ARTIFICIAL INTELLIGENCE

NOTE:

MAKAUT course structure and syllabus of 5th semester has been changed from 2020. Previously **ARTIFICIAL INTELLIGENCE** was in 7th Semester. This subject has been shifted in 5th semester in present curriculum. Subject organization has been changed slightly. Taking special care of this matter we are providing chapterwise relevant MAKAUT university solutions and some model questions & answers for newly introduced topics, so that students can get an idea about university questions patterns.

INTELLIGENT AGENT

Multiple Choice Type Questions

1. Agents are [WBUT 2017]
   a) autonomous
   b) adaptive
   c) both (a) and (b)
   d) none of these

Answer: (c)

2. Artificial Intelligence is [WBUT 2019]
   a) programming with intelligence
   b) putting more memory to a computer
   c) making machine intelligent
   d) playing games

Answer: (c)

3. Which element in agent is used for selecting external actions? [WBUT 2019]
   a) Perceive
   b) Performance
   c) Learning
   d) Actuator

Answer: (d)

Short Answer Type Questions

1. a) What is perception? [WBUT 2008]

Answer:

In perception the environment is scanned by means of various sensory organs, real or artificial, and the scene is decomposed into separate objects in various spatial relationships.

b) Define intelligent agent. [WBUT 2008]

OR,

What is an agent? [WBUT 2011]

Explain different types of environment related to intelligent agent. [WBUT 2008]

OR,

What is an agent? Describe various agent types. [WBUT 2016]

OR,

What is an agent in AI? What are the types of agent? Discuss about environment for agent. [WBUT 2017, 2018]

Answer:

Intelligent Agent:

An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**. An agent is a computer system situated in some environment, and that is capable of *autonomous action* in this environment in order to meet its design objectives.

An intelligent agent is a computer system that is capable of *flexible* autonomous action in order to meet its design objectives. By *flexible,* we mean that the system must be:

- *responsive*: agents should perceive their environment (which may be the physical world, a user, a collection of agents, the Internet, etc.) and respond in a timely fashion to changes that occur in it,
- *proactive*: agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behaviour and take the initiative where appropriate, and
- *social*: agents should be able to interact, when they deem appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities.

Intelligent agents are usually classified into five classes based on their degree of perceived intelligence and capability:

1. simple reflex agents
2. model-based reflex agents
3. goal-based agents
4. utility-based agents
5. learning agents.

The environment can be classified into the following:

Accessible* vs. *inaccessible

An accessible environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state. Most moderately complex environments (including, for example, the everyday physical world and the Internet) are inaccessible. The more accessible an environment is, the simpler it is to build agents to operate in it.

Deterministic* vs. *non-deterministic

As we have already mentioned, a deterministic environment is one in which any action has a single guaranteed effect. The physical world can to all intents and purposes be regarded as non-deterministic. Non-deterministic environments present greater problems for the agent designer.

Episodic* vs. *non-episodic

In an episodic environment, the performance of an agent is dependent on a number of discrete episodes, with no link between the performance of an agent in different scenarios. Episodic environments are simpler from the agent developer's perspective because the agent can decide what action to perform based only on the current episode - it need not reason about the interactions between this and future episodes.

Static* vs. *dynamic

A static environment is one that can be assumed to remain unchanged except by the performance of actions by the agent. A dynamic environment is one that has other processes operating on it, and which hence changes in ways beyond the agent's control. The physical world is a highly dynamic environment.

Discrete vs. continuous
An environment is discrete if there are a fixed, finite number of actions and percepts in it. Russell and Norvig give a chess game as an example of a discrete environment, and taxi driving as an example of a continuous one.

2. What are the disadvantages of table driven agent? **[WBUT 2011]**
Answer:
Disadvantages of Table driven agent
1. The table needed for something as simple as an agent that can only play chess would be about 3510° entries.
2. It would take quite a long time for the designer to build the table.
3. The agent has no autonomy at all, because the calculation of best actions is entirely built in.
   So if the environment changed in some unexpected way, the agent would be lost.
4. Even if we gave the agent a learning mechanism as well, so that it could have a degree of autonomy, it would take forever to learn the right value for all the table entries.

3. a) What is percept sequence? **[WBUT 2016]**
b) What is agent system?
Answer:
a) A percept is an input that an intelligent agent receives at a given moment. The **percept sequence** is the complete history of every and any percept that has been received. Importantly, a rational agent's choice of action can only depend on its percept sequence and - conversely - not on anything it hasn't perceived.

b) Multi-agent systems consist of agents and their environment. A multi-agent system is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems can be used to solve problems that are difficult or impossible for an individual agent or a monolithic system to solve.

Long Answer Type Questions

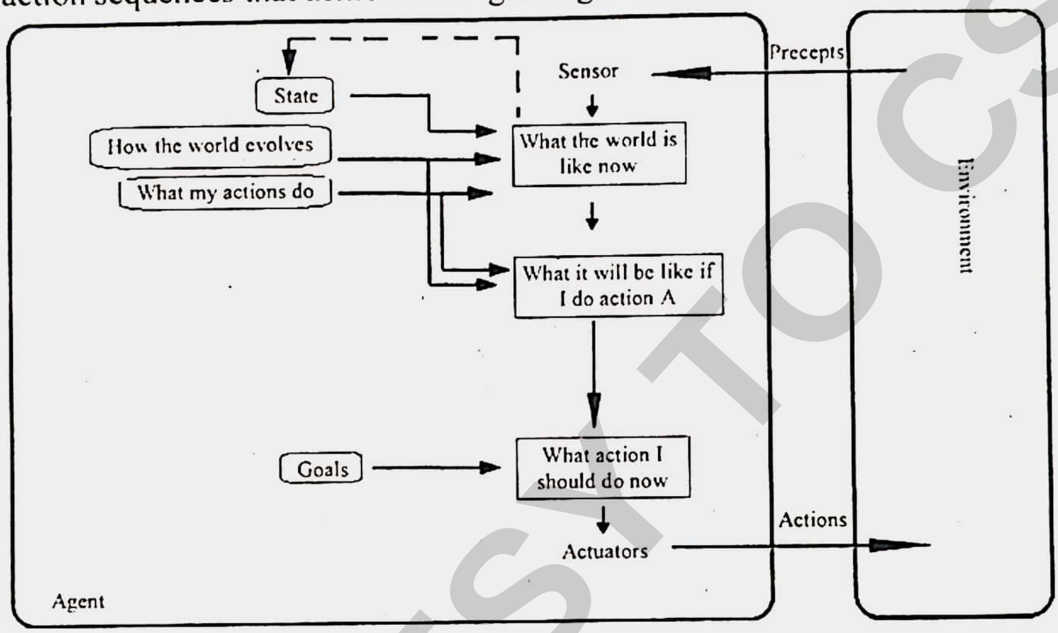
1. Discuss on 'agents as search procedure'. **[WBUT 2011]**
Answer:
This notion of search is computation inside the agent. It is different from searching in the world, when it may have to act in the world, for example, an agent searching for its keys, lifting up cushions, and so on. It is also different from searching the web, which involves searching for information. Searching in this chapter means searching in an internal representation for a path to a goal.
The idea of search is straightforward: the agent constructs a set of potential partial solutions to a problem that can be checked to see if they truly are solutions or if they could lead to solutions. Search proceeds by repeatedly selecting a partial solution, stopping if it is a path to a goal, and otherwise extending it by one more arc in all possible

ways. When an agent is given a problem, it is usually given only a description that lets it recognize a solution, not an algorithm to solve it. It has to search for a solution.

2. a) Describe goal based agent system. **[WBUT 2017]**

Answer:

Goal-based agents expand on the capabilities of the model-based agents, by using "goal" information. Goal information describes situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state. Search and planning are the subfields of artificial intelligence devoted to finding action sequences that achieve the agent's goals.



b) What do you mean by a table driven agent? What is the problem of this agent? **[WBUT 2017]**

Answer:

1st part: In table driven agent, the agent uses a lookup table for actions to be taken for every possible state of the environment.

2nd part: ***Refer to Question No. 3(b) of Short Answer Type Questions.***

4. Write short note on Intelligent Agents. **[WBUT 2006, 2013]**

Answer:

Refer to Question No. 1(b) of Short Answer Type Questions.

PROBLEM SOLVING & SEARCHING

Multiple Choice Type Questions

1. Heuristic search has [WBUT 2006, 2013]
   a) minimization of function value
   b) maximization of function value
   c) both (a) & (b)
   d) none of these

Answer: (c)

2. Decomposable problem can be represented by [WBUT 2006, 2007, 2011]
   a) OR graph
   b) AND graph
   c) AND-OR graph
   d) None of these

Answer: (c)

3. Which is not heuristic search? [WBUT 2006, 2008, 2010, 2013]
   a) Constrained satisfaction search
   b) Depth first search
   c) Simulated annealing
   d) Steepest ascent Hill climbing

Answer: (b)

4. Algorithm that gives optimal solution [WBUT 2007 2009, 2010, 2012, 2013, 2017, 2018]
   a) Hill climbing
   b) BFS
   c) Blind search
   d) A *

Answer: (d)

5. Which of the following is not a conflict resolution strategy in production system? [WBUT 2008, 2014, 2015]
   a) Production rules
   b) Recency
   c) Refractoriness
   d) Specificity

Answer: (a)

6. Uninformed search is also known as [WBUT 2010, 2017]
   a) Brute force search
   b) Hill climbing search
   c) Worst case search
   d) Blind search

Answer: (d)

7. The time complexity and space complexity for bidirectional breadth first search technique, respectively are (with branching factor b and depth d) [WBUT 2011]
   a) $O(b^d), O(b^d)$
   b) $O(b^{\frac{d}{2}}), O(b^{\frac{d}{2}})$
   c) $O(b^{\frac{d}{2}}), O(b^d)$
   d) $O(b^d), O(b^{\frac{d}{2}})$

Answer: (b)

8. The term 'Optimality', so far one of the performance measuring indices of any search technique is concerned, refers to [WBUT 2011]
   a) time complexity
   b) space complexity
   c) both (a) and (b)
   d) none of these

Answer: (c)

9. The main advantage of any heuristic search algorithm over blind search one is with respect to [WBUT 2011]

a) time complexity
b) space complexity
c) completeness
d) optimality

Answer: (b)

10. Depth first search procedure uses [WBUT 2011, 2017]

a) AND graph
b) OR graph
c) AND-OR graph
d) none of these

Answer: (d)

11. In Minimax algorithm search process obeys [WBUT 2011, 2017, 2018]

a) breadth first search fashion
b) depth first search fashion
c) best first search fashion
d) none of these

Answer: (b)

12. Iterative Deepening search procedure is [WBUT 2011]

a) optimal with respect to time consumption
b) optimal with respect to space consumption
c) both (a) and (b)
d) none of these

Answer: (c)

13. Searching techniques are used for [WBUT 2012]

a) goal node searching
b) optimization of search space
c) finding goal distance of the goal node from start node
d) all of these

Answer: (d)

14. Hill climbing has potential problems like [WBUT 2012]

a) lake
b) foothill trap
c) garden
d) all of these

Answer: (b)

15. The form of heuristic function of A* is [WBUT 2012]

a) $f^*(n)=g^*(n)*h^*(n)$
b) $f^*(n)=g^*(n)+h^*(n)$
c) $f^*(n)=g^*(n)+h(n)$
d) none of these

Answer: (d)

16. Inheritable knowledge is best presented by [WBUT 2012]

a) OR graph
b) AND graph
c) AND-OR graph
d) none of these

Answer: (d)

17. If in a problem the number of initial states is much more than the number of final states we should use [WBUT 2014, 2015]

a) backward reasoning
b) forward reasoning
c) both (A) and (B)
d) none of these

Answer: (a)

18. Which is NOT a heuristic search? [WBUT 2015]
   a) A* search
   b) steepest ascent Hill-climbing
   c) Simulated annealing
   d) Depth first search

Answer: (d)

19. What is meant by simulated annealing in artificial intelligence? [WBUT 2016]
   a) Returns an optimal solution when there is a proper cooling schedule
   b) Returns an optimal solution when there is no proper cooling schedule
   c) It will not return an optimal solution when there is a proper cooling schedule
   d) None of the mentioned

Answer: (a)

20. A* algorithm is based on [WBUT 2016]
   a) Breadth-first-search
   b) Depth-first-search
   c) Best-first-search
   d) Hill climbing

Answer: (c)

21. Which search agent operates by interleaving computation and action? [WBUT 2016]
   a) Offline search
   b) Online search
   c) Breadth-first search
   d) Depth-first search

Answer: (b)

22. Which search method takes less memory? [WBUT 2016]
   a) Depth-First Search
   b) Breadth-First search
   c) Both (a) and (b)
   d) Linear Search

Answer: (a)

23. Which is the most straight forward approach for planning algorithm? [WBUT 2016]
   a) Best-first search
   b) State-space search
   c) Depth-first search
   d) Hill-climbing search

Answer: (b)

24. Uniformed search is also known as: [WBUT 2018]
   a) Brute force search
   b) Hill climbing search
   c) Worst case search
   d) Blind search

Answer: (d)

25. Depth first search producers uses [WBUT 2018]
   a) AND graph
   b) OR graph
   c) AND-OR graph
   d) None of these

Answer: (d)

26. An algorithm is complete if [WBUT 2019]
   a) it terminates with a solution when one exists
   b) it starts with a solution
   c) it does not terminate with a solution
   d) it has a loop

Answer: (a)

27. Heuristic function h (n) is defined as the [WBUT 2019]
   a) Lowest path cost
   b) Cheapest path from root to goal node
   c) Estimated cost of cheapest path from root to goal node
   d) Average path cost

Answer: (c)

28. In many problems the path to goal is irrelevant, this class of problems can be solved using [WBUT 2019]
   a) informed search techniques
   b) uniformed search techniques
   c) local search techniques
   d) only (a) and (b)

Answer: (c)

29. Which type of mathematical problems are defined as a set of objects whose state must satisfy a number of constraints or limitations? [WBUT 2019]
   a) Constraints satisfaction Problems
   b) Uniformed Search Problems
   c) Local Search Problems
   d) Only (a) and (b)

Answer: (a)

30. What is the full form of STRIPS? [WBUT 2019]
   a) STandard Research Institute Problem Solver
   b) STanford Research Institute Problem Solver
   c) STdard Refinement of Information for Problem Solving
   d) None of these

Answer: (b)

Short Answer Type Questions

1. Discuss the benefits of a production system. [WBUT 2006, 2010, 2012]

OR,

What is a production system? [WBUT 2016]

Answer:

A production system is a method of modeling human problem solving. When trying to solve everyday problems, people may respond to an external condition with appropriate actions; they also form plans to guide their behavior. One way to uncover a person's problem-solving strategies is to collect a protocol: the problem solver's own account of thoughts and intentions, spoken while solving the problem. From these 'think aloud' protocols, the researcher tries to construct a problem space consisting of the person's states of knowledge and the operators needed to move from one state to another. A production system is one means of encoding a problem space, in a form that can be run on a computer. It has three main components: a database, called the working memory that represents a person's current state of knowledge; a set of production rules that operate on the working memory; and an interpreter that decides which rule to fire in a particular state.

Production systems have certain advantages over conventional computer programs that make them suitable for use in expert systems: programs that simulate the performance of human experts.

The benefits are:

- **Flexibility:** Production systems use the same basic IF-THEN format to represent knowledge in very different domains.
- **Modularity:** The functional units of a production system are independent, self-contained chunks of knowledge, any one of which can be altered or replaced without disabling the entire production system and without requiring the modification of other rules. Such alterations might modify or restrict the behavior of the system, but will not cripple it. This is because the rules in a production system are separate from the program that runs them: the rules do not interact with one another directly but only through changes to the working memory.
- **Cognitive plausibility:** Production systems can be made to reason either forwards, from initial evidence towards a conclusion, or backwards, from a hypothesis to the uncovering of the right kind of evidence that would support that hypothesis, or by a combination of the two.

2. What is combinatorial explosion? **[WBUT 2007]**

OR,

Briefly discuss combinatorial explosion. **[WBUT 2016, 2017, 2019]**

Answer:

The naive way of solving combinatorial problems can be paraphrased as `generate and test': In a first step one enumerates all combinations from which one selects all solutions in the second step. In most cases however, `generate and test' is simply not feasible. This is obvious if the set of combinations is infinite. But even if it is finite then it is usually very large, i.e. exponentially large in size of the problem description. In this case, the generation step runs into a combinatorial explosion (from which it usually returns only several billions of years later).

3. What is the hill-climbing technique? Describe it. **[WBUT 2007]**

OR,

Briefly explain the problems with Hill Climbing search. How it is different from Gradient Descent Search? **[WBUT 2014]**

Answer:

Hill climbing is an optimization technique which belongs to the family of local search. It is a relatively simple technique to implement, making it a popular first choice. Although more advanced algorithms may give better results, there are situations where hill climbing works well. Hill climbing attempts to maximize (or minimize) a function $f(x)$, where x are discrete states. These states are typically represented by vertices in a graph, where edges in the graph encode nearness or similarity of a graph. Hill climbing will follow the graph from vertex to vertex, always locally increasing (or decreasing) the value of f, until a local maximum (or local minimum) x_m is reached. Hill climbing can

also operate on a continuous space: in that case, the algorithm is called gradient ascent (or gradient descent if the function is minimized).
The algorithm is started with a random (potentially bad) solution to the problem. It sequentially makes small changes to the solution, each time improving it a little bit. At some point the algorithm arrives at a point where it cannot see any improvement anymore, at which point the algorithm terminates. Ideally, at that point a solution is found that is close to optimal, but it is not guaranteed that hill climbing will ever come close to the optimal solution.
Gradient descent methods can move in any direction that the ridge or alley may ascend or descend. Hence, gradient descent is generally preferred over hill climbing when the target function is differentiable. Hill climbers, however, have the advantage of not requiring the target function to be differentiable, so hill climbers may be preferred when the target function is complex.

4. What do you mean by completeness of a search? Why DFS is not always complete? [WBUT 2008, 2012, 2015]

OR,

What do you mean by completeness of a search method? When do you think BFS & DFS can be incomplete? [WBUT 2013, 2014]

Answer:

A search strategy is said to be *complete* if it is guaranteed to find a solution when there is one.
DFS progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found, or until it hits a node that has no children. Then the search backtracks, returning to the most recent node it hasn't finished exploring. No matter how deep the current node is, DFS will always go deeper if it has a child.
The major weakness of DFS is that it will fail to terminate if there is an infinite path "to the left of" the path to the first solution. In other words, for many problems DFS is not complete: a solution exists but DFS cannot find it.
Both Depth First Search and Breadth First Search are complete for finite state spaces. Both are systematic. They will explore the entire search space before reporting failure. This is because the termination criterion for both is the same. Either they pick the goal node and report success, or they report failure when OPEN (we assume a data structure called OPEN to store the candidates that we have generated.) becomes empty. The only difference is where the new nodes are placed in the OPEN list. Since for every node examined, all unseen successors are put in OPEN, both searches will end up looking at all reachable nodes before reporting failure. If the state space is infinite, but with finite branching then depth first search may go down an infinite path and not terminate. Breadth First Search, however, will find a solution, if there exists one. If there is no solution, both algorithms will not terminate for infinite state spaces.

5. What are the pitfalls of hill-climbing algorithm? [WBUT 2008, 2014]

Answer:
Problems of Hill Climbing includes local maxima, ridges and plateau.
Local maxima: A problem with hill climbing is that it will find only local maxima. Unless the heuristic is convex, it may not reach a global maximum.
Ridges: A ridge is a curve in the search place that leads to a maximum, but the orientation of the ridge compared to the available moves that are used to climb is such that each move will lead to a smaller point. In other words, each point on a ridge looks to the algorithm like a local maximum, even though the point is part of a curve leading to a better optimum.
Plateau: Another problem with hill climbing is that of a plateau, which occurs when we get to a "flat" part of the search space, i.e. we have a path where the heuristics are all very close together. This kind of flatness can cause the algorithm to cease progress and wander aimlessly.

6. Compare blind search and heuristic search. **[WBUT 2011]**

Answer:
Sometimes we may not get much relevant information to solve a problem. Suppose we lost our car key and we are not able to recall where we left, we have to search for the key with some information such as in which places we used to place it. It may be our pant pocket or may be the table drawer. If it is not there then we have to search the whole house to get it. The best solution would be to search in the places from the table to the wardrobe. Here we need to search blindly with fewer clues. This type of search is called uninformed search or **blind search**. There are two popular AI search techniques in this category: breadth first search and depth first search.
We ca-. olve the problem in an efficient manner if we have relevant information, clues or hints. The clues that help solve the problem constitute heuristic information. So informed search is called **heuristic search**. Instead of searching one path or many paths just like that informed search uses the given heuristic information to decide whether or not to explore the current state further. Hill climbing is an AI search algorithm that explores the neighboring states and chooses the most promising state as successor and continue searching for the subsequent states. Once a state is explored, hill climbing algorithm simply discards it. Hill climbing search technique can make substantial savings if it has reliable information. It has to face three challenges: foothill, ridge and plateau. Best first search is a heuristic search technique that stores the explored states as well so that it can backtrack if it realizes that the present path proves unworthy.

7. Discuss on the components of AI production system. **[WBUT 2011]**
OR,
Describe different components of AI. **[WBUT 2019]**

Answer:
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 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.

8. Did depth limited search always show the completeness property? Explain. **[WBUT 2013, 2014]**

Answer:
Even though depth-limited search cannot follow infinitely long paths, nor can it get stuck in cycles, in general the algorithm is not complete since it does not find any solution that lies beyond the given search depth. But if the maximum search depth is chosen to be greater than the depth of a solution the algorithm becomes complete.

9. Define α-cutoff & β-cutoff of a game tree. **[WBUT 2013, 2014]**

Answer:
When we traverse the search tree in depth-first order:

- At each MAX node n, alpha(n) = maximum value found so far
- At each MIN node n, beta(n) = minimum value found so far

Alpha cutoff: Stop searching below MIN node n if beta(*n*) <= alpha(*i*) for some MAX node ancestor i of n.
Beta cutoff: Given a MAX node n, cut off the search below n (i.e., don't generate or examine any more of n's children) if alpha(*n*) >= beta(*i*) for some MIN node ancestor *i* of *n*.

10. Explain the phenomenon 'Stuck at local optimum' from the perspective of Hill climbing. How can it be resolved using Simulated Annealing? **[WBUT 2013]**

OR,

Explain the working principle of simulated annealing algorithm. **[WBUT 2019]**

Answer:
Hill climbing attempts to iteratively improve the current state by means of an evaluation function. It always attempts to make changes that improve the current state. Hill-climbing can only advance if there is a higher point in the adjacent landscape. The main problem that hill climbing can encounter is that of "stuck at local optima". This occurs when the algorithm stops making progress towards an optimal solution; mainly due to the lack of immediate improvement in adjacent states.
Simulated annealing solves the problem of "stuck at local minima or maxima" by allowing worse moves (lesser quality) to be taken some of the time. That is, it allows some uphill steps so that it can escape from local minima.
Unlike hill climbing, simulated annealing chooses a random move from the neighbourhood (hill climbing chooses the best move from all those available – at least when using steepest descent (or ascent)).If the move is better than its current position

then simulated annealing will always take it. If the move is worse (i.e. lesser quality) then it will be accepted based on some probability.

The law of thermodynamics state that at temperature, t, the probability of an increase in energy of magnitude, δE, is given by

$$P(\delta E) = \exp(-\delta/kt)$$

where k is a constant known as Boltzmann's constant.

The simulation in the algorithm calculates the new energy of the system. If the energy has decreased then the system moves to this state. If the energy has increased then the new state is accepted using the probability returned by the above formula. A certain number of iterations are carried out at each temperature and then the temperature is decreased. This is repeated until the system freezes into a steady state.

This equation is directly used in simulated annealing, although it is usual to drop the Boltzmann constant as this was only introduced into the equation to cope with different materials. Therefore, the probability of accepting a worse state is given by the equation

$$P = \exp(-c/t) > r$$

where

c	=	the change in the evaluation function
t	=	the current temperature
r	=	a random number between 0 and 1

The probability of accepting a worse move is a function of both the temperature of the system and of the change in the cost function. It can be appreciated that as the temperature of the system decreases the probability of accepting a worse move is decreased. This is the same as gradually moving to a frozen state in physical annealing. Also note, that if the temperature is zero then only better moves will be accepted which effectively makes simulated annealing act like hill climbing.

11. Justify the statement:
DFS can be viewed as a special case of Depth-limited search. **[WBUT 2014]**

Answer:

Depth-first search will not find a goal if it searches down a path that has infinite length. So, in general, depth-first search is not guaranteed to find a solution, so it is not complete. This problem is eliminated by limiting the depth of the search to some value l. However, this introduces another way of preventing depth-first search from finding the goal: if the goal is deeper than l it will not be found Regular depth-first search is a special case, for which $l = \infty$.

12. a) What is the difference between Greedy best-first search and A* search?
b) Under what condition is breadth-first search optimal?
c) Show that any monotonic heuristic is admissible. **[WBUT 2015]**

Answer:

a) A* is like Greedy Best-First-Search in that it can use a heuristic to guide itself. In the simple case, it is as fast as Greedy Best-First-Search. However, A* is better as it combines the pieces of information that Dijkstra's algorithm uses (favoring vertices that are close to the starting point) *and* information that Greedy Best-First-Search uses (favoring vertices that are close to the goal).

b) Breadth-first search is optimal if the path cost is a non-decreasing function of the depth of the node. The most common such scenario is that all actions have the same cost.

c) A heuristic function is said to be consistent, or monotone, if it is always at most equal to the estimated distance from any neighboring vertex plus the step cost of reaching that neighbor.

Formally, for every node N and each successor P of N, the estimated cost of reaching the goal from N is no greater than the step cost of getting to P plus the estimated cost of reaching the goal from P. That is: $H(n) \le C(N,P) + h(P)$ and $h(G) = 0$ where

- h is the consistent heuristic function
- N is any node in the graph
- P is any descendant of N
- G is any goal node
- c(N,P) is the cost of reaching node P from N

A consistent heuristic is also admissible, i.e. it never overestimates the cost of reaching the goal (the opposite however is not always true!). This is proved by induction on m, the length of the best path from node to goal. By assumption, $h(N_m) \le h^*(N_m)$, where $h^*(n)$ denotes the cost of the shortest path from n to the goal. Therefore,

$$h(N_{m+1}) \le c(N_{m+1}, N_m) + h(N_m) \le c(N_{m+1}, N_m) + h^*(N_m) = h^*(N_{m+1}),$$

making it admissible. (N_{m+1} is any node whose best path to the goal, of length m+1, goes through some immediate child N_m whose best path to the goal is of length m.)

13. When does BFS give optimal solution? **[WBUT 2016]**

Answer:

Breadth-first search is optimal if the path cost is a non-decreasing function of the depth of the node. The most common such scenario is that all actions have the same cost.

14. What is blind search technique? Explain with examples. **[WBUT 2017, 2018]**

Answer:

Refer to Question No. 6 (1st part) of Short Answer Type Questions.

15. Show that if a heuristic is consistent then f(n) is monotonically non decreasing along any path. **[WBUT 2017]**

Answer:

A heuristic is consistent if for every node n, every successor n' of n generated by any action a,

h(n) ≤ c(n,a,n') + h(n')

If h is consistent, we have

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

$= g(n) + c(n,a,n') + h(n')$

$\geq g(n) + h(n)$

$= f(n)$

i.e., $f(n)$ is non-decreasing along any path.

Hence, if $h(n)$ is consistent, f along any path is non-decreasing.

16. Write iterative deepening algorithm with example. **[WBUT 2018]**

Answer:

Iterative deepening search calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth. So basically, we do DFS in a BFS fashion. First do DFS to depth 0 (i.e., treat start node as having no successors), then, if no solution found, do DFS to depth 1, etc.

Algorithm

Begin

until solution found

    do DFS with depth cutoff c

    c = c+1

End

Successive depth-first searches are conducted – each with depth bounds increasing by 1 as shown in the figure below:

Depth bound = 1   Depth bound = 2   Depth bound = 3   Depth bound = 4

17. Describe depth limited search. **[WBUT 2019]**

Answer:

Depth limited search is the new search algorithm for uninformed search. The unbounded tree problem happens to appear in the depth-first search algorithm, and it can be fixed by imposing a boundary or a limit to the depth of the search domain. We will say that this limit as the depth limit which makes the DFS search strategy more refined and organized into a finite loop. We denote this limit by l and thus this provides the solution to the infinite path problem that originated earlier in the DFS algorithm. Thus, Depth limited search can be called an extended and refined version of the DFS algorithm. In a nutshell, we can say that to avoid the infinite loop status while execution of the codes, depth limited search algorithm is being executed into a finite set of depth called depth limit.

18. Does BFS ensure completeness and optimality? Justify. [WBUT 2019]

Answer:

BFS is complete — if the shallowest goal node is at depth d, it will eventually find it after expanding all the nodes shallower than d.

As far as optimality of the solution is concerned, the BFS algorithm stops at the shallowest goal found. We must remember that the shallowest goal node need not necessarily be the optimal goal node.

BFS is optimal if the path cost is a non-decreasing function of d. Usually, BFS is applied when all the actions have the same cost.

Long Answer Type Questions

1. a) How are production system and control strategies applied in solving AI problems?

b) You are given two jars – *a* 4-litre one and *a* 3-litre one. Neither has any measuring mark on it. How can you get 2 litres of water into the 4-litre jug. With the help of state-space diagram, find a solution. [WBUT 2006, 2007, 2012]

OR,

You are given two jugs, a 4-gallon one and a 3-gallon one. Neither have any 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? Give the state-space diagram, describe the production rules and give a possible solution. [WBUT 2016]

Answer:

a) ***Refer to Question No. 1 of Short Answer Type Questions.***

b) ***State Space***

The problem can be described as set of ordered pairs of integers (x, y)

x = 0, 1, 2, 3 or 4 where x number of gallons of water in the 4- gallon jug

y = 0, 1, 2, or 3 where y the quantity of water in the 3- gallon jug

start state : (0, 0)

goal state : (2, *n*) for any value of n

We can represent the problem using production rules. The general form of productions looks like this:

<antecendent1, antecedent2, ... antecedentN>
(condition1, condition2, ...)
---> (consequent1, consequent2, consequentM)
where the conditions on the antecedents specify the applicability of an operation

Production Rules

1 $(x,y) \rightarrow (4,y)$ if $x < 4$	Fill the 4-gallon jug
2 $(x,y) \rightarrow (x,3)$ if $y < 3$	Fill the 3-gallon jug
3 $(x,y) \rightarrow (x-d,y)$ if $x > 0$	Pour some water out of the 4-gallon jug
4 $(x,y) \rightarrow (x,y-d)$ if $x > 0$	Pour some water out of the 3-gallon jug
5 $(x,y) \rightarrow (0,y)$ if $x > 0$	Empty the 4-gallon jug on the ground
6 $(x,y) \rightarrow (x,0)$ if $y > 0$	Empty the 3-gallon jug on the ground
7 $(x,y) \rightarrow (4,y-(4-x))$ if $x+y \geq 4$ and $y > 0$	Pour water from the 3-gallon jug into the 4-gallon jug until the 4-gallon jug is full
8 $(x,y) \rightarrow (x-(3-y),3)$ if $x+y \geq 3$ and $x > 0$	Pour water from the 4-gallon jug into the 3-gallon jug until the 3-gallon jug is full
9 $(x,y) \rightarrow (x+y, 0)$ if $x+y \leq 4$ and $y > 0$	Pour all the water from the 3-gallon jug into the 4-gallon jug
10 $(x,y) \rightarrow (0, x+y)$ if $x+y \leq 3$ and $x > 0$	Pour all the water from the 4-gallon jug into the 3-gallon jug

One solution

Gallons in the 4-Gallon Jug	Gallons in the 3-Gallon Jug	Rule Applied
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5
0	2	9
2	0	

2. Discuss and compare hill climbing and best-first search technique.
[WBUT 2006, 2008, 2016]

OR,

Compare and contrast hill climbing and best-first search procedures.
[WBUT 2009, 2012]

OR,

Compare and contrast Best-First and Hill climbing search. [WBUT 2018]

Answer:

1st Part:

Hill climbing technique & its Drawbacks:

Refer to Question No. 3 & 5 of Short Answer Type Questions.

2nd Part:

The **best first search** allows us to switch between paths thus gaining the benefit of both approaches. At each step the most promising node is chosen. If one of the nodes chosen generates nodes that are less promising it is possible to choose another at the same level and in effect the search changes from depth to breadth. If on analysis these are no better than this previously unexpanded node and branch is not forgotten and the search method reverts to the descendants of the first choice and proceeds, backtracking as it were.

This process is very similar to steepest ascent, but in hill climbing once a move is chosen and the others rejected the others is never reconsidered whilst in best first they are saved to enable revisits if an impasse occurs on the apparent best path. Also the best available state is selected in best first even its value is worse than the value of the node just explored whereas in hill climbing the progress stops if there are no better successor nodes. The best first search algorithm will involve an OR graph which avoids the problem of node duplication and assumes that each node has a parent link to give the best node from which it came and a link to all its successors. In this way if a better node is found this path can be propagated down to the successors. This method of using an OR graph requires 2 lists of nodes.

3. a) What do you mean by admissibility and consistency of a heuristic function?
[WBUT 2007]

OR,

What do you mean by consistency of a heuristic? [WBUT 2016]

b) Validate each of the following statements giving brief explanation.

i) The heuristic function "Sum of Manhattan distances" for 8-puzzle problem is consistent.

ii) If heuristic is consistent then the heuristic is admissible but the converse is not true. [WBUT 2007]

Answer:

a) A heuristic h(n) is admissible if it never overestimates the cost to the goal from node n; i.e. it is always optimistic .

A heuristic h(n) is consistent if for any nodes A and B h(B) >=h(A) +c(A,B)

Intuitively, this says that our heuristic will become more accurate (less optimistic) as we approach the goal.

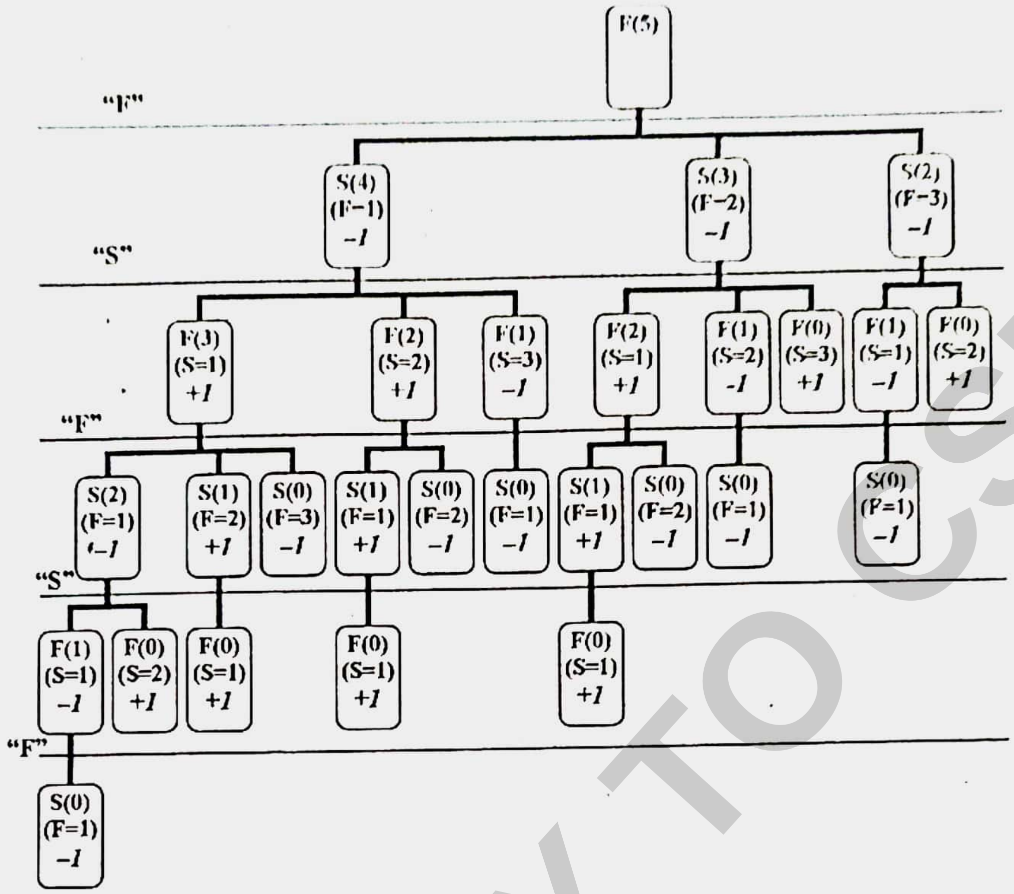
b) i) sum of Manhattan distances of tiles to their goal location is a heuristic that can be used for 8-puzzle problem. The Manhattan distance is the number of moves required if no other tiles are in the way. An heuristic is said to be consistent if one step takes us from n to n', then $h(n) \le h(n')$ + cost of step from n to n'. The minimum number of moves to get from some position $<i_1,j_1>$ to some other position $<i_2,j_2>$ is the Manhattan distance, so this heuristic is consistent and A* is admissible.

ii) Proof: **If heuristic is consistent then the heuristic is admissible**
This is proved by induction on m, the length of the best path from node to goal. By assumption, $h(N_m) \le h^*(N_m)$, where $h^*(n)$ denotes the cost of the shortest path from n to the goal.
Therefore, $h(N_{m+1}) \le c(N_{m+1}, N_m) + h(N_m) \le c(N_{m+1}, N_m) + h^*(N_m) = h^*(N_{m+1})$
making it admissible. (N_{m+1} is any node whose best path to the goal, of length $m + 1$, goes through some immediate child N_m whose best path to the goal is of length m.)
Consider a search problem in which the states are nodes along a path $P = n_0, n_1 \ldots n_m$ where n_0 is the start state, n_m is the goal state and for all i there is an action from n_i to n_{i+1} of cost 1. The cheapest cost from any node n_i to the goal is then $k(n_i) = m - i$. Now we define a heuristic function $h(n) = m - 2 \times [i/2]$. Clearly this is admissible since for all states n_i, $h(n_i) \le k(n_i)$ but is not consistent.

4. The game of NIM is played as follows. Two players alternate in removing one, two, or three pennies from a stack initially containing five pennies. The player who picks up the last penny loses.
   a) Draw the full game tree.
   b) Show that the player who has the second move can always win.
   c) Execute $\alpha - \beta$ pruning procedure on the game tree. How many terminal nodes are examined?
[WBUT 2007, 2012]

Answer:

a) F(X) → X pieces left for first Player; if X=0, first player wins.
S(Y) → Y pieces left for second Player; if Y=0, second player wins.
F = a → F picks up a sticks.
S = b → S picks up b sticks.



b) Let us now assign +1 to each game that F wins and -1 to each game that S wins. The values are assigned to each leaf node. Using minmax method as shown in the figure, we conclude that S will win the game when both F and S use optimal strategies.

From the tree it is pretty clear that

i. when first player picks 3 sticks second player can pick 1 penny to ensure victory.
ii. when first player picks 2 sticks second player can pick 2 pennies to ensure victory.
iii. when first player picks 1 stick second player can pick 3 pennies to ensure victory.

c) Let us use alpha-beta pruning from left to right. Here we know that the value of each node can either be +1 or -1. It can be seen that 5 nodes are not expanded.

5. Consider the following 8-puzzle problem: **[WBUT 2009]**

Given the critical state:

2	8	3
1		4
7	6	5

and Goal the state:

1	2	3
8		4
7	6	5

i) List the operators.
ii) Select a heuristic function for the 8-puzzle problem
iii) Solve the problem by A^* algorithm with your selected heuristic function.

OR,

Consider the following problem:
The 8-puzzle consists if 3×3 board with 8 numbered tiles and a blank space. Each tile has a number on it. A tile adjacent to the blank space can only slide into that space. The game consists of a starting position and a specified goal position. The goal is to transform the starting position into the goal position by sliding the tiles around. Convert this problem to a state space search problem. Show that it works on the following example:

Start		
2	8	3
1	6	4
7		5

			Goal
2	2	3	
1		4	
7	6	5	

[WBUT 2014]

OR,

Consider the following arrangement and solve the problem using A* search. Define the State space, write the operations, define the heuristic and also find whether this heuristic is admissible or not. Also show the solution. [WBUT 2016, 2017]

Initial State:

2	8	3
1	6	4
7		5

Final State:

1	2	3
8		4
7	6	5

Answer:

i) *Operators*: blank moves left, right, up, or down.

ii) The "Tiles Out of Place heuristic" or "*Hamming priority function*" algorithm can be used.

The heuristic involves summing the number of tiles that are not in their correct location for the Goal State. This heuristic is admissible, because it is clear that any tile that is out of place must be moved at least once. Below is the result of this heuristic when tiles 1 to 8 are in the Start State as shown in the figure.

```
Tiles Out of Place = 1 + 1 + 0 + 0 + 0 + 0 + 0 + 1 = 3
```

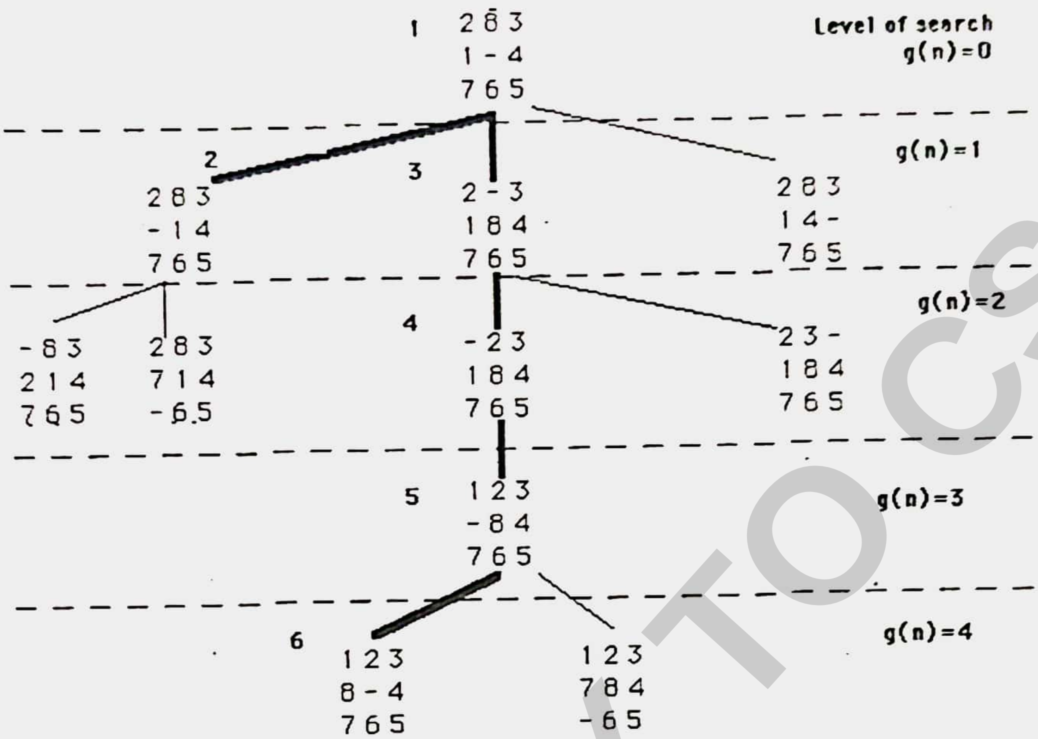
iii) A* uses the evaluation function:

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

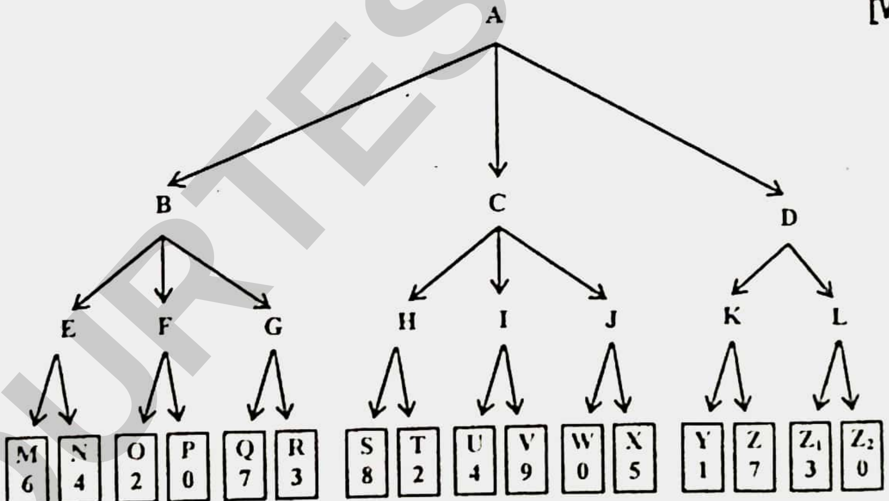
where *g(n)* measures the actual length of the path from the start state to the state *n*, and *h(n)* is a heuristic estimate of the distance from a state *n* to a goal state.

The following figure shows the full best-first search of the eight puzzle graph using the value of *h(n)* as the "number of tiles out of place". The number at the top of each state represents the order in which it was taken off of the open list. The levels of the graph are used to assign *g(n)*.

f(n) = g(n) + h(n)
g(n) = distance from start
h(n) = # tiles out of place



6. Given a game tree for a two-ply game, where the evaluation function for winding are given at the leaf nodes. Assume that the game is opened by the maximizer. [WBUT 2009]

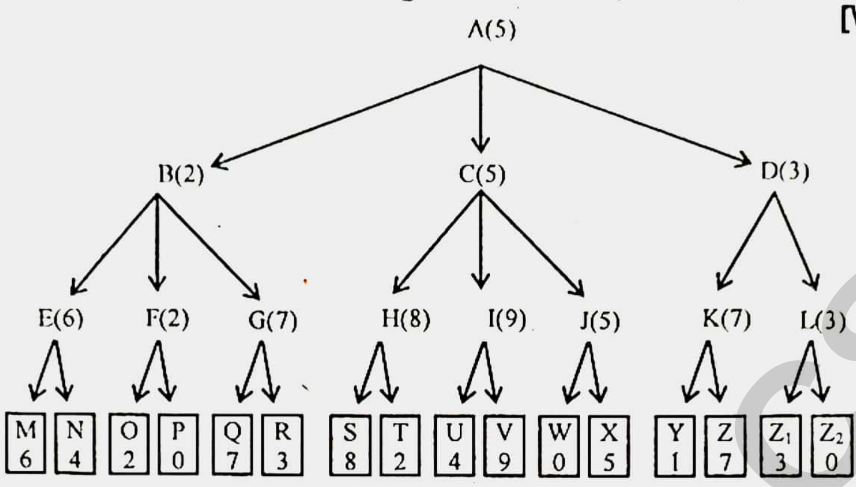


a) Using Minimax algorithm, determine which nodes the maximizer and the minimize should select in their first turn.

b) Identify the nodes that will be pruned by invoking Alpha-Beta algorithms.

OR,

Consider the following game tree where the evaluation function values for winning is given at leaf nodes. Assume that the game tree is opened by the maximizer. **[WBUT 2017]**

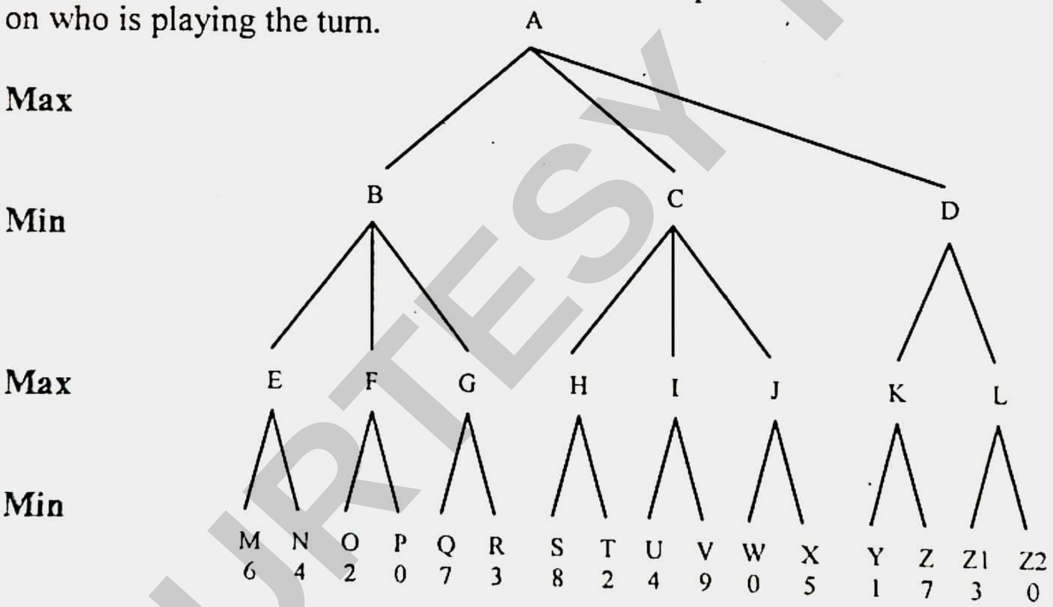


i) **Applying minimax algorithm determine which nodes the maximizer and minimizer would select in their turns.**

ii) **How many nodes will be pruned using alpha-beta pruning?**

Answer:

In the figure below we write down the backed up values in the internal nodes depending on who is playing the turn.



a) The moves are as follows:

Max chooses C
Min chooses J
Max chooses X

b) Node R will not be pruned and all the rest would be pruned.

7. Prove each of the following statements:
a) Breadth first search is a special case of uniform cost search.
[WBUT 2010, 2014, 2015]

OR,

Is BFS identical to uniform cost search? Justify your answer. [WBUT 2017]

b) Breadth first, depth first and uniform cost search are special cases of Best First Search. [WBUT 2010, 2014]
c) Uniform cost search is a special case of A* search. [WBUT 2010, 2015]

Answer:

a) When all step costs are equal, the cost g(n) of a path from the start node to a node n is proportional to depth(n). Then uniform-cost search reproduces BFS with the modified search algorithm (a node is tested to be the goal when it is removed from the fringe).

b) Best-first search is the general term for an algorithm that expands the node with the least cost according to some evaluation function by keeping nodes to expand on a priority queue. Let f(n) is the cost function applied to node n.
Breadth first uses priority based on minimum path length and order of expansion. Thus Breadth first is best-first with f(n)=depth(n).
Depth first is prioritized on maximum path length and reverse order of expansion uniform-cost is prioritized on path cost from the start node. Thus Depth first is best-first with f(n)=-depth(n) or f(n) = 1/depth(n).
Uniform-cost is prioritized on path cost from the start node. Uniform-cost search expands the node with the lowest path cost.
Hence, breadth first, depth first and uniform cost search are special cases of Best First Search.

c) A* has the objective function = $g(n) + h(n)$, and uniform-cost is a special case with $h(n) = 0$ (thus f(n) = g(n); Contrariwise, A* with g(n) = 0 (thus f(n) = h(n)) is greedy.

8. a) How do you evaluate any search technique? [WBUT 2011]

Answer:

Evaluating the performance of an AI-based search technique can be complicated. We are concerned with mainly the measurements:

- How quickly a solution is found
- How good the solution is

There are several types of problems for which all that matters is that a solution, any solution, be found with the minimum effort. For these problems, the first measurement is especially important. In other situations, the quality of the solution is more important.
The speed of a search is affected both by the size of the search space and by the number of nodes actually traversed in the process of finding the solution. Because backtracking from dead ends is wasted effort, we want a search that seldom retraces its steps.
In AI-based searching, there is a difference between finding the best solution and finding a good solution. Finding the best solution can require an exhaustive search because sometimes this is the only way to know that the best solution has been found. Finding a

good solution, in contrast, means finding a solution that is within a set of constraints— it does not matter if a better solution might exist.
Some search techniques work better in certain situations than in others, so it is difficult to say whether one search method is always superior to another. But some search techniques have a greater probability of being better for the average case. In addition, the way a problem is defined can sometimes help one choose an appropriate search method.

b) Discuss on Bidirectional search technique. **[WBUT 2011]**

Answer:
Bidirectional Search, as the name implies, searches in two directions at the same time: one forward from the initial state and the other backward from the goal. This is usually done by expanding tree with branching factor b and the distance from start to goal is d. The search stops when searches from both directions meet in the middle. Bidirectional search is a brute-force search algorithm that requires an explicit goal state instead of simply a test for a goal condition. Once the search is over, the path from the initial state is then concatenated with the inverse of the path from the goal state to form the complete solution path.
Bidirectional search still guarantees optimal solutions. Assuring that the comparisons for identifying a common state between the two frontiers can be done in constant time per node by hashing. The time complexity of Bidirectional Search is O(b^d/2) since each search need only proceed to half the solution path. Since at least one of the searches must be breadth-first in order to find a common state, the space complexity of bidirectional search is also O(b^d/2). As a result, it is space bound in practice.

Advantages
The merit of bidirectional search is its speed. Sum of the time taken by two searches (forward and backward) is much less than the $O(b^d)$ complexity.
It requires less memory.

Disadvantages
Implementation of bidirectional search algorithm is difficult because additional logic must be included to decide which search tree to extend at each step.
One should have known the goal state in advance.
The algorithm must be too efficient to find the intersection of the two search trees.
It is not always possible to search backward through possible states.

9. a) Write down the disadvantages of hill climbing search procedure. **[WBUT2011]**
OR,
What are the three major problems of hill-climbing technique? **[WBUT 2016]**
b) When does simulated annealing algorithm behave like hill climbing?
[WBUT 2011]

Answer:
The various disadvantages are:
1. Local maxima
A surface with two local maxima. (Only one of them is the global maximum.) If a hill-climber begins in a poor location, it may converge to the lower maximum.

A problem with hill climbing is that it will find only local maxima. Unless the heuristic is convex, it may not reach a global maximum. Other local search algorithms try to overcome this problem such as stochastic hill climbing, random walks and simulated annealing.

2. Ridges and alleys

Ridges are a challenging problem for hill climbers that optimize in continuous spaces. Because hill climbers only adjust one element in the vector at a time, each step will move in an axis-aligned direction. If the target function creates a narrow ridge that ascends in a non-axis-aligned direction (or if the goal is to minimize, a narrow alley that descends in a non-axis-aligned direction), then the hill climber can only ascend the ridge (or descend the alley) by zig-zagging. If the sides of the ridge (or alley) are very steep, then the hill climber may be forced to take very tiny steps as it zig-zags toward a better position. Thus, it may take an unreasonable length of time for it to ascend the ridge (or descend the alley).

By contrast, gradient descent methods can move in any direction that the ridge or alley may ascend or descend. Hence, gradient descent or conjugate gradient method is generally preferred over hill climbing when the target function is differentiable. Hill climbers, however, have the advantage of not requiring the target function to be differentiable, so hill climbers may be preferred when the target function is complex.

3. Plateau

Another problem that sometimes occurs with hill climbing is that of a plateau. A plateau is encountered when the search space is flat, or sufficiently flat that the value returned by the target function is indistinguishable from the value returned for nearby regions due to the precision used by the machine to represent its value. In such cases, the hill climber may not be able to determine in which direction it should step, and may wander in a direction that never leads to improvement.

b) If the starting temperature is not too hot then the ending solution will be very close to the starting solution. Thus this will become a simple climbing algorithm.

10. Consider the 3-puzzle problem shown in figure.

2	3
1	

Initial

1	2
3	

Final

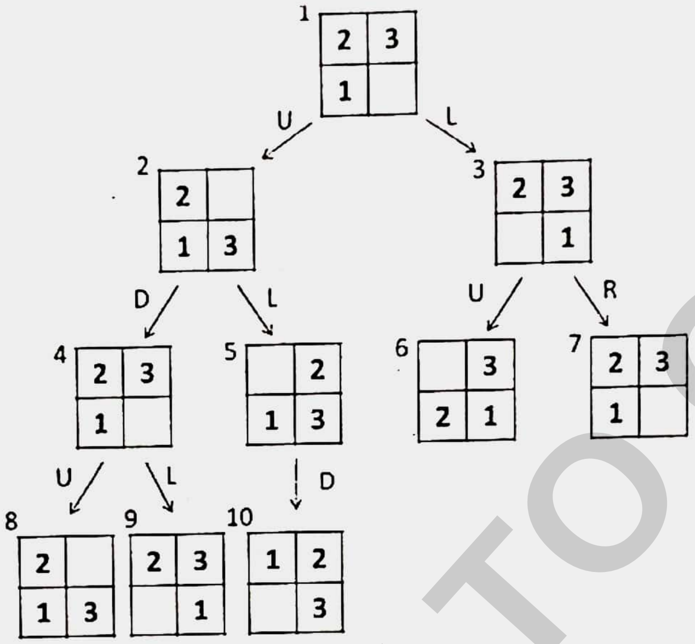
Figure

Possible operators (in order) are: up, down, left, right; Assume that repeated states are not detected.

a) Draw the search tree using breadth first search. [WBUT 2011, 2015]
b) Would depth first search find the goal? Explain. [WBUT 2011, 2015]
c) How many nodes would be generated if Iterative Deepening is used starting with depth increment one? [WBUT 2011]
d) A* search with the heuristic being the number of misplaced tiles. [WBUT 2015]

Answer:

a) Breadth first search will start from the root node, then expands all the successors of the root node, and then all their successors and so on. Breadth first search stops when first solution is found.



b) Depth-first search always expands the deepest node in the search tree. Notice that there was no mechanism to remember states that have been visited earlier. Depth first will not find a solution as it will start oscillating between movements U and D.

c) Iterative deepening will prevent the looping.

The no. of nodes generated would be: $(d+1)b^0 + (d)b^1 + (d-1)b^2 + \ldots b^d$

where b is the branching factor and d is the depth.

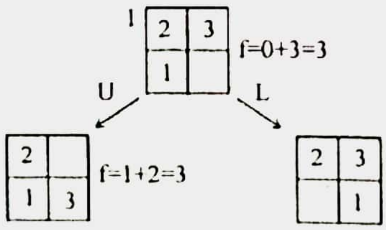
d) f(n) = g(n) + h(n)

h(n) = no. Of misplaced tiles

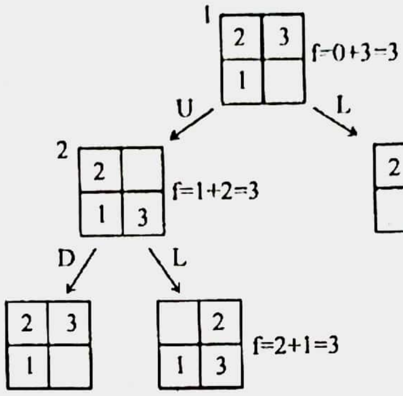
g(n)= distance from start

At the start state, h=-3 (all 3 tiles are misplaced)

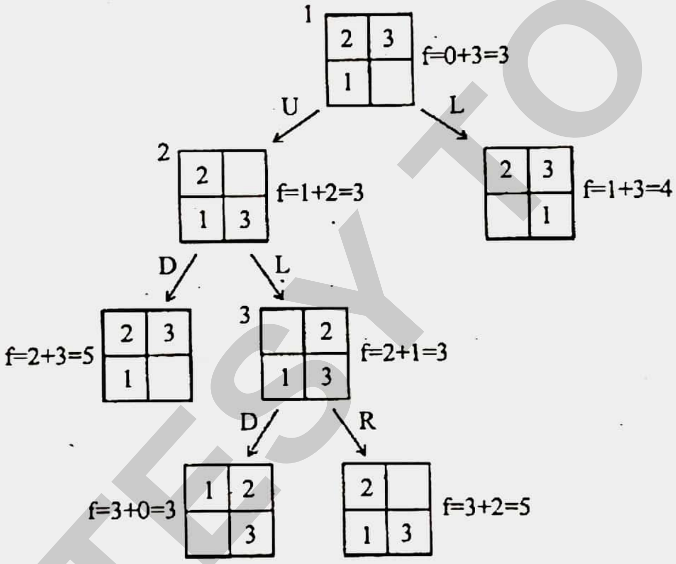
When the start state is opened we see two states as shown below.



The leftmost state has a lower cost so that one is chosen and opened. We now see states with costs 5, 3 and 4 as shown below.



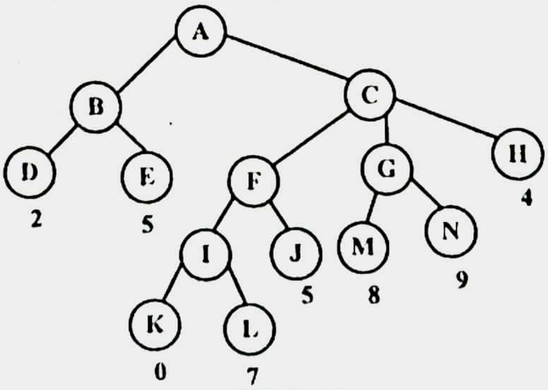
The state with lowest cost is opened and we see two more nodes including the goal node as shown below. We notice that the goal node has the lowest cost so we choose that and can finish the search. If some other node with a lower cost function value was still visible, A* search would choose that instead of the higher cost goal node. This is because A* tries to find the path with lowest cost.



11. Consider the following game tree in which static scores are all from first player's point of view. **[WBUT 2011]**

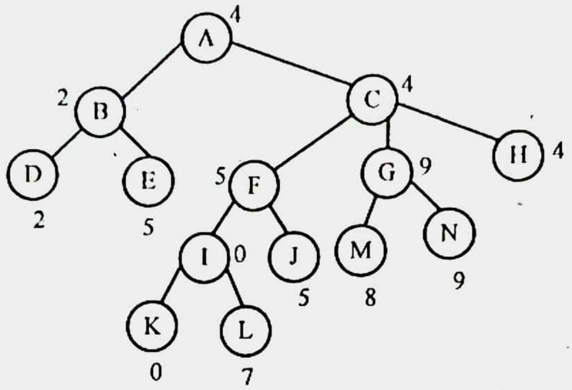
i) Which would be his best first move if MINIMAX algorithm is used?

ii) Which branches will be pruned if $\alpha - \beta$ pruning algorithm is used?

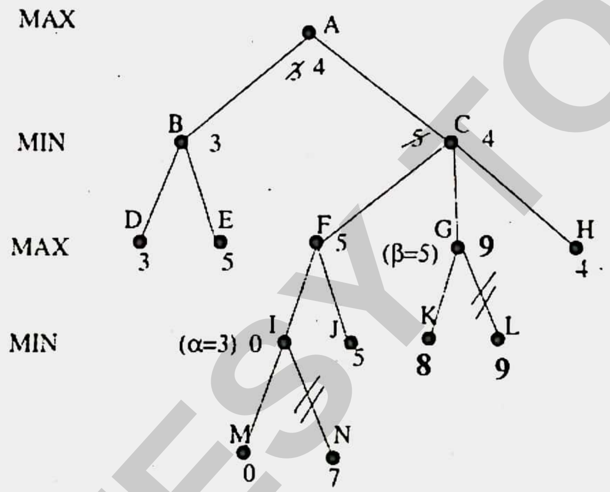


Answer:

i) The following graph shows the values associated with each of the nodes using MINIMAX algorithm. The best move for the first layer will be C.



ii) Alpha-beta pruning left to right



12. a) Consider the following problem:

A farmer is on the left bank of a river with a boat, a cabbage, a goat, and a wolf. The task is to get everything to the right bank of the river. The restrictions are as follows:

Only the farmer can handle the boat, when he is in the boat, there is only space for one more item and the farmer can't leave the goat alone either with the wolf, or with the cabbage.

Represent this problem as a state space search problem & show at least one solution path.

b) Define admissible & consistent heuristic. For a heuristic h, prove the following:

i) if h is consistent, then prove that $h(n) \le c(n,n') + h(n')$ is applicable for any descendent n' of n.

ii) if h is consistent, then prove that h is admissible also. **[WBUT 2013]**

Answer:

a) In this problem information can be represented in the form of the following 4-tuple. : (A, B, C, D) for farmer, wolf, goat and cabbage respectively. Each variable position stores the location of a particular object. For instance, the initial state where the farmer, the goat, the wolf and the cabbage are on the south bank is given by (S,S,S,S). The goal state is represented by (N,N,N,N).

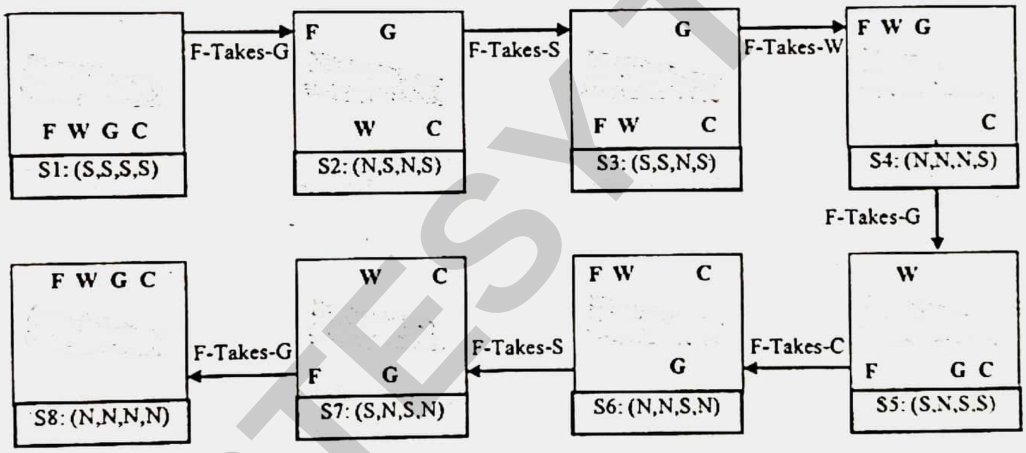
- There are 4 transforming operators:
  - Farmer-Takes-Self

    (A,B,C,D) (Opposite(A), B,C,D?)
  - Farmer-Takes-Wolf

    (A,B,C,D) (Opposite(A), Opposite(B),C,D)
  - Farmer-Takes-Goat

    (A,B,C,D) (Opposite(A), B, Opposite(C),D)
  - Farmer-Takes-Cabbage

    (A,B,C,D) (Opposite(A), B,C, Opposite(D))
- Finally, a state is safe as long as the wolf and the goat or the cabbage are not left unattended – this is the precondition to the above operators. While there are other possibilities here is one 7 step solution to the river problem.



b) A heuristic function is said to be **admissible** if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path. Formally, 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.

A consistent (or monotone) heuristic function is a function that estimates the distance of a given state to a goal state, and that is always at most equal to the estimated distance from any neighboring vertex plus the step cost of reaching that neighbor.

Formally, for every node N and every successor P of N generated by any action a, the estimated cost of reaching the goal from N is no greater than the step cost of getting to P plus the estimated cost of reaching the goal from P. In other words:

$$h(N) \leq c(N,P) + h(P)$$

where
h is the consistent heuristic function
N is any node in the graph
P is any descendant of N
G is any goal node
c(N,P) is the cost of reaching node P from N

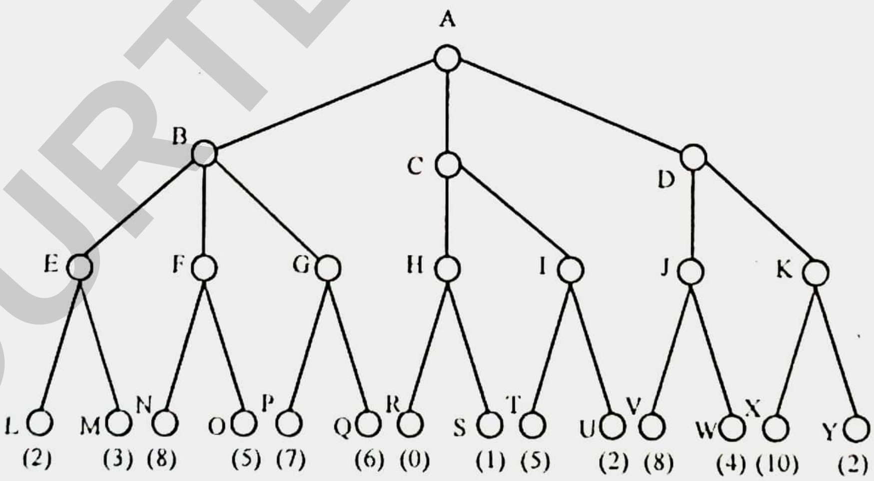
i) Let k(n) be the cost of the cheapest path from n to the goal node. We will prove by induction on the number of steps to the goal that h(n) <= k(n).
Base case: If there are 0 steps to the goal from node n, then n is a goal and therefore h(n) = 0 <= k(n).

Induction step: If n is i steps away from the goal, there must exist some successor n' of n generated by some action a such that n' is on the optimal path from n to the goal (via action a) and n' is i – 1 steps away from the goal.

Therefore, h(n) <=c(n, n') + h(n')
But by the induction hypothesis, h(n') <= k(n').
Therefore, h(n) _ c(n, n') + k(n') = k(n)
since n' is on the optimal path from n to the goal via action a.

ii) ***Refer to Long Answer Type Question No. 3(b)(ii).***

13. a) Define Constraint Satisfaction Problem. 4-queens problem seeks to place 4-queens in a 4×4 chess board such that no two queens will attack each other in either horizontal or vertical or diagonal way. Formulate this problem as CSP.
b) Consider the following game tree in which the static scores (in parentheses at the leaf nodes) are all from the first player's point of view. Assume that the first player is the maximizing player.



i) What move should the first player choose?
ii) What nodes would not need to be examined using alpha-beta cutoff algorithm, assuming that nodes are examined in left-to-right order? [WBUT 2013]

Answer:

a) 1st Part:

Constraint satisfaction problems (or CSPs) consist of variables with constraints on them. Many important real-world problems can be described as CSPs.

Examples of some of the most common problems include: The n-Queen problem i.e., the local condition is that no two queens attack each other, i.e. are on the same row, or column, or diagonal, a cryptography problem, map colouring problem etc.

Formally speaking, a constraint satisfaction problem (or CSP) is defined by a set of variables, X1;X2; : : : ;Xn, and a set of constraints, C1;C2; : : : ;Cm. Each variable Xi has a nonempty domain Di of possible values. Each constraint Ci 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, {Xi = vi, Xj =vj ,...} An assignment that does not violate any constraints is called a consistent or legal assignment. A complete assignment is one in which every variable is mentioned, and a solution to a CSP is a complete assignment that satisfies all the constraints. Some CSPs also require a solution that maximizes an objective function.

2nd Part:

4 queens problem as CSP

Variables: Qi (i is the column)

• Domains: Di = {1, 2, 3, 4} (the row)

• Constraints

– Qi≠Qj (cannot be in same row)

– |Qi - Qj| ≠ |i - j| (or same diagonal)

b) (i) Evaluating the minimax algorithm, we find the values at all the node by moving upwards from the leaf node.

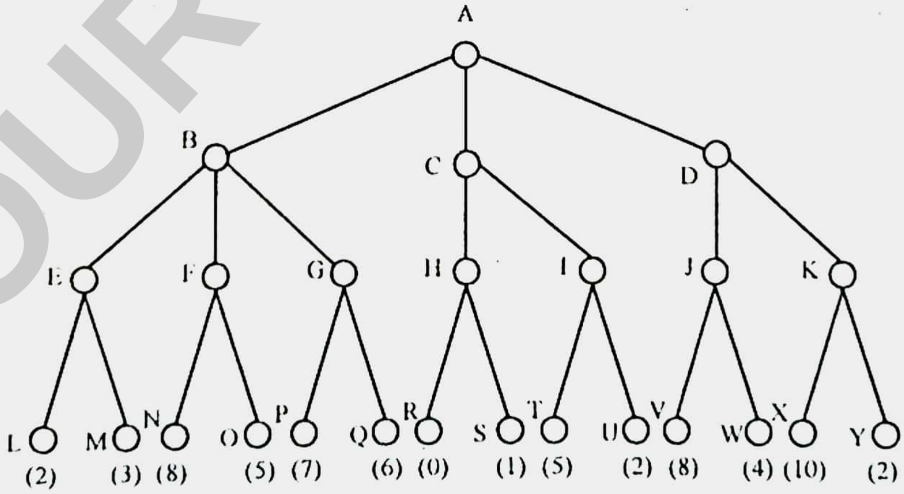
E=3, F=8, G=7, H=1, I=5, J=8, K=10

B=3, C=1, D= 8

A=8

We therefore find that the 1st player will choose D, which gives an outcome of 8 at node V.

(ii)



While examining the nodes from left to right MAX will explore L, M and N. Since value at N=8 and E has a value 3, it will not examine O. Again since P=7 which is more than E, MAX will not examine Q. So value at B =3. Again nodes R and S are examined and since the value at H=1. Thus MAX will ignore I, T and U since B=3 and H=1. Again similarly after examining V, W and X the node left out will be Y.
Hence the nodes that are not examined are: O, Q, I, T, U, Y

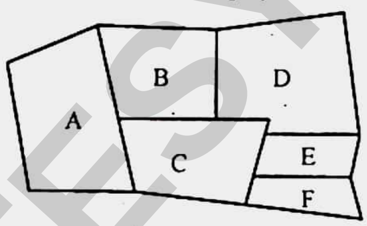
14. A farmer has a wolf, goat and cabbage. He wants to cross a river with all its possessions. He has a single boat that can carry at most one of his possessions on any trip. Moreover, an unattended wolf eats goat and unattended goat eats cabbage. How does he do the transportation?
Describe a production system stating clearly the production rules and control strategies. Now, show that your proposed production system solution can solve this problem. [WBUT 2014]

Answer:
States of the problem space are represented using the predicate
state(F,W,G,C)
Initial state given by the fact state(w,w,w,w) and goal state is (e,e,e,e) where w and e represent west and east of the river.
Rule to operate on state with F and W on the same side to produce state with
F and W on the other side :
move(state(X,X,G,C), state(Y,Y,G,C)) : -opp(X,Y).
opp(e,w).
/* ensures e/w swap */
opp(w,e).
/* ensures w/e swap *
The unsafe states may be represented by the rules
/* wolf eats goat */
unsafe(state(X,Y,Y,C)) :- opp(X,Y).
/* goat eats cabbage */
unsafe(state(X,W,Y,Y)) : - opp(X,Y)
Using above rules to modify the move rule:
/* farmer takes wolf to other side */
(1)
move(state(X,X,G,C), state(Y,Y,G,C)):
opp(X,Y), not (unsafe(state(Y,Y,G,C))).
Similarly,
/* farmer takes goat to other side */
(2)
move(state(X,W,X,C), state(Y,W,Y,C)):
opp(X,Y), not (unsafe(state(Y,W,Y,C))).
/* farmer takes cabbage to other side */
(3)
move(state(X,W,G,X), state(Y,W,G,Y)):

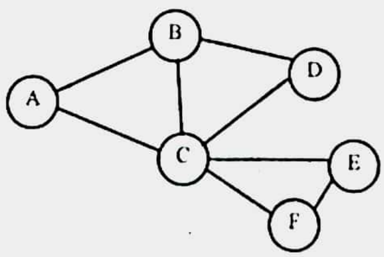
opp(X,Y), not (unsafe(state(Y,W,G,Y))).
/* farmer takes himself to other side */
(4)
move(state(X,W,G,C), state(Y,W,G,C)):
opp(X,Y), not (unsafe(state(Y,W,G,C))).
/* gets here when none of the above fires, system predicate 'fail' causes a backtrack*/
move(state(F,W,G,C), state(F,W,G,C)): fail.
Following the above rules:

Try farmer takes goat	e w e w
Try farmer takes self	w w e w
Try farmer takes wolf	e e e w
Try farmer takes goat	w e w w
Try farmer takes cabbage	e e w e
Try farmer takes wolf	w w w e
Try farmer takes goat	e w e e
BACKTRACK from	e w e e
BACKTRACK from	w w w e
Try farmer takes self	w e w e
Try farmer takes goat	e e e e

15. Consider the "map colouring problem" where a given map is to be coloured in a manner so that no neighbouring states of a country contain the same colour. Give a solution to following map colouring problem viewing it as a Constraint Satisfaction Problem. [WBUT 2014]



Answer:
The constraint satisfaction problem consists of 6 variables {A,B,C,D,E,F}. Each variable has the same domain {red, green, blue}. No two adjacent variables have the same value, so the constraints are: A≠B, A≠C, B≠C, B≠D, C≠D, C≠E, C≠F,D≠E, E≠F. Let us consider the domain as {Red, Green, Blue}.
This can be represented using a constraint graph as follows:



There are only 18 solutions (valid assignments). They can be counted as follows: Fix the color of SA (chosen because it is the most constraining variable). For each color of SA it remains 2 valid colors for each state, except T that still has 3. So let the backtracking algorithm a value for WA. Each color of SA (combined with the color already chosen for

WA) leaves only one color for NT, then one color for Q, then one for NSW, then one for V. So, there are 6 valid assignments for the states.
Final Solution could be:
A: Red, B: Green, C: Blue, D: Red, E: Green, F: Red

16. Prove that A* is admissible. **[WBUT 2015]**

Answer:
A* is both admissible and considers fewer nodes than any other admissible search algorithm with the same heuristic, because A* works from an "optimistic" estimate of the cost of a path through every node that it considers — optimistic in that the true cost of a path through that node to the goal will be at least as great as the estimate. But, critically, as far as A* "knows", that optimistic estimate might be achievable.
When A* terminates its search, it has, by definition, found a path whose actual cost is lower than the estimated cost of any path through any open node. But since those estimates are optimistic, A* can safely ignore those nodes. In other words, A* will never overlook the possibility of a lower-cost path and so is admissible.
Suppose now that some other search algorithm B terminates its search with a path whose actual cost is not less than the estimated cost of a path through some open node. Algorithm B cannot rule out the possibility, based on the heuristic information it has, that a path through that node might have a lower cost. So while B might consider fewer nodes than A*, it cannot be admissible. Accordingly, A* considers the fewest nodes of any admissible search algorithm that uses a no more accurate heuristic estimate.

17. Consider the following game tree. **[WBUT 2015]**

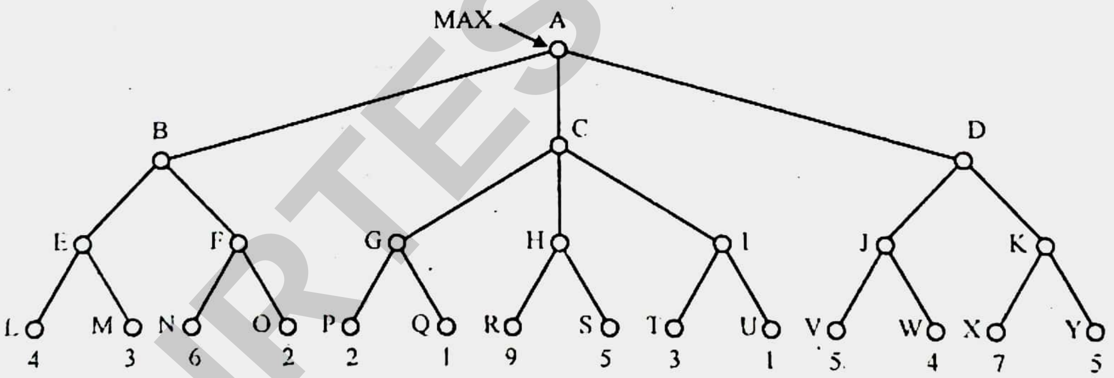
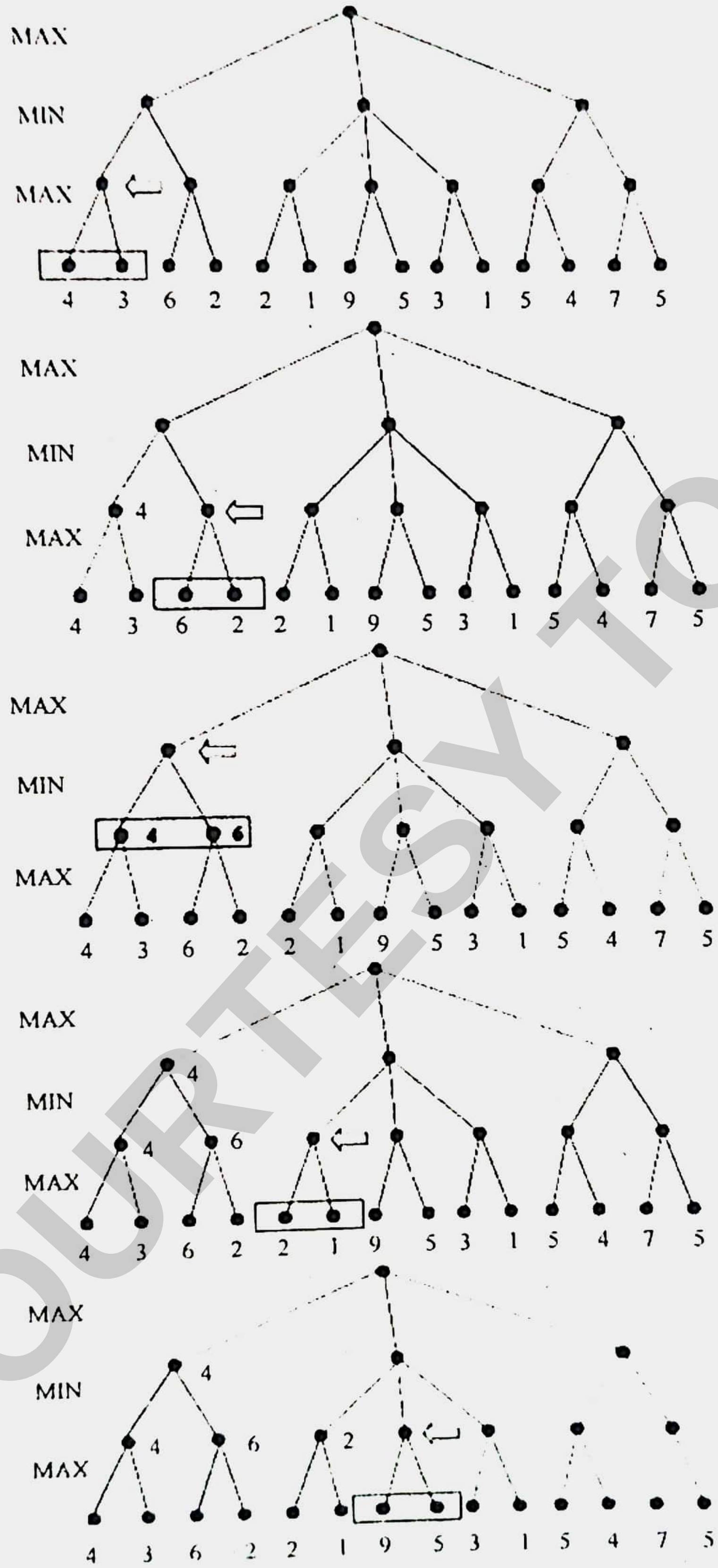


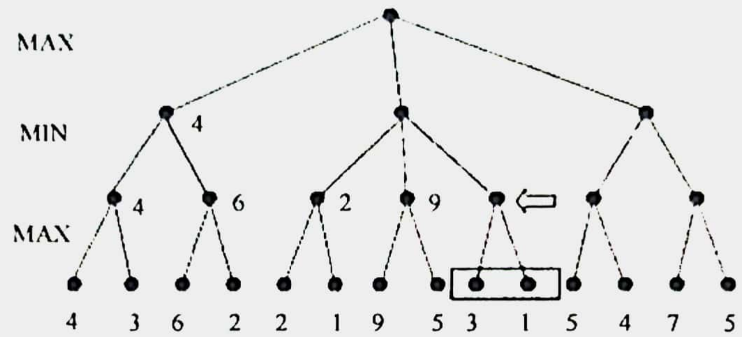
Fig: 1

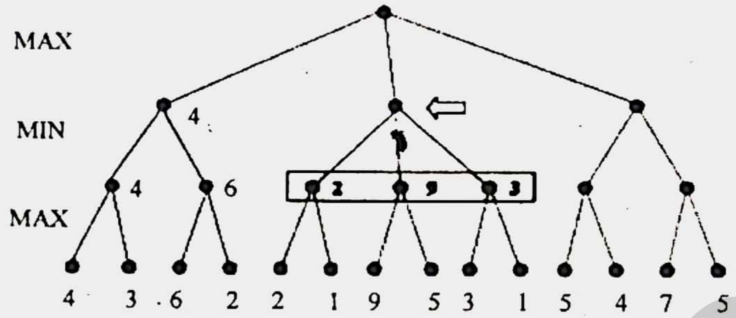
(i) **Using MINIMAX procedure, determine what moves should be chosen by the maximizer in his first turn.**
(ii) **Execute Alpha-Beta pruning on the above game tree. How many terminal nodes are examined? For each cutoff specify whether it is an Alpha-cutoff or Beta-cutoff.**

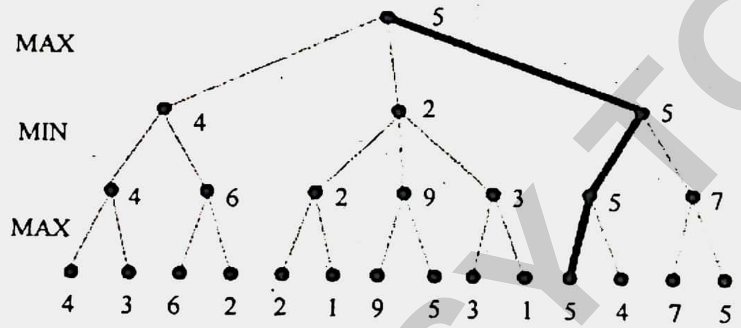
Answer:

i) The steps for minimax is as shown below.



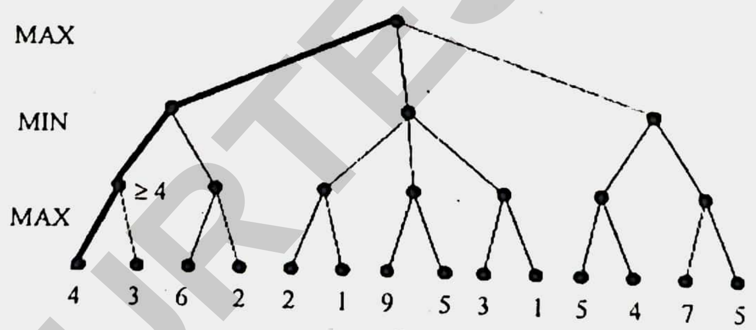


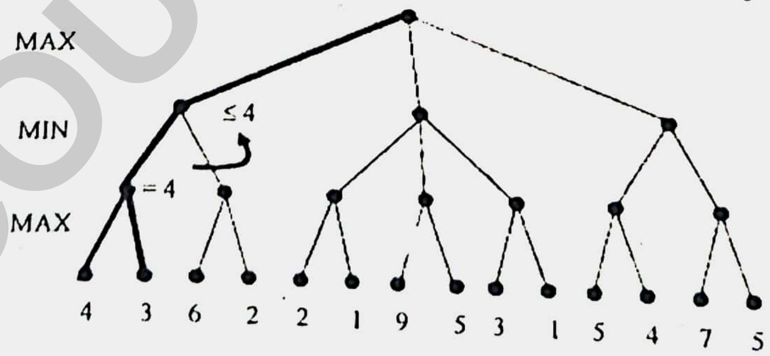


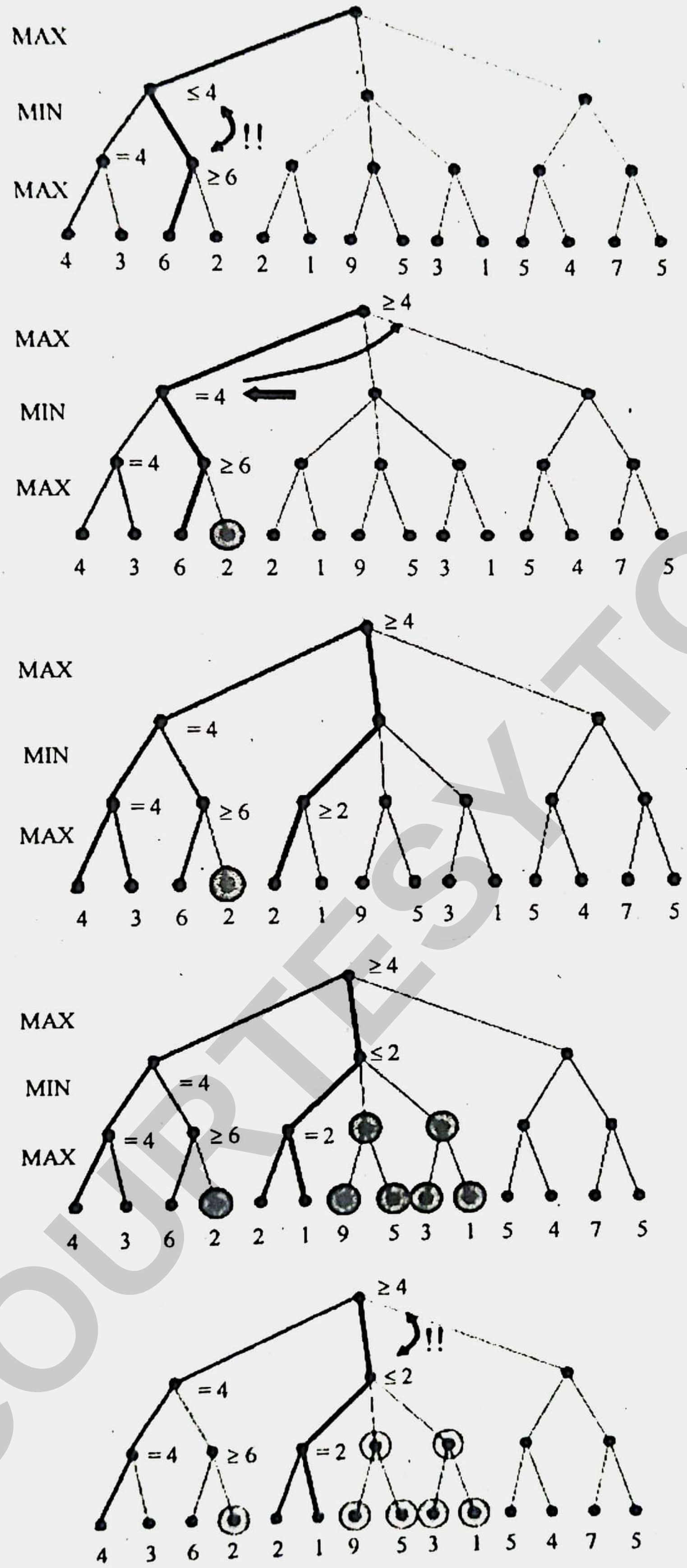


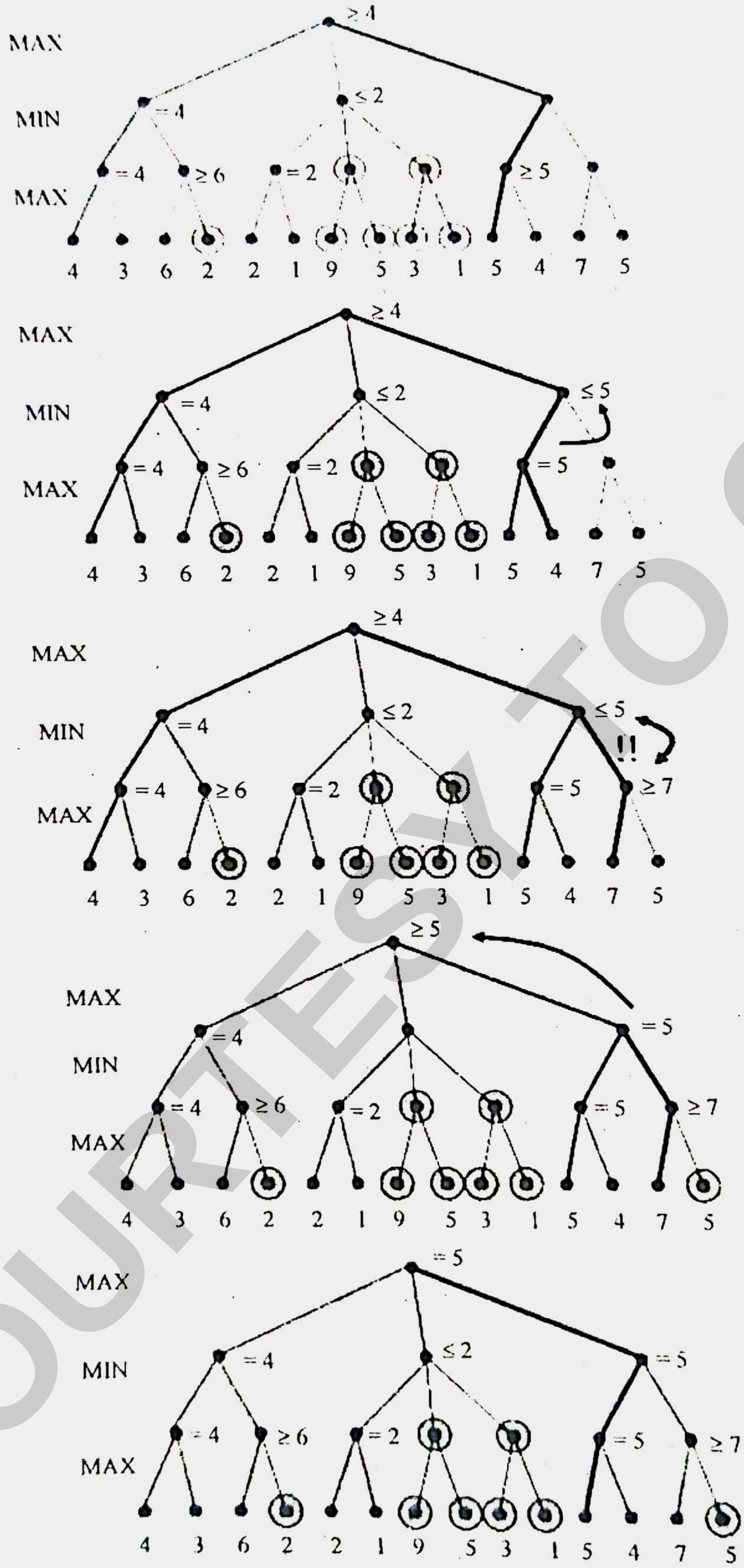
Thus the maximizer chooses node D.

ii) The steps for alpha-beta pruning are shown below:






MAX
MIN
MAX
≤ 4
= 4
≥ 6
!!
4 3 6 2 2 1 9 5 3 1 5 4 7 5
≥ 4
= 4
≥ 2
≤ 2
= 2



nodes that were never explored !!!

Alpha-cutoff occurs at when Min makes a move and Beta-cutoff occurs when Max makes a move as shown in the figure above.

18. a) Is uniform cost search a special case of Best First Search? Justify your answer.
b) Describe local beam search.
c) Three missionaries and three cannibals are standing at the left bank of a river. There is a boat having a capacity of taking two people and it can be driven by a missionary or a cannibal. If the number of missionaries is less than the number of cannibals at any bank, then cannibal will eat missionary. How is it possible for all the missionaries and cannibals to cross the river so that no missionary is getting eaten? Describe the state space of the problem. Describe the production rules for solving the problem. Show one solution of the problem. [WBUT 2017]

Answer:

a) *Refer to Question No. 7 (b) of Long Answer Type Questions.*

b) In "local beam search", the search begins with k randomly generated states and does the following:
- At each step, all the successors of all k states are generated
- If any one of the successors is a goal, the algorithm halts
- Otherwise, it selects the k best successors from the complete list and repeats
- The parallel search of beam search leads quickly to abandoning unfruitful searches and moves its resources to where the most progress is being made

c) **Each state space can be represented by**
State(no_of_missionaries, no_of_cannibals, side_of_the_boat)
where no_of_missonaries are the number of missionaries at left side of river, no_of_cannibals are the number of cannibals at the left side of river and side_of_the_boat is the side of the boat at particular state.
For our case
Initial State => State(3, 3, L) and
Final State => State(0, 0, R).
Where L represents left side and R represents right side of river.

Production Rules for Missionaries and Cannibals problem.

- Rule 1: (i, j) : Two missionaries can go only when i-2>=j or i-2=0 in one bank and i+2>=j in the other bank.
- Rule 2: (i, j) : Two cannibals can cross the river only when j-2<=i or i=0 in one bank and j+2 <= i or i+0 or i=0 in the other.
- Rule 3: (i, j) : One missionary and one cannibal can go in a boat only when i-1>=j-1 or i=0 in one bank and i+1>=j+1 or i=0 in the other.
- Rule 4: (i, j) : one missionary can cross the river only when i-1>=j or i=0 in one bank and i+1>=j in the other bank.
- Rule 5: (i, j) : One cannibal can cross the river only when j-1 < i or i=0 in one bank and j+1<=i or j=0 in the other bank of the river.

Let missionaries, cannibals and boat be denoted by M, C and B. The following table gives one possible solution.

	Left side	*Right side*
0. Initial setup:	CCCMMM B	
1. Two cannibals *cross over:*	MMMC	B CC
2. One comes back:	MMMCC	B C
3. Two cannibals go over again:	MMM	B CCC
4. One cannibal comes back:	MMMC	B CC
5. Two missionaries *cross:*	MC	B MMCC
6. A missionary *& a* cannibal *return:*	MMCC	B MC
7. Two missionaries *cross again:*	CC	B MMMC
8. A cannibal *returns:*	CCC	B MMM
9. Two cannibals *cross:*	C	B MMMCC
10. One cannibal returns:	CC	B MMMC
11. And brings over the third cannibal:	—	B MMMCCC

19. What do you mean by constraint satisfaction problem? Solve the following cryptography problem using constraint satisfaction search: **[WBUT 2018]**

S E N D
M O R E
=========
M O N E Y

Answer:

1st part: ***Refer to Question No. 13(a) (1st Part) of Long Answer Type Questions.***

2nd part:

Constraints:

- No two digits can be assigned to same letter.
- Only single digit number can be assigned to a letter.
- No two letters can be assigned same digit.
- Rule of arithmetic may be followed.

Initial state of problem.

D=?
E=?
Y=?
N=?
R=?
O=?
S=?
M=?
C1=?
C2=?
C1, C2, C3 stands for the carry variables respectively.

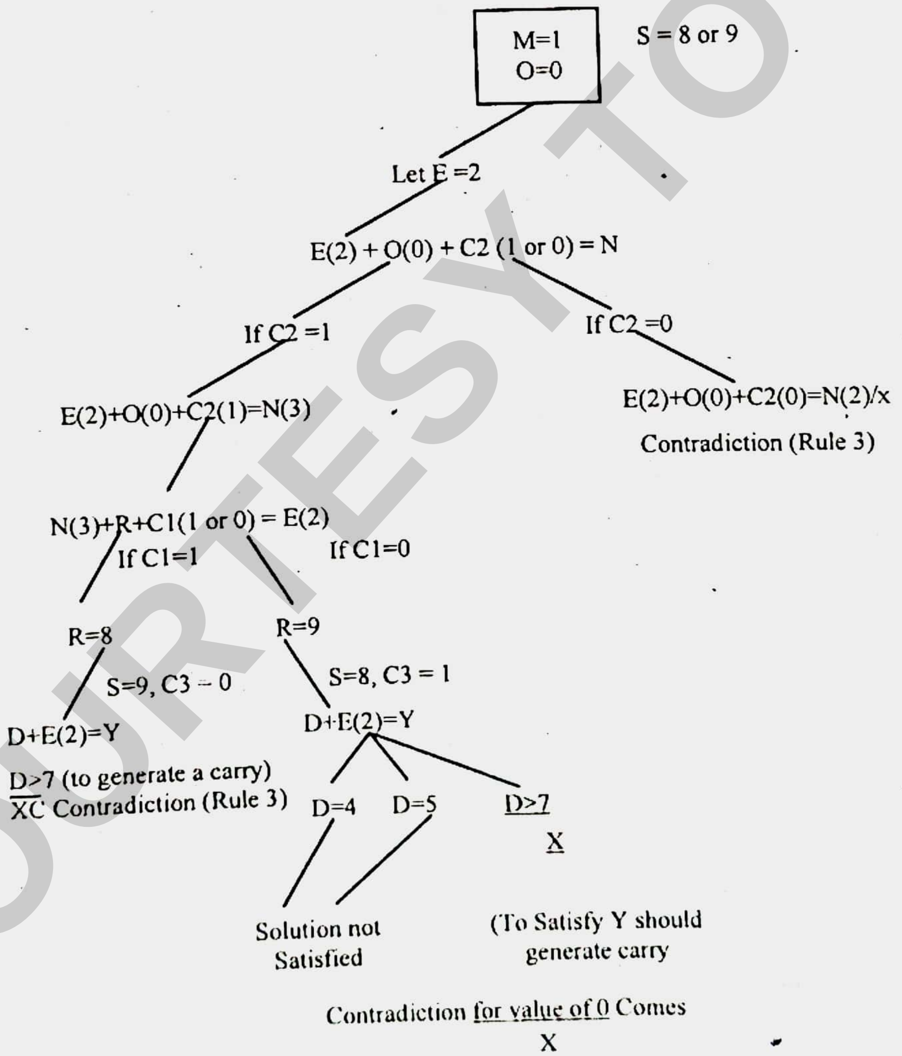
Goal State: the digits to the letters must be assigned in such a manner so that the sum is satisfied.

We are following the depth-first method to solve the problem.

1. initial guess m=1 because the sum of two single digits can generate at most a carry '1'.
2. When N=1 O=0 or 1 because the largest single digit number added to M=1 can generate the sum of either 0 or 1 depend on the carry received from the carry sum. By this we conclude that O=0 because m is already 1 hence we cannot assign same digit another letter (rule no.)
3. We have M=1 and O=0 to get O=0 we have S=8 or 9, again depending on the carry received from the earlier sum.

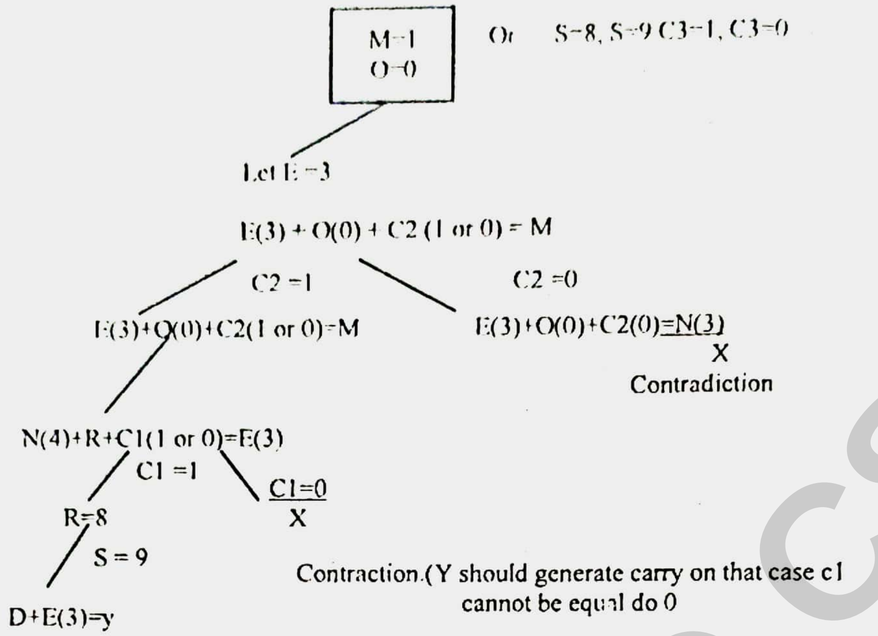
The same process can be repeated further. The problem must be composed into various constraints. And each constraint is to be satisfied by guessing the possible digits that the letters can be assumed that the initial guess has been already made. Rest of the process is being shown in the form of a tree in the figure below, using depth-first search for the clear understandability of the solution process.

Step-1



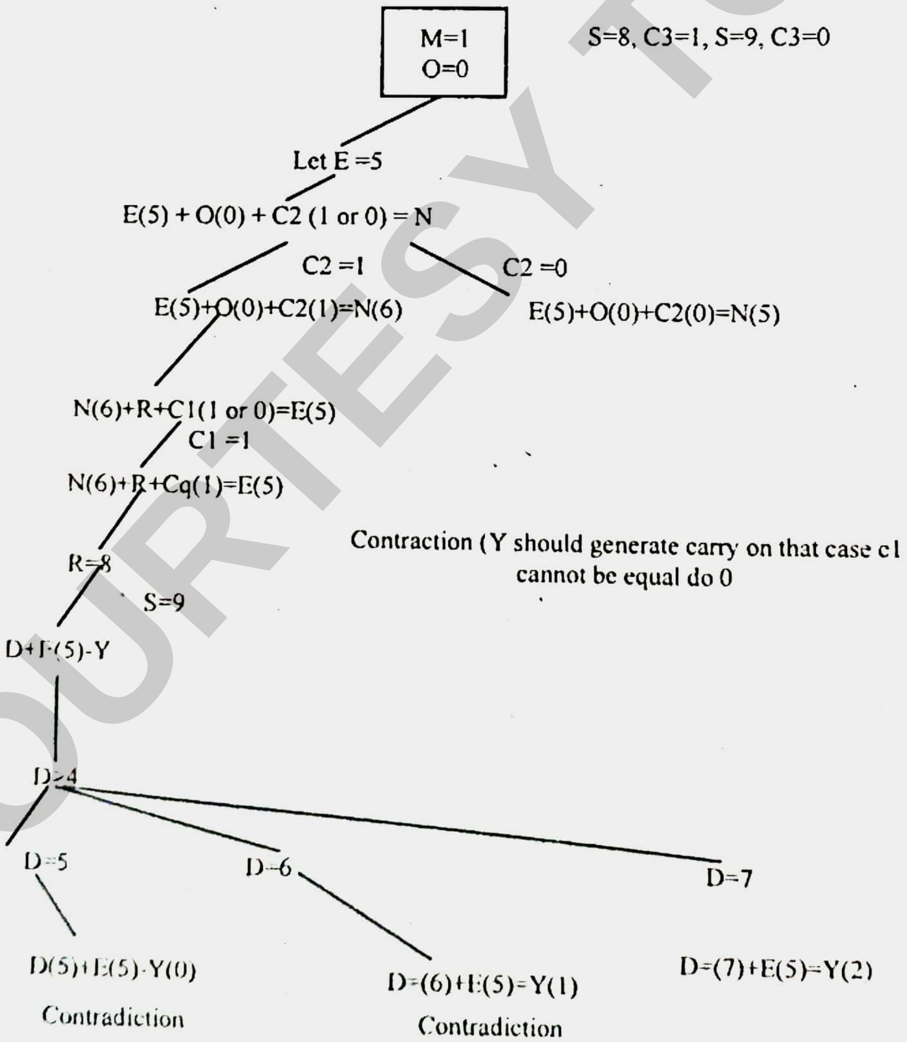
After Step 1 we derive are more conclusion that Y contradiction should generate a Carry. That is D+2>9

Step-2



After Step 2, we found that C1 cannot be Zero. Since Y has to generate a carry to satisfy goal state. From this step onwards, no need to branch for C1 = 0.

Step-3



At step (4) we have assigned a single digit to every letter in accordance with the constraints and production rules.
Now by backtracking, we find the different digits assigned to different letters and hence reach the solution state.

Solution State:
Y=2
D=7
S=9
R=8
N=6
E=5
O=0
M=1
C1=1
C2=0
C3=0

	C3(0)	C2(1)	C1(1)	
	S(9)	E(5)	N(6)	D(7)
+	M(1),	O(0)	R(8)	E(5)
M(1)	O(0)	N(6)	E(5)	Y(2)

20. Describe DFS. Does DFS always ensure completeness and optimality? Justify. Consider the following arrangement and solve the 4 puzzle problem applying DFS. **[WBUT 2019]**

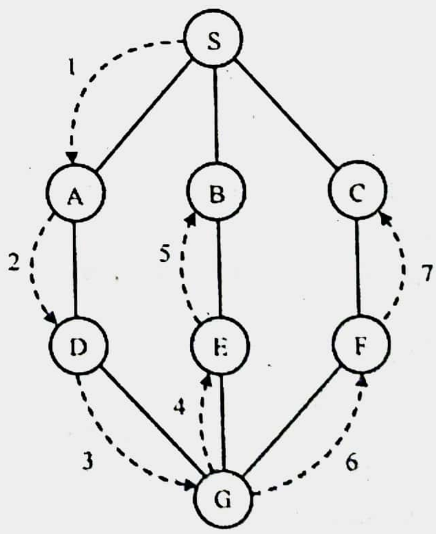
2	3
1	

Start State

1	2
3	

Goal State

Answer:
1st part:
Depth First Search (DFS) algorithm traverses a graph in a depth ward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration.



As in the example given above, DFS algorithm traverses from S to A to D to G to E to B first, then to F and lastly to C. It employs the following rules.

- **Rule 1** – Visit the adjacent unvisited vertex. Mark it as visited. Display it. Push it in a stack.
- **Rule 2** – If no adjacent vertex is found, pop up a vertex from the stack. (It will pop up all the vertices from the stack, which do not have adjacent vertices.)
- **Rule 3** – Repeat Rule 1 and Rule 2 until the stack is empty.

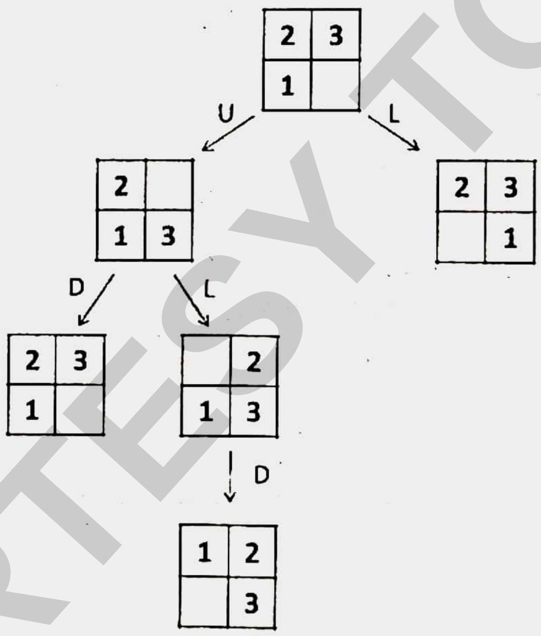
2nd part:

DFS is complete if the search tree is finite, meaning for a given finite search tree, DFS will come up with a solution if it exists.

DFS is not optimal, as the number of steps in reaching the solution, or the cost spent in reaching it is high.

3rd part:

In the problem, we order successors by applicable moves as follows: up, down, left, and right.



21. Describe local beam search. Describe minimax procedure. **[WBUT 2019]**

Answer:

1st part: ***Refer to Question No. 18(b) of Long Answer Type Questions.***

2nd part:

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.
Every board state has a value associated with it. In a given state if the maximizer has upper hand then, the score of the board will tend to be some positive value. If the minimizer has the upper hand in that board state then it will tend to be some negative value. The values of the board are calculated by some heuristics which are unique for every type of game.

22. Describe state space search. What are the advantages of BFS over DFS and vice-versa? Does uniform cost search ensure completeness and optimality? Justify. [WBUT 2019]

Answer:

1st part:

A **state** contains all the information necessary to predict the effects of an action and to determine if it is a goal state. State-space searching assumes that

- the agent has perfect knowledge of the state space and can observe what state it is in (i.e., there is full observability);
- the agent has a set of actions that have known deterministic effects.
- some states are goal states, the agent wants to reach one of these goal states, and the agent can recognize a goal state; and
- a **solution** is a sequence of actions that will get the agent from its current state to a goal state.

2nd part:

Advantages of BFS:

1. Solution will be found out by BFS If there are some solution.
2. BFS will never get trapped in blind alley, means unwanted nodes.
3. If there are more than one solution then it will find solution with minimal steps.

Disadvantages of BFS:

1. Memory Constraints As it stores all the nodes of present level to go for next level.
2. If solution is far away then it consumes time.

Advantages of DFS:

1. Memory requirement is Linear WRT Nodes.
2. Less time and space complexity rather than BFS.
3. Solution can be found out by without much more search.

Disadvantage of DFS:

1. Not Guaranted that it will give you solution.
2. Cut-off depth is smaller, so time complexity is more.
3. Determination of depth until the search has proceeds.

3rd part:

Uniform-cost search does not care about the number of steps a path has, but only the total path cost. It will get stuck in an infinite loop if there is a path with infinite sequence of zero-cost actions. Completeness is guaranteed only if the cost of every step is some positive number.

Uniform-cost search is optimal. This is because, at every step the path with the least cost is chosen, and paths never gets shorter as nodes are added, ensuring that the search expands nodes in the order of their optimal path cost.

23. Write short notes on the following:

a) Constraint satisfaction problems [WBUT 2009, 2011]

b) Simulated Annealing [WBUT 2013, 2015]

c) Heuristic Search [WBUT 2013]

d) A* search [WBUT 2018]

Answer:

a) Constraint satisfaction problems:

Refer to Question No. 13(a) (1st Part) of Long Answer Type Questions.

b) Simulated Annealing:

Refer to Question No. 10 of Short Answer Type Questions.

c) Heuristic Search: *Refer to Question No. 6 of Short Answer Type Questions.*

d) A* search: *Refer to Question No. 16 of Long Answer Type Questions.*

KNOWLEDGE & REASONING

Multiple Choice Type Questions

1. Skolem function is used in [WBUT 2006, 2007, 2012, 2013, 2014]
   a) unification algorithms
   b) natural deduction
   c) conversion to clause form
   d) none of these

Answer: (c)

2. Inheritable knowledge is best represented by [WBUT 2006, 2007, 2008, 2009, 2013, 2014, 2015, 2017, 2018]
   a) semantic net
   b) first order logic
   c) database
   d) none of these

Answer: (a)

3. Which of the following is a *declarative knowledge*? [WBUT 2006, 2007, 2009, 2010, 2013]
   a) A set of production rules
   b) Using LISP code to define a value
   c) Describing the objects using a set of attributes and associated values
   d) A knowledge about the order in which to pursue the subgoals

Answer: (b)

4. Which of the following statements is NOT true about *backward chaining*? [WBUT 2006, 2007, 2010]
   a) Backward chaining is a goal-directed reasoning process
   b) Backward chaining would be much better to use when trying to prove theorems
   c) For arriving at a new fact, backward chaining is more natural
   d) A medical diagnostic program is a query system that would probably use backward chaining

Answer: (c)

5. Resolution can be used for [WBUT 2006, 2009, 2010, 2013]
   a) question answering
   b) theorem proving
   c) both (a) & (b)
   d) none of these

Answer: (c)

6. Find out the most appropriate representation for "Alive means not dead" [WBUT 2007, 2013]
   a) $\forall x: \exists y:[alive(x,y) \rightarrow \neg dead(x,y)]$
   b) $\forall x: \forall y:[alive(x,y) \rightarrow \neg dead(x,y)]$
   c) $\forall x: \forall y:[alive(x,y) \rightarrow \neg dead(x,y)] \wedge [\neg dead(x,y) \rightarrow alive(x,y)]$
   d) $\exists x: \forall y:[alive(x,y) \rightarrow \neg dead(x,y)] \wedge [\neg dead(x,y) \rightarrow alive(x,y)]$

Answer: (c)

7. For a given proposition p, $p \vee \neg p$ = true is a [WBUT 2008, 2014, 2015]
   a) Tautology
   b) Contradiction
   c) Boolean expression
   d) Satisfiable formula

Answer: (a)

8. The first order logic is [WBUT 2008, 2014, 2015]
   a) both sound and complete
   b) sound but not complete
   c) complete but not sound
   d) neither sound nor complete

Answer: (a)

9. Horn clause is a clause with ....positive literals. [WBUT 2010, 2017, 2018]
   a) At most one
   b) At most two
   c) At least one
   d) At most four

Answer: (a)

10. Which of the following is tautology? [WBUT 2011]
   a) $(P \wedge Q) \wedge \sim Q$
   b) $(P \wedge Q) \Rightarrow P$
   c) $P \wedge \sim Q$
   d) none of thee

Answer: (b)

11. Frame is a collection of [WBUT 2011, 2017, 2018]
   a) Slots
   b) Filler
   c) Resolution
   d) Knowledge

Answer: (a)

12. Resolution-refutation is best associated with [WBUT 2011]
   a) sound rule of inference
   b) complete rule of inference
   c) both (a) and (b)
   d) none of these

Answer: (c)

13. NLP (with respect of AI) stands for [WBUT 2012]
   a) Natural Linear Processing
   b) Natural Language Processing
   c) Natural Linear Programming
   d) Natural Language Programming

Answer: (b)

14. Find out the most appropriate predicate representation for "every child like to play game". [WBUT 2012]
   a) $\exists x:[\text{CHILD}(x) \rightarrow [\forall y:[\text{GAME}(y) \wedge \text{LIKES}(x,y)]]]$
   b) $\forall x:[\text{CHILD}(x) \rightarrow [\exists y:[\text{GAME}(y) \wedge \text{LIKES}(x,y)]]]$
   c) $\forall x:[\text{CHILD}(x) \rightarrow [\forall y:[\text{GAME}(y) \wedge \text{LIKES}(x,y)]]]$
   d) $\exists x:[\text{CHILD}(x) \rightarrow [\exists y:[\text{GAME}(y) \wedge \text{LIKES}(x,y)]]]$

Answer: (b)

15. Knowledge consists of [WBUT 2012]
   a) concepts and procedures
   b) facts and rules
   c) both (a) and (b)
   d) none of these

Answer: (c)

16. "Mary is slightly ill". This statement can be completely expressed in [WBUT 2014, 2015]

a) FOPL
b) propositional logic
c) fuzzy logic
d) none of these

Answer: (c)

17. Uninformed search is also known as [WBUT 2014, 2015]

a) brute force search
b) hill climbing search
c) blind search
d) none of these

Answer: (c)

18. How do you represent "all dogs have tails"? [WBUT 2016]

a) $v_x : \mathrm{dog}(x) \rightarrow$ has tail (x)
b) $v_x : \mathrm{dog}(x) \rightarrow$ has tail (y)
c) $v_x : \mathrm{dog}(y) \rightarrow$ has tail (x)
d) $v_x : \mathrm{dog}(x) \rightarrow$ has $\rightarrow$ tail (x)

Answer: (a).

19. Which condition is used to cease the growth of forward chaining? [WBUT 2016]

a) Atomic sentences
b) Complex sentences
c) No further inference
d) All of the mentioned

Answer: (c)

20. Inheritable knowledge is best represented by [WBUT 2019]

a) semantic net
b) FOPL
c) database
d) none of these

Answer: (a)

21. PV ~ P is a [WBUT 2019]

a) tautology
b) contradiction
c) all false proposition
d) none of these

Answer: (a)

22. Resolution system can be used for [WBUT 2019]

a) question answering
b) theorem proving
c) both (a) and (b)
d) none of these

Answer: (c)

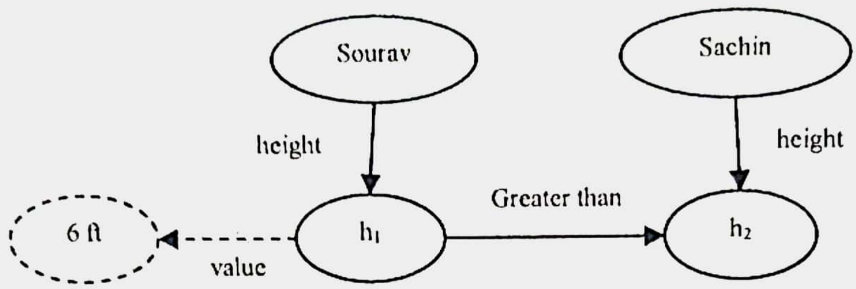
Short Answer Type Questions

1. With the help of semantic net, represent the fact that Sourav is 6 feet tall and that he is taller than Sachin. [WBUT 2007, 2015]

OR,

With the help of semantic net, prove that Sourav is 6 feet tall and he is taller than Sachin. [WBUT 2018]

Answer:



2. a) Define Horn Clause. **[WBUT 2007, 2015, 2016]**

OR,

What is 'Horn Clause'? **[WBUT 2018]**

b) Is $p \rightarrow q$ a Horn Clause? Justify your answer. **[WBUT 2007, 2015, 2016]**

c) What is meant by "tautology in propositional logic"? **[WBUT 2007, 2015]**

Answer:

a) A *Horn clause* is a clause with at most one positive literal.
Any Horn clause therefore belongs to one of four categories:

- A *rule*: 1 positive literal, at least 1 negative literal. A rule has the form "~P1 V ~P2 V ... V ~Pk V Q". This is logically equivalent to "[P1^P2^ ... ^Pk] => Q"; thus, an if-then implication with any number of conditions but one conclusion.
- A *fact* or *unit*: 1 positive literal, 0 negative literals. Examples: "man(socrates)", "parent(elizabeth,charles)", "ancestor(X,X)".
- A *negated goal*: 0 positive literals, at least 1 negative literal. In virtually all implementations of Horn clause logic, the negated goal is the negation of the statement to be proved; the knowledge base consists entirely of facts and goals. The statement to be proven, therefore, called the goal, is therefore a single unit or the conjuction of units; an *existentially* quantified variable in the goal turns into a *free* variable in the negated goal.
- The null clause: 0 positive and 0 negative literals. Appears only as the end of a resolution proof.

b) The above can be re-written in the form ~pVq. Thus it has 1 negative literal and one positive literal which satisfies the condition of a Horn Clause of having at most 1 positive literal.

c) Some propositions are always true regardless of the truth value of its component propositions. For example (P V¬P) is always true regardless of the value of the proposition. A proposition that is always true called a **tautology**.

3. Find all the interpretations of $P \rightarrow Q$, where *P* and *Q* are two propositions and → is an implication sign. **[WBUT 2008, 2012]**

Answer:

P → Q is only FALSE when the Premise(P) is TRUE AND Consequence(Q) is FALSE.

P → Q is always TRUE when the Premise(P) is FALSE OR the Consequence(Q) is TRUE.

P → Q can be formulated in the following ways:

1. If P then Q
2. P only if Q
3. P is a sufficient condition for Q
4. Q is a necessary condition for P
5. Q if P
6. Q follows P
7. Q provided P
8. Q is a logical consequence of P
9. Q whenever P

However, in some cases the Premise(P) and the Consequence(Q) need not be related. The interpretation of the truth table depends on the formulation P and Q.

4. How procedural knowledge differs from declarative knowledge? [WBUT 2008]

OR,

Distinguish between Declarative and Procedural Knowledge. [WBUT 2016]

Answer:

Declarative knowledge is knowledge *about* something. Declarative knowledge enables a student to describe a rule of grammar and apply it in pattern practice drills.

Procedural knowledge is knowledge of how to do something. Procedural knowledge enables a student to apply a rule of grammar in communication.

For example, declarative knowledge is what you have when you read and understand the instructions for programming the DVD player. Procedural knowledge is what you demonstrate when you program the DVD player.

Procedural knowledge does not translate automatically into declarative knowledge; many native speakers can use their language clearly and correctly without being able to state the rules of its grammar. Likewise, declarative knowledge does not translate automatically into procedural knowledge; students may be able to state a grammar rule, but consistently fail to apply the rule when speaking or writing.

5. Given the following text 'Everyone who enters in a theatre has to buy a ticket. Person who doesn't have money can't buy a ticket. Vinod enter a theater'. Prove by resolution that "Vinod buys a ticket". [WBUT 2008, 2015]

Answer:

Let us define the following predicates

b(x) : x buys ticket.

m(x) x has money.

e(x,t): x enters theatre.

So we have the following:

1. $\forall x\ [e(x,t) \rightarrow b(x)]$

2. $\forall x[\neg m(x) \rightarrow \neg b(x)]$
3. e(vinod,t)
4. b(vinod) from 1 and 3

Therefore Vinod buys a ticket.

6. A problem – solving search can proceed in either the forward or the backward direction. What factors determine the choice of direction for a particular problem? Justify your answer. **[WBUT 2009, 2010, 2016]**

Answer:

For logical reasoning tasks, one can either search forward from the initial state (forward) or search backward from the goal state (backward).

For example, suppose that the goal is to conclude the color of a pet named Fritz, given that he croaks and eats flies, and that the rule base contains the following four rules:

1. **If** X croaks and eats flies - **Then** X is a frog
2. **If** X chirps and sings - **Then** X is a canary
3. **If** X is a frog - **Then** X is green
4. **If** X is a canary - **Then** X is yellow

In **forwards reasoning** we use a 'forwards chaining' search process to recursively generate conclusions. Taking whatever facts are initially established, we check to see which rules may then be used, i.e., which rules have all their conditions satisfied by the facts. We then add the relevant conclusion(s) and repeat the operation, continuing on until no new conclusions can be produced.

In this example, the rule base would be searched and the first rule would be selected, because its antecedent (**If** Fritz croaks and eats flies) matches our data. Now the consequents (**Then** X is a frog) is added to the data. The rule base is again searched and this time the third rule is selected, because its antecedent (**If** Fritz is a frog) matches our data that was just confirmed. Now the new consequent (**Then** Fritz is green) is added to our data. Nothing more can be inferred from this information, but we have now accomplished our goal of determining the color of Fritz.

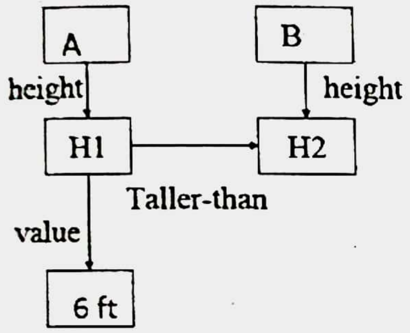
Backwards reasoning is the process of searching for a solution path connecting some final conclusion with one or more initial facts. States are combinations of required conclusions and the transitions are defined by the rules. Each rule is a method for generating further `required conclusions' from existing required conclusions.

In this case, the rule base in the example would be searched and the third and fourth rules would be selected, because their consequents (**Then** Fritz is green, **Then** Fritz is yellow) match the goal (to determine Fritz's color). It is not yet known that Fritz is a frog, so both the antecedents (**If** Fritz is a frog, **If** Fritz is a canary) are added to the goal list. The rule base is again searched and this time the first two rules are selected, because their consequents (**Then** X is a frog, **Then** X is a canary) match the new goals that were just added to the list. The antecedent (**If** Fritz croaks and eats flies) is known to be true and therefore it can be concluded that Fritz is a frog, and not a canary. The goal of determining Fritz's color is now achieved (Fritz is green if he is a frog, and yellow if he is a canary, but he is a frog since he croaks and eats flies; therefore, Fritz is green).

If there is a single goal (conclusion), backward chaining will normally be more efficient, as there is no wasteful generation of irrelevant conclusions. But if there are many different ways of demonstrating any particular fact, backwards chaining may be wasteful. Forward chaining is likely to be more efficient if there are many conclusions to be drawn or where we have a small set of initial facts. It may also be preferable if conclusions tend to have many rules. Backward chaining is likely to be more efficient where there is a single conclusion to be drawn or where the initial set of facts is large.

7. With the help of semantic net, represent the fact than Mr. A is 6 feet tall and he is taller than Mr. B. **[WBUT 2011]**

Answer:



8. Explain the following terms with examples:
(i) Tautology **[WBUT 2012, 2016]**
(ii) Contradiction **[WBUT 2012]**

Answer:

i) A tautology is a proposition that is always true.
   Example: $p \vee \neg p$
ii) A contradiction is a proposition that is always false.
   Example: $p \wedge \neg p$

9. With an example show a simplified Frame system. **[WBUT 2013]**

Answer:

The contents of the frame are certain slots, which have values. All frames of a given situation constitute the system. A frame can be defined as a data structure that has slots for various objects and a collection of frames consists of exceptions for a given situation. A frame structure provides facilities for describing objects, facts about situations, procedures on what to do when a situation is encountered. Because of this facilities a frames are used to represent the two types of knowledge, viz., declarative/factual and procedural.

In the higher levels of the frame hierarchy, typical knowledge about the class is stored and the value in a slot may be a range or a condition. In the lower levels, the value in a slot may be a specific value, to override the value which would otherwise be inherited from a higher frame. An instance of an object is joined to its class by an 'instance of' relationship. A class is joined to its super class by a 'subclass of' relationship.

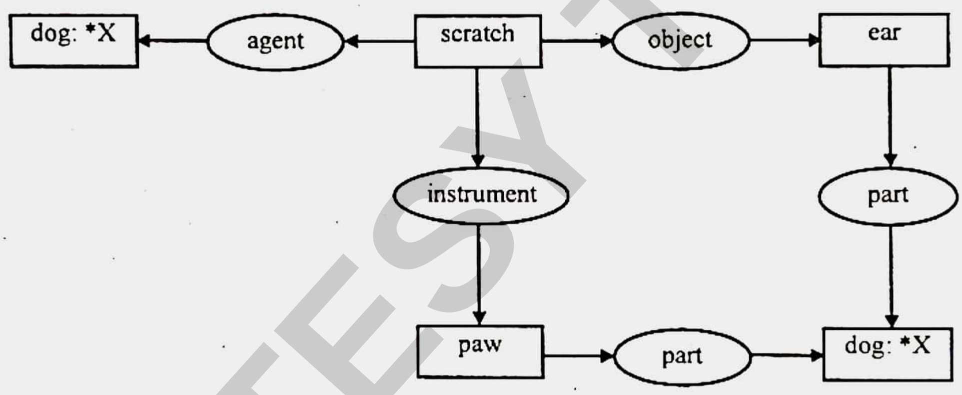
Frames may contain both procedural and declarative knowledge. Slot values normally amount to declarative knowledge, but a daemon is in effect a small program. So a slot with a daemon in it amounts to procedural knowledge.
A frames system may allow multiple inheritance but, if it does so, it must make provision for cases when inherited values conflict. A typical book frame is shown in the figure below which consists of frame name, attributes (slots) and values (fillers: list of values, range, string, or procedures etc.)

Book Frame
Slot → Filler
Title → Romeo and Juliet Author → William Shakespeare

Fig: A book frame

10. Draw Conceptual graph for the following sentence:
"The dog scratches its ear with its paw". [WBUT 2014]

Answer:



11. Convert the following sentences into first order predicate logic: [WBUT 2015]
(i) Everyone likes Ram
(ii) No one is perfect
(iii) Someone ate everything
iv) All basketball players are tall.

Answer:

i) ∀x likes (x, Ram)
ii) ∀(x)~ P(x) where P(x) mans "x is perfect"
iii) ∃x ∀y·ate(x, y)
iv) ∀x, basketball_player(x) → tall(x)

12. What do you mean by contradiction and contingency? [WBUT 2016]

Answer:

Contradiction: *Refer to Question No. 8 (ii) of Short Answer Type Questions.*

A **contingency** is a proposition that is neither a tautology nor a contradiction.
Example: p V q → ¬r

13. Write the predicate logic representations for the following sentences:
(i) If it is bird, it can fly **[WBUT 2016]**
(ii) Every father is parent

Answer:

i) $\forall x\left[B(x)\rightarrow F(x)\right]$

ii) $(x)\left((\exists y)Fxy\rightarrow(\exists y)Pxy\right)$

14. What is tautology? Prove that $(((P\rightarrow Q)\rightarrow P)\rightarrow P)$ is a tautology. What are Modus Ponens and Modus Tollens? **[WBUT 2017, 2018]**

Answer:

1st Part:

Refer to Question No. 2(c) of Short Answer Type Questions.

2nd Part:

$(((P\rightarrow Q)\rightarrow P)\rightarrow P)$

⇔ ((((~P V Q)->P)->P)
⇔ (((~(~P V Q) V P)->P)
⇔ (((P ^ ~Q) V P)->P)
⇔ ((P^ ~Q)->P)
⇔ (~(P^~Q) V P)
⇔ (~P VQ V P)
⇔ True (**tautology** proved)

3rd Part:

Modus ponens means "the way that affirms by affirming") and modus tollens" means the way that denies by denying". Here is how they are constructed:
Modus Ponens: "If A is true, then B is true. A is true. Therefore, B is true."
Modus Tollens: "If A is true, then B is true. B is not true. Therefore, A is not true."

15. What do you mean by natural language processing (NLP)? What is parsing in NLP? What are the types of parsing? Draw the parsed tree of the sentence "The white dog crossed the road". **[WBUT 2017]**

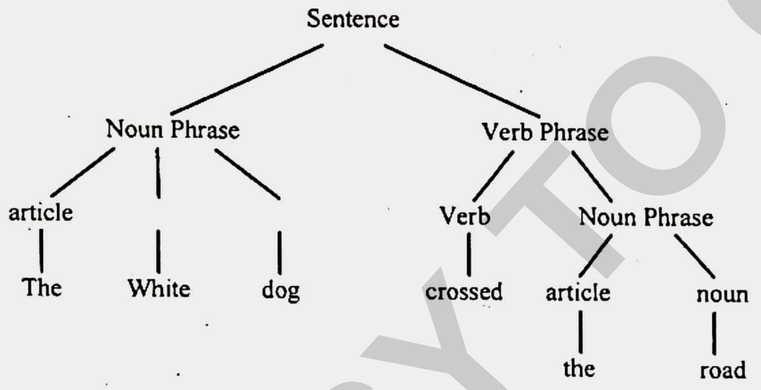
Answer:

Natural-language processing (NLP) is an area of artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

Breaking a sentence as per grammar is Parsing. A Sentence is broken into Noun Phrase and Verb Phrase. Similarly, a Noun Phrase could be again Article + Noun. Such breaking of a sentence into grammar is called NLP parsing.
When the parser starts constructing the parse tree from the start symbol and then tries to transform the start symbol to the input, it is called top-down parsing.

- ⇨ Recursive descent parsing: It is a common form of top-down parsing. It is called recursive as it uses recursive procedures to process the input. Recursive descent parsing suffers from backtracking.
- ⇨ Backtracking: It means, if one derivation of a production fails, the syntax analyzer restarts the process using different rules of same production. This technique may process the input string more than once to determine the right production.

Parsed tree of the sentence "The white dog crossed the road" is as given below:



16. What is clause? **[WBUT 2017]**

Answer:
A clause is a disjunction of literals (positive or negative) with no literal repeated

17. a) Prove that PV ~ P is a Tautology.
b) What is contradiction? **[WBUT 2017]**

Answer:
1st Part:
The truth table is as given below:

P	¬ P	P V ¬ P
T	F	T
F	T	T

As we can see, PV ~ P is always true and hence is a tautology.

2nd Part:
Refer to Question No. 8(ii) of Short Answer Type Questions.

18. Briefly describe the issues in knowledge representation. **[WBUT 2019]**

Answer:

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

Important Attributes

-- Are there any attributes that occur in many different types of problem?

There are two instance and isa and each is important because each supports property inheritance.

Relationships

-- What about the relationship between the attributes of an object, such as, inverses, existence, techniques for reasoning about values and single valued attributes.

Granularity

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

Long Answer Type Questions

1. What is semantic net? **[WBUT 2006, 2007, 2011, 2015]**

Represent the fact that "Sourav is taller than Sachin" with the help of semantic net. **[WBUT 2006]**

OR,

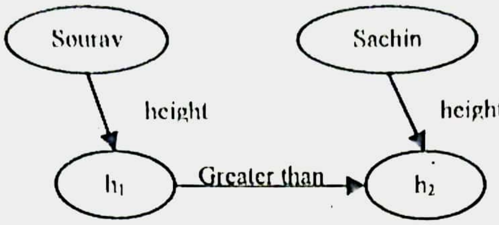
Explain semantic network with proper example. **[WBUT 2016]**

Answer:

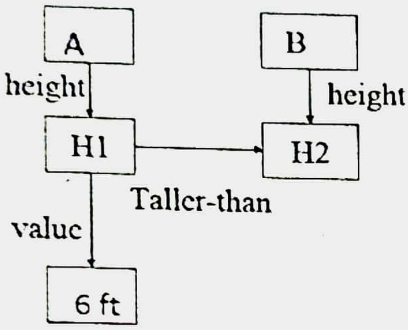
1st Part:

A semantic network is a directed graph consisting of nodes (also termed points or vertices) which represent concepts and edges (also termed lines or arcs) which represent semantic relations between the concepts. They are knowledge representation schemes involving nodes and links (arcs or arrows) between nodes. The nodes represent objects or concepts and the links represent relations between nodes. The links are directed and labeled; thus, a semantic network is a directed graph. In print, the nodes are usually represented by circles or boxes and the links are drawn as arrows between the circles. This represents the simplest form of a semantic network, a collection of undifferentiated objects and arrows. The structure of the network defines its meaning. The meanings are merely which node has a pointer to which other node. The network defines a set of binary relations on a set of nodes.

2nd Part:



3rd Part:



2. Convert the following into clausal forms: **[WBUT 2006, 2014, 2015, 2019]**

$\forall x\{P(x) \rightarrow \{(\forall y)P(y) \rightarrow P(f(x, y))]\wedge^{\sim}\{(\forall y)[Q(x, y) \rightarrow P(Y)]\}\}\}.$

Answer:

The question is wrong, because the left & right parentheses don't match.
We assume the following is one of the possible corrected version.

$\forall x\{P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(f(x,y))) \wedge \neg\{(\forall y)[Q(x,y) \rightarrow P(y)]\})).$

Conversion steps

1. $\forall x\{P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(f(x,y))) \wedge \neg\{(\forall y)[Q(x,y) \rightarrow P(y)]\})).$
2. $\forall x\{\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge \neg((\forall y)[\neg Q(x,y) \vee P(y)]))).$
3. $\forall x\{\neg P(x) \vee \{(\forall y)(\neg P(y) \vee P(f(x,y))) \wedge ((\exists y)[Q(x,y) \wedge \neg P(y)]))).$
4. $\forall x\{\neg P(x) \vee \{(\forall y)(\neg P(y) \vee P(f(x,y))) \wedge ((\exists w)[Q(x,w) \wedge \neg P(w)]))).$
5. $\forall x(\forall y)(\exists w)(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y))) \wedge [Q(x,w) \wedge \neg P(w)]))).$
6. $\forall x(\forall y)(\exists w)$

   $((\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge$

   $(\neg P(x) \wedge Q(x,w)) \wedge$

   $(\neg P(x) \wedge \neg P(w)).$
7. $\forall x \forall y$

   $((\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge$

   $(\neg P(x) \wedge Q(x,f_w(x))) \wedge$

   $(\neg P(x) \wedge \neg P(f_w(x))).$ [f_w is the skolem functor]

Final set of clauses:

$\{(\neg P(x) \vee \neg P(y) \vee P(f(x,y))), (\neg P(x) \wedge Q(x,f_w(x))), (\neg P(x) \wedge \neg P(f_w(x))\}.$

3. Write down the difference between the following:
i) Associative network and conceptual graph. **[WBUT 2007, 2008]**

Answer:

Associative network is a means of representing relational knowledge as a labeled directed graph. Each vertex of the graph represents a concept and each label represents a relation between concepts. Access and updating procedures traverse and manipulate the graph.

Conceptual graphs are a system of logic based on the existential graphs of Charles Sanders Peirce and the semantic networks of artificial intelligence. They express meaning in a form that is logically precise, humanly readable, and computationally tractable. With a direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages. Square vertices are used to represent concepts and oval vertices represent relations. Edges are used to link concepts with relations.

ii) Frame and script. **[WBUT 2007, 2009]**

Answer:

A frame is a data structure that includes all the knowledge about a particular object. In a frame, knowledge is organized in a special hierarchical structure that permits a diagnosis of knowledge independence.

A script is a knowledge representation scheme similar a frame, but instead of describing an object, the script describes a sequence of events. Like the frame, the script portrays a stereotype. Unlike the frame, it is usually presented in a context. To describe a sequence of events, the script uses a series of slots containing information about the people, objects and actions they are involved in the events.

iii) Forward and backward reasoning. **[WBUT 2007, 2008, 2015]**

Answer:

Refer to Question No. 6 of Short Answer Type Questions.

4. Give the predicate logic expression of the following sentences and then represent them in semantic net.
i) Every man has beaten the thief **[WBUT 2008, 2016]**

Answer:

$\forall x \exists y(\text{man}(x) \rightarrow \text{thief}(y) \wedge \text{beaten}(x,y))$.

ii) All persons in the party loved every child. **[WBUT 2008]**

OR,

Every person in the party loves every child. **[WBUT 2016]**

Answer:

$\forall x \forall y\ \exists z\ (\text{person}(x) \wedge \text{party}(z) \wedge \text{child}(y) \rightarrow \text{loved}(x,y))$.

Drawing of semantic networks is left as an exercise to the students.

5. Represent the following using predicate logic and draw the conclusions as required. **[WBUT 2009, 2017]**

- **i)** ***X* is an Indian**
- **ii)** ***Y* is an Indian**
- **iii)** ***X* is a leader**
- **iv)** **Every Indian is a man**
- **v)** **Everyone is loyal to someone**
- **vi)** **Every man is either loyal to a leader or hate a leader**
- **vii)** **Man tries to assassinate a leader if he is not loyal to him**
- **viii)** ***Y* assassinated *X***

Conclude that *Y* hated *X*.

Answer:

i. Indian(X)

ii. Indian(Y)

iii. Leader(X)

iv. $\forall$x Indian(x))=> Man(x)

v. $\forall$x $\exists$y loyal_to(x,y)

vi. $\forall$x Man(x) $\rightarrow$ (loyal_to(x,Leader) $\vee$ hates(x,Leader)) $\wedge$(~ (loyal_to(x,Leader) $\wedge$ hates(x,Leader))

vii. $\forall$x$\forall$x Man(x) $\wedge$ Leader(y) $\wedge$ try_assassinate(x,y) $\rightarrow$ ~loyal_to(x,y)

viii. Assassinated (Y,X)

To prove Hates(Y,X):
From 1), ii), iii) and iv) we conclude Man(Y).
From vii) we substitute X and Y for y and x to conclude ~loyal_to(Y,X) assuming try_assassinate(Y,X) (from viii).
From vi) we know Man(Y) is true if and only if Hates(Y,X) is true. Hence proved.

6. a) Represent the following sentences by default logic. Also mention the sets *D* and *W*.

- **i)** **Typically molluscs are shell-bearers.**
- **ii)** **Cephalopods are molluscs**
- **iii)** **Cephalopods are not shell-bearers.** **[WBUT 2010]**

Answer:

mollusc(X) : shellBearer(X) / shellBearer(X)

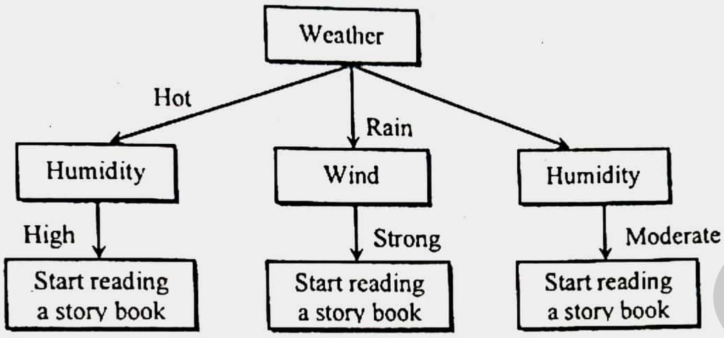
cephalopod(X) → mollusc(X) $\wedge$ ¬ shellBearer(X)

$$W = \{\text{Cephaload}(X) \rightarrow \text{Molluscs}(X)\}$$

$$D = \left\{\frac{\text{Molluscs}(X):\text{Shell-bearer}(X)}{\text{Shell-bearer}(X)}\right\}$$

b) Draw a decision tree corresponding to the following expression: [WBUT 2010]
If (Weather = Hot ^ Humidity = High) ∨
(Weather = Cool ^ Humidity = Moderate) ∨
(Weather = Rainy ^ Wind = Strong).
Then start reading a story book.

Answer:



7. Consider the following sentences: **[WBUT 2011]**
John likes all kinds of food.
Apples are food.
Anything anyone eats and is not killed by is food.
Mary eats peanuts and is still alive. Sam eats everything.
Mary eats.
Use resolution to answer "What food does Sam eat?"

Answer:

Writing in FOL:

i) John likes all kinds of foods.
$\forall_x$ .food(x) → likes(John, x)

ii) Apples are food.
food(Apple)

iii) Anything anyone eats and and is not killed is food.
$\forall_{x,y}$:(eats(x, y) ^ alive(x)) → food(y)

iv) Mary eats peanuts and is still alive.
eats(Mary, Peanut) ^ alive(Mary)

v) Sam eats everything that Mary eats.
$\forall_x$ eats(Mary, x) → eats(Sam, x)

Converting the FOL sentences into CNF:

(a) ¬food(x) V likes(John, x)
(b) food(Apple)
(c) ¬eats(x; y) V ¬alive(x) V food(y)
(d) eats(Mary, Peanut)
(e) alive(Mary)
(f) ¬eats(Mary, x) V eats(Sam, x)

To answer "what food does Sam eat?":
g) ¬eats(Sam, x) – negating the query
h) ¬eats(Mary, x) – from f and g
i) Nil – from (d) and (h) unifying peanuts/x
Thus Sam eats peanuts.

8. a) What is Turing test? **[WBUT 2012]**
b) Art is the father of John. Bob is the father of Kim. Fathers are parents. Prove that Art is the parent of John. **[WBUT 2012, 2014]**
c) Convert the following sentences into first order predicate logic: **[WBUT 2012]**
   i) Everyone loves Ram.
   ii) Not everyone loves Ravana.
   iii) Not everyone came for all meetings.
   iv) Some people did not come for all meetings.
   v) Only one person spoke at the meeting.
d) With the help of semantic net, represent the following facts: **[WBUT 2012]**
   i) Tweety is a bird.
   ii) Tweety has two wings.
   iii) If a bird has wings and no broken wing, it can fly.
e) What is the difference between semantic net and frame? **[WBUT 2012]**

Answer:

a) The Turing test is a test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of an actual human. In the original illustrative example, a human judge engages in a natural language conversation with a human and a machine designed to generate performance indistinguishable from that of a human being. All participants are separated from one another. If the judge cannot reliably tell the machine from the human, the machine is said to have passed the test. The test does not check the ability to give the correct answer; it checks how closely the answer resembles typical human answers. The conversation is limited to a text-only channel such as a computer keyboard and screen so that the result is not dependent on the machine's ability to render words into audio.

b) father(art, john)
2. father(bob,kim)
3. (¬father(Y,Z), parent(Y,Z))
   i.e. all fathers are parents
4. (¬ parent(X,john), answer(X)) i.e. the query is: who is parent of john?
   Here is a resolution proof:
5. R[4,3b]{Y=X,Z=john}
   (¬father(X,john), answer(X))
6. R[5,1]{X=art} answer(art) - so art is parent of john

c) i) $\forall x$ love (x, Ram)

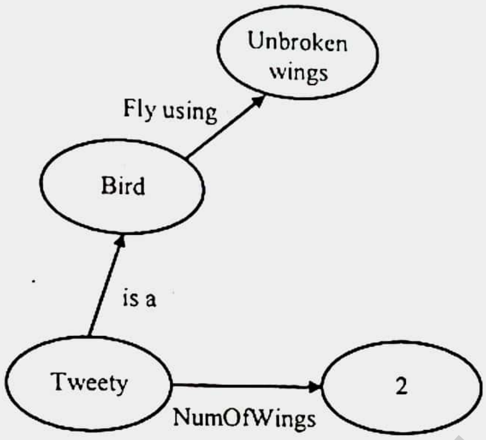
ii) $\exists x$ ¬love(x,Ravana)

iii) $\neg\forall x \forall y$ attend_meeting(x, y)

iv) $\exists x \forall y$ attend_meeting(x, y)

v) $\exists p1$ spokeMeeting(p1) $\wedge$ ($\forall p2$ spokeMeeting (p2) $\Rightarrow$ p2=p1)

d)



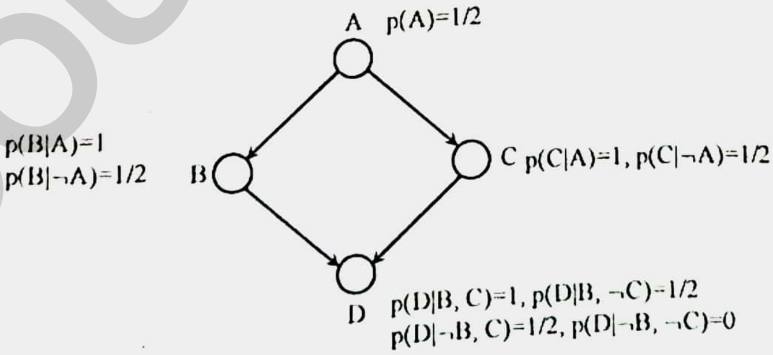
e) The idea of semantic networks started out as a natural way to represent labeled connections between entities. But, as the representations are expected to support increasingly large ranges of problem solving tasks, the representation schemes necessarily become increasingly complex. In particular, it becomes necessary to assign more structure to nodes, as well as to links. For example, in many cases we need node labels that can be computed, rather than being fixed in advance. It is natural to use database ideas to keep track of everything, and the nodes and their relations begin to look more like frames.

In the literature, the distinction between frames and semantic networks is actually rather blurred. However, the more structure a system has, the more likely it is to be termed a frame system rather than a semantic network.

9. a) What are the different approaches to knowledge representation?

b) State the difference between Inheritable knowledge and Inferential knowledge.

c) An admission committee for a college is trying to determine the probability that an admitted candidate is really qualified. The relevant probabilities are given in the following Bayes network. Find p(A|D).



[WBUT 2013, 2015]

Answer:

a) There are several approaches to knowledge representation in AI, which can be seen as subcategories of the symbol-representation approach. They all share the conditions the knowledge is explicated by the use of some kind of symbolic language and is installed in the system "manually", piece by piece.

The four most important kinds of symbolic systems may be
- a) logic based systems
- b) semantic networks and
- c) frame-based systems.

b) 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.

c) A = applicant is qualified
B = applicant has high grade point average
C = applicant has excellent recommendations
D = applicant is admitted
$P(A=1/D=1) = P(A=1,D=1)/P(D=1)$.
Now, $P(D) = \sum_{A,B,C} P(A)P(B|A)P(C|A)P(D=1|B,C)$
$P(A/D) = P(A=1)\sum_{B,C} P(B=1|A=1)P(B=1|A=1)P(D=1|B=1,C=1)/P(D=1)$.
Calculating for the $P(A/D)$ we get $P(A/D)=2/3$.

10. a) State the basic principle of Resolution method for both Proposition & Predicates.
b) Consider the following FOPL sentences from the domain of Monkey-Banana problem:
(i) in_room (bananas), (ii) in_room (chair), (iii) in_room (monkey), (iv) tall (chair), (v) ¬ close (bananas, floor), (vi) can_move (monkey, chair, bananas), (vii) can_climb (monkey, chair), (viii) close (X, Y) → can_reach (X, Y), (ix) (get_on (X, Y) ∧ under (Y, bananas) ∧ tall (Y)) → close (X, bananas), (x) (in_room) X ∧ in_room (Y) ∧ in_room (Z) ∧ can_move (X, Y, Z)) → close (Z, floor) ∨ under (Y, Z), (xi) can_climb (X, Y) → get_on (X, Y)
Now from those given sentences prove that 'monkey can reach to the bananas' using Resolution method. [WBUT 2013]

Answer:

a) Resolution for propositional logic

Consider a set of formulas S in clausal form. Suppose we are given a formula G, also in clausal form, for which we have to prove that it can be derived from S by applying resolution. Proving $S \vdash G$ is equivalent to proving that the set of clauses W, consisting of the clauses in S supplemented with the negation of the formula G, that is $W = S \cup \{\neg G\}$, is unsatisfiable. Resolution on W now proceeds as follows: first, it is checked whether or not W contains the empty clause ⊢; if this is the case, then W is unsatisfiable, and G is a logical consequence of S. If the empty clause ⊢ is not in W, then the resolution rule is applied on a suitable pair of clauses from W, yielding a new clause. Every clause derived this way is added to W, resulting in a new set of clauses on which the same resolution procedure is applied. The entire procedure is repeated until some generated set of clauses has been shown to contain the empty clause ⊢, indicating unsatisfiability of W, or until all possible new clauses have been derived.

Resolution for predicate logic

Assume that a set of given statements F and a statement to be proved P:

Algorithm

1. Convert all the statements of F to clause form
2. Negate P and convert the result to clause form. Add it to the set of clauses obtained in step 1.
3. Repeat until either a contradiction is found or no progress can be made or a predetermined amount of effort has been expended:

a) Select two clauses. Call these the parent clauses.

b) Resolve them together. The resulting clause, called the resolvent, will be the disjunction of all of the literals of both the parent clauses with appropriate substitutions performed and with the following exception: If there is one pair of literals T1 and T2 such that one of the parent clauses contains T1 and the other contains T2 and if T1 and T2 are unifiable, then neither T1 nor T2 should appear in the resolvent. We call T1 and T2 complementary literals. Use the substitution produced by the unification to create the resolvent. If there is one pair of complementary literals, only one such pair should be omitted from the resolvent.

c) If the resolvent is the empty clause, then a contradiction has been found. If it is not then add it to the set of clauses available to the procedure.

b) Clausal form of knowledge base

In_room(monkey)
In_room(bananas)
In_room(chair)
Tall(chair)
Can_move(monkey,chair,bananas)
Can_climb(monkey,chair)
~Close(banans,floor)
~Can_climb(x,y)V get_on(x,y)
~ close(x,y) V can_reach(x,y)
~get_on(x,y) V ~under(y,bananas) V~tall(y) V close(x,bananas)
~in_room(x) V ~in_room(y) V ~in_room(z) V ~can_move(x,y,z) V close(y,floor) V under(y,z)
~can_reach(monkey,bananas)

~can_move(monkey,chair,bananas) V close(bananas,floor) V under(chair,bananas)
Close(bananas,floor) V under(chair,bananas)
Under(chair,bananas)
~get_on(x,chair) V ~tall(chair) V close(x,bananas)
~get_on(x,chair) V close(x,bananas)
Get_on(monkey,chair)
Close(monkey,bananas)
~close(monkey,y) V can_reach(monkey,y)
Reach(monkey,bananas)

11. Convert the following sentences into first order predicate logic: [WBUT 2014]
(i) Everyone loves Ram.
(ii) Not everyone likes Ravana.
(iii) No frogs are birds.
(iv) Some vehicles are not motorcycles.
(v) No person buys an expensive policy.

Answer:
i) & ii) ***Refer to Question No. 8(c)(i) & (ii) of Long Answer Type Questions.***
iii) $(\forall \text{ frogs } x)(\sim Bx)$
iv) $(\exists x)(Vx, Mx)$
v) $\forall$ x, y, z Person(x) $\wedge$ Policy(y) $\wedge$ Expensive(y) $\Rightarrow$ $\neg$Buys(x, y, z).

12. By using the predicate logic principles prove that "Marcus hated Caeser". Given below are the list of statements. [WBUT 2016]
Marcus was a man.
Marcus was Pompeian.
All Pompeians were Romans.
Caeser was a ruler.
All Romans were either loyal to Caeser or hated him.
Everyone is loyal to someone.
People only try to assassinate rulers they are not loyal to.
Marcus tried to assassinate Caeser.
Explain the answer thoroughly.

Answer:
Using predicate logic the sentences in FOPL are as follows:
[1] man(Marcus)
[2] Pompeian(Marcus)
[3] $\forall$ X Pompeian(X) => Roman(X)
[4] ruler(Caesar)
[5] $\forall$ X Roman(X) => loyalto(X, Caesar) V hate(X,Caesar)
[6] $\forall$ X $\exists$ Y loyalto(X, Y)
[7] $\forall$ X $\forall$ Y man(X) $\wedge$ ruler(Y) $\wedge$ try_assassinate(X, Y)=> ~loyalto(X, Y)
[8] try_assassinate(Marcus, Caesar)

Converted to CNF

[1] man(Marcus)
[2] Pompeian(Marcus)
[3] ~Pompeian(X1) V Roman(X1)
[4] ruler(Caesar)
[5] ~Roman(X2) V loyalto(X2, Caesar) V hate(X2, Caesar)
[6] loyalto(X3, f1(X3))
[7] ~man(X4) V ~ruler(Y1) V~ try_assassinate(X4,Y1) V~loyalto(X4, Y1)

Resolution

- [G] hate(Marcus, Caesar)?
- Assume negation: ~hate(Marcus, Caesar) + [5]
- ~Roman(Marcus) V loyalto(Marcus, Caesar) + [3]
- ~Pompeian(Marcus) V loyalto(Marcus, Caesar) +[2]
- loyalto(Marcus, Caesar) + [7]
- ~man(Marcus) V ~ruler(Caesar) V~try_assassinate(Marcus, Caesar) + [1]
- ~ruler(Caesar) V ~try_assassinate(Marcus, Caesar) + [4]
- ~try_assassinate(Marcus, Caesar) + [8]
- [] #contradiction!
- Therefore, negation is false, and so G is true.

tryassassinate(Marcus, Caesar).

13. Suppose you have the following search space:

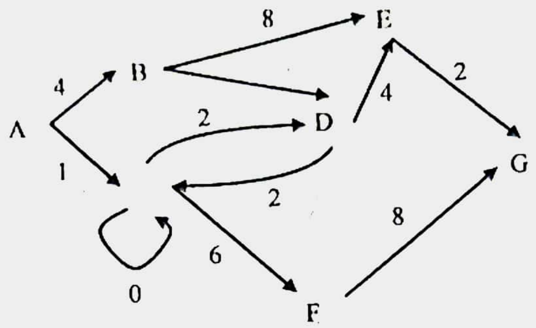
State	Next	Cost
A	B	4
A	C	1
B	D	3
B	E	8
C	C	[illegible]
C	D	2
C	F	6
D	C	2
D	E	4
E	G	2
F	G	8

Assume that the initial state is A and the goal state is G. Show how each of the following search strategies would create a search tree to find a path from the initial state to the goal state and the cost of the solution:

i) Breadth-first search
ii) Depth-first search
iii) Iterative deepening search **[WBUT 2018]**

Answer:

The graphs is as shown below:



14. a) What is Skolemisation? **[WBUT 2018]**

Answer:

Skolemization is a way of removing existential quantifiers from a formula. Variables bound by existential quantifiers which are not inside the scope of universal quantifiers can simply be replaced by constants: $\exists x[x<3]$ can be changed to $c<3$, with c a suitable constant.

When the existential quantifier is inside a universal quantifier, the bound variable must be replaced by a Skolem function of the variables bound by universal quantifiers. Thus $\forall x[x=0 \lor \exists y[x=y+1]]$ becomes $\forall x[x=0 \lor x=f(x)+1]$.

In general, the functions and constants symbols are new ones added to the language for satisfying these formulas, and are often denoted by the formula they realize, for instance $c\exists x\square(x)$.

This is used in second order logic to move all existential quantifiers outside the scope of first order universal quantifiers. This can be done since second order quantifiers can quantify over functions. For instance $\forall 1x\forall 1y\exists 1z\square(x,y,z)$ is equivalent to $\exists 2F\forall 1x\forall 1y\square(x,y,F(x,y))$.

b) Given the following text 'Everyone who enters in a theatre has to buy a ticket. Person who doesn't have money can't buy a ticket. Vinod enters a theatre.' Prove by resolution that 'Vinod has money'. **[WBUT 2018]**

Answer:

Let us define the following predicates

b(x): x buys ticket.

m(x) x has money.

e(x,t): x enters theatre.

So we have the following:

1. $\forall x\,[e(x,t) \rightarrow b(x)]$
2. $\forall x[\neg m(x) \rightarrow \neg b(x)]$
3. e(vinod,t)
4. b(vinod) from 1 and 3
5. m(vinod) from 2 and 4

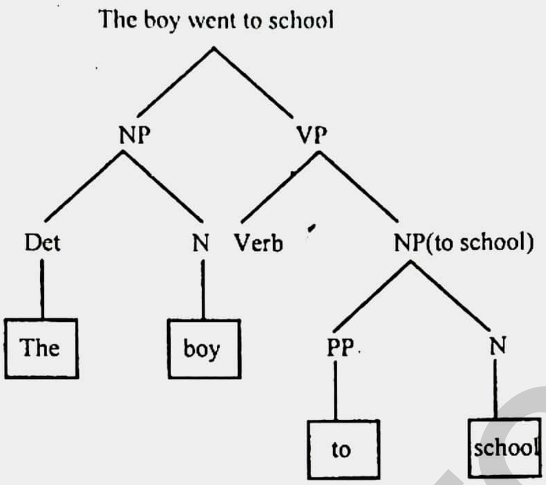
Therefore 'Vinod has money'.

15. a) Briefly explain the steps of Natural language Processing. [WBUT 2018]
b) Generate the parse tree for the sentence 'The boy went to School'.
c) Explain AO* algorithm with a suitable example.

Answer:

a) *Refer to Question No. 15 (1st part) of Short Answer Type Questions.*

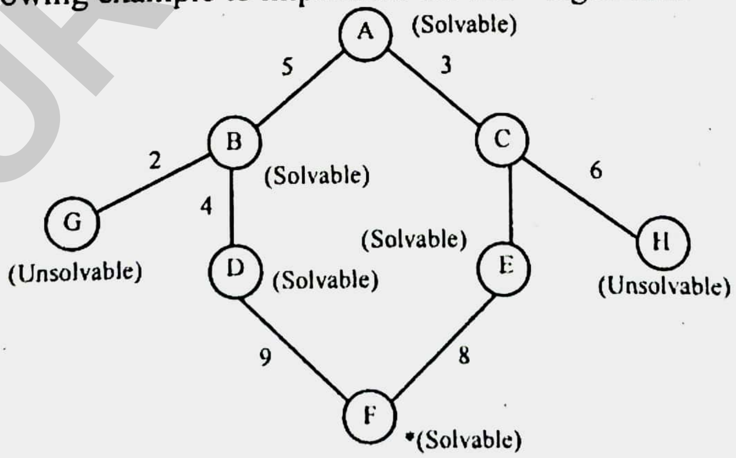
b)



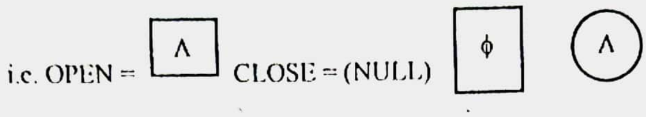
c) AO* *Algorithm*

1. Initialize the graph to start node
2. Traverse the graph following the current path accumulating nodes that have not yet been expanded or solved
3. Pick any of these nodes and expand it and if it has no successors call this value *FUTILITY* otherwise calculate only *f* for each of the successors.
4. If *f* is 0 then mark the node as *SOLVED*
5. Change the value of *f* for the newly created node to reflect its successors by back propagation.
6. Wherever possible use the most promising routes and if a node is marked as *SOLVED* then mark the parent node as *SOLVED*.
7. If starting node is *SOLVED* or value greater than *FUTILITY*, stop, else repeat from 2.

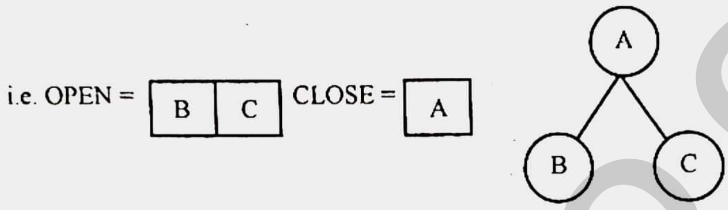
Let us take the following example to implement the AO* algorithm.



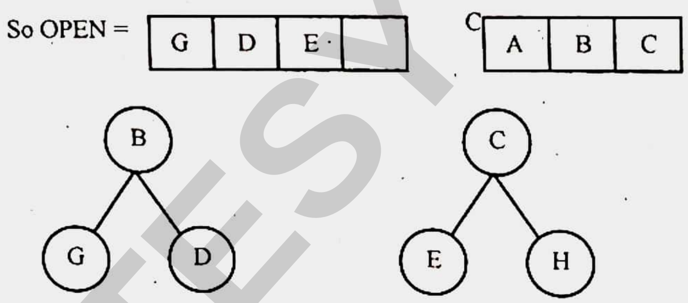
Step 1:
In the above graph, the solvable nodes are A, B, C, D, E, F and the unsolvable nodes are G, H. Take A as the starting node. So, place A into OPEN.



Step 2
The children of A are B and C which are solvable. So place them into OPEN and place A into the CLOSE.

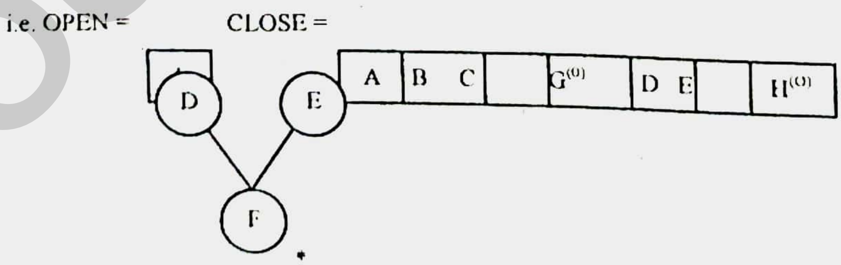


Step 3
Now process the nodes B and C. The children of B and C are to be placed into OPEN. Also remove B and C from OEPN and place them into CLOSE.



(O)
'O' indicated that the nodes G and H are unsolvable.

Step 4:
As the nodes G and H are unsolvable, so place them into CLOSE directly and process the nodes D and E.

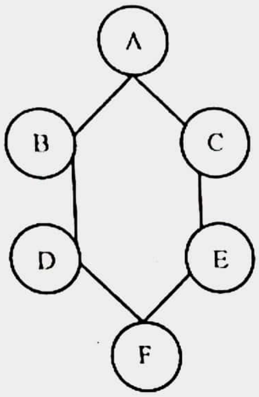


Step 5:
Now we have been reached at our goal state. So place F into CLOSE.

i.e. CLOSE =

A	B C		$G^{(0)}$	D E		$H^{(0)}$	F

Step 6:
Success and Exit
AO* Graph



16. Describe alpha-beta pruning procedure. **[WBUT 2019]**

Answer:
Refer to Question No. 17(c) of Long Answer Type Questions.

17. Write short notes on the following:
a) Semantic net **[WBUT 2015]**
b) Conceptual graph **[WBUT 2018]**
c) Alpha-Beta pruning in min-max search **[WBUT 2018]**
d) The steps for transforming into Clause Form **[WBUT 2018]**

Answer:
a) Semantic net:
Refer to Question No. 1 of Long Answer Type Questions.

b) Conceptual graph:
Refer to Question No. 3(i) of Long Answer Type Questions.

c) Alpha-Beta pruning in min-max search:
Alpha-beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the mini-max algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two player games Such as Tic-tac-toe, Chess etc. It stops evaluating a move when at-least one possibility has been found that proves the moves to be worst.
Example: ***Refer to Question No. 2 of Long Answer Type Questions.***

d) Algorithm for transforming into a clause form:
Step I: Elimination of if-then operator:
Replace"→" operator by ¬ & Voperator.

By replacing 'if-then' operator by negation and OR operator, we find.
∀X (¬ (Child (X)∧ ∃Y (Takes (X, Y) ∧ Biscuit (Y))) ∨ Loves (john, X)

Step II: Reduction of the scope of negation:
Replace ¬ sign by choosing any of the following:

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

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

c) $\neg(\neg P) = P$

d) $\neg(\exists X\, P) = \forall X \neg P$

e) $\neg(\forall X\, P) = \exists X \neg P$

In the present context, we rewrite the sentences as:

∀X (¬ (Child (X) ∨ ¬ (∃Y (Takes (X, Y) ∧ Biscuit (Y))) ∨ Loves (john, X))
⇒∀X (¬ (Child (X) ∨ ∀Y (¬Takes (X, Y) ∨¬ Biscuit (Y)) ∨ Loves

Step III: Renaming the variable within the scope of quantifiers:
Rename ∃X by ∃Y when {∃X} is a subset/proper subset of {∀X}. In the present context, since X and Yare distinct, the above operation cannot be carried out.

Step IV: Moving of quantifiers in the front of the expression:
Bring all quantifiers at the front of the expression.
Applying this on the example yields.

∀X∀Y ¬ Child (X) ∨ ¬ Takes (X, Y) ∨¬ Biscuit (Y) ∨ Loves (john, X)

Step V: Replacing existential quantifier as Skolem function of essential quantifiers:
When an existential quantifier (Y) precedes an essential quantifier (X), replace Y as S (X), where S is the Skolem function. In this example, since Y is not a subset of X, such a situation does not arise. Also the essential quantifier is dropped from the sentence.

Step VI: Putting the resulting expression in conjunctive normal form (CNF):
For example, if the original expression is in the from P ∨ (Q ∧ R), then replace it by (P ∨ Q) ∧ (Q ∧ R).
In the present context, the resulting expression corresponding to expression (3) being in CNF, we need not do any operation at this step.

Step VII: Writing one clause per line:
If the original expression is of the following CNF, then rewrite each clause/line, as illustrated below.

Original Expression:

$$\left(\neg P_{11} \vee \neg P_{12\cdots} \vee \neg P_{1n} \vee Q_{11} \vee \neg Q_{12} \vee Q_{1m}\right) \wedge$$

$$\underline{\left(\neg P_{21} \vee \neg P_{22\cdots} \vee \neg P_{2n} \vee Q_{21} \vee \neg Q_{22\cdots} \vee Q_{2m}\right) \wedge}$$

$$\left(\neg P_{11} \vee \neg P_{12\cdots} \vee \neg P_{1n} \vee Q_{11} \vee Q_{12\cdots} \vee Q_{1m}\right)$$

PROBABILISTIC REASONING

Multiple Choice Type Questions

1. A Baysian network is a [WBUT 2008, 2014, 2015, 2017, 2018]
   a) tree b) directed graph c) undirected graph d) none of these

Answer: (b)

2. Let a and b be any two events. Which of the following must be true? [WBUT 2014]
   a) $P(a \cup b) = P(a) + P(b)$
   b) $P(a) \leq 0$
   c) $P(a) + P(\neg a) = 1$
   d) none of these

Answer: (c)

3. Fuzzy logic is a form of [WBUT 2016]
   a) Two-valued logic
   b) Crisp set logic
   c) Many-valued logic
   d) Binary set logic

Answer: (c)

4. The truth values of traditional set theory is .... and that fuzzy set is .... [WBUT 2016]
   a) Either 0 or 1, between 0 & 1
   b) Between 0 and 1, either 0 or 1
   c) Between 0 and 1, between 0 and 1
   d) Either 0 or 1, either 0 or 1

Answer: (a)

5. The process of eliminating existential quantifiers is known as [WBUT 2017, 2018]
   a) Resolution
   b) Skolemisation
   c) Unification
   d) none of these

Answer: (b)

Short Answer Type Questions

1. What is Dempster Shafer Theory? [WBUT 2006, 2012]

Answer:

The Dempster-Shafer theory, also known as the theory of belief functions, is a generalization of the Bayesian theory of subjective probability. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ from probabilities will depend on how closely the two questions are related. The Dempster-Shafer theory is based on two ideas: the idea of obtaining degrees of belief for one question from subjective probabilities for a related question, and

Dempster's rule for combining such degrees of belief when they are based on independent items of evidence.

2. A box contains 10 screws out of which 3 are defective. Two screws are drawn at random. Let A = first drawn screw is non-defective, B = second drawn screw is non-defective.
Using the concept of sampling without replacement evaluate $P(B/A)$ and $P(A \cap B)$. [WBUT 2009]

Answer:

P(A) = 7/10

P(B/A) = 6 / 9 =2 / 3

P (A ∩B) = P(A).P (B/A) = 14 / 30 = 0.47

3. When one has cold, one usually has a high temperature (80%) of the time. At any time around 1 in every 10000 people has a cold and 1 in every 1000 people has a high temperature.
Now, suppose that you have a high temperature. What is the probability that you have cold? [WBUT 2014]

Answer:

Let A denote "Have a high temperature" and B denote "Have a cold". Thus

P(A)=0.001

P(B)=0.0001

P(A/B)= 0.8

By using Bayes theorem we calculate.

$$P(B|A) = \frac{P(B|A) \cdot P(B)}{P(A)} = \frac{0.8 \cdot 0.0001}{0.001} = 0.008$$

Long Answer Type Questions

1. Write down the differences between conventional set and fuzzy set.[WBUT 2011]

Answer:

A set is a collection of things, for example the room temperature, the set of all real numbers, etc. Such collection of things is called the Universe of Discourse, X, and is defined as the range of all possible values for a variable. Universe of Discourse can be divided into sets or subsets. For Example, consider a set A of the real numbers between 5 and 8 from the universe of discourse X. Conventional sets are called crisp sets. In classic sets, the transition of an element in the universe between being a member and non member in a given set is abrupt.

In fuzzy sets, this transition occurs gradually. A fuzzy set is a set containing elements that have varying degree of membership in the set. Accordingly, elements in a fuzzy sets can be members of other fuzzy set on the same universe. Elements of fuzzy sets are mapped to a universe of membership values using a function-theoretic form. The same operations of the classical sets are still valid for the fuzzy sets.

2. a) What is fuzzy set? What is the difference between fuzzy set and crisp set? Explain different fuzzy operations using examples.
b) What do you mean by conflict resolution strategy? Design a search space for the given set of production rules.

$p \cap q \rightarrow \text{goal}$
$r \cap s \rightarrow p$
$w \cap r \rightarrow q$
$t \cap u \rightarrow q$
$v \rightarrow s$

start $\rightarrow v \cap r \cap q$ **Resolution act strategy : Conflict resolution strategies fire the most recently added rule in the working memory.**
c) What do you mean by Skolem constant and Skolem function? Explain Inductive Learning. **[WBUT 2012]**

Answer:

1st Part:

Fuzzy sets are sets whose elements have degrees of membership. In mathematical notation, a fuzzy set is a pair (U, m) where U is a set and $m: U \rightarrow [0,1]$.

2nd Part:

Crisp sets are the sets that we have used most of our life. In a crisp set, an element is either a member of the set or not. For example, a jelly bean belongs in the class of food known as candy. Mashed potatoes do not.
Fuzzy sets allow elements to be *partially* in a set. Each element is given a degree of membership in a set. This membership value can range from 0 (not an element of the set) to 1 (a member of the set). It is clear that if one only allowed the extreme membership values of 0 and 1, that this would actually be equivalent to crisp sets. A membership function is the relationship between the values of an element and its degree of membership in a set.

3rd Part:

The different fuzzy set operations are union, intersection and complement.

- **Union**

The membership function of the Union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions. This is called the *maximum* criterion.

$\mu_{A \cup B} = \max(\mu_A, \mu_B)$

The Union operation in Fuzzy set theory is the equivalent of the **OR** operation in Boolean algebra.

- **Intersection**

The membership function of the Intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions. This is called the *minimum* criterion.

$\mu_{A \cap B} = \min(\mu_A, \mu_B)$

The Intersection operation in Fuzzy set theory is the equivalent of the AND operation in Boolean algebra.

- **Complement**

The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as the negation of the specified membership function. This is called the *negation* criterion.

$\mu_{\bar{A}} = 1 - \mu_A$

The Complement operation in Fuzzy set theory is the equivalent of the **NOT** operation in Boolean algebra.

The following rules which are common in classical set theory also apply to Fuzzy set theory: De Morgans law, Associativity, Commutativity, Distributivity.

b) 1st part:

Conflict resolution in a forward-chaining inference engine decides which of several rules that *could* be fired (because their condition part matches the contents of working memory should actually be fired. Conflict resolution proceeds by sorting the rules into some order, and then using the rule that is first in that particular ordering. There are quite a number of possible orderings that could be used:

Specificity Ordering
Rule Ordering
Data Ordering
Size Ordering
Recency Ordering
Context Limiting

2nd Part:

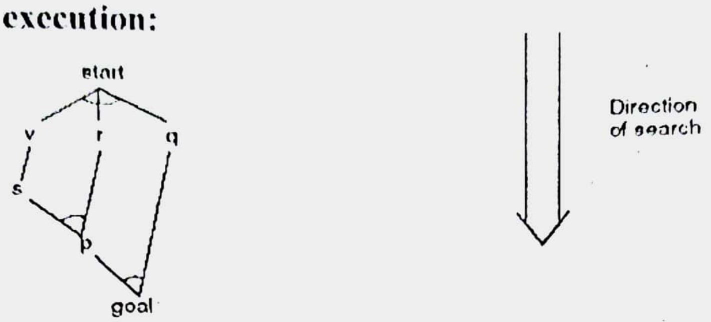
Production set:

1. p ^ q → goal
2. r ^ s → p
3. w ^ r → q
4. t ^ u → q
5. v → s
6. start → v ^ r ^ q

Trace of execution:

Iteration #	Working Memory	Conflict set	Rule fired
0	start	6	6
1	start, v, r, q	6, 5	5
2	start, v, r, q, s	6, 5, 2	2
3	start, v, r, q, s, p	6, 5, 2, 1	1
4	start, v, r, q, s, p, goal	6, 5, 2, 1	halt

Space searched by execution:



c) 1st part:

A *Skolem constant* is a new constant that is substituted for a variable when eliminating an existential quantifier from a fact or a universal quantifier from a conjecture.

Consider a sentence like $\exists x[dog(x)]$ from which we would like to get rid of the variables. What we can do is simply to create a new object constant, called a Skolem constant, and re-express the sentence with this: i.e. convert $\exists x[dog(x)]$ to dog(Sk1).

Consider a case where an existential quantifier is under the scope of a universal quantifier e.g., $\forall x[\exists y[loves(x;y)]]$ The object loved might be different for each x. So we cannot just replace y with a Skolem constant. However, if the sentence is true, then it must be possible to define a function that takes an object x and returns an object that x loves. We can refer to this function by giving it a name (e.g. beloved_of) even when we do not know how it is defined. This kind of invented function is called a *Skolem function*. We can eliminate existential quantifiers using a Skolem function as given below:

$$\forall x[loves(x;beloved\_of(x))] \text{ or alternatively } \forall x[loves(x;sk1(x))]$$

2nd part:

An inductive approach does not require a complete and tractable domain theory and has the potential to create more effective rules by learning from more than one example at a time.

The idea behind this form of learning is extrapolate from a given set of examples so that we can make accurate predictions about future examples. Inductive learning is an inherently conjectural process because any knowledge created by generalization from specific facts cannot be proven true; it can only be proven false. Hence, inductive inference is falsity preserving, not truth preserving.

In supervised learning we have a set of $\{xi, f(xi)\}$ for $1 \le i \le n$, and our aim is to determine 'f' by some adaptive algorithm. The inductive learning is a special class of the supervised learning techniques, where given a set of $\{xi, f(xi)\}$ pairs, we determine a hypothesis $h(xi)$ such that $h(xi) \approx f(xi), \ \forall i$.

A natural question that may be raised is how to compare the hypothesis h that approximates f. For instance, there could be more than one $h(xi)$ where all of which are approximately close to f(xi). Let there be two hypothesis $h1$ and $h2$, where $h1(xi) \approx f(xi)$ and $h2(xi) = f(xi)$. We may select one of the two hypotheses by a preference criterion, called bias. Two types of biases are commonly used in machine learning:

- Restricted Hypothesis Space Bias Allow only certain types of f functions, not arbitrary ones
- Preference Bias Define a metric for comparing fs so as to determine whether one is better than another

A completely unbiased inductive algorithm could only memorize the training examples and could not say anything more about other unseen examples.

Inductive Learning Framework

Raw input data from sensors are preprocessed to obtain a feature vector, x, that adequately describes all of the relevant features for classifying examples. Each x is a list of (attribute, value) pairs. For example,

x = (Person = Sue, Eye-Color = Brown, Age = Young, Sex = Female)

The number of attributes (also called features) is fixed (positive, finite). Each attribute has a fixed, finite number of possible values. Each example can be interpreted as a *point* in an n-dimensional feature space, where n is the number of attributes.

3. Describe the fuzzy set operations like: union, intersection and complement. **[WBUT 2017]**

Answer:

Refer to Question No. 2(a) (3rd part) of Long Answer Type Questions.

4. Write short notes on the following:

a) Bayesian network **[WBUT 2009]**

b) Dempster-Shafer evidence theory **[WBUT 2011, 2012, 2013]**

c) Fuzzy set **[WBUT 2015]**

Answer:

a) Bayesian network:

Bayesian networks (BNs), also known as belief networks (or Bayes nets for short), belong to the family of probabilistic graphical models (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BNs combine principles from graph theory, probability theory, computer science, and statistics.

BNs correspond to the GM structure known as a *directed acyclic graph* (DAG). BNs are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the joint probability distribution (JPD) over a set of random variables. A BN reflects a simple conditional independence statement. Namely that each variable is independent of its nondescendents in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters that are required to characterize the JPD of the variables.

b) Dempster-Shafer evidence theory:
Refer to Question No. 1 of Short Answer Type Questions.

c) Fuzzy set:
Refer to Question No. 2(a) (1st & 3rd Part) of Long Answer Type Questions.

NATURAL LANGUAGE PROCESSING

Multiple Choice Type Questions

1. Which of the following includes major tasks of NLP? [MODEL QUESTION]
   a) Discourse Analysis
   b) Automatic Summarization
   c) Machine Translation
   d) All of these

Answer: (d)

2. "He lifted the beetle with red cap." contain which type of ambiguity? [MODEL QUESTION]
   a) Lexical ambiguity
   b) Syntax Level ambiguity
   c) Referential ambiguity
   d) None of these

Answer: (b)

3. In NLP, The process of removing words like "and", "is", "a", "an", "the" from a sentence is called as [MODEL QUESTION]
   a) Stemming
   b) Lemmatization
   c) Stop word
   d) All of these

Answer: (c)

4. Natural language processing can be divided into the two subfields of: [MODEL QUESTION]
   a) context and expectations
   b) generation and understanding
   c) semantics of pragmatics
   d) recognition and synthesis

Answer: (b)

5. Which of the following techniques can be used for the purpose of keyword normalization, the process of converting a keyword into its meaningful base form? [MODEL QUESTION]
   a) Lemmatization
   b) Levenshtein distance
   c) Morphing
   d) Stemming

Answer: (a)

Short Answer Type Questions

1. For a compositional semantics model, show the semantics you would attach to the rule S-> NP VP, the semantic entry for the verb "sleep" and the semantic analysis that would result for the sentence "John slept" assuming that the semantics for the NP "John" is "JOHN1". [MODEL QUESTION]

Answer:
Sleep: λx Эe is-a(e, sleeping) and e(x)
Semantics for S-> NP VP: Sem(VP)(Sem(NP))
Semantic analysis of sentence: Эe is-a(e, sleeping) and e(JOHN1)

2. Explain Dependency Parsing in NLP? **[MODEL QUESTION]**

Answer:

Dependency Parsing, also known as Syntactic parsing in NLP is a process of assigning syntactic structure to a sentence and identifying its dependency parses. This process is crucial to understand the correlations between the "head" words in the syntactic structure. The process of dependency parsing can be a little complex considering how any sentence can have more than one dependency parses. Multiple parse trees are known as ambiguities. Dependency parsing needs to resolve these ambiguities in order to effectively assign a syntactic structure to a sentence. Dependency parsing can be used in the semantic analysis of a sentence apart from the syntactic structuring.

3. State two advantages of partial parsers over parsers that provide in-depth syntactic information. **[MODEL QUESTION]**

Answer:

First, partial parsers can be more robust than the regular parsers, because partial parsers work on easier tasks. Second, for some NLP applications such as information extraction, information derived from partial parsers can be more relevant than that from regular parsers.

4. What are some ways we can preprocess text input? **[MODEL QUESTION]**

Answer:

Here are several pre-processing steps that are commonly used for NLP tasks:

- case normalization: we can convert all input to the same case (lowercase or uppercase) as a way of reducing our text to a more canonical form
- punctuation/stop word/white space/special characters removal: if we don't think these words or characters are relevant, we can remove them to reduce the feature space
- lemmatizing/stemming: we can also reduce words to their inflectional forms (i.e. walks → walk) to further trim our vocabulary
- generalizing irrelevant information: we can replace all numbers with a <NUMBER> token or all names with a <NAME> token

Long Answer Type Questions

1. The Lesk algorithm is a simple method for word sense disambiguation that relies on the use of dictionaries. Here are glosses and examples from Wiktionary for three different senses of the word *course*: **[MODEL QUESTION]**

1. A normal or customary sequence.

2. A learning program, as in university. ***I need to take a French course.***

3. The direction of movement of a vessel at any given moment. ***The ship changed its course 15 degrees towards south.***

a) Which of the three senses does the word *course* have in the following sentence? In the United States, the normal length of a course is one academic term.

b) Which of the three senses does the Lesk algorithm predict based on the given glosses and examples? Ignore the word *course*, punctuation, and stop words. Explain your answer.

c) Change the sentence such that the word *course* maintains its original sense, but the Lesk algorithm now predicts a different sense than in the previous item.

Answer:

a) Sense 2

b) Sense 1. The algorithm chooses the sense whose lexicon entry (gloss plus example) has the highest number of overlapping words with the sentence. In this case there is 1 overlapping word with the entry for sense 1 (normal) but no overlap with the other entries.

c) Example: In the United States, the typical length of a university course is one academic term. (One overlap with sense 2 [university], no overlap with sense 1 [normal changed to typical]) or sense 3.

2. A transition-based dependency parser analyses the sentence *I booked a flight from L.A.* Here is the gold-standard tree for this sentence. [MODEL QUESTION]

a) Suppose that the parser starts in the initial configuration for the sentence and takes the transitions SH, SH, LA. State the new configuration. To represent the partial dependency tree, list the arcs contained in it.

b) State a complete sequence of transitions that takes the parser all the way from the initial configuration to a terminal configuration, and that recreates all arcs of the gold-standard tree.

c) Provide a modified transition sequence where the parser mistakingly predicts the arc *booked → from*, but gets the other dependencies right

Answer:

a) Configuration after SH, SH, LA:

stack: [booked] **buffer:** [a, flight, from, L.A.]

Partial dependency tree: booked → I

b) SH SH LA SH SH LA SH SH RA RA RA

c) SH SH LA SH SH LA RA SH SH RA RA

LEARNING

Multiple Choice Type Questions

1. Deriving the concept of "luxury item" from a diamond ring, dinner at Mainland China, a Mercedes-Benz car falls under [WBUT 2008]
   a) inductive learning
   b) analogical learning
   c) supervised learning
   d) learning automata

Answer: (b)

2. How the new states are generated in genetic algorithm? [WBUT 2016]
   a) Composition
   b) Mutation
   c) Cross-over
   d) Both (b) and (c)

Answer: (d)

3. The rule used to charge weight in Neural Network (NN) is [WBUT 2017]
   a) Kirchoff's rule
   b) Hebb's rule
   c) Boehm's rule
   d) none of these

Answer: (b)

4. The rule used to change weight in Neural Network (NN) is [WBUT 2018]
   a) Kirchoff's rule
   b) Hebb's rule
   c) Boehm's rule
   d) None of these

Answer: (b)

Short Answer Type Questions

1. Write down the advantages and disadvantages of Genetic Algorithm. [WBUT 2016]

OR,

Write the advantages of Genetic algorithm. [WBUT 2017]

Answer:

Advantages:

- Search is unconstrained neither by the continuity of the function under investigation, nor the existence of a derivative function.
- Efficient means of investigating large combinatorial problems- can solve combinatorial problems many orders of magnitude faster than exhaustive 'brute force' searches

Disadvantages:

- Computationally expensive- some problems require many days or weeks to run often still faster than brute force, however
- Blind, undirected search -difficult to direct a GA towards optimal solution area if known

- Can be sensitive to initial parameters- parameters such as mutation can significantly influence the search
- Stochastic process - not guaranteed to find an optimal solution, just highly likely to

2. Describe neural network based learning. **[WBUT 2019]**

Answer:

Refer to Question No. 2(b) of Long Answer Type Questions.

Long Answer Type Questions

1. In Genetic algorithm, how do you obtain the new chromosome (solution) from the old one? **[WBUT 2011]**

Answer:

Refer to Question No. 2(a) ("algorithm" part) of Long Answer Type Questions.

2. Write short notes on the following:

a) Genetic Algorithm **[WBUT 2009, 2015]**

b) Neural network **[WBUT 2009, 2015]**

Answer:

a) Genetic Algorithm:

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

GAs were introduced as a computational analogy of adaptive systems. They are modelled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness.

The Algorithm

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
   i) **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
   ii) **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

iii) **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).

iv) **[Accepting]** Place new offspring in a new population

4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

The paradigm of GAs described above is usually the one applied to solving most of the problems presented to GAs. Though it might not find the best solution. more often than not, it would come up with a partially optimal solution.

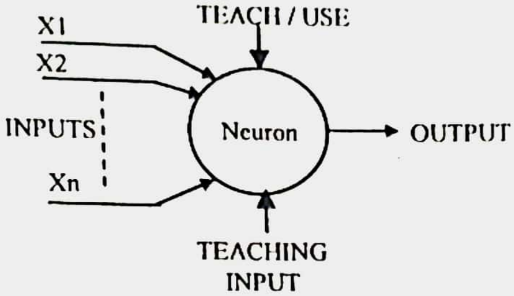
b) Neural network:

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.



ANN uses neurons to train and use them in computations. An artificial neuron shown in the figure below is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

3. Explain the cycle of genetic algorithm. Discuss different types of crossover techniques. [WBUT 2012]

Answer:

1st Part: ***Refer to Question No. 2(a) of Long Answer Type Questions.***

2nd Part:

Many crossover techniques exist for organisms which use different data structures to store themselves.

One-point crossover

A single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children.

Two-point crossover

Two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

"Cut and splice"

Another crossover variant, the "cut and splice" approach, results in a change in length of the children strings. The reason for this difference is that each parent string has a separate choice of crossover point.

Uniform Crossover and Half Uniform Crossover

The Uniform Crossover uses a fixed mixing ratio between two parents. Unlike one- and two-point crossover, the Uniform Crossover enables the parent chromosomes to contribute the gene level rather than the segment level.

If the mixing ratio is 0.5, the offspring has approximately half of the genes from first parent and the other half from second parent, although cross over points can be randomly chosen.

The Uniform Crossover evaluates each bit in the parent strings for exchange with a probability of 0.5. Even though the uniform crossover is a poor method, empirical evidence suggest that it is a more exploratory approach to crossover than the traditional exploitative approach that maintains longer schemata. This results in a more complete search of the design space with maintaining the exchange of good information. In the uniform crossover scheme (UX) individual bits in the string are compared between two parents. The bits are swapped with a fixed probability, typically 0.5.
In the half uniform crossover scheme (HUX), exactly half of the non-matching bits are swapped. Thus first the Hamming distance (the number of differing bits) is calculated. This number is divided by two. The resulting number is how many of the bits that do not match between the two parents will be swapped.

Three parent crossover
In this technique, the child is derived from three parents. They are randomly chosen. Each bit of first parent is checked with bit of second parent whether they are same. If same then the bit is taken for the offspring otherwise the bit from the third parent is taken for the offspring. For example, the following three parents:
parent1 1 1 0 1 0 0 0 1 0
parent2 0 1 1 0 0 1 0 0 1
parent3 1 1 0 1 1 0 1 0 1
produces the following offspring:
offspring 1 1 0 1 0 0 0 0 1[3]

Crossover for Ordered Chromosomes
Depending on how the chromosome represents the solution, a direct swap may not be possible. One such case is when the chromosome is an ordered list, such as an ordered list of the cities to be travelled for the traveling salesman problem. There are many crossover methods for ordered chromosomes. The already mentioned N-point crossover can be applied for ordered chromosomes also, but this always need a corresponding repair process, actually, some ordered crossover methods are derived from the idea. However, sometimes a crossover of chromosomes produces recombination which violate the constraint of ordering and thus need to be repaired. Several examples for crossover operators (also mutation operator) preserving a given order are given below

- partially matched crossover (PMX): In this method, two crossover points are selected at random and PMX proceeds by position wise exchanges. The two crossover points give matching selection. It affects cross by position-by-position exchange operations. In this method parents are mapped to each other, hence we can also call it partially mapped crossover.

- cycle crossover (CX): Beginning at any gene i in parent 1, the ith gene in parent 2 becomes replaced by it. The same is repeated for the displaced gene until the gene which is equal to the first inserted gene becomes replaced (cycle).
- order crossover operator (OX1): A portion of one parent is mapped to a portion of the other parent. From the replaced portion on, the rest is filled up by the remaining genes, where already present genes are omitted and the order is preserved.
- order-based crossover operator (OX2)
- position-based crossover operator (POS)
- voting recombination crossover operator (VR)
- alternating-position crossover operator (AP)
- sequential constructive crossover operator (SCX)

EXPERT SYSTEMS

Multiple Choice Type Questions

1. The inference engine in an expert system interpretes **[WBUT 2008]**
a) database with the help of a knowledge-base
b) knowledge-base using a database
c) database using working memory
d) database using a Graphical User Interface (GUI)

Answer: (a)

2. MYCIN is an example of **[WBUT 2017]**
a) expert system **b) knowledge base**
c) conceptual graph **d) semantic net**

Answer: (a)

3. When do we call the states are safely explorable? **[WBUT 2019]**
a) A goal state is unreachable from any state
b) A goal state is denied access
c) A goal state is reachable from every state
d) None of the mentioned

Answer: (c)

Short Answer Type Questions

1. What is expert system? Why is it required? **[WBUT 2008, 2009, 2015]**

Answer:

An expert system is a computer programme where data is stored and manipulated by the programme to come up with advises, hints, directions in reaction of input by users of data acquisition devices.

Expert systems are used in professional areas like Diagnostics (Medicines), Construction, and Simulation. Since the growth rate in this specific field is 100 percent it is to be expected that expert systems will be applied in other areas. In principle is an expert system a further development of the (conventional) third generation programming languages. And as such a system like that can hardly be called an extension of Artificial Intelligence.

The advantage of an expert system can be seen in the fact that a separation takes place between collected knowledge and Problem Solving Strategy. Because of this differentiation maintenance costs of the system can be at a minimum. A complete Expert System consists of the following components:

1. KNOWLEDGE BASE composed by experts in a particular field
2. COMMUNICATION components
3. SITUATION dependent or CONTEXT related HELP system
4. PROBLEM SOLVING components. (line interpreter)

Only a combination of all these components allows the system to be called an EXPERT SYSTEM.

2. What is expert system? What is expert system shell? **[WBUT 2012]**

Answer:

1st Part: ***Refer to Question No 1 of Short Answer Type Questions.***

2nd Part:

Compared to the wide variation in domain knowledge, only a small number of AI methods are known that are useful in expert systems. That is, currently there are only a handful of ways in which to represent knowledge, or to make inferences, or to generate explanations. Thus, systems can be built that contain these useful methods without any domain-specific knowledge. Such systems are known as Expert system *shells*.

Building expert systems by using shells offers significant advantages. A system can be built to perform a unique task by entering into a shell all the necessary knowledge about a task domain. The inference engine that applies the knowledge to the task at hand is built into the shell. If the program is not very complicated and if an expert has had some training in the use of a shell, the expert can enter the knowledge himself.

Many commercial shells are available today, ranging in size from shells on PCs, to shells on workstations, to shells on large mainframe computers. They range in complexity from simple, forward-chained, rule-based systems requiring two days of training to those so complex that only highly trained knowledge engineers can use them to advantage. They range from general-purpose shells to shells custom-tailored to a class of tasks, such as financial planning or real-time process control. Although shells simplify programming, in general they do not help with knowledge acquisition. The power of an expert system lies in its store of knowledge about the task domain -- the more knowledge a system is given, the more competent it becomes.

3. What do you mean by knowledge acquisition? **[WBUT 2012]**

Answer:

Knowledge acquisition refers to the task of endowing expert systems with knowledge, a task currently performed by knowledge engineers. Knowledge acquisition includes the elicitation, collection, analysis, modeling and validation of knowledge for knowledge engineering and knowledge management projects.

Some of the most important issues in knowledge acquisition are as follows:

- Most knowledge is in the heads of experts
- Experts have vast amounts of knowledge
- Experts have a lot of tacit knowledge
  - They don't know all that they know and use
  - Tacit knowledge is hard (impossible) to describe
- Experts are very busy and valuable people
- Each expert doesn't know everything
- Knowledge has a "shelf life"

Many techniques have been developed to help elicit knowledge from an expert. These are referred to as knowledge elicitation or knowledge acquisition (KA) techniques. The term "KA techniques" is commonly used.

The following list gives a brief introduction to the types of techniques used for acquiring, analyzing and modeling knowledge:

- Protocol-generation techniques include various types of interviews (unstructured, semi-structured and structured), reporting techniques (such as self-report and shadowing) and observational techniques
- Protocol analysis techniques are used with transcripts of interviews or other text-based information to identify various types of knowledge, such as goals, decisions, relationships and attributes. This acts as a bridge between the use of protocol-based techniques and knowledge modeling techniques.
- Hierarchy-generation techniques, such as laddering, are used to build taxonomies or other hierarchical structures such as goal trees and decision networks.
- Matrix-based techniques involve the construction of grids indicating such things as problems encountered against possible solutions. Important types include the use of frames for representing the properties of concepts and the repertory grid technique used to elicit, rate, analyze and categorize the properties of concepts.
- Sorting techniques are used for capturing the way people compare and order concepts, and can lead to the revelation of knowledge about classes, properties and priorities.
- Limited-information and constrained-processing tasks are techniques that either limits the time and/or information available to the expert when performing tasks. For instance, the twenty-question technique provides an efficient way of accessing the key information in a domain in a prioritized order.
- Diagram-based techniques include the generation and use of concept maps, state transition networks, event diagrams and process maps. The use of these is particularly important in capturing the "what, how, when, who and why" of tasks and events.

Long Answer Type Questions

1. Write short note on Expert System. **[WBUT 2013, 2018]**

Answer:

Refer to Question No. 1 Short Answer Type Questions.

QUESTION 2015

Group – A

(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:

i) A Bayesian network is a

a) tree
✓b) directed graph
c) undirected graph
d) none of these

ii) Which is NOT a heuristic search?

a) A* search
b) steepest ascent Hill-climbing
c) Simulated annealing
✓d) Depth first search

iii) "Lata is slightly ill". This statement can be completely expressed in

a) FOPL
b) Propositional logic
✓c) Fuzzy logic
d) None of these

iv) Uninformed search is also know as

a) brute force search
b) hill climbing search
✓c) blind search
d) none of these

v) For a given proposition q, $q \vee \neg q$ is a

✓a) tautology
b) contradiction
c) satisfiable formula
d) none of these

vi) Which of the following is there in Prolog?

a) Existential quantifier
b) Universal quantifier
✓c) Conjunction
d) Disjunction

vii) Inheritable knowledge is best represented by

✓a) semantic net
b) database
c) first order logic
d) none of these

viii) The first order logic is

✓a) both sound and complete
b) sound but not complete
c) complete but not sound
d) neither sound nor complete

ix) If in a problem the number of initial states is much more than the number of final states we should use

✓a) backward reasoning
b) forward reasoning
c) both (a) and (b)
d) none of these

x) Which of the following is NOT a conflict resolution strategy in production system?

✓a) Production rules
b) Recency
c) Refraction
d) Specificity

Group – B

(Short Answer Type Questions)

2. a) What is the difference between Greedy best-first search and A* search?
b) Under what condition is breadth-first search optimal?
c) Show that any monotonic heuristic is admissible.
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 12.

3. a) What is semantic net?
b) With the help of semantic net, represent the fact that Sourav is 6 feet tall and that he is taller than Sachin.
a) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 1.
b) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 1.

4. What do you mean by completeness of a search? Why is DFS not always complete?
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 4.

5. a) Define Horn Clause.
b) Is $p \rightarrow q$ a Horn Clause? Justify your answer.
c) What is meant by tautology in prepositional logic?
See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 2.

6. What is an expert system? Why is it required?
See Topic: EXPERT SYSTEM, Short Answer Type Question No. 1.

Group – C
(Long Answer Type Questions)

7. a) Consider the following 3-puzzle problem:

Start State

2	3
1	

Goal State

1	2
	3

Possible operators (in order) are up, down, left and right, assume that repeated states are not detected. Label each visited node with a number indicating the order in which they are visited.

(i) Draw the search tree using BFS

(ii) Would DFS find the goal? Explain.

(iii) A* search with the heuristic being the number of misplaced tiles.

b) Prove that A* is admissible.

a) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 10.

b) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 16.

8. a) Consider the following game tree.

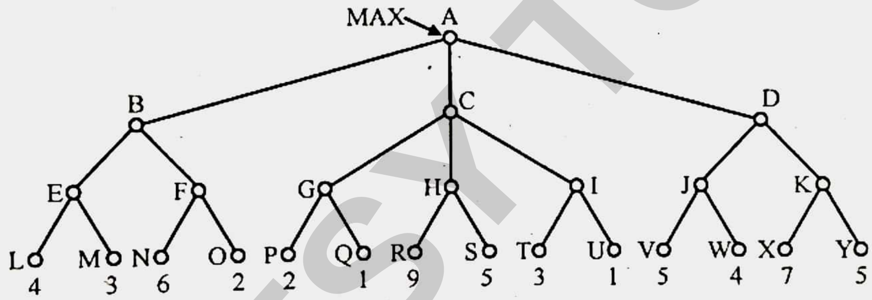


Fig: 1

(i) Using MINIMAX procedure, determine what moves should be chosen by the maximizer in his first turn.

(ii) Execute Alpha-Beta pruning on the above game tree. How many terminal nodes are examined? For each cutoff specify whether it is an Alpha-cutoff or Beta-cutoff.

b) Justify each of the following statements:

(i) BFS is a special case of Uniform-Cost search

(ii) Uniform-Cost search is a special case of A* search

a) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 17.

b) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 7(a) & (c).

9. a) Convert the following sentences into first order predicate logic:

(i) Everyone likes Ram

(ii) No one is perfect

(iii) Someone ate everything

iv) All basketball players are tall.

b) An admission committee for a college is trying to determine the probability that an admitted candidate is really qualified. The relevant probabilities are given in the Bayes network shown below. Calculate $P(A|D)$.

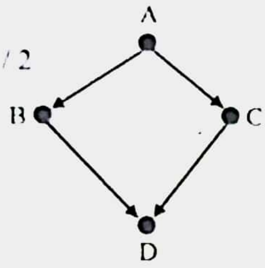
$P(A) = 1/2$

$P(B|A) = 1$

$P(B|\neg A) = 1/2$

$P(C|A) = 1$

$P(C|\neg A) = 1/2$



$P(D|B, C) = 1$

$P(D|B, \neg C) = 1/2$

$P(D|\neg B, C) = 1/2$

$P(D|\neg B, \neg C) = 0$

A = applicant is qualified.
B = applicant has high grade point average.
C = applicant has excellent recommendations.
D = applicant is admitted.

c) Compare and Contrast between

(i) Forward and Backward reasoning

(ii) Inheritable knowledge and inferential knowledge

a) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 9.

b) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 9(c).

c) i) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 3(b)(iii).

ii) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 12(b).

10. a) Translate the following into clausal form:

$$(\forall x)\Big(P(x) \rightarrow \big((\forall y)\big(P(y) \rightarrow P(f(x, y))\big) \wedge \sim (\forall y)\big(Q(x, y) \rightarrow P(y)\big)\big)\Big)$$

b) Given the following text "Everyone who enters in a theatre has bought a ticket. Person who does not have money can't buy ticket. Vinod enters a theatre. "Prove by resolution that "Vinod buys a ticket".

c) Write a program in PROLOG or LISP clause for having DOUBLE (L, LL). Each element in the list L appears twice in the list LL. For example DOUBLE ([1, 2], [1, 1, 2, 2]) is true.

a) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 2.

b) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 5.

c) OUT OF SYLLABUS

11. Write short notes on any three of the following:

a) Genetic Algorithm

b) Semantic net

c) Neural Network

d) Fuzzy set

e) Simulated Annealing

a) See Topic: LEARNING, Long Answer Type Question No. 2(a).
b) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 17(a).
c) See Topic: LEARNING, Long Answer Type Question No. 2(b).
d) See Topic: PROBALISTIC REASONING, Long Answer Type Question No. 4(c).
e) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 23(b).

QUESTION 2016

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for the following:

i) What is meant by simulated annealing in artificial intelligence?
- ✓a) Returns an optimal solution when there is a proper cooling schedule
- b) Returns an optimal solution when there is no proper cooling schedule
- c) It will not return an optimal solution when there is a proper cooling schedule
- d) None of the mentioned

ii) A* algorithm is based on
- a) Breadth-first-search
- b) Depth-first-search
- ✓c) Best-first-search
- d) Hill climbing

iii) Which search agent operates by interleaving computation and action?
- a) Offline search
- ✓b) Online search
- c) Breadth-first search
- d) Depth-first search

iv) How the new states are generated in genetic algorithm?
- a) Composition
- b) Mutation
- c) Cross-over
- ✓d) Both (b) and (c)

v) Which search method takes less memory?
- ✓a) Depth-First Search
- b) Breadth-First search
- c) Both (a) and (b)
- d) Linear Search

vi) How do you represent "all dogs have tails"?
- ✓a) $v_x : \mathrm{dog}(x) \to$ has tail (x)
- b) $v_x : \mathrm{dog}(x) \to$ has tail (y)
- c) $v_x : \mathrm{dog}(y) \to$ has tail (x)
- d) $v_x : \mathrm{dog}(x) \to$ has $\to$ tail (x)

vii) Which condition is used to cease the growth of forward chaining?

a) Atomic sentences
b) Complex sentences
✓c) No further inference
d) All of the mentioned

viii) Which is the most straight forward approach for planning algorithm?

a) Best-first search
✓b) State-space search
c) Depth-first search
d) Hill-climbing search

ix) Fuzzy logic is a form of

a) Two-valued logic
b) Crisp set logic
✓c) Many-valued logic
d) Binary set logic

x) The truth values of traditional set theory is ………… and that fuzzy set is …………

✓a) Either 0 or 1, between 0 & 1
b) Between 0 and 1, either 0 or 1
c) Between 0 and 1, between 0 and 1
d) Either 0 or 1, either 0 or 1

Group – B

(Short Answer Type Questions)

2. What do you mean by contradiction and contingency? Explain semantic network with proper example.

1st Part: See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 12.
2nd Part: See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 1.

3. What is an agent? Describe various agent types.

See Topic: INTELLIGENT AGENT, Short Answer Type Question No. 1(b).

4. A problem solving search can proceed either in the forward or the backward direction. Justify.

See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 6.

5. Write a prolog program to find out the Factorial of a number.

OUT OF SYLLABUS

6. Distinguish between Declarative and Procedural Knowledge. What is a production system?

See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 4.
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 1.

Group – C
(Long Answer Type Questions)

7. a) What do you mean by consistency of a heuristic?
b) Compare Hill climbing and Best First search.
c) Consider the following arrangement and solve the problem using A* search. Define the State space, write the operations, define the heuristic and also find whether this heuristic is admissible or not. Also show the solution.

Initial State:

2	8	3
1	6	4
7		5

Final State:

1	2	3
8		4
7	6	5

a) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 3(a) (2nd Part).
b) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 2.
c) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 8.

8. By using the predicate logic principles prove that "Marcus hated Caeser". Given below are the list of statements.
Marcus was a man.
Marcus was Pompeian.
All Pompeians were Romans.
Caeser was a ruler.
All Romans were either loyal to Caeser or hated him.
Everyone is loyal to someone.
People only try to assassinate rulers they are not loyal to.
Marcus tried to assassinate Caeser.
Explain the answer thoroughly.
See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 12.

9. a) What is percept sequence?
b) What is agent system?
c) Write down the advantages and disadvantages of Genetic Algorithm.
d) Briefly discuss combinatorial explosion.

a) & b) See Topic: INTELLIGENT AGENT, Short Answer Type Question No. 4(a) & (b).
c) See Topic: LEARNING, Short Answer Type Question No. 1.
d) See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 2.

10. a) When does BFS give optimal solution?
b) What are the three major problems of hill-climbing technique?
c) You are given two jugs, a 4-gallon one and a 3-gallon one. Neither have any 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? Give the state-space diagram, describe the production rules and give a possible solution.
a) See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 12.
b) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 9(a).
c) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 1(b).

11. a) What is a tautology? Explain with an example.
b) Write the predicate logic representations for the following sentences:
(i) If it is bird, it can fly
(ii) Every father is parent
(iii) Every man has beaten the thief
(iv) Every person in the party loves every child.
c) What is horn clause? Show that p → q is a horn clause.
a) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 8(i).
b) i) & ii) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 13.
iii) & iv) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 4(i) & (ii).
c) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 2(a) & (b).

QUESTION 2017

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternatives for any ten of the following:

i) Frame is a collection of
✓a) Slots b) Filler
c) Resolution d) Knowledge

ii) A Baysian network is a
a) tree✓ b) directed graph
c) undirected graph d) none of these

iii) Horn clause is a clause with .... positive literals
✓a) at most one
b) at most two
c) at least one
d) at most four

iv) An algorithm that gives optimal solution is
a) Hill climbing
b) BFS
c) Blind search
✓d) A*

v) Agents are
a) autonomous
b) adaptive
✓c) both (a) and (b)
d) none of these

vi) Uninformed search is also known as
a) brute force search
b) hill climbing search
c) worst case search
✓d) blind search

vii) The process of eliminating existential quantifiers is known as
a) Resolution
✓b) Skolemisation
c) Unification
d) none of these

viii) The rule used to charge weight in Neural Network (NN) is
a) Kirchoff's rule
✓b) Hebb's rule
c) Boehm's rule
d) none of these

ix) Inheritable knowledge is best represented by
✓a) semantic net
b) FOPL
c) database
d) none of these

x) MYCIN is an example of
✓a) expert system
b) knowledge base
c) conceptual graph
d) semantic net

xi) Minimax algorithm search process obeys
a) Breadth first search fashion
✓b) Depth first search fashion
c) Best first search fashion
d) Blind search fashion

xii) Depth first search procedure uses
a) AND graph
b) OR graph
c) AND-OR graph
✓d) none of these

Group – B
(Short Answer Type Questions)

2. Write a program in PROLOG of LISP to find GCD of N number.
OUT OF SYLLABUS

3. What is an agent in AI? What are the types of agent? Discuss about environment for agent.
See Topic: INTELLIGENT AGENT, Short Answer Type Question No. 1(b).

4. What is blind search technique? Explain with examples.
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 14.

5. What is tautology? Prove that $\left(\left(\left(P \rightarrow Q\right) \rightarrow P\right) \rightarrow P\right)$ is a tautology. What are Modus Ponens and Modus Tollens?
See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 14.

6. What do you mean by natural language processing (NLP)? What is parsing in NLP? What are the types of parsing? Draw the parsed tree of the sentence "The white dog crossed the road".
See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 15.

Group – C
(Long Answer Type Questions)

7. a) Is BFS identical to uniform cost search? Justify your answer.
b) Show that if a heuristic is consistent then f(n) is monotonically non decreasing along any path.
c) Consider the following arrangement and solve the problem using A* search. Define the State space, write the operations, define the heuristic and also find whether this heuristic is admissible or not. Also show the solution.

Initial State:

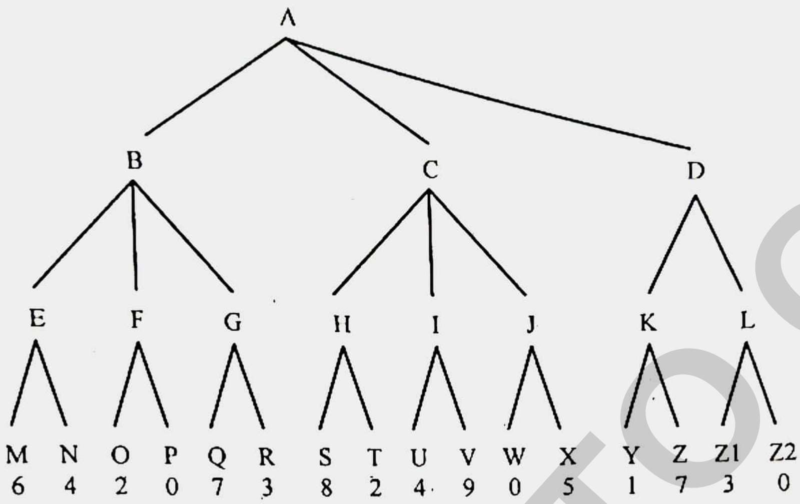
2	8	3
1	6	4
7		5

Final state:

1	2	3
8		4
7	6	5

a) See Topic: **PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 7(a).**
b) See Topic: **PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 15.**
c) See Topic: **PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 8.**

8. a) Consider the following game tree where the evaluation function values for winning is given at leaf nodes. Assume that the game tree is opened by the maximizer.



i) Applying minimax algorithm determine which nodes the maximizer and minimizer would select in their turns.

ii) How many nodes will be pruned using alpha-beta pruning?

b) Describe the fuzzy set operations like: union, intersection and complement.

c) What is clause?

a) See Topic: **PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 6.**
b) See Topic: **PROBABILISTIC REASONING, Long Answer Type Question No. 3.**
c) See Topic: **KNOWLEDGE & REASONING, Short Answer Type Question No. 16.**

9. a) Write the advantages of Genetic algorithm.
b) Describe goal based agent system.
c) What do you mean by a table driven agent? What is the problem of this agent?
d) Briefly discuss combinatorial explosion.

a) See Topic: **LEARNING, Short Answer Type Question No. 1.**
b) See Topic: **INTELLIGENT AGENT, Long Answer Type Question No. 2.a).**
c) See Topic: **INTELLIGENT AGENT, Long Answer Type Question No. 2.b).**
d) See Topic: **PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 2.**

10. a) Is uniform cost search a special case of Best First Search? Justify your answer.
b) Describe local beam search.

c) Three missionaries and three cannibals are standing at the left bank of a river. There is a boat having a capacity of taking two people and it can be driven by a missionary or a cannibal. If the number of missionaries is less than the number of cannibals at any bank, then cannibal will eat missionary. How is it possible for all the missionaries and cannibals to cross the river so that no missionary is getting eaten? Describe the state space of the problem. Describe the production rules for solving the problem. Show one solution of the problem.

See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 18.

11. a) Prove that PV ~ P is a Tautology.

b) What is contradiction?

c) Represent the following sentences using Predicate logic or using FOPL or find the wffs of the following sentences and draw the conclusion as required:

(i) X is an Indian, (ii) Y is an Indian, (iii) X is a leader, (iv) Every Indian is a man, (v) Every one is loyal to someone, (vi) Every man is either loyal to a leader or hate a leader, (vii) Man tries to assassinate a leader if he is not loyal to him, (viii) Y assassinated X.

Now conclude that Y hated X.

a) & b) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 17.

c) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 5.

QUESTION 2018

Group – A

(Multiple Choice Type Questions)

1. Choose the correct alternatives of the following:

i) Frame is a collection of

✓a) Slots | b) Fillers
c) Resolutions | d) Knowledges

ii) A Baysian network is a

a) tree | ✓b) directed graph
c) undirected graph | d) None of these

iii) Horn clause is a clause with ____________ positive literals.

✓a) at most one | b) at most two
c) at least one | d) at most four

iv) An algorithm that gives optimal solution is
a) Hill climbing
b) BFS
c) Blind search
✓d) A*

v) Uniformed search is also known as:
a) Brute force search
b) Hill climbing search
c) Worst case search
✓d) Blind search

vi) The process of eliminating existential quantifiers is known as
a) Resolution
✓b) Skolemisation
c) Unification
d) None of these

vii) The rule used to change weight in Neural Network (NN) is
a) Kirchoff's rule
✓b) Hebb's rule
c) Boehm's rule
d) None of these

viii) Inheritable knowledge is best represented by:
✓a) Semantic net
b) FOPL
c) Database
d) None of these

ix) Minimax algorithm search process obeys
a) breadth first search fashion
✓b) depth first search fashion
c) best first search fashion
d) blind search fashion

x) Depth first search producers uses
a) AND graph
b) OR graph
c) AND-OR graph
✓d) None of these

Group – B
(Short Answer Type Questions)

2. Compare and contrast Best-First and Hill climbing search.
See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 2.

3. What is an agent in AI? What are the types of agent? Discuss about environment for agent
See Topic: INTELLIGENT AGENT, Short Answer Type Question No. 1(b).

4. What is blind search technique? Explain with examples.
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 14.

5. What is tautology? Prove that $(((P \to Q) \to P) \to P)$ is a tautology. What are Modus Ponens and Modus Tollens?
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 14.

6. Write iterative deepening algorithm with example.
See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 16.

Group – C
(Long Answer Type Questions)

7. Suppose you have the following search space:

State	Next	Cost
A	B	4
A	C	1
B	D	3
B	E	8
C	C	0
C	D	2
C	F	6
D	C	2
D	E	4
E	G	2
F	G	8

Assume that the initial state is A and the goal state is G. Show how each of the following search strategies would create a search tree to find a path from the initial state to the goal state and the cost of the solution:

i) Breadth-first search
ii) Depth-first search
iii) Iterative deepening search

See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 13.

8. a) What do you mean by constraint satisfaction problem? Solve the following cryptography problem using constraint satisfaction search:

```
  S E N D
 M O R E
=========
M O N E Y
```

b) Write a program in PROLOG to compute the GCD of two numbers.

a) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 19.
OUT OF SYLLABUS

9. a) What is 'Horn Clause'?

b) What is Skolemisation?

c) Given the following text 'Everyone who enters in a theatre has to buy a ticket. Person who doesn't have money can't buy a ticket. Vinod enters a theatre.' Prove by resolution that 'Vinod has money'.

d) With the help of semantic net, prove that Sourav is 6 feet tall and he is taller than Sachin.

a) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 2(a).

b) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 14(a).

c) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 14(b).

d) See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 1.

10. a) Briefly explain the steps of Natural language Processing.

b) Generate the parse tree for the sentence 'The boy went to School'.

c) Explain AO* algorithm with a suitable example.

See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 15.

11. Write short notes any three of the following:

a) Conceptual graph

b) Alpha-Beta pruning in min-max search

c) A* search

d) The steps for transforming into Clause Form

e) Expert Systems

a) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 17(b).

b) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 17(c).

c) See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 23(d).

d) See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 17(d).

e) See Topic: EXPERT SYSTEMS, Long Answer Type Question No. 1.

QUESTION 2019

Group – A

(Multiple Choice Type Questions)

1. Choose the correct alternatives of the following:

i) Inheritable knowledge is best represented by

✓a) semantic net

b) FOPL

c) database

d) none of these

ii) Artificial Intelligence is
- a) programming with intelligence
- b) putting more memory to a computer
- ✓c) making machine intelligent
- d) playing games

iii) Which element in agent is used for selecting external actions?
- a) Perceive
- b) Performance
- c) Learning
- ✓d) Actuator

iv) When do we call the states are safely explorable?
- a) A goal state is unreachable from any state
- b) A goal state is denied access
- ✓c) A goal state is reachable from every state
- d) None of the mentioned

v) An algorithm is complete if
- ✓a) it terminates with a solution when one exists
- b) it starts with a solution
- c) it does not terminate with a solution
- d) it has a loop

vi) Heuristic function $h(n)$ is defined as the
- a) Lowest path cost
- b) Cheapest path from root to goal node
- ✓c) Estimated cost of cheapest path from root to goal node
- d) Average path cost

vii) In many problems the path to goal is irrelevant, this class of problems can be solved using
- a) informed search techniques
- b) uniformed search techniques
- ✓c) local search techniques
- d) only (a) and (b)

viii) Which type of mathematical problems are defined as a set of objects whose state must satisfy a number of constraints or limitations?
- ✓a) Constraints satisfaction Problems
- b) Uniformed Search Problems
- c) Local Search Problems
- d) Only (a) and (b)

ix) $PV \sim P$ is a
- ✓a) tautology
- b) contradiction
- c) all false proposition
- d) none of these

x) Resolution system can be used for
- a) question answering
- b) theorem proving
- ✓c) both (a) and (b)
- d) none of these

xi) What is the full form of STRIPS?

a) STandard Research Institute Problem Solver

✓b) STanford Research Institute Problem Solver

c) STdard Refinement of Information for Problem Solving

d) None of these

Group – B

(Short Answer Type Questions)

2. Explain the working principle of simulated annealing algorithm.

See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 10.

3. Describe different components of AI.

See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 7.

4. Briefly discuss combinatorial explosion.

See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 2.

5. Describe depth limited search.

See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 17.

6. Describe neural network based learning.

See Topic: LEARNING, Short Answer Type Question No. 2.

Group – C

(Long Answer Type Questions)

7. Describe DFS. Does DFS always ensure completeness and optimality? Justify. Consider the following arrangement and solve the 4 puzzle problem applying DFS.

2	3
1	

Start State

1	2
3	

Goal State

See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 20.

8. Describe local beam search. Describe minimax procedure. Describe alpha-beta pruning procedure.

1st & 2nd Part: See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 21.

3rd Part: See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 16.

9. Convert the following into clausal form and explain each step

$(\forall x)\{P(x) \Rightarrow \{(\forall y)[P(y) \Rightarrow P(f(x, y))] \wedge \sim (\forall y)[Q(x, y) \Rightarrow P(y)]\}\}$

See Topic: KNOWLEDGE & REASONING, Long Answer Type Question No. 2.

10. Describe state space search. What are the advantages of BFS over DFS and vice-versa? Does uniform cost search ensure completeness and optimality? Justify.

See Topic: PROBLEM SOLVING & SEARCHING, Long Answer Type Question No. 22.

11. Briefly describe the issues in knowledge representation. Given

P (A|B) =0.20, P(A |~B)=0.65, P(~A|B)=0.12, P(~A|~B)=0.03. Now find P (~A|B). Does BFS ensure completeness and optimality? Justify.

1st part: See Topic: KNOWLEDGE & REASONING, Short Answer Type Question No. 18.

2nd part: Data is not sufficient to solve this problem.

3rd part: See Topic: PROBLEM SOLVING & SEARCHING, Short Answer Type Question No. 18.