CS6659-ARTIFICIAL INTELLIGENCE

UNIT 1
INTRODUCTION TO Al AND PRODUCTION SYSTEMS

SYLLABUS: Introduction to AI- Problem formulation, Problem Definition - Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics - Specialized production system- Problem solving methods - Problem graphs, Matching, Indexing and Heuristic functions -Hill Climbing-Depth first and Breath first, Constraints satisfaction - Related algorithms, Measure of performance and analysis of search algorithms.

COURSE OBJECTIVE: Learn the methods of solving problems using Artificial Intelligence.

PART – A

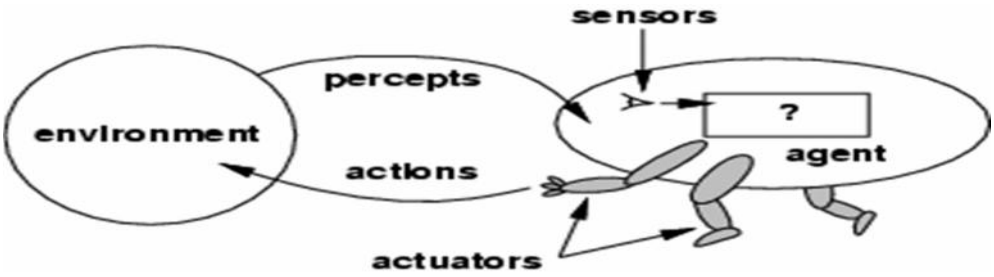
1. What is AI?

Artificial Intelligence is the branch of computer science concerned with making computers behave like humans.

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

2. Define an agent.

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.



3. What is an agent function? Differentiate an agent function and an agent program.

An agent's behavior is described by the agent function that maps any given percept sequence to an action.

AGENT FUNCTION	AGENT PROGRAM
An abstract mathematical description	A concrete implementation, running on the agent Architecture.

4. What can AI do today?

- Autonomous Planning and Scheduling

- Game Planning
- Autonomous Control
- Diagnosis
- Logistics Planning
- Robotics

5. What is a task environment? How it is specified?

Task environments are essentially the "problems" to which rational agents are the "solutions" . A Task environment is specified using PEAS (Performance, Environment, Actuators, and Sensors) description.

6. List the properties of task environments.

- Fully observable vs. partially observable**.**
- Deterministic vs. stochastic.
- Episodic vs sequential
- Static vs dynamic.
- Discrete vs. continuous.
- Single agent vs. multiagent**.**
-

7. What are the four different kinds of agent programs?

- Simple reflex agents;
- Model-based reflex agents;
- Goal-based agents; and
- Utility-based agents.

8. Explain a simple reflex agent with a diagram.

The simplest kind of agent is the **simple reflex agent.** These agents select actions on the basis AGENT of the *current* percept, ignoring the rest of the percept history.

9. **Explain goal based reflex agent.**

Knowing about the current state of the environment is not always enough to decide what to do. For example, at a road junction, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to. In other words, as well as a current state description, the agent needs some sort of **goal** information that describes situations that are desirable-for example, being at the passenger's destination.

10. **What are utility based agents?**

Goals alone are not really enough to generate high-quality behavior in most environments.
For example, there are many action sequences that will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others. A **utility function** maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.

11. **What are learning agents?**

A learning agent can be divided into four conceptual components. The most important distinction is between the learning element, which is re-ELEMENT possible for making improvements, and the

performance element, which is responsible for selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions. The learning element uses CRITIC feedback from the critic on how the agent is doing and determines how the performance element should be modified to do better in the future.

12. Define the problem solving agent.

A Problem solving agent is a **goal-based** agent. It decides what to do by finding sequence of actions that lead to desirable states. The agent can adopt a goal and aim at satisfying it. Goal formulation is the first step in problem solving.

13. List the steps involved in simple problem solving agent.

- Goal formulation
- Problem formulation
- Search
- Search Algorithm
- Execution phase

14. Define search and search algorithm.

The process of looking for sequences actions from the current state to reach the goal state is called **search.** The **search algorithm** takes a **problem** as **input** and returns a **solution** in the form of **action sequence.** Once a solution is found, the **execution phase** consists of carrying out the recommended action..

15. What are the components of well-defined problems?

- The **initial state** that the agent starts in . The initial state for our agent of example problem is described by *In(Arad)*
- A **Successor Function** returns the possible **actions** available to the agent. Given a state successor-FN(x) returns a set of {action, successor} ordered pairs where each action is one of the legal actions in state x,and each successor is a state that can be reached from x by applying the action.

  For example, from the state In(Arad),the successor function for the Romania problem would return

  { [Go(Sibiu),In(Sibiu)],[Go(Timisoara),In(Timisoara)],[Go(Zerind),In(Zerind)] }

- The **goal test** determines whether the given state is a goal state.
- A **path cost** function assigns numeric cost to each action. For the Romania problem the cost of path might be its length in kilometers.

16. Differentiate toy problems and real world problems.

TOY PROBLEMS	REAL WORLD PROBLEMS
A **toy problem** is intended to illustrate various problem solving methods. It can be easily used by different researchers to compare the performance of algorithms.	A **real world problem** is one whose solutions people actually care about.

17. Give examples of real world problems.

(ii) Touring problems
(iii) Travelling Salesperson Problem(TSP)
(iv) VLSI layout
(v) Robot navigation
(vi) Automatic assembly sequencing
(vii) Internet searching

18. List the criteria to measure the performance of different search strategies.

- **Completeness**: Is the algorithm guaranteed to find a solution when there is one?
- **Optimality**: Does the strategy find the optimal solution?
- **Time complexity**: How long does it take to find a solution?
- **Space complexity**: How much memory is needed to perform the search?

19. Compare Uninformed Search (Blind search) and informed Search (Heuristic Search) strategies.

Uninformed or Blind Search	Informed or Heuristic Search
• No additional information beyond that provided in the problem definition • Not effective • No information about number of steps or path cost	• More effective • Uses problem-specific knowledge beyond the definition of the problem itself.

20. Define 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 the evaluation function f(n). Traditionally, the node with the *lowest* evaluation function is selected for expansion.

21. Differentiate Uninformed Search(Blind search) and Informed Search(Heuristic Search) strategies.

Uninformed or Blind Search	Informed or Heuristic Search
• No additional information beyond that provided in the problem definition • Not effective • No information about number of steps or path cost	• Uses problem-specific knowledge beyond the definition of the problem itself. • More effective

22. What is a heuristic function?

A **heuristic function** or simply a heuristic is a function that ranks alternatives in various search algorithms at each branching step basing on available information in order to make a decision which branch is to be followed during a search.

For example, for shortest path problems, a *heuristic* is a function, *h* (*n*) defined on the nodes of a search tree, which serves as an estimate of the cost of the cheapest path from that node to the goal node. Heuristics are used by informed search algorithms such as Greedy best-first search and A* to choose the best node to explore.

23. What is admissible heuristic?

Admissible Heuristic is a heuristic h(n) that never overestimates the cost from node n the goal node

24. What are relaxed problems?

- A problem with fewer restrictions on the actions is called a *relaxed problem*
- The cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem
- If the rules of the 8-puzzle are relaxed so that a tile can move *anywhere*, then *hoop(n)* gives the shortest solution
- If the rules are relaxed so that a tile can move to *any adjacent square*, then *hmd(n)* gives the shortest solution

25. What is greedy best-first-search?

Greedy best-first-search tries to expand the node that is closest to the goal, on the grounds that that is likely to lead to a solution quickly. For example, it evaluates nodes by using just the heuristic function: f (n) = h (n)

26. What is A^* search?

A^* search is the most widely-known form of best-first search. It evaluates the nodes by combining g(n),the cost to reach the node, and h(n),the cost to get from the node to the goal:

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

Where f (n) = estimated cost of the cheapest solution through n.

g (n) is the path cost from the start node to node n.

h (n) = heuristic function

A^* search is both complete and optimal.

27. What is Recursive best-first search?

Recursive best-first search is a simple recursive algorithm that attempts to mimic the operation of standard best-first search, but using only linear space.

28. What are local search algorithms?

Local search algorithms operate using a single current state (rather than multiple paths) and generally move only to neighbors of that state. The local search algorithms are not systematic. The

key two advantages are (i) they use very little memory – usually a constant amount, and (ii) they can often find reasonable solutions in large or infinite (continuous) state spaces for which systematic algorithms are unsuitable.

29. What are the advantages of local search?

- Use very little memory – usually a constant amount
- Can often find reasonable solutions in large or infinite state spaces (e.g., continuous)
  - Unsuitable for systematic search
- Useful for pure optimization problems
  - Find the best state according to an objective function
  - Traveling salesman

30. What are optimization problems?

In optimization problems, the aim is to find the best state according to an **objective function** the optimization problem is then: Find values of the variables that minimize or maximize the objective function while satisfying the constraints**.**

31. What is Hill-climbing search?

The **Hill-climbing** algorithm is simply a loop that continually moves in the direction of increasing value –that is uphill. It terminates when it reaches a "peak" where no neighbor has a higher value. The algorithm does not maintain a search tree so the current node data structure need only record the state and its objective function value. Hill-climbing does not look ahead beyond the immediate neighbors of the current state.

32. What is the problem faced by hill-climbing search?

Hill-climbing often get stuck for the following reasons:

i. **Local maxima** – A local maxima is a peak that is higher than each of its neighboring states, but lower than the local maximum. Hill climbing algorithm that reach the vicinity of a local maximum will be drawn upwards towards the peak, but will then be stuck with nowhere else to go.
ii. **Ridges** – Ridges result in a sequence of local maxima that is very difficult for greedy algorithms to navigate.
iii. **Plateaux:** a plateau is an area of state space landscape where the evaluation function is flat. A hill-climbing search might be unable to find its way off the plateau.

33. What is local beam search?

The **local beam search** algorithm keeps track of k states rather than just one. It begins with k randomly generated states. At each step, all the successors of all k states are generated. If anyone is a goal, the algorithm halts. Otherwise, it selects the k best successors from the complete list and repeats.

34. What are the variants of hill-climbing?

- **Stochastic hill-climbing**
  - Random selection among the uphill moves.
  - The selection probability can vary with the steepness of the uphill move.

- **First-choice hill-climbing**
  - cfr. Stochastic hill climbing by generating successors randomly until a better one is found.
- **Random-restart hill-climbing**
  - Tries to avoid getting stuck in local maxima.

35. Explain briefly simulated annealing search.

Simulated annealing is an algorithm that combines hill climbing with random walk in some way that yields both efficiency and completeness.

36. What is genetic algorithm?

A genetic algorithm (or GA) is a variant of stochastic beam search in which successor states are generated by combining two parent states, rather than by specifying a single state

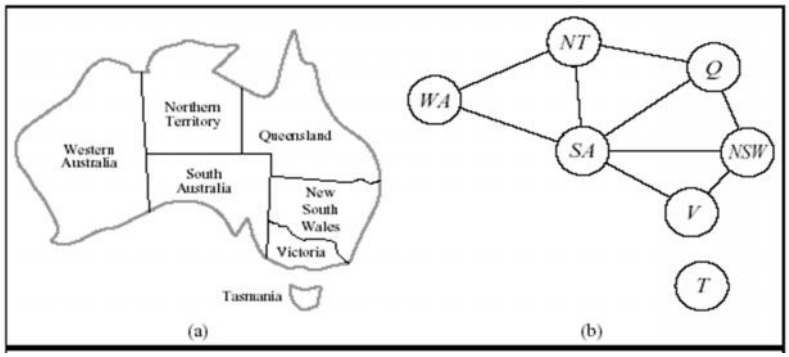
37. Define constraint satisfaction problem. (NOV/DEC 2015-REG 2008)

A **Constraint Satisfaction problem** (or CSP) is defined by a set of **variables** X_1, X2,…..,X_n, and a set of **constraints,** C_1,C_2,…..,C_m. Each variable X_i has a nonempty **domain** D_i of possible **values.** Each constraint C_i involves some subset of the variables and specifies the allowable combinations of values for that subset. A **state** of the problem is defined by an assignment of values to some or all of the variables,$\{X_i = v_i, X_j = v_j, \ldots\}$ A **solution** to a CSP is a complete assignment that satisfies all the constraints.

38. What is a constraint graph?

It is helpful to visualize the Constraint Satisfaction Problem as a Constraint Graph. A Constraint Graph is a graph where the **nodes** of the graph correspond to variables of the problem and the **arcs** corresponds to **constraints.**

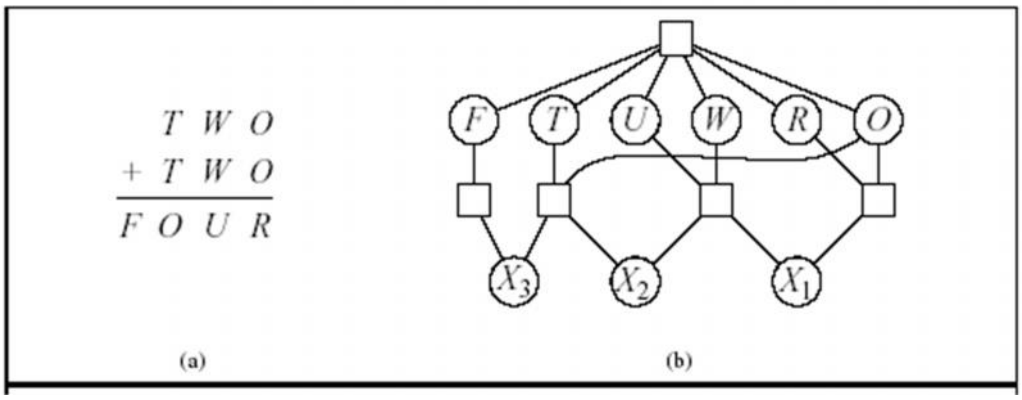
Constraint graph – Example:



The map coloring problem represented as a constraint graph

39. What are crypt arithmetic problem? Give an example.

Verbal arithmetic, also known as alphabetic, crypt arithmetic, cryptarithm or word addition, is a type of mathematical game consisting of a mathematical equation among unknown numbers, whose digits are represented by letters. The goal is to identify the value of each letter.



(a) A **crypt arithmetic problem**. Each letter stands for a distinct digit; the aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeroes are allowed.

(b) The constraint hyper graph for the crypt arithmetic problem, showing the All diff constraint as well as the column addition constraints. Each constraint is a square box connected to the variables it constrains.

40. What is backtracking search?

Backtracking search is a depth-first search that chooses values for one variable at a time and backtracks when a variable has no legal values left to assign.

41. What is adversarial search?

Competitive environments, in which the agents' goals are in conflict, give rise to adversarial search problems – often known as games.

42. Define a game.

Formal Definition of Game

We will consider games with two players, whom we will call **MAX** and **MIN**. MAX moves first, and then they take turns moving until the game is over. At the end of the game, points are awarded to the winning player and penalties are given to the loser. A **game** can be formally defined as a **search problem** with the following components:

- The **initial state**, which includes the board position and identifies the player to move.
- A **successor function**, which returns a list of (*move*, *state*) pairs, each indicating a legal move and the resulting state.
- A **terminal test**, which describes when the game is over. States where the game has ended are called **terminal states**.
- A **utility function** (also called an objective function or payoff function), which give a numeric value for the terminal states. In chess, the outcome is a win, loss, or draw,with values +1,-1,or 0. he payoffs in backgammon range from +192 to -192.

43. Explain briefly the minimax algorithm.

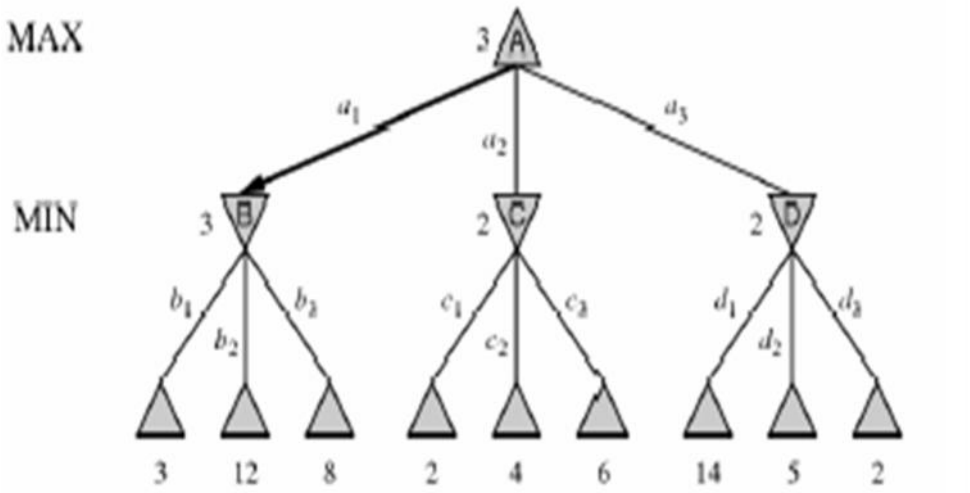


Fig Shows that A two-ply game tree. The △ nodes are "MAX nodes",in which it is MAX's turn to move,and the ▽ nodes are "MIN nodes". The terminal nodes show the utility values for MAX; the other nodes are labeled with their minimax values. MAX's best move at the root is a_1,because it leads to the successor with the highest minimax value,and MIN's best reply is b_1,because it leads to the successor with the lowest minimax value.

The minimax Algorithm

The minimax algorithm computes the minimax decision from the current state. It uses a simple recursive computation of the minimax values of each successor state,directly implementing the defining equations.

The recursion proceeds all the way down to the leaves of the tree ,and then the minimax values are **backed up** through the tree as the recursion unwinds**.**

For example in Figure 2.19,the algorithm first recourses down to the three bottom left nodes,and uses the utitliy function on them to discover that their values are 3,12,and 8 respectively.

Then it takes the minimum of these values,3,and returns it as the backed-up value of node B. A similar process gives the backed up values of 2 for C and 2 for D. Finally,we take the maximum of 3,2,and 2 to get the backed-up value of 3 at the root node.

The minimax algorithm performs a complete depth-first exploration of the game tree. If the maximum depth of the tree is *m*,and there are b legal moves at each point,then the time complexity of the minimax algorithm is $O(b^m)$. The space complexity is O(bm) for an algorithm that generates successors at once.

44. What is Alpha-Beta pruning?

The problem with minimax search is that the number of game states it has to examine is exponential in the number of moves. Unfortunately, we can't eliminate the exponent, but we can effectively cut it in half. By performing pruning,we can eliminate large part of the tree from

consideration. We can apply the technique known as alpha beta pruning, when applied to a minimax tree, it returns the same move as minimax would,but prunes away branches that cannot possibly influence the final decision.

Alpha Beta pruning gets its name from the following two parameters that describe bounds on the backed-up values that appear anywhere along the path:

- α: the value of the best(i.e. Highest-value) choice we have found so far at any choice point along the path of MAX.
- β: the value of best (i.e., lowest-value) choice we have found so far at any choice point along the path of MIN.

Alpha Beta search updates the values of α and β as it goes along and prunes the remaining branches at anode(i.e. Terminates the recursive call) as soon as the value of the current node is known to be worse than the current α and β value for MAX and MIN, respectively

49. List the various AI Application Areas

- natural language processing - understanding,
- generating, translating;
- planning;
- vision - scene recognition, object recognition, face
- recognition;
- robotics;
- theorem proving;
- speech recognition;
- game playing;
- problem solving;
- expert systems etc

50. What is heuristic search strategy? (Nov/Dec 2014)

Heuristic search is an Ai search technique employs heuristics for its move. Heuristic is a rule of thumb probably leads to solution. It play a major role in search strategies because of exponential nature of the most problems. It is also helps to reduce the number of alternatives from an exponential number to a polynomial number.

51. What is constraint satisfaction problem? (Nov/Dec 2014)

Constraint satisfaction problems (CSPs) are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which are solved by constraint satisfaction methods.

52. What are the functionalities of an agent function? (Nov/Dec 2012)

An agent is anything that can be viewed as perceiving (aware of) its environment through sensors and SENSOR acting upon that environment through actuators

- A human agent has eyes, ears, and other organs for sensors and hands, legs, mouth, and other body parts for actuators.

- A robotic agent might have cameras and infrared range finders for sensors and various motors for actuators.
- A software agent receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

53. How can we avoid ridge and plateau in hill climbing? (Nov/Dec2012)

A ridge is a special kind of local maximum. It is an area of the search space that is higher that the surrounding areas and that it have a slope. But the orientation of the high region, compared to the set of available moves and the directions in which they move, makes it impossible to traverse a ridge by single moves. Any point on a ridge can look like peak because movement in all probe directions is downward. A plateau is a flat area of the search space in which a whole set of neighboring states have the same value. On a plateau, it i not possible to determine the best direction in which to move by making local comparisons.

54. What are software agents? (Nov/Dec 2013)

A **software agent** is an piece of **software** that functions as an **agent** for a user or another program, working autonomously and continuously in a particular environment. It is inhibited by other processes and **agents**, but is also able to learn from its experience in functioning in an environment over a long period of time.

55. Define the effect of heuristics accuracy on performance (Nov/Dec 2013)

- A heuristic *h(n)* is admissible if for every node *n*,
- $h(n) \leq h^*(n)$, where $h^*(n)$ is the true cost to reach the goal state from *n*.
- An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic

Example: *hSLD(n)* (never overestimates the actual road distance)

56. Define Ideal Rational agents? (April/May 2015)

An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of −

- Its percept sequence
- Its built-in knowledge base

Rationality of an agent depends on the following four factors −

- The **performance measures**, which determine the degree of success.
- Agent's **Percept Sequence** till now.
- The agent's **prior knowledge about the environment**.
- The **actions** that the agent can carry out.

57. Why problem formulation must follow goal formulation (April/May 2015)

In goal formulation, we decide which aspects of the world we are interested in, and which can be ignored or abstracted away. Then in problem formulation we decide how to manipulate the important aspects (and ignore the others). If we did problem formulation first we would not know what to include and what to leave out. That said, it can happen that there is a cycle of iterations

between goal formulation, problem formulation, and problem solving until one arrives at a sufficiently useful and efficient solution.

58. Application of AI

- Hospitals and Medicine
- Heavy industry
- Game playing
- Speech Recognition
- Understanding the natural language
- Computer vision
- Experts systems
- Heuristics classification

59. What is a rational agent? (NOV/DEC 2011)

A rational agent is one that does the right thing. Here right thing is one that will cause agent to be more successful. That leaves us with the problem of deciding how and when to evaluate the agent's success.

60. State the significance of using heuristic functions? (NOV/DEC 2011)

The path cost from the current state to goal state is calculated, to select the minimum path cost as the next state.

Find the shortest solution using heuristic function that never over estimates the number of steps to the goal.

60. What is ridge? (MAY/JUNE 2016)

The orientation of the high region, compared to the set of available moves, makes it impossible to climb up. However, two moves executed serially may increase the height

61. How much knowledge would be required by a perfect program for the problem of playing chess? (MAY/JUNE 2016)

Computer chess is a game of computer architecture encompassing hardware and software capable of playing chess autonomously without human guidance

PART- B

1. Explain brute force's algorithm. **(R)**
2. Explain AI application areas in detail. **(R)**
3. Explain depth limited search algorithm in detail. **(R)**
4. Explain Water jug problem with an example. **(Ap)**
5. Explain various search strategies. **(U)**
6. What is an agent? Explain the basic kinds of agents program **(U)(Nov/Dec 2014)**
7. Explain the components necessary to define a problem **(U)(Nov/Dec 2014)**
8. What is depth limited search? Give the recursive implementation of depth limited search **(U) (Nov/Dec 2014)**
9. Discuss recursive best first search algorithm **(U) (Nov/Dec 2014) (Nov/Dec 2015)**
10. Explain in detail the structure of different intelligent agents. (U) **(Nov/Dec 2012)**

11. Explain AO* algorithm with an example. **(Ap) (Nov/Dec 2012) (Nov/Dec 2015)**
12. What are the five uninformed search strategies? Explain any two in detail with example. **(Ap) (Nov/Dec 2013)**
13. Explain the approach of formulation for constraint satisfaction problems with example. (Ap) **(Nov/Dec 2013)**
14. Explain any two informed search strategies. **(R)** (**April/May 2015)**
15. Discuss about constraint satisfaction problem. **(R)**(**April/May 2015)**
16. Explain the following uninformed search strategies. (**April/May 2015) (NOV/DEC 2015)**
    a. Depth first search **(U)**
    b. Iterative Deepening Depth First Search **(U)**
    c. Bidirectional Search **(U)**
17. Explain the Heuristic functions with examples. **(Ap)** (**May/June 2016)**
18. Write the algorithm for Generate and Test and Simple Hill Climbing. **(U) (May/June 2016)**
19. Solve the given problem. Describe the operators involved in it.
    Consider a water jug problem: you are given two jugs, a 4 gallon one and a 3-gallon one. Neither has ant measuring markers on it.There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into the 4-gallon jug? Explicit Assumptions: A jug can be filled from the pump, water can be poured out of a jug onto the ground, water can be poured from one jug to another and that there are no other measuring devices available **(Ap & E) (May/June 2016)**

COURSE OUTCOME: Identify problems that are amenable to solution by AI methods.

UNIT II
REPRESENTATION OF KNOWLEDGE

SYLLABUS: Game playing - Knowledge representation, Knowledge representation using Predicate logic, Introduction to predicate calculus, Resolution, Use of predicate calculus, Knowledge representation using other logic-Structured representation of knowledge.

COURSE OBJECTIVE: Learn the Knowledge representation using Predicate logic.

PART – A

1. What are Logical Agents?
Logical agents apply inference to a knowledge base to derive new information and make decisions.

2. What are the limitations of simple reflex agents?
Reflex agents are also unable to avoid infinite loops
In Wumpus world -A pure reflex agent cannot know for sure when to climb, because neither having the gold nor being in the start square is part of the percept; they are things the agent knows by forming a representation of the world.

3. What are Causal Rules?
Causal rules reflect the assumed direction of causality in the world:

(Al1,l2,s) At(Wumpus,l1,s) ^ Adjacent(l1,l2) => Smelly(l2)
(A l1,l2,s) At(Pit,l1,s) ^ Adjacent(l1,l2) => Breezy(l2)
Systems that reason with causal rules are called model-based reasoning systems

4. What is Situation Calculus?
A situation is a snapshot of the world at an interval of time during which nothing changes
• Every true or false statement is made with respect to a particular situation.
– Add situation variables to every predicate.
– at(hunter,1,1) becomes at(hunter,1,1,s0): at(hunter,1,1) is true in situation (i.e., state) s0.
• Example: The action agent-walks-to-location-y could be represented by (
x)(y)(s) (at(Agent,x,s) ^ ~onbox(s)) -> at(Agent,y,result(walk(y),s))

5. What are Diagnostic rules?
Diagnostic rules infer the presence of hidden properties directly from the percept-derived information. We have already seen two diagnostic rules:
(A l,s) At(Agent,l,s) ^ Breeze(s) => Breezy(l)
(A l,s) At(Agent,l,s) ^ Stench(s) => Smelly(l)

6. Give an example rule for Goal Based Agent.
Once the gold is found, it is necessary to change strategies. So now we need a new set of action values. We could encode this as a rule:
a. (s) Holding (Gold,s) => GoalLocation([1,1]),s)

7. What are the components of Propositional Logic?
- Logical constants: true, false
- Propositional symbols: P, Q, S, ... (atomic sentences)
- Wrapping parentheses: (…)
- Sentences are combined by connectives:
- and [conjunction]

...or [disjunction]
...implies [implication / conditional]
..is equivalent [biconditional]
..not [negation]
Literal: atomic sentence or negated atomic sentences

8. What is Horn Clause?
A Horn sentence or Horn clause has the form:

P1 P2	P3 ...	Pn	Q
or alternatively			
P1	P2	P3 ...	Pn Q

where Ps and Q are non-negated atoms
• To get a proof for Horn sentences, apply Modus Ponens repeatedly until nothing can be done

9. Define First Order Logic.

• First-order logic (FOL) models the world in terms of
  – Objects, which are things with individual identities
  – Properties of objects that distinguish them from other objects
  – Relations that hold among sets of objects
  – Functions, which are a subset of relations where there is only one "value" for any given "input"

• Examples:
  – Objects: Students, lectures, companies, cars...
  – Relations: Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes.
  – Properties: blue, oval, even, large
  – Functions: father-of, best-friend, second-half, one-more-than...

10. What are the types of Quantifiers?

- Universal Quantifiers ∀
- Existential Quantifiers ∃

11. What is Universal Quantification? Universal quantification

. (x)P(x) means that P holds for all values of x in the domain associated with that variable
b. E.g., (x) dolphin(x) mammal(x)

12. What is Existential Quantification?

a. (x) P(x) means that P holds for some value of x in the domain associated with that variable
b. E.g., (x) mammal(x) lays-eggs(x)
b. Permits one to make a statement about some object without naming it
Connections between all and Exists.

13. 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.

14. Define a Sentence?

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

15. Define an inference procedure

An inference procedure reports whether or not a sentence is entitled 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. KBAlpha is derived from KB or i derives alpha from KB

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

Logical level
Implementation level
Knowledge level or epistemological level

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

18. 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 within an agent, the agent believes the corresponding sentence.

19. 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.

20. What is entailment?
The relation between sentences is called entailment. The formal definition
of entailment is this: if and only if in every model in which is true, is also true or if is true then must also be true. Informally the truth of is contained in the truth of .

21. What is truth preserving?
An inference algorithm that derives only entailed sentences is called
sound or truth preserving.

22. Define a Proof
A sequence of application of inference rules is called a proof. Finding proof is exactly finding solution to search problems. If the successor function is defined to generate all possible applications of inference rules then the search algorithms can be applied to find proofs.

23. Define a Complete inference procedure
An inference procedure is complete if it can derive all true conditions from a set of premises.

24. Define Interpretation
Interpretation specifies exactly which objects, relations and functions are referred to by the constant predicate, and function symbols.

25. Define Validity of a sentence
A sentence is valid or necessarily true if and only if it is ue under all possible interpretation in all possible worlds.

26. Define Statist ability of a sentence
A sentence is Statist ability **"**if and only if there is some interpretation in some world for which it is true".

27. Define true sentence
A sentence is true under a particular interpretation if the state of affairs it represents is the case

28. What are the basic Components of propositional logic?

i. Logical Constants (True, False)

29. 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.

In propositional logic, modus ponendo ponens (Latin for "the way that affirms by affirming"; generally abbreviated to MP or modus ponens) or implication elimination is a valid, simple argument form and rule of inference. It can be summarized as "P implies Q; P is asserted to be true, so therefore Q must be true."

30. Define atomic sentence and complex sentence (Nov/Dec2014)

- An **atomic sentence** (which has value true or false) is an n-place predicate of n terms
- A **complex sentence** is formed from atomic sentences connected by the logical connectives:
- $\neg P$, $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P \leftrightarrow Q$ where P and Q are sentences

31. What is unification? (November/December 2014), (Nov/Dec 2012)(April/May 2015)

The idea of unification is to describe values by logical equations which can be resolved automatically by some unification algorithm. First-order unification is a form of unification whose values are trees. There are several kinds of first-order unification which differ in the choice of the notion of trees. Trees may be ground terms (tuples) or feature trees (records), they may be finite or infinite. First-order unification for feature trees is called feature unification.

32. Represent the following sentence in predicate form "All the children like sweets" (Nov/Dec 2012)

$\forall$ (children) ->sweets

33. Define the first order definite clause (Nov/Dec 2013)

A Horn clause with exactly one positive literal is a **definite clause**; a definite clause with no negative literals is sometimes called a **fact**; and a Horn clause without a positive literal is sometimes called a **goal clause** (note that the empty clause consisting of no literals is a goal clause). These three kinds of Horn clauses are illustrated in the following propositional example:

	Disjunction form	**Implication form**	**Read intuitively as**
Definite clause	$\neg p \vee \neg q \vee ... \vee \neg t \vee u$	$u \leftarrow p \wedge q \wedge ... \wedge t$	assume that, if *p* and *q* and ... and *t* all hold, then also *u* holds

Fact	u	u	assume that u holds
Goal clause	$\neg p \vee \neg q \vee ... \vee \neg t$	$false \leftarrow p \wedge q \wedge ... \wedge t$	show that p and q and ... and t all hold [note 1]

34. State the expressiveness extension (Nov/Dec 2013)

With Semantic Expressiveness it is possible to use simple references in form of a new syntax to place the same information in various places without writing redundant information

35. Differentiate forward chaining and backward chaining (April/May 2015)

Backward Chaining is more appropriate when the conclusion is already known. It will seem to run more efficiently than its counterpart, but can only be used in that context.

Forward Chaining is used when a conclusion is not known before hand and it has to reason its way through to one. The way they are chosen to be used, is usually relative to the context and how an actual expert would work to solve a problem in the system's stead.

36. What is the use of online search agent in unknown environment? (April/May 2015)

An online search problem can be solved by an agent executing actions, so these functions are necessary.

37. Distinguish between predicate and propositional logic. (November/December 2011) (November/Decemeber 2015)

Propositional logic (also called sentential logic) is the logic the includes sentence letters (A, B, C) and logical connectives, but not quantifiers. The semantics of propositional logic uses truth assignments to the letters to determine whether a compound propositional sentence is true.

Predicate logic is usually used as a synonym for first-order logic, but sometimes it is used to refers to other logics that have similar syntax. Syntactically, first-order logic has the same connectives as propositional logic, but it also has variables for individual objects, quantifiers, symbols for functions, and symbols for relations.

The semantics include a domain of discourse for the variables and quantifiers to range over, along with interpretations of the relation and function symbols.

38. What factors justify whether the reasoning is to be done in forward or backward reasoning? (November/December 2011)

Selection of forward reasoning or backward reasoning depends on which direction offers less branching factor and justifies its reasoning process to the user. Most of the search techniques can be used to search either forward or backward. One exception is the means-ends analysis technique which proceeds by reducing differences between current and goal states, sometimes reasoning forward and sometimes backward.

37. **What is alpha-beta pruning? (May/June 2016)**

Alpha–beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two-player games (Tic-tac-toe, Chess, Go, etc.).

38. **For the given sentence "All Pompeians were Romans" Write a well formed formula in predicate logic (May/June 2016)**

∀ (pompeianns) ->Romans

PART- B

1. Explain alpha-beta pruning algorithm with its procedure. **(U)**
2. Explain the resolution for first order logic and inference rule (**R)**
3. Illustrate the use of first-order-logic to represent the knowledge **(Ap).**
4. Explain the unification algorithm with an example. **(U)**
5. Explain the unification algorithm with an example. **(U)**
6. Write the algorithm for deciding entailment in propositional login **(U) (Nov/Dec 2014)**
7. Explain standard quantifiers of first order login with example **(U**) **(Nov/Dec 2014)**
8. Explain the forward chaining algorithm with the help of the pseudo –code (Ap) **(Nov/Dec 2014)**
9. Give the completeness proof of resolution **(An) (Nov/Dec 2014)**
10. Consider the following facts and represent them in predicate form **(Ap & E) (Nov/Dec 2012) (Nov/Dec 2015)**
    - F1 There are 500 employees in ABC Company
    - F2 Employees earning more than Rs.5000 pay tax
    - F3 John is a manager in ABC Company
    - F4 Manager earns Rs 10,000

    Convert the facts in predicate form to clauses and then prove by resolution "John pays tax"
11. Explain with an example the use of unification algorithm to prove the concept of resolution **(U) (Nov/Dec 2012)**
12. Explain the forward chaining process and efficient forward chaining with example. State its usage **(U) (Nov/Dec 2013)**
13. State and explain the various steps in knowledge engineering process **(An)** (**Nov/Dec 2013)**
14. Explain forward chaining and backward chaining algorithm with an example **(U)** (**April/May 2015) (November/December 2015)**
15. Illustrate the use of first order logic to represent knowledge **(Ap)** (**April/May 2015)**
16. Explain the backward chaining algorithm**.(R) (May/June 2016)**
17. Convert the following well formed formula into clause form with sequence of steps

    ∀x:[Roman(x) ∧ Know(x,Marcus)]->[hate(x,Caesar) ∨(∀y: ∃z:hate(y,z)->thinkcrazy(x,y))] **(E)**
18. (i) Write the resolution procedure for prepositional logic. **(Ap)(May/June 2016)**

    (ii) Explain the iterative Deepening Algorithm. **(U) May/June 2016) (Nov/Dec 2015)**

COURSE OUTCOME: Identify appropriate AI methods to solve a given problem in representation of knowledge.

UNIT III
KNOWLEDGE INFERENCE

SYLLABUS: Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network- Dempster - Shafer theory.

COURSE OBJECTIVE: Introduce the concepts of knowledge inference.

PART - A

1. What is the use of Fuzzy set theory?
Fuzzy set theory is a means of specifying how well an object satisfies a vague description.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. Define probability distribution:
If we want to have probabilities of all the possible values of a random variable probability distribution is used.
E.g. (weather) = (0.7,0.2,0.08,0.02). This type of notations simplifies many equations.

9. 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.

10. 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=variables.

11. How Knowledge is represented?

A variety of ways of knowledge (facts) have been exploited in AI programs. Facts: truths in some relevant world. These are things we want to represent.

12. What is propositional logic?

It is a way of representing knowledge. In logic and mathematics, a propositional calculus or logic is a formal system in which formulae representing propositions can be formed by combining atomic propositions using logical connectives

Sentences considered in propositional logic are not arbitrary sentences but are the ones that are either true or false, but not both. This kind of sentences is called propositions.

Example Some facts in propositional logic:

It is raning. - RAINING

It is sunny - SUNNY

It is windy - WINDY

If it is raining, then it is not sunny - RAINING -> $\neg$ SUNNY

13. What are the elements of propositional logic?

Simple sentences which are true or false are basic propositions. Larger and more complex sentences are constructed from basic propositions by combining them with **connectives**. Thus **propositions** and **connectives** are the basic elements of propositional logic. Though there are many connectives, we are going to use the following **five basic connectives** here:

NOT, AND, OR, IF_THEN (or IMPLY), IF_AND_ONLY_IF.

They are also denoted by the symbols: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, respectively.

14. What is inference?

Inference is deriving new sentences from old.

15. What are modus ponens?

There are standard patterns of inference that can be applied to derive chains of conclusions that lead to the desired goal. These patterns of inference are called **inference rules.**

16. What is entailment?

Propositions tell about the notion of truth and it can be applied to logical reasoning.

We can have logical entailment between sentences. This is known as entailment where a sentence follows logically from another sentence. In mathematical notation we write :

$$\alpha \models \beta$$

17. What is knowledge based agents?

The central component of a knowledge-based agent is its knowledge base, or KB. Informally, a knowledge base is a set of sentences. Each sentence is expressed in a language called a knowledge representation language and represents some assertion about the world.It takes a percept as input and returns an action. The agent maintains a knowledge base, KB, which may initially contain some **background knowledge.** Each time the agent program is called, it does three things. First, it TELLS the knowledge base what it perceives. Second, it ASKS the knowledge base what action it should perform. In the process of answering this query, extensive reasoning may be done about the current state of the world, about the outcomes of possible action sequences, and so on.

18. Define 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,

Functions: father of, best friend, one more than, plus,

19. Compare different knowledge representation languages.

Language	**Ontological Commitment (What Exists in the world)**	**Epistemological Commitment (What an agent believes about facts)**
Propositional Logic	Facts	True/false/unknown
First-order Logic	Facts,objects,relations	True/false/unknown
Temporal Logic	Facts,objects,relations,times	True/false/unknown
Probability Theory	facts	Degree of belief ϵ [0,1]
Fuzzy Logic	Facts with degree of truth ϵ[0,1]	Known interval value

20. What are the syntactic elements of First Order Logic?

The basic syntactic elements of first-order logic are the symbols that stand for objects, relations, and functions. The symbols come in three kinds:
a) Constant symbols, which stand for objects;
b) Predicate symbols, which stand for relations;
c) and function symbols, which stand for functions.
We adopt the convention that these symbols will begin with uppercase letters.
Example:
Constant symbols :
Richard and John;
predicate symbols :Brother, OnHead, Person, King, and Crown;
function symbol :LeftLeg.

21. What are quantifiers?

There is need to express properties of entire collections of objects, instead of enumerating the objects by name. Quantifiers let us do this.

FOL contains two standard quantifiers called

a) Universal (?**)** and
b) Existential (A**)**

Universal quantification

(?x) P(x) : means that P holds for **all** values of x in the domain associated with that variable

E.g., (?x) dolphin(x) => mammal(x)

Existential quantification

(A x)P(x) means that P holds for **some** value of x in the domain associated with that variable

E.g., (A x) mammal(x) ^ lays-eggs(x)

Permits one to make a statement about some object without naming it

22. Explain Universal Quantifiers with an example.

Rules such as "All kings are persons," is written in first-order logic as

?x King(x) => Person(x)

where ? is pronounced as " For all .."

Thus, the sentence says, "For all *x*, if *x* is a king, then z is a person."

The symbol x is called a variable (lower case letters)

The sentence ?**x** P, where P is a logical expression says that P is true for every object x.

23. Explain Existential quantifiers with an example.

Universal quantification makes statements about every object.

It is possible to make a statement about some object in the universe without naming it,by using an existential quantifier.

Example

"King John has a crown on his head"

∃x Crown(x) ^ OnHead (x,John)

Ax is pronounced "There exists an x such that .." or " For some x .."

24. Explain the connection between ? and A

"Everyone likes ice-cream "is equivalent

"There is no one who does not like ice cream"

This can be expressed as:

∀x Likes(x,IceCream) is equivalent to

∃ ¬Likes(x,IceCream)

25. What are the steps associated with the knowledge Engineering process?
Discuss them by applying the steps to any real world application of your choice.

Knowledge Engineering: The general process of knowledge base construction a process is called knowledge engineering.

A knowledge engineer is someone who investigates a particular domain, learns what concepts are important in that domain, and creates a formal representation of the objects and relations in the domain. We will illustrate the knowledge engineering process in an electronic circuit domain that should already be fairly familiar,

The steps associated with the knowledge engineering process are:

- Identify the task.
- Assemble the relevant knowledge.
- Decide on a vocabulary of predicates, functions, and constants. That is, translate the
- Encode general /knowledge about the domain.
- Encode a description of the specific problem instances.
- Pose queries to the inference procedure and get answers.
- Debug the knowledge base.

26. Give examples on usage of First Order Logic.
The best way to find usage of First order logic is through examples. The examples can be taken from some simple **domains.** In knowledge representation, a domain is just some part of the world about which we wish to express some knowledge.

Assertions and queries in first-order logic

Sentences are added to a knowledge base using TELL, exactly as in propositional logic. Such sentences are called **assertions.**

For example, we can assert that John is a king and that kings are persons:

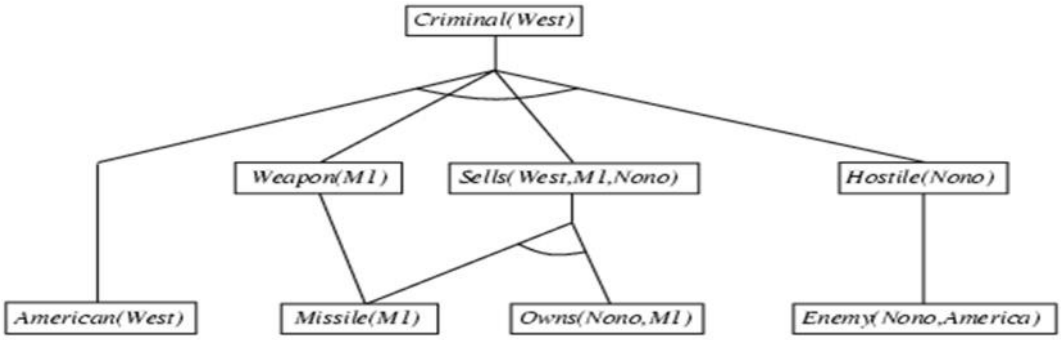
TELL (KB, King (John)) .

Where KB is knowledge base.

TELL (KB, ?x King(x) => Person*(x))*.

27. What is forward chaining? Explain with an example.
Using a deduction to reach a conclusion from a set of antecedents is called forward chaining. In other words, the system starts from a set of facts, and a set of rules, and tries to find the way of using these rules and facts to deduce a conclusion or come up with a suitable course of action. This is known as data driven reasoning.

EXAMPLE:



The proof tree generated by forward chaining.

Example knowledge base

- The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- Prove that Col. West is a criminal

Note:

(a) The initial facts appear in the bottom level
(b) Facts inferred on the first iteration is in the middle level

(c) The facts inferered on the 2nd iteration is at the top level

Forward chaining algorithm

```
function FOL-FC-ASK(KB, α) returns a substitution or false
  repeat until new is empty
    new ← { }
    for each sentence r in KB do
      (p1 ∧ ... ∧ pn ⇒ q) ← STANDARDIZE-APART(r)
      for each θ such that (p1 ∧ ... ∧ pn)θ = (p'1 ∧ ... ∧ p'n)θ
                  for some p'1, ..., p'n in KB
        q' ← SUBST(θ, q)
        if q' is not a renaming of a sentence already in KB or new then do
          add q' to new
          φ ← UNIFY(q', α)
          if φ is not fail then return φ
    add new to KB
  return false
```

28. What is backward chaining? Explain with an example.

Forward chaining applies a set of rules and facts to deduce whatever conclusions can be derived.

In backward chaining, we start from a conclusion, which is the hypothesis we wish to prove, and we aim to show how that conclusion can be reached from the rules and facts in the data base.

The conclusion we are aiming to prove is called a goal and the reasoning in this way is known as goal-driven.

Backward chaining example

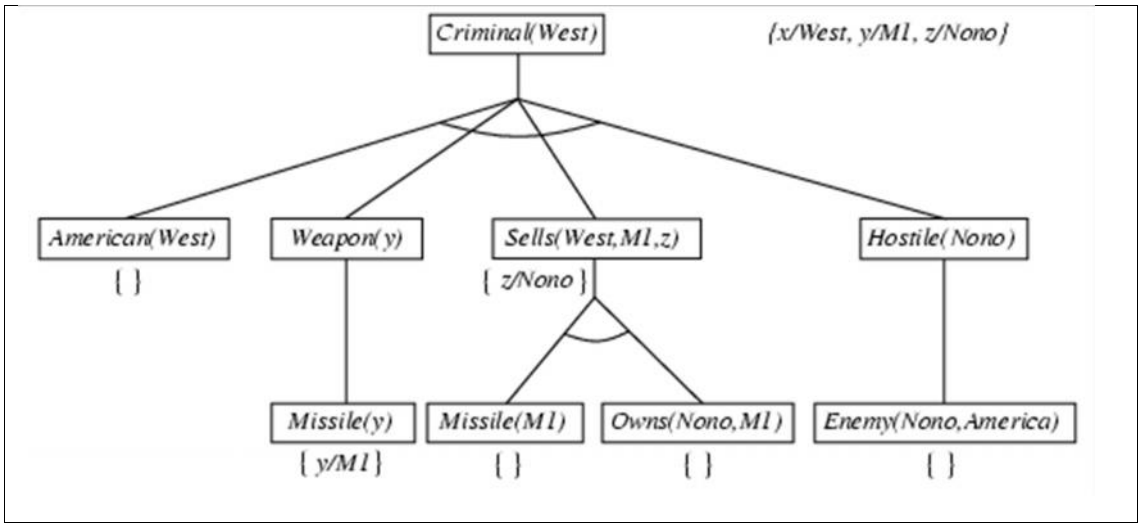


Fig : Proof tree constructed by backward chaining to prove that West is criminal.

Note:

(a) To prove Criminal (West), we have to prove four conjuncts below it.
(b) Some of which are in knowledge base, and others require further backward chaining.

29. What is Ontological Engineering?

Ontology refers to organizing everything in the world into hierarch of categories. Representing the abstract concepts such as Actions, Time, Physical Objects, and Beliefs is called Ontological Engineering.

30. How categories are useful in Knowledge representation?

Categories and Objects: The organization of objects into **categories** is a vital part of knowledge representation. Although interaction with the world takes place at the level of individual objects, much reasoning takes place at the level of categories.

31. What is taxonomy?

Subclass relations organize categories into a taxonomy, or taxonomic hierarchy. Taxonomies have been used explicitly for centuries in technical fields. For example, systematic biology aims to provide a taxonomy of all living and extinct species; library science has developed a taxonomy of all fields of knowledge, encoded as the Dewey Decimal system.

32. Explain the Ontology of Situation calculus.

Situations are logical terms consisting of the initial situation (usually called *So)* andall situations that are generated by applying an action to a situation. The function *Result(a, s)* (sometimes called *Do)* names the situation that results when action *a* is executed in situation *s.* Figure 10.2 illustrates this idea.

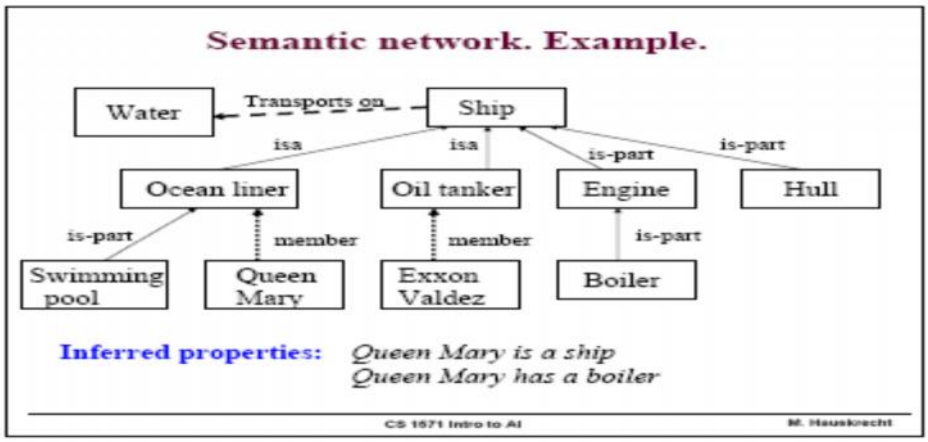
Fluents are functions and predicates that vary from one situation to the next, such as the location of the agent or the aliveness of the wumpus. The dictionary says a fluent is something that flows, like a liquid. In this use, it means flowing or changing across situations. By convention, the situation is always the last argument of a fluent. For example, *lHoldzng(G1, So)* says that the agent is not holding the gold GI in the initial situation *So. Age(Wumpus, So)* refers to the wumpus's age in *So.* **A temporal** or **eternal** predicates and functions are also allowed. Examples include the predicate Gold (GI) and the function *LeftLeg Of (Wumpus).*

33. What is event calculus?

Event calculus Situation calculus works well when there is a single agent performing instantaneous, discrete actions. When actions have duration and can overlap with each other, situation calculus becomes somewhat awkward. Therefore, we will cover those topics with an alternative for- EVENTCALCULUS malism known as **event calculus,** which is based on points in time rather than on situations.

34. What are semantic networks?

Semantic networks are capable of representing individual objects, categories of objects, and relation among objects. Objects or Category names are represented in ovals and are connected by labeled arcs. Semantic network example



37. What is the function of a re- planning agent?

A re planning agent knows what to do when something unexpected happens. Example call a planner again to come up with a new plan to reach a goal

38. Define consistent plan? (Nov/Dec2014)

Consistent Planning

- Data is stored at the most detailed level of the planning hierarchy.
- Planning levels are interdependent: changes made at one planning level immediately affect all other planning levels. The system performs aggregation and disaggregation at runtime.
- Advantages of Consistent Planning
- Ease of use: you enter planning figures at one level and can rely on data consistency at all other levels.

39. Define critical path (Nov/Dec 2014)

An extension of critical path scheduling to a case where there are alternate methods for completing a project is introduced. Three solution methods are presented including one based on the heuristic search algorithm used inartificial intelligence.

Critical path heuristics are a method to relax planning tasks, and thus automatically .... $g_n \subseteq G$, and for all i $g_i \subseteq regr(g_{i+1}, a_i)$ and $|g_i| = 1$ (resp. $|g_i| \leq m$).

40. What are the characteristics of partial order planer? (Nov/Dec 2012) & define partial order planner (April/May 2015)

A **partial-order plan** or **partial plan** is a plan which specifies all actions that need to be taken, but does not specify an exact order for the actions when the order does not matter. It is the result of a partial-order planner. A partial-order plan consists of four components:

- A set of **actions** (also known as **operators**).
- A **partial order** for the actions. It specifies the conditions about the order of some actions.
- A set of **causal links**. It specifies which actions meet which preconditions of other actions. Alternatively, a set of **bindings** between the variables in actions.
- A set of **open preconditions**. It specifies which preconditions are not fulfilled by any action in the partial-order plan.

41. Define the bi-directed search? (Nov/Dec 2013)

Bidirectional search is a graph search algorithm that finds a shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: one forward from the initial state and one backward from the goal, stopping when the two meet in the middle.

42. What are continuous random variables? (Nov/Dec 2013)

A continuous random variable is a random variable where the data can take infinitely many values. For example, a random variable measuring the time taken for something to be done is continuous since there are an infinite number of possible times that can be taken.

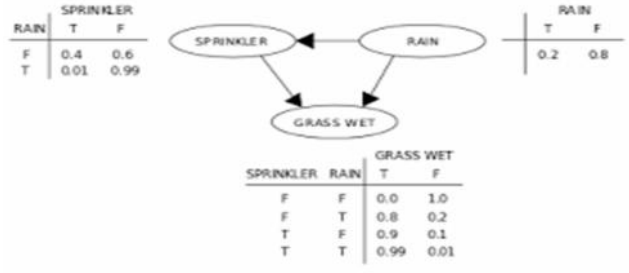
$$F(t) = P(X \leq t) = \int_a^t f(x)\,dx$$

43. Define knowledge acquisition. (April/May 2015)

The knowledge engineer might already be an expert in the domain, ***or*** might need to work with real experts to extract what they know-a process called **knowledge acquisition.**

44. What is Bayesian Networks? (May /June 2016)

A Bayesian network, Bayes network, belief network, Bayes model or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).



RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99

RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

45. Write the properties of fuzzy sets. (May/June 2016)

- Complementation
- Intersection
- Union

PART- B

1. Explain about partial order planning with an example. **(U)**
2. Explain about the different types of state space searches. **(U)**
3. Explain about partial order planning algorithm. **(U)**
4. Describe in detail about planning graphs. (**An)**
5. Explain in detail about graph plan algorithm. **(U)**
6. Explain in detail about conditional planning with an example. **(Ap)**
7. Explain about re planning agent algorithm. **(U)**
8. Explain the concepts of forward and backward state space search in detail. **(An) (Nov/Dec 2014)**
9. Describe Graph plan algorithm in detail with example **(U) (Nov/Dec 2014)**
10. Describe planning methods for handling indeterminacy **(R) (Nov/Dec 2014)**
11. Describe a planning method based on hierarchical task networks with an example **(Ap) (Nov/Dec 2014)**
12. Explain the concept behind partial order planning with suitable examples **(U**) **(Nov/Dec 2012) (Nov/Dec 2015)**
13. Explain the use of planning graphs in providing better heuristics estimates with suitable examples **(U) (Nov/Dec 2012)**
14. Explain the procedure of planning with state space search with example. **(U) (Nov/Dec 2013)**
15. Explain the process of scheduling with resource constraints in detail with suitable example. **(Ap) (Nov/Dec 2013)**

16. Explain the concept of planning with state space search using suitable example **(An)** (**April/May 2015)**
17. Explain the use of planning graph in providing better heuristic estimation with suitable example **(U) (April/May 2015)**
18. Explain the hidden markov model. **(R)** (**Nov/Dec 2013)**
19. Explain the need of fuzzy set and fuzzy logic with example. **(An) (Nov/Dec 2013)**
20. Explain the inference in temporal models. **(R)** (**April/May 2015)**
21. Write short notes on Hidden Markov model. **(U)** (**April/May 2015) (Nov/Dec 2015)**
22. Explain about the exact inference in Bayesian networks **(R)**(**April/May 2015) (Nov/Dec 2015)**
23. (a) Briefly explain how reasoning is done using fuzzy logic **(Ap) (May/June 2016)**
(b) Explain Dempster –Shafer Theory **(U) (May/June 2016)**

24. What is Forward Chaining and how does it work?Explain the forwarded Chaining algorithm with an example **(U) (May/June 2016)**

COURSE OUTCOME: Formalize a given problem in the language/framework of different AI methods.

UNIT-IV

PLANNING AND MACHINE LEARNING

SYLLABUS: Basic plan generation systems - Strips -Advanced plan generation systems – K strips - Strategic explanations -Why, Why not and how explanations. Learning - Machine learning, adaptive Learning.

COURSE OBJECTIVE: Introduce the concepts of planning and machine learning.

PART – A

1. What is meant by learning?
Learning is a goal-directed process of a system that improves the knowledge or the knowledge representation of the system by exploring experience and prior knowledge.

2. Define informational equivalence.
A transformation from on representation to another causes no loss of information; they can be constructed from each other.

3. Define computational equivalence.
The same information and the same inferences are achieved with the same amount of effort.
4. List the difference between knowledge acquisition and skill refinement.
• knowledge acquisition (example: learning physics) — learning new symbolic information coupled with the ability to apply that information in an effective manner
• skill refinement (example: riding a bicycle, playing the piano) — occurs at a subconscious level by virtue of repeated practice
5. What is meant by analogical reasoning?
Instead of using examples as foci for generalization, one can use them directly to solve new problems.

6. Define Explanation-Based Learning.
The background knowledge is sufficient to explain the hypothesis. The agent does not learn anything factually new from the instance. It extracts general rules from single examples by explaining the examples and generalizing the explanation

7. What is meant by Relevance-Based Learning?
• uses prior knowledge in the form of determinations to identify the relevant attributes
• generates a reduced hypothesis space

8. Define Knowledge-Based Inductive Learning.
Knowledge-Based Inductive Learning finds inductive hypotheses that explain set of observations with the help of background knowledge.

9. What is truth preserving?
An inference algorithm that derives only entailed sentences is called sound or truth preserving.

10. Define Inductive learning.
Learning a function from examples of its inputs and outputs is called inductive learning.

11. How the performance of inductive learning algorithms can be measured?
It is measured by their learning curve, which shows the prediction accuracy as a function of the number of observed examples.

12. List the advantages of Decision Trees
• It is one of the simplest and successful forms of learning algorithm.
• It serves as a good introduction to the area of inductive learning and is easy to implement.

13. What is the function of Decision Trees?
A decision tree takes as input an object or situation by a set of properties, and outputs a yes/no decision. Decision tree represents Boolean functions.

14. List some of the practical uses of decision tree learning.
• Designing oil platform equipment
• Learning to fly

15. Define reinforcement learning.
The task of reinforcement learning is to use rewards to learn a successful agent function.

16. Differentiate between Passive learner and Active learner.
A passive learner watches the world going by, and tries to learn the utility of being in various states. An active learner acts using the learned information, and can use its problem generator to suggest explorations of unknown portions of the environment.

17. State the design issues that affect the learning element.
• Which components of the performance element are to be improved
• What representation is used for those components

- What feedback is available
- What prior information is available

18. State the factors that play a role in the design of a learning system.
- Learning element
- Performance element
- Critic
- Problem generator

19. What is memorization?
The technique of memorization is used to speed up programs by saving the results of computation. The basic idea is to accumulate a database of input/output pairs; when the function is called, it first checks the database to see if it can avoid solving the problem from scratch.

20. Define Q-Learning.
The agent learns an action-value function giving the expected utility of taking a given action in a given state. This is called Q-Learning.

21. Differentiate between supervised learning & unsupervised learning.
Any situation in which both inputs and outputs of a component can be perceived is called supervised learning. Learning when there is no hint at all about the correct outputs is called unsupervised learning.

22. Define Ockham's razor.
Extracting a pattern means being able to describe a large number of cases in a concise way. Rather than just trying to find a decision tree that agrees with example, try to find a concise one, too.

23. Define Bayesian learning
Bayesian learning simply calculates the probability of each hypothesis, given the data,
and makes predictions on that basis. That is, the predictions are made by using all the hypotheses, weighted by their probabilities, rather than by using just a single "best" hypothesis.

24. What is meant by hidden variables?
Many real-world problems have hidden variables (sometimes called latent variables) which are not observable in the data that are available for learning.

25. Define Cross validation.
The basic idea behind Cross validation is try to eliminate how well the current hypothesis will predict unseen data.

26. What are the operations in Genetic algorithms?
It starts with a set of one or more individuals and applies selection and reproduction operators to evolve an individual that is successful, as measured by a fitness function.

27. List the various Components of the performance element
1. A direct mapping from conditions on the current state to actions.

2. A means to infer relevant properties of the world from the percept sequence.
3. Information about the way the world evolves.
4. Information about the results of possible actions the agent can take.
5. Utility information indicating the desirability of world states.
6. Action-value information indicating the desirability of particular actions in particular states.
7. Goals that describe classes of states whose achievement maximizes the agent\'s utility.

28. Differentiate between Parity function and majority function.

If the function is the parity function, which returns 1 if and only if an even number of inputs are 1, then an exponentially large decision tree will be needed.

A majority function, which returns 1 if more than half of its inputs are 1.

29. What is the function of a performance element?

The performance element is responsible for selecting external actions.

30. What is the function of a learning element?

Learning element is responsible for making improvements.

31. List the 3 approaches that can be used to learn utilities.

- Least-mean-square Approach
- Adaptive Dynamic Programming Approach
- Temporal Difference Approach

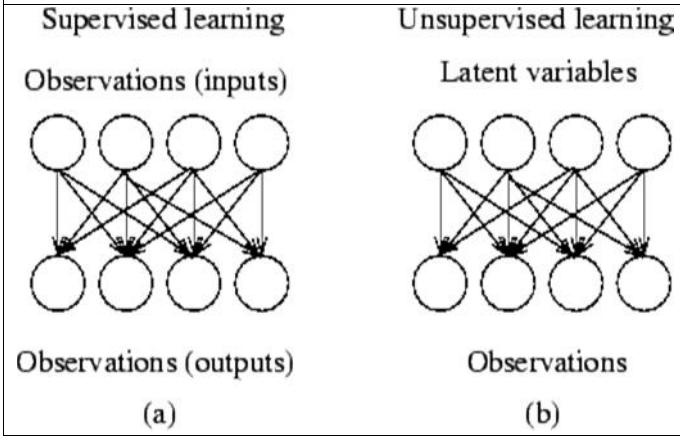
32. Distinguish between supervised learning and unsupervised learning (Nov/Dec2014)

In supervised learning, the output datasets are provided which are used to train the machine and get the desired outputs whereas in unsupervised learning no datasets are provided, instead the data is clustered into different classes.

Example: Face recognition

Supervised learning : Learn by examples as to what a face is in terms of structure, color, etc so that after several iterations it learns to define a face

Unsupervised learning : since there is no desired output in this case that is provided therefore categorization is done so that the algorithm differentiates correctly between the face of a horse, cat or human (clustering of data)



33. What is over fitting (Nov/Dec2014)?

Over fitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Over fitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A model that has been over fit will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data.

34. Distinguish between supervised learning and reinforcement learning (Nov/Dec 2012)

"Reinforcement learning (RL) and supervised learning are usually portrayed as distinct methods of learning from experience. RL methods are often applied to problems involving sequential dynamics and optimization of a scalar performance objective, with online exploration of the effects of actions. Supervised learning methods, on the other hand, are frequently used for problems involving static input-output mappings and minimization of a vector error signal, with no explicit dependence on how training examples are gathered.

35. Define computational learning theory (Nov/Dec 2012)

Computational learning theory is the analysis of computational complexity of machine learning algorithms. It is the intersection of theory of computation and machine learning.

36. 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.

37. 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.

38. 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.

39. What is called as multiple connected graphs?

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

40. List the 3 basic classes of algorithms for evaluating multiply connected graphs.

- Clustering methods
- Conditioning methods
- Stochastic simulation methods

41. Define Uncertainty.

Uncertainty means that many of the simplifications that are possible with deductive inference are no longer valid.

42. What is meant by deterministic nodes?
A deterministic node has its value specified exactly by the values of its parents, with no uncertainty.

43. What are all the various uses of a belief network?
• Making decisions based on probabilities in the network and on the agent\'s utilities.
• Deciding which additional evidence variables should be observed in order to gain useful information.
• Performing sensitivity analysis to understand which aspects of the model have the greatest impact on the probabilities of the query variables (and therefore must be accurate).

44. What is the function of cut set conditioning method?
This method transforms the network into several simpler poly trees.

45. State Bayes' rule (Nov/Dec 2014) & Define the Bayes rule? (Nov/Dec2012)

Bayes rule

True Bayesians actually consider conditional probabilities as more basic than joint probabilities . It is easy to define P(A|B) without reference to the joint probability P(A,B). To see this note that we can rearrange the conditional probability formula to get:

P(A|B) P(B) = P(A,B)

but by symmetry we can also get:

P(B|A) P(A) = P(A,B)

It follows that:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

This is the so-called Bayes Rule.

46. What is first order Markov process (Nov/Dec2014)
A Markov process is a stochastic model that has the Markov property. It can be used to model a random system that changes states according to a transition rule that only depends on the current state.

47. What do you mean hybrid Bayesian Networks? (Nov/Dec 2012)
Definition: BN = (DAG, CPD)
- DAG: directed acyclic graph (BN's structure)
  - Nodes: random variables (typically binary or discrete, but methods also exist to handle continuous variables)
  - Arcs: indicate probabilistic dependencies between nodes (*lack* of link signifies conditional independence)
- CPD: conditional probability distribution (BN's parameters)
  - Conditional probabilities at each node, usually stored as a table (conditional probability table, or CPT)
- Root nodes are a special case – no parents, so just use priors in CPD

48. What is partial order planning (Nov/Dec 2013)

Partial-order planning is an approach to automated planning that leaves decisions about the ordering of actions as open as possible. It contrasts with total-order planning, which produces an exact ordering of actions.

49. Define inference in temporal models (Nov/Dec 2013)

Inference is the act or process of deriving logical conclusions from premises known or assumed to be true.[1] The conclusion drawn is also called an idiomatic. The laws of valid inference are studied in the field of logic.

Q-learning uses temporal differences to estimate the value of $Q^*(s,a)$. In Q-learning, the agent maintains a table of $Q[S,A]$, where S is the set of states and A is the set of actions. $Q[s,a]$ represents its current estimate of $Q^*(s,a)$.

50. List down applications of Bayesian network (April/May 2015)

- image processing
- gaming
- decision support systems
- information retrieval
- semantic search

51. Define uncertainty. How it is solved APRIL/MAY 2015

the "reasoning with uncertainty" (or "reasoning under uncertainty") research in AI has been focused on the uncertainty of *truth value*, that is, to allow and process truth values other than "true" and "false".

- There is uncertainty in the facts we know:
  - What's the temperature? Imprecise measures
  - Is Bush a good president? Imprecise definitions
  - Where is the pit? Imprecise knowledge

52. Define Inductive learning. (NOVEMBER/DECEMBER 2011)

Learning a function from examples of its inputs and outputs is called inductive learning.

53. Distinguish between supervised learning and unsupervised learning. (NOVEMBER/DECEMBER 2011)

- Any situation in which both inputs and outputs of a component can be perceived is called supervised learning.
- Learning when there is no hint at all about the correct outputs is called unsupervised learning.

54. What is rote learning? (MAY/JUNE 2016)

Rote learning is a memorization technique based on repetition. The idea is that one will be able to quickly recall the meaning of the material the more one repeats it. Some of the alternatives to rote learning include meaningful learning, associative learning, and active learning.

55. Brief frame problem? (MAY/JUNE 2016)

In artificial intelligence, the frame problem describes an issue with using first-order logic (FOL) to express facts about a robot in the world. The frame problem is the problem of finding adequate collections of axioms for a viable description of a robot environment.

PART- B

1. Explain the learning decision tree with algorithm **(U)**
2. Explain the explanation based learning? **(R)**
3. Explain how learning with complete data is achieved? **(An)**
4. Discuss learning with hidden variables? **(An)**
5. Explain all the statistical learning method available in AI. **(U)**
6. Explain about Reinforcement learning. **(R)**
7. Explain decision tree learning algorithm **(U) (Nov/Dec 2014)**
8. Discuss back propagation algorithm for learning in multilayer neural networks **(U) (Nov/Dec 2014)**
9. Explain the basic concept of support vector machine **(R)(Nov/Dec 2014)**
10. Give the complete agent design for an exploratory Q learning agent **(U)(Nov/Dec 2014)**
11. Explain the concept of learning using Decision tree **(Ap)(Nov/Dec 2012)**
12. Write short notes on **(Nov/Dec 2012)**
    i) Reinforcement learning **(R)**
    ii) Explanation based learning **(R)**
13. Explain the process of learning on action utility function **(U) (Nov/Dec 2013)**
14. Explain the temporal difference learning with example **(Ap) (Nov/Dec 2013)**
15. What are various approaches for instance based learning .Explain any one with example **(Ap) (Nov/Dec 2013)**
16. The following table consists of training data from an employee database. The data have been generalized. Let status be the class label attribute. Construct Decision tree from the given data **(C)** (**April/May 2015)**

	Department	Age	Salary	Count	Status
b.	Sales	31..35	46k..50k	30	Senior
c.	Sales	26..30	26k..30k	40	Junior
d.	Sales	31..35	31k..35k	40	Junior
e.	Systems	31..35	31k..35k	42	Senior
f.	Systems	42..45	36k..40k	40	Junior
g.	Sales	31..35	31k..35k	40	Junior

17. Explain in detail about Active and Passive Reinforcement learning **(An)** (**April/May 2015)**
18. Explain variable elimination algorithm for answering queries on Bayesian networks **(U) (Nov/Dec 2014)**
19. Discuss forward-backward algorithm in details. **(U) (Nov/Dec 2014)**
20. Discuss the different design issues to be solved to use HMM for real world applications. **(U) (Nov/Dec 2014)**
21. Explain the method of performing exact inference in Bayesian networks. **(Ap) (Nov/Dec 2012)**
22. Explain the concept of inference in temporal models. **(U) (Nov/Dec 2012)**
23. How to handle uncertain knowledge with example .**(An) (Nov/Dec 2013)**

24. How to represent knowledge in uncertain domain **(An)** (**Nov/Dec 2013)**
25. (i) Describe the components of a planning system **(U) (MAY/JUNE 2016)**
(ii) What is ID3? Write the drawback of ID3? **(An) (MAY/JUNE 2016)**
26. (i) Describe the Hierarchical planning method with an example. **(U) (MAY/JUNE 2016)**
(ii) Describe the Learning with macro operators. **(Ap) (MAY/JUNE 2016)**

COURSE OUTCOME: Implement basic AI algorithms to solve a given problem in planning and machine learning by AI methods.

UNIT V
EXPERT SYSTEMS

SYLLABUS: Expert systems - Architecture of expert systems, Roles of expert systems - Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems - MYCIN, DART, XOON, Expert systems shells.

COURSE OBJECTIVE: Introduce the concepts of Expert Systems.

PART - A

1. What are Expert Systems?

The expert systems are the computer applications developed to solve complex problems in a particular domain, at the level of extra-ordinary human intelligence and expertise.

2. What are the Characteristics of Expert Systems

- High performance
- Understandable
- Reliable
- Highly responsive

3. List out the Capabilities of Expert Systems

The expert systems are capable of Instructing and assisting human in decision making

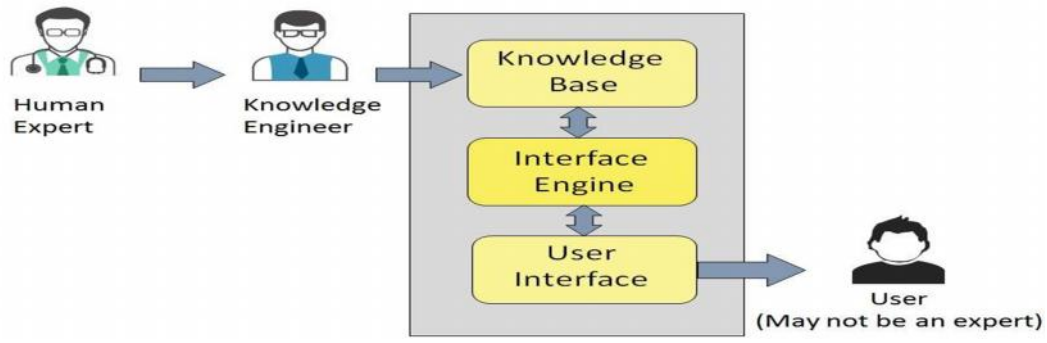
- Demonstrating
- Deriving a solution
- Diagnosing
- Explaining
- Interpreting input
- Predicting results
- Justifying the conclusion
- Suggesting alternative options to a problem

4. What are the Components of Expert Systems

The components of ES include −

- Knowledge Base
- Interface Engine

- User Interface



5. What are the Requirements of Efficient ES User Interface?

- It should help users to accomplish their goals in shortest possible way.
- It should be designed to work for user's existing or desired work practices.
- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

6. Limitations of Expert Systems?

No technology can offer easy and complete solution. Large systems are costly, require significant development time, and computer resources. ESs has their limitations which include −

- Limitations of the technology
- Difficult knowledge acquisition
- ES are difficult to maintain
- High development costs

7. Write are the Applications of Expert System?

The following table shows where ES can be applied.

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on humans.

Monitoring Systems	Comparing data continuously with observed system or with prescribed behavior such as leakage monitoring in long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

8. List out the steps of Development of Expert Systems

The process of ES development is iterative. Steps in developing the ES include −

- Identify Problem Domain
- Design the System
- Develop the Prototype
- Test and Refine the Prototype
- Develop and Complete the ES
- Maintain the ES

9. What are the Benefits of Expert Systems

- Availability − they are easily available due to mass production of software.
- Less Production Cost − Production cost is reasonable. This makes them affordable.
- Speed − they offer great speed. They reduce the amount of work an individual puts in.
- Less Error Rate − Error rate is low as compared to human errors.
- Reducing Risk − they can work in the environment dangerous to humans.
- Steady response − they work steadily without getting motional, tensed or fatigued.

10. List out the expert System Technology

- Expert System Development Environment
- Tools
- Shells

11. Give an importance of Expert System Shells

The part of an expert system that does not contain any domain specific or case specific knowledge is the expert system shell. A single expert system shell can be used to build a number of different expert systems. An example of an expert system shell is CLIPS.

12. Write notes on Knowledge Engineering of experts system?

- Takes knowledge from experts and inputs it into the expert system.
- Usually choose which expert system shell to use.
- Responsible for entering meta-rules.

13. What is CLIPS?

- A CLIP is C Language Integrated Production System – an expert system shell.
- A CLIP uses a LISP-like notation to enter rules.

14. Explain Backward Chaining in Expert Systems

Backward chaining is often used in expert systems that are designed for medical diagnosis:

- For each hypothesis, H:
- If H is in the facts database, it is proved.
- Otherwise, if H can be determined by asking a question, then enter the user's answer in the facts database. Hence, it can be determined whether H is true or false, according to the user's answer**.**

15. Write a Simple Medical Expert System?

Rules

- If headache then prescribe pain killer
- If headache and sore throat and coughing than diagnose flu
- If tired and headache then diagnose glandular fever
- If tired and sore throat then diagnose tonsillitis
- If tired than diagnose stress.

16. Define Knowledge acquisition in AI?

- Knowledge acquisition from human experts the "paradox of expertise"

19. **What are the Activities in AI**

- Knowledge acquisition
- Knowledge representation
- Knowledge inferencing
- Knowledge transfer to the user

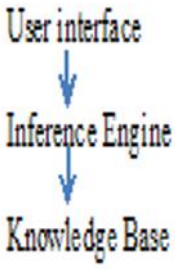
20. **Write MYCIN rule example:**

1.IF the infection is meningitis
2.AND patient has evidence of serious skin or soft tissue infection
3.AND organisms were not seen on the stain of the culture

4.AND type of infection is bacterial

*5.*THEN There is evidence that the organism (other than those seen on cultures or smears) causin the infection is *Staphylococus coagpus.*

21. **Draw the Three Major ES Components in AI**



22. **List out all ES Components in AI**

- Knowledge Acquisition Subsystem
- Knowledge Base
- Inference Engine
- User Interface
- Blackboard (Workplace)
- Explanation Subsystem (Justifier)
- Knowledge Refining System
- User
- Most ES *do not have* a Knowledge Refinement Component

23. **List out the two Basic Knowledge Base Elements**

- Facts
- Special heuristics, or rules that direct the use of knowledge
- Knowledge is the primary raw material of ES
- Incorporated knowledge representation

24. **How Expert Systems Work in Artificial intelligence**

- Development
- Consultation
- Improvement

25. **What are the Problem Areas Addressed by Expert Systems**

- Interpretation systems
- Prediction systems
- Diagnostic systems
- Design systems
- Planning systems

- Monitoring systems
- Debugging systems
- Repair systems
- Instruction systems
- Control systems

26. What is Meta knowledge? How Meta knowledge is represented in rule based expert systems**? (May/June 2016)**

In ES, Meta knowledge refers to knowledge about the operation of knowledge-based systems

- Meta knowledge is knowledge about knowledge and expertise.
- Most successful expert systems are restricted to as small a domain as possible.
- In an expert system, ontology is the Meta knowledge that describes everything known about the problem domain.
- Wisdom is the meta knowledge of determining the best goals of life and how to obtain them

27. Write any four earliest expert systems**? (May/June 2016)**
    - MYCIN
    - DART
    - XOON
    - Expert systems shells.

PART-B

1. What is an expert system shell **(R)**
2. What are common pitfalls in planning an expert system **(R)**
3. What is knowledge acquisition? Explain in detail **(R)**
4. Discuss briefly about meta knowledge **(U)**
5. i) Discuss briefly about the EMYCIN in detail **(U)**
   ii) Illustrate Heuristics with an example. **(An)**
   iii) Classify the XOON and DART in detail and write its application. **(Ap)**
6. Draw the schematic diagram of an expert system. Explain all the relevant components. **(Ap)**
7. Explain the various stages of expert system development? **(An)**
8. (i) Explain about the knowledge acquisition **(U) (May/June 2016)**

(ii) Write the Characteristics features of Expert systems **(U) (May/June 2016)**

9.(i) Explain the basic components of an expert systems **(U) (May/June 2016)**

(ii) Write any six applications of expert systems. **(U) (May/June 2016)**

COURSE OUTCOME: Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.