DS-5-29-11-23: 40(202)+26(203)=66

[Time: 3 hrs] **[Total Marks: 80]**

Note : 1. Question 1 is compulsory
2. Answer any three out of remaining question
3. assume suitable data where required

Q1	Attempt any **4**	**[20]**
[A]	Explain problems faced by Hill Climbing algorithm.	**[05]**
[B]	Write PEAS descriptor for Shopping for used AI books on the Internet.	**[05]**
[C]	Write a program in Prolog to create a family tree.	**[05]**
[D]	Draw and explain architecture of Expert System.	**[05]**
[E]	Discuss different types of environments for Intelligent Agents.	**[05]**
Q2		**[20]**
[A]	Explain A algorithm with an example. Also discuss its performance.	**[10]**
[B]	What are the different types of agents? Explain Goal based agent with a diagram.	**[10]**
Q3		**[20]**
[A]	What is formulation of a problem? Formulate 8-Puzzle problem in terms of following components: initial state, actions, successor function, goal test and path cost.	**[10]**
[B]	Define chromosome, selection, fitness function, cross over and mutation as used in genetic algorithm. Explain the working of genetic algorithm.	**[10]**
Q4		**[20]**
[A]	"As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen." Prove that "Robert is criminal." Using forward and backward Chaining.	**[10]**
[B]	What is planning in AI? Explain partial order planning with an example.	**[10]**

21DE865CC12274237C4C708323D59E32

Q5 **[20]**

[A] Write first order statements for following **[10]**

(i) Every dolphin is Mammal

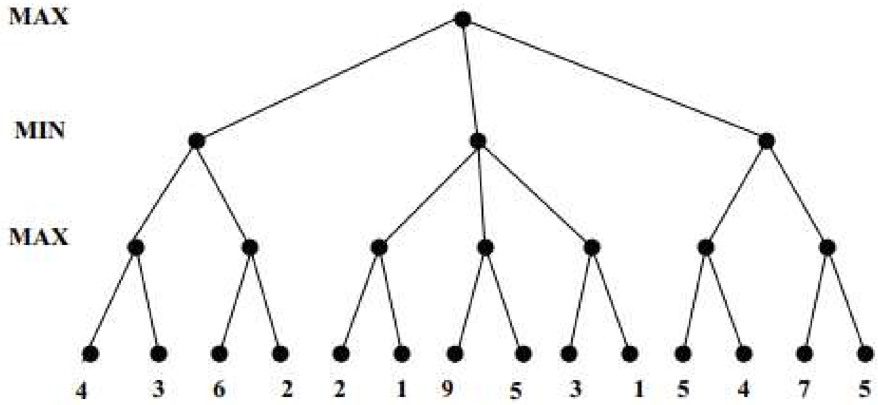
(ii) No purple mushroom is poisonous.

(iii) Every gardener loves sun.

(iv) You can fool someone all the time.

(v) All Romans were either loyal to ceaser or hated him.

[B] Explain Alpha-beta pruning algorithm. Apply alpha beta pruning on the following example considering the first node as MAX. **[10]**



Q6 **[20]**

[A] Explain Bayesian Belief Networks with an example. **[10]**

[B] Explain different types of learning in AI. **[10]**

****

21DE865CC12274237C4C708323D59E32

TE SEM V Subject: CSC503 Artificial Intelligence

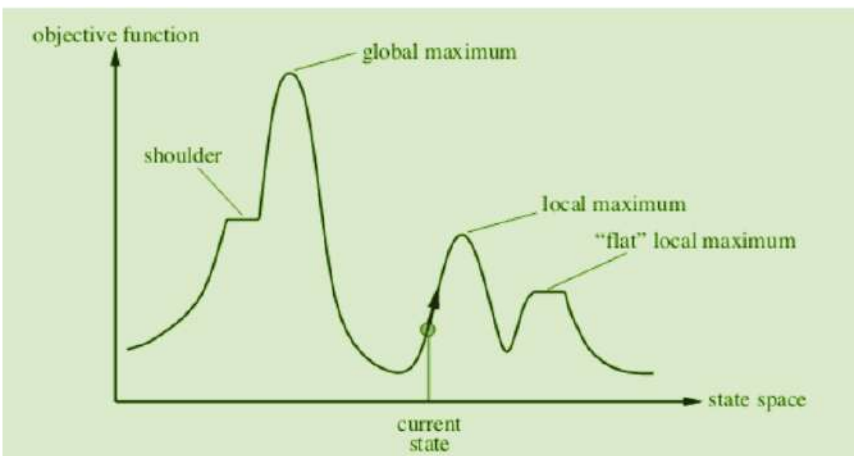
University Exam
December 2023
Artificial Intelligence
Solution

Question 1 is compulsory
Answer any 3 out of remaining

Q.1 Attempt any 4 **[20]**

A. Explain problems faced by Hill Climbing Algorithm. **[05]**

Ans:



Hill climbing cannot reach the optimal/best state(global maximum) if it enters any of the following regions :

Local maximum: At a local maximum all neighboring states have a value that is worse than the current state. Since hill-climbing uses a greedy approach, it will not move to the worse state and terminate itself. The process will end even though a better solution may exist.

To overcome the local maximum problem: Utilize the backtracking technique. Maintain a list of visited states. If the search reaches an undesirable state, it can backtrack to the previous configuration and explore a new path.

Plateau: On the plateau, all neighbors have the same value. Hence, it is not possible to select the best direction.

To overcome plateaus: Make a big jump. Randomly select a state far away from the current state. Chances are that we will land in a non-plateau region.

Ridge: Any point on a ridge can look like a peak because movement in all possible directions is downward. Hence the algorithm stops when it reaches this state.

To overcome Ridge: In this kind of obstacle, use two or more rules before testing. It implies moving in several directions at once.

B. Write PEAS descriptor for Shopping for used AI books on the internet. [05]

Ans:

PEAS description

Performance measure: price, quality, appropriateness, efficiency

Environment: current and future WWW sites, vendors, shippers

Actuators: display to user, follow URL, fill in form

Sensors: HTML pages (text, graphics, scripts)

C. Write a program in prolog to create a family tree. [05]

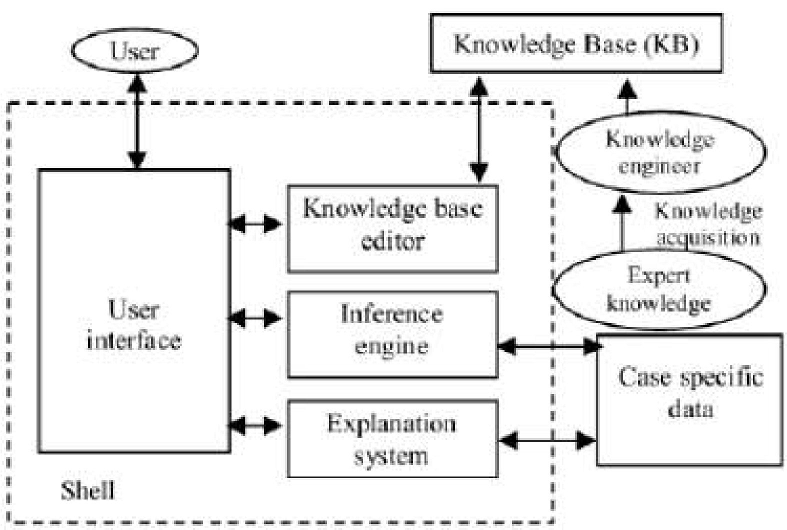
Ans:

```
female(pammi).
female(lizza).
female(patty).
female(anny).
male(jimmy).
male(bobby).
male(tomy).
male(pitter).
parent(pammi,bobby).
parent(tomy,bobby).
parent(tomy,lizza).
parent(bobby,anny).
```

```
parent(bobby,patty).
parent(patty,jimmy).
parent(bobby,pitter).
parent(pitter,jimmy).
mother(X,Y):- parent(X,Y),female(X).
father(X,Y):- parent(X,Y),male(X).
haschild(X):- parent(X,_).
sister(X,Y):- parent(Z,X),parent(Z,Y),female(X),X\==Y.
brother(X,Y):-parent(Z,X),parent(Z,Y),male(X),X\==Y.
Input:
parent(X,jimmy).
mother(X,Y).
haschild(X).
sister(X,Y).
```

D. Draw and explain architecture of Expert System. **[05]**

Ans:



- **Knowledge Base –**

  The knowledge base represents facts and rules. It consists of knowledge in a particular domain as well as rules to solve a problem, procedures and intrinsic data relevant to the domain.

- **Inference Engine –**
  The function of the inference engine is to fetch the relevant knowledge from the knowledge base, interpret it and to find a solution relevant to the user's problem. The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts. Inference engines can also include an explanation and debugging abilities.
- **Knowledge Acquisition and Learning Module –**
  The function of this component is to allow the expert system to acquire more and more knowledge from various sources and store it in the knowledge base.
- **User Interface –**
  This module makes it possible for a non-expert user to interact with the expert system and find a solution to the problem.
- **Explanation Module –**
  This module helps the expert system to give the user an explanation about how the expert system reached a particular conclusion.

E. Discuss different types of environments for Intelligent Agents. [05]

Ans:

1. Fully Observable vs Partially Observable

- When an agent sensor is capable to sense or access the complete state of an agent at each point in time, it is said to be a fully observable environment else it is partially observable.
- Maintaining a fully observable environment is easy as there is no need to keep track of the history of the surrounding.
- An environment is called **unobservable** when the agent has no sensors in all environments.

- **Examples:**
    - **Chess –** the board is fully observable, and so are the opponent's moves.
    - **Driving –** the environment is partially observable because what's around the corner is not known.

2. Deterministic vs Stochastic

- When a uniqueness in the agent's current state completely determines the next state of the agent, the environment is said to be deterministic.
- The stochastic environment is random in nature which is not unique and cannot be completely determined by the agent.
- **Examples:**
    - **Chess –** there would be only a few possible moves for a coin at the current state and these moves can be determined.
    - **Self-Driving Cars-** the actions of a self-driving car are not unique, it varies time to time.

3. Competitive vs Collaborative

- An agent is said to be in a competitive environment when it competes against another agent to optimize the output.
- The game of chess is competitive as the agents compete with each other to win the game which is the output.
- An agent is said to be in a collaborative environment when multiple agents cooperate to produce the desired output.
- When multiple self-driving cars are found on the roads, they cooperate with each other to avoid collisions and reach their destination which is the output desired.

4. Single-agent vs Multi-agent

- An environment consisting of only one agent is said to be a single-agent environment.
- A person left alone in a maze is an example of the single-agent system.
- An environment involving more than one agent is a multi-agent environment.
- The game of football is multi-agent as it involves 11 players in each team.

5. Dynamic vs Static

- An environment that keeps constantly changing itself when the agent is up with some action is said to be dynamic.
- A roller coaster ride is dynamic as it is set in motion and the environment keeps changing every instant.
- An idle environment with no change in its state is called a static environment.
- An empty house is static as there's no change in the surroundings when an agent enters.

6. Discrete vs Continuous

- If an environment consists of a finite number of actions that can be deliberated in the environment to obtain the output, it is said to be a discrete environment.
- The game of chess is discrete as it has only a finite number of moves. The number of moves might vary with every game, but still, it's finite.
- The environment in which the actions are performed cannot be numbered i.e. is not discrete, is said to be continuous.
- Self-driving cars are an example of continuous environments as their actions are driving, parking, etc. which cannot be numbered.

7.Episodic vs Sequential

- In **an Episodic task environment**, each of the agent's actions is divided into atomic incidents or episodes. There is no dependency between current and previous incidents. In each incident, an agent receives input from the environment and then performs the corresponding action.

- **Example:** Consider an example of **Pick and Place robot**, which is used to detect defective parts from the conveyor belts. Here, every time robot(agent) will make the decision on the current part i.e. there is no dependency between current and previous decisions.
- In a **Sequential environment**, the previous decisions can affect all future decisions. The next action of the agent depends on what action he has taken previously and what action he is supposed to take in the future.
- **Example:**
    - **Checkers-** Where the previous move can affect all the following moves.

8. Known vs Unknown

- In a known environment, the output for all probable actions is given. Obviously, in case of unknown environment, for an agent to make a decision, it has to gain knowledge about how the environment works.

Q.2 **[20]**

A. Explain A* algorithm with an example. Also discuss its performance. **[10]**

Ans:

A* search is the most commonly known form of best-first search. It uses the heuristic function h(n), and costs to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solves the problem efficiently. A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal results faster. A* algorithm is similar to UCS except that it uses g(n)+h(n) instead of g(n).

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as follows, and this sum is called a **fitness number**.

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

f(n): estimated cost of the cheapest solution

g(n): cost to reach node n from start state

h(n): cost to reach from node n to goal node

Algorithm of A* search:

Step1: Place the starting node in the OPEN list.

Step 2: Check if the OPEN list is empty or not, if the list is empty then return failure and stops.

Step 3: Select the node from the OPEN list which has the smallest value of evaluation function (g+h), if node n is goal node then return success and stop, otherwise

Step 4: Expand node n and generate all of its successors, and put n into the closed list. For each successor n', check whether n' is already in the OPEN or CLOSED list, if not then compute evaluation function for n' and place into Open list.

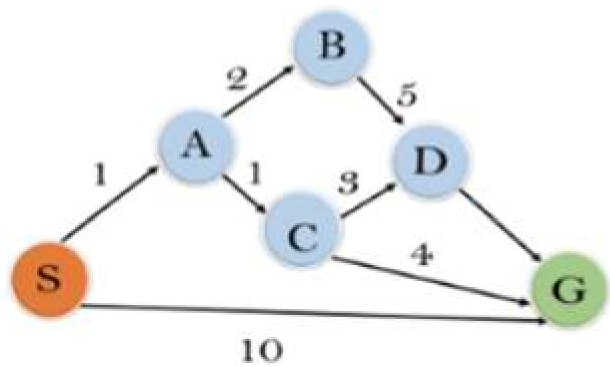
Step 5: Else if node n' is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to **Step 2**.

Example:

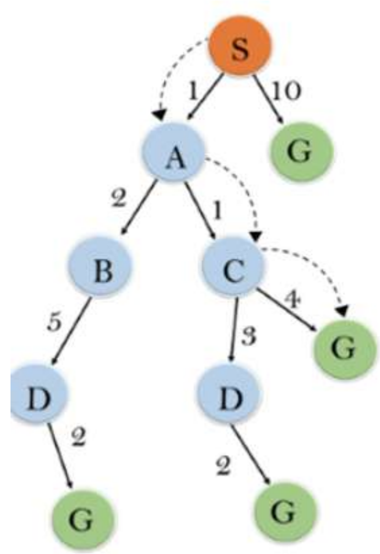
In this example, we will traverse the given graph using the A* algorithm. The heuristic value of all states is given in the below table so we will calculate the f(n) of each state using the formula f(n)= g(n) + h(n), where g(n) is the cost to reach any node from start state.

Here we will use OPEN and CLOSED list.



State	h(n)
S	5
A	3
B	4
C	2
D	6
G	0

Solution:



Complete: A* algorithm is complete as long as:

- Branching factor is finite.

- Cost at every action is fixed.

Optimal: A* search algorithm is optimal if it follows below two conditions:

- **Admissible:** the first condition requires for optimality is that h(n) should be an admissible heuristic for A* tree search. An admissible heuristic is optimistic in nature.
- **Consistency:** Second required condition is consistency for only A* graph-search.

If the heuristic function is admissible, then A* tree search will always find the least cost path.

Time Complexity: The time complexity of A* search algorithm depends on heuristic function, and the number of nodes expanded is exponential to the depth of solution d. So the time complexity is O(b^d), where b is the branching factor.

Space Complexity: The space complexity of A* search algorithm is **O(b^d)**

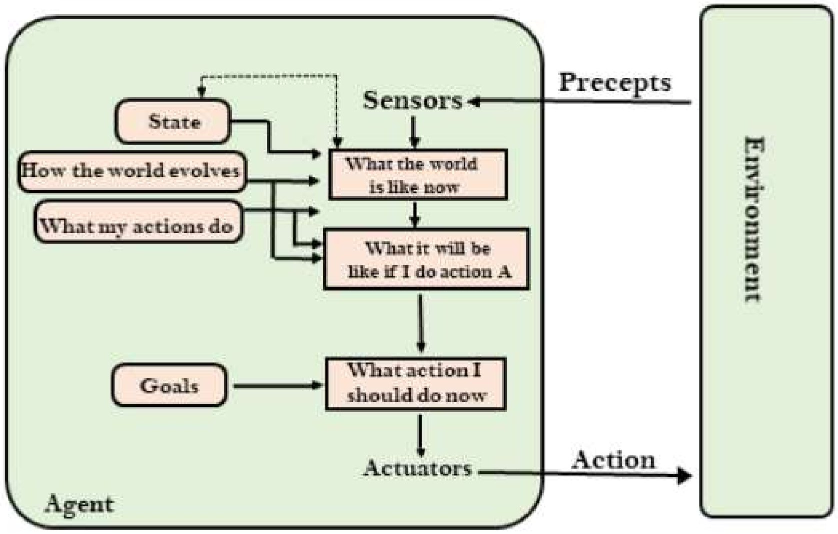
B. What are the different types of agents? Explain Goal based agent with a Diagram. **[10]**

Ans:

Agents can be grouped into five classes based on their degree of perceived intelligence and capability. All these agents can improve their performance and generate better action over the time. These are given below:

- Simple Reflex Agent
- Model-based reflex agent
- Goal-based agents
- Utility-based agent
- Learning agent

Goal-based agents



The knowledge of the current state environment is not always sufficient to decide for an agent what to do.

- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenarios are called searching and planning, which makes an agent proactive.

Q.3 **[20]**

A. What is the formulation of a problem? Formulate 8-puzzle problem in terms of following components: initial state, actions, successor function, goal test and path cost. **[10]**

Ans:

Components to formulate the associated problem:

- **Initial State:** This state requires an initial state for the problem which starts the AI agent towards a specified goal. In this state new methods also initialize problem domain solving by a specific class.
- **Action:** This stage of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this stage.
- **Transition:** This stage of problem formulation integrates the actual action done by the previous action stage and collects the final stage to forward it to their next stage.
- **Goal test:** This stage determines that the specified goal achieved by the integrated transition model or not, whenever the goal achieves stop the action and forward into the next stage to determines the cost to achieve the goal.
- **Path costing:** This component of problem-solving numerical assigned what will be the cost to achieve the goal. It requires all hardware software and human working cost.

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

Initial state: Any state can be designated as the initial state. Note that any given goal can be reached from exactly half of the possible initial states.

Successor function: This generates the legal states that result from trying the four actions

Actions: blank moves left, right, up or down

Goal test: This checks whether the state matches the goal configuration shown in figure

Path cost: Each step costs 1, so the path cost is the number of steps in the path.

B. **Define chromosome, selection, fitness function, cross over and mutation As used in genetic algorithm. Explain the working of genetic algorithm.** **[10]**

Ans:

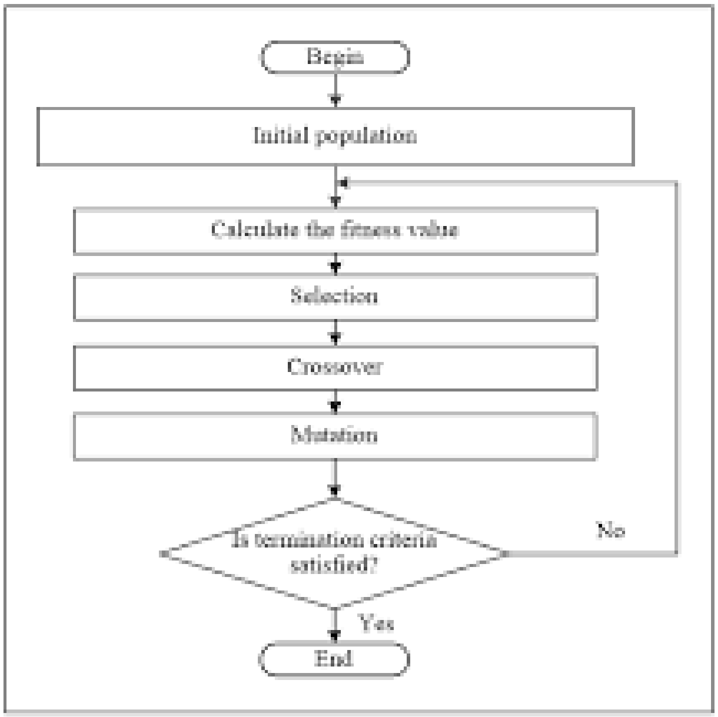
Chromosomes: A chromosome is one such solution to the given problem.

Selection: The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to successive generations.

Fitness Function − A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output. In some cases, the fitness function and the objective function may be the same, while in others it might be different based on the problem.

Crossover: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring).

Mutation: The key idea is to insert random genes in offspring to maintain the diversity in the population to avoid premature convergence.



Genetic algorithms provide a natural model for parallelism because each neuron or segment of a solution is an independent unit.

- Genetic algorithms begin with a population of candidate problem solutions.
- Candidate solutions are evaluated according to their ability to solve problem instances: only the fittest survive and combine with each other to produce the next generation of possible solutions
- In the genetic algorithm model, for example, a population of patterns represents the candidate solutions to a problem.
- As the algorithm cycles, this population of patterns "evolves" through operations which mimic reproduction, mutation, and natural selection.
- Genetic algorithms are also applied to more complex representations, including production rules, to evolve rule sets adapted to interacting with an environment.

For example, genetic programming combines and mutates fragments of computer code in an attempt.

- Like neural networks, genetic algorithms are based on a biological metaphor: they view learning as a competition among a population of evolving candidate problem solutions.

- A "fitness" function evaluates each solution to decide whether it will contribute to the next generation of solutions.
- Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions.

In the genetic algorithm, process is as follows:

Step 1. Determine the number of chromosomes, generation, and mutation rate and crossover rate value

Step 2. Generate chromosome-chromosome number of the population, and the initialization value of the genes chromosome-chromosome with a random value

Step 3. Process steps 4-7 until the number of generations is met

Step 4. Evaluation of fitness value of chromosomes by calculating objective function

Step 5. Chromosomes selection

Step 5. Crossover

Step 6. Mutation

Step 7. New Chromosomes (Offspring)

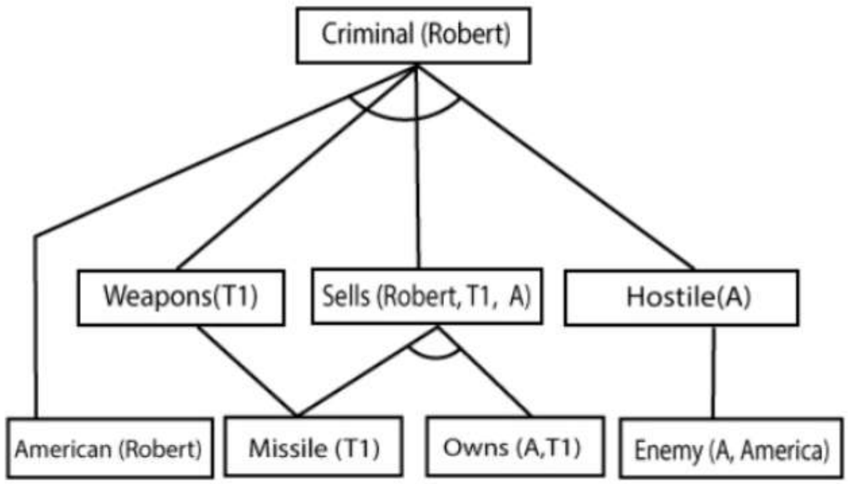
Step 8. Solution (Best Chromosomes)

Q.4 **[20]**

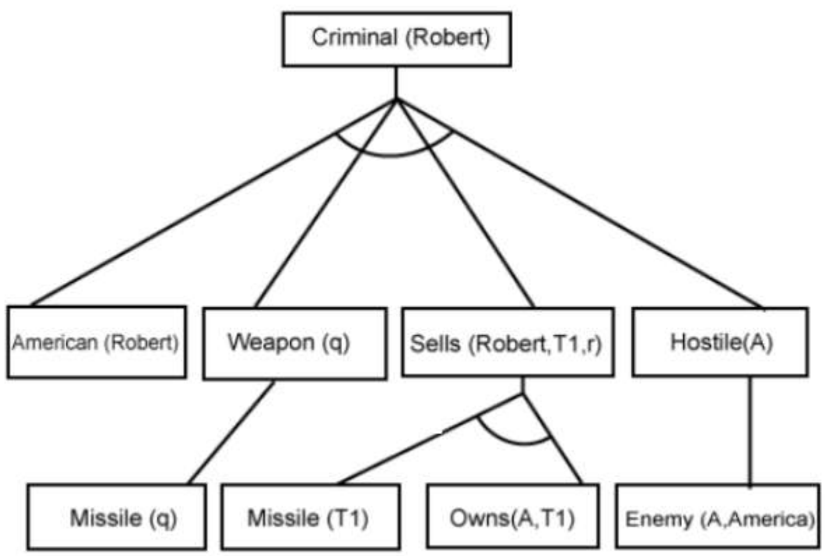
A. "As per the law, it is a crime for an American to sell weapons to hostile nations.[10]
Country A, an enemy of America, has some missiles and all the missiles were sold to it by Robert, who is an American citizen."
Prove that "Robert is criminal". Using forward and backward chaining.

Ans:

Forward Chaining (Bottom Up)



Backward Chaining(Top Down)



B. What is planning in AI?Explain partial order planning with an example. [10]

Ans:

Planning is the task of finding a procedural course of action for a declaratively described system to reach its goals while optimizing overall performance measures.

In partial ordered planning (POP), ordering of the actions is partial. Also partial ordered planning don't specify which action will come first out of the two actions which are placed.

With partial ordered planning, problem can be decomposed, so it can work well in case the environment is non-cooperative.

It combines two action sequence:

i. First branch covers left sock and left shoe.

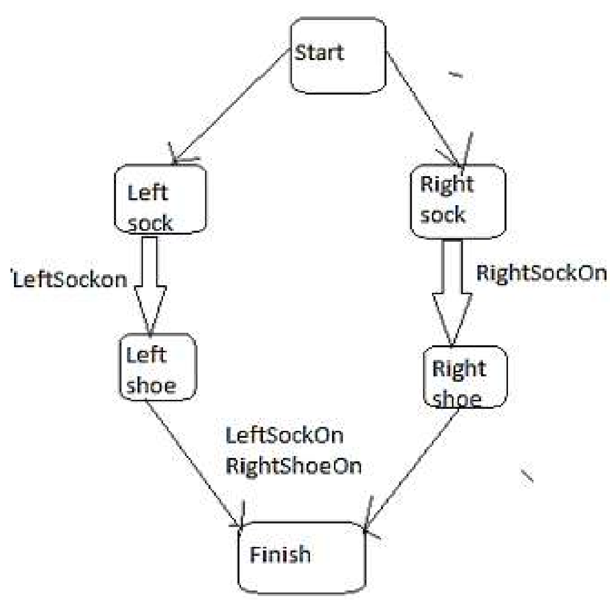
ii. In case, to wear a left shoe, wearing left sock is precondition, similarly.

iii. Second branch covers right shock and right shoe.

iv. Here, wearing a right shock is precondition for wearing the right shoe.

Once these actions are taken we achieve our goal and reach the finish state.

E.g. Partial order planning of Wearing Shoe



Pop as search problem:

Set of actions:

These are the steps of plan.

For e.g.: Set of Actions = {Start, Rightsock, Rightshoe ,Leftsock, Leftshoe, Finish}

Set of ordering constraints/preconditions:

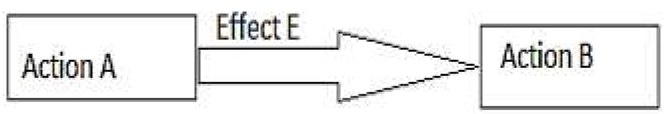
i. Preconditions are considered as ordering constraints.(i.e. without performing action "x" we cannot perform action "y")

ii. For e.g.: Set of ordering = {Right-sock <right-shoe; left-sock<left-shoe}="" that="" is="" in="" order="" to="" wear="" shoe,="" first="" we="" should="" wear="" a="" sock.<="" p="">

Set of causal links:

Action A achieves effect "E" for action B

a.



b.

i. You can understand that if you buy an apple its effect can be eating an apple and the precondition of eating an apple is cutting apple.

ii. For e.g. Set of Causal Links = {Right-sock-> Right-sock-on

Set of open preconditions:

i. Preconditions are called open if it cannot be achieved by some actions in the plan.

ii. Consistent Plan is a Solution for POP Problem

iii. A consistent Plan doesn't have cycle of constraints; it doesn't have conflicts in the causal links and doesn't have open preconditions so it can provide a solution for POP problem.

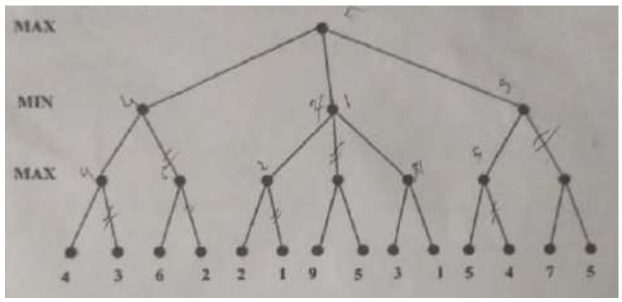
Q.5 **[20]**

A. Write first order statements for following **[10]**

1. Every dolphin is Mammal.

2. No purple mushroom is poisonous.
3. Every gardener loves sun.
4. You can fool someone all the time.
5. All Romans were either loyal to ceasar or hated him.

B. Explain Alpha-Beta pruning algorithm. Apply alpha beta pruning on the following example considering the first node as MAX. **[10]**



Q.6 **[20]**

A. Explain Bayesian Belief Networks with an example. **[10]**
Ans:
Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty. We can define a Bayesian network as:

"A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."

It is also called a Bayes network, belief network, decision network, or Bayesian model.

Bayesian networks are probabilistic, because these networks are built from a probability distribution, and also use probability theory for prediction and anomaly detection.

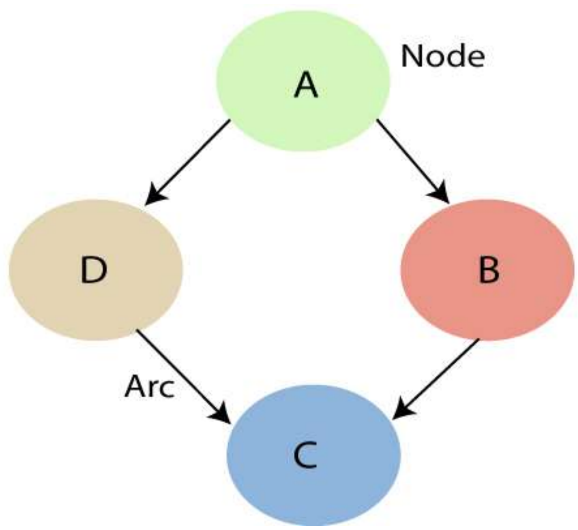
Real world applications are probabilistic in nature, and to represent the relationship between multiple events, we need a Bayesian network. It can also be used in various tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction, and decision making under uncertainty.

Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:

- ○ Directed Acyclic Graph
- ○ Table of conditional probabilities.

The generalized form of Bayesian network that represents and solve decision problems under uncertain knowledge is known as an Influence diagram.

A Bayesian network graph is made up of nodes and Arcs (directed links), where:



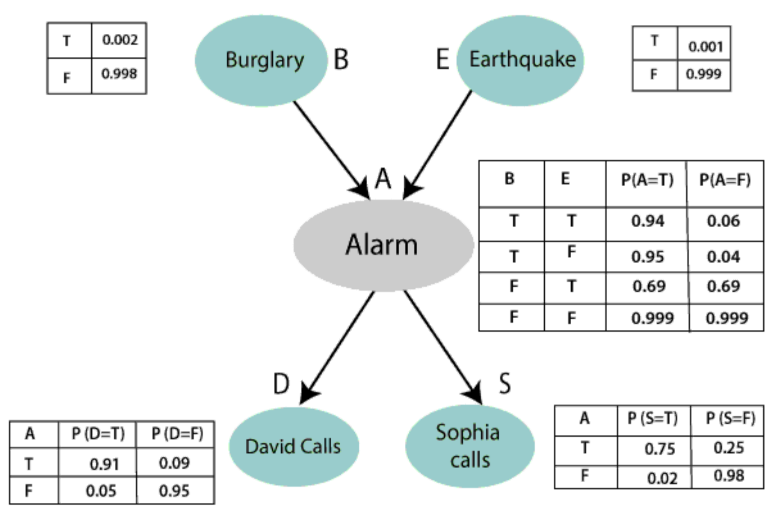
- ○ Each node corresponds to the random variables, and a variable can be continuous or discrete.
- ○ Arc or directed arrows represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.
  These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other
  - ○ In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.

- If we are considering node B, which is connected with node A by a directed arrow, then node A is called the parent of Node B.
- Node C is independent of node A.

Example: Harry installed a new burglar alarm at his home to detect burglary. The alarm reliably responds at detecting a burglary but also responds for minor earthquakes. Harry has two neighbors David and Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm. David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too. On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm. Here we would like to compute the probability of Burglary Alarm.

Problem:

Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.



From the formula of joint distribution, we can write the problem statement in the form of probability distribution:

P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E).

= 0.75* 0.91* 0.001* 0.998*0.999

= 0.00068045.

Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

B. Explain different types of learning in AI. **[10]**

Ans:

Supervised Learning

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting. Supervised learning can be further classified into two types - Regression and Classification.

Regression trains on and predicts a continuous-valued response, for example predicting real estate prices.

Classification attempts to find the appropriate class label, such as analyzing positive/negative sentiment, male and female persons, benign and malignant tumors, secure and unsecure loans etc.

In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called labeled data. The learned rule is then used to label new data with unknown outputs.

Supervised learning involves building a machine learning model that is based on labeled samples. For example, if we build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, we first need to create a database and label it. We need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new examples. There are many supervised learning algorithms such as Logistic Regression, Neural networks, Support Vector Machines (SVMs), and Naive Bayes classifiers.

Common examples of supervised learning include classifying e-mails into spam and not-spam categories, labeling webpages based on their content, and voice recognition.

Unsupervised Learning

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. There is no labeled data here.

When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. This kind of learning data is called unlabeled data.

Suppose that we have a number of data points, and we want to classify them into several groups. We may not exactly know what the criteria of classification would be. So, an unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way.

Unsupervised learning algorithms are extremely powerful tools for analyzing data and for identifying patterns and trends. They are most commonly used for clustering similar input into logical groups. Unsupervised learning algorithms include Kmeans, Random Forests, Hierarchical clustering and so on.

Semi-supervised Learning

If some learning samples are labeled, but some other are not labeled, then it is semi-supervised learning. It makes use of a large amount of unlabeled data for training and a small amount of labeled data for testing. Semi-supervised learning is applied in cases where it is expensive to acquire a fully labeled dataset while more practical to label a small subset. For example, it often requires skilled experts to label certain remote sensing images, and lots of field experiments to locate oil at a particular location, while acquiring unlabeled data is relatively easy.

Reinforcement Learning

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly. The best known instances include self-driving cars and chess master algorithm AlphaGo.