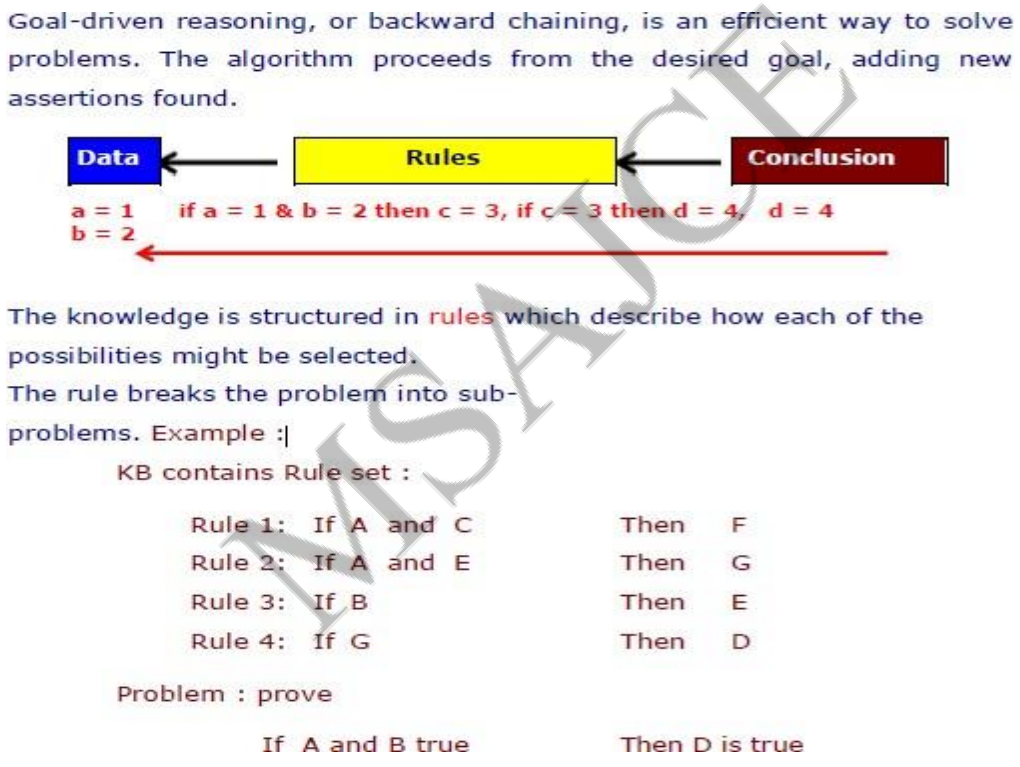
■ User Interface

That portion of the code which creates an easy to use system;

■ Explanations

The ability of the system to explain the reasoning process that it used to reach a recommendation.

4. Explain in details about Goal Driven Reasoning



Goal-driven reasoning, or backward chaining, is an efficient way to solve problems. The algorithm proceeds from the desired goal, adding new assertions found.

The knowledge is structured in rules which describe how each of the possibilities might be selected.

The rule breaks the problem into sub-problems. Example :

KB contains Rule set :

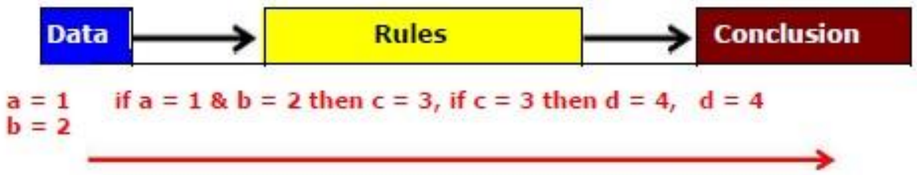
Rule 1:	If A and C	Then	F
Rule 2:	If A and E	Then	G
Rule 3:	If B	Then	E
Rule 4:	If G	Then	D

Problem : prove

If A and B true Then D is true

5. Explain in details about Data Driven Reasoning

The data driven approach, or Forward chaining, uses rules similar to those used for backward chaining. However, the inference process is different. The system keeps track of the current state of problem solution and looks for rules which will move that state closer to a final solution. The Algorithm proceeds from a given situation to a desired goal, adding new assertions found.



The knowledge is structured in rules which describe how each of the possibilities might be selected. The rule breaks the problem into sub-problems.

Example :

KB contains Rule set :

Rule 1:	If A and C	Then F
Rule 2:	If A and E	Then G
Rule 3:	If B	Then E
Rule 4:	If G	Then D

Problem : prove

If A and B true Then D is true