What are the major hill climbing techniques?

1. **Simple Hill Climbing:** It examines the neighbouring nodes one by one and selects the first neighbouring node which optimizes the current cost as the next node.
2. **Steepest-Ascent Hill climbing:** It first examines all the neighbouring nodes and then selects the node closest to the solution state as of the next node.
3. **Stochastic hill climbing:** It does not examine all the neighbouring nodes before deciding which node to select. It just selects a neighbouring node at random and decides (based on the amount of improvement in that neighbour) whether to move to that neighbour or to examine another.

Why BFS is slower than DFS?

- BFS uses a Queue data structure that operates on a "First In First Out" (FIFO) principle. This requires constantly adding and removing elements from the data structure, leading to higher computational overhead and slower execution.
- BFS requires storing all unvisited nodes in the queue at a given level. This can lead to significant memory consumption, especially for large graphs or deep search spaces.
- BFS explores all nodes at a given level before moving to the next level. This can lead to redundant exploration of irrelevant branches, especially when the target is located deep within the search space.

What are Expert systems? Give one example. What are the main parts of an Expert System? And, how they interact with one another. Discuss concept of uncertainty

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. Example: CaDet: It is a clinical support system that could identify cancer in its early stages in patients.

Parts of Expert System:

- **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
- **Inference Engine:** The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts.
- **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 helps 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.

Reasons for uncertainty

- **Incomplete information:** The knowledge base might not have all the relevant information for a specific case.
- **Inaccurate information:** Outdated information can affect the accuracy of conclusions.
- **Vague knowledge:** Experts themselves might have differing opinions when describing problems.

Ways to handle uncertainty:

- **Fuzzy logic:** Allows for degrees of truth instead of just true or false
- **Probabilistic reasoning:** Assigns probabilities to different solutions based on available evidence.
- **Bayesian networks:** Use conditional probabilities to represent relationships between variables and update them as new information arrives.

Explain the tic tac toe problem in artificial intelligence.

Problem Description: Tic Tac Toe is a two-player game played on a 3x3 grid. Players take turns marking squares with their symbol (X or O), aiming to get three consecutive marks in a row (horizontally, vertically, or diagonally) to win.

Challenges:

- State Space: The search space for all possible game states is vast, requiring efficient search algorithms to navigate.
- Decision Making: The AI needs to evaluate different move options, considering both immediate and long-term consequences.
- Adaptability: The AI must adapt to its opponent's playing style and unpredictable moves.

Approaches:

- Minimax Algorithm: Evaluates all possible game states and selects the move that maximizes the AI's potential while minimizing the opponent's.
- Alpha-Beta Pruning: Improves the efficiency of the minimax algorithm by pruning unnecessary branches of the search tree.
- Heuristics: Utilizes domain knowledge to guide the search process towards promising moves, reducing the search space.

What is first-order logic in artificial intelligence?

First-order-logic is an extension of propositional logic, which deals with simple statements and logical operators.

Parts of First-order-logic:

1. Syntax: Syntax represents the rules to write an expression
2. Semantics: Semantics represents techniques to evaluate an expression

Basic Elements of First Order Logic:

Element	**Example**	**Meaning**
Constant	1, 2, A, John, Mumbai, cat, ...	Values that can not be changed
Variables	x, y, z, a, b, ...	Can take up any value and can also change
Predicates	Brother, Father, >, ...	Defines a relationship between its input terms
Function	sqrt, LeftLegOf, ...	Computes a defined relation of input term
Connectives	$\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$	Used to form complex sentences using atomic sentences
Equality	==	Relational operator that checks equality
Quantifier	$\forall$, $\exists$	Imposes a quantity on the respective variable

Show that DFS is neither complete nor optimal search.

DFS is not complete: DFS can explore infinitely deep paths without reaching a solution, making it incomplete in the following cases:

1. **Infinite search space:** If the search space contains infinite branches, DFS can get stuck exploring one branch forever, failing to discover the solution even though it exists in another branch.
2. **Cycles:** If the search space contains cycles, DFS can keep revisiting the same state repeatedly, never reaching a new state or finding the solution.

DFS is not optimal: DFS may not find the best solution because it prioritizes depth over breadth. Consider the following scenarios:

1. **Shallow solutions:** When the solution is located close to the starting state, DFS might waste time exploring deeper branches, ignoring the optimal path closer to the surface.
2. **Multiple paths to the solution:** If there are multiple paths leading to the goal state, DFS might choose a longer path first, missing the optimal solution that exists on a shorter path.

What are the main aspects considered before solving a complex AI problem?

- **Problem definition:** Clearly define the problem objective, success criteria, input data, and expected output.
- **State space representation:** Choose appropriate techniques to represent the problem state space
- **Search algorithms:** Select efficient algorithms based on optimality, completeness, and resource requirements.
- **Heuristics:** Use domain knowledge to guide the search process towards promising solutions and reduce the search space.

What is state space representation in AI?

State space Representation involves defining an INITIAL STATE and a GOAL STATE and then determining a sequence of actions, called states, to follow.

- **State:** A state can be an Initial State, a Goal State, or any other possible state that can be generated by applying rules between them.
- **Space:** Space refers to the exhaustive collection of all conceivable states.
- **Search:** This technique moves from the beginning state to the desired state by applying good rules while traversing the space of all possible states.
- **Search Tree:** Search tree is a tree-like structure that represents the problem. The initial state is represented by the root node of the search tree, which is the starting point of the tree.
- **Transition:** Transition describe how the system moves from one state to another based on the chosen action.

How state space strategy is useful in problem solving in AI?

1. Clear Problem Representation: By explicitly defining the states, operators, initial state, and goal state(s), it simplifies the problem and makes it easier to design effective solutions.

2. Efficient Search Algorithms: By exploring different states and applying operators, we can find the optimal path or configuration.

3. Systematic Exploration: State space strategy allows for a systematic exploration of all possible solutions. This ensures that the search process is thorough and no promising options are overlooked

4. Performance Analysis: By comparing the efficiency, completeness, and optimality of various methods, developers can identify the best approach for a specific problem.

5. Applicability to Diverse Domains: State space strategy is applicable to a wide range of AI problems, including game playing, planning and scheduling, robotics, constraint satisfaction problems, and expert systems.

In what situations is depth first search more efficient than breadth first search in problem solving for intelligent agents?

1. **Shallow Solutions:** If the goal state is located close to the starting state, DFS can explore the path directly and reach the solution quickly.

2. **Vast Search Spaces:** In vast search spaces with many branches, DFS, only keeps track of the current path, making it less resource-intensive.

3. **Heuristic Guidance:** When combined with a good heuristic function, DFS can be more efficient than BFS.

4. **Specific Problem Structures:** In tree structures or games with limited branching factors, DFS quickly reaches the goal state by navigating the tree or exploring all possible moves efficiently.

5. **Limited Resources:** If resources like memory or processing power are limited, DFS is better than BFS due to smaller memory footprint and lesser computation

What do you mean by rule-based approach and learning based approach?

Rule Based Approach refers to the AI modelling where the relationship or patterns in data are defined by the developer. The machine follows the rules or instructions mentioned by the developer, and performs its task accordingly. Whereas in **Learning based approach**, the relationship or patterns in data are not defined by the developer. In this approach, random data is fed to the machine and it is left to the machine to figure out patterns and trends out of it

Differentiate between Uninformed and Informed Search technique.

Informed Search	Uninformed Search
It is also known as Heuristic Search	It is also known as Blind Search
It uses knowledge for the searching process	It doesn't use knowledge for the searching process

Fast	Slow
May or may not be complete	Always complete
Low cost	High cost
More efficient	Less efficient
Less lengthy	More Lengthy
Less computational requirements	More computational requirements
• A* Search • AO* Search • Hill Climbing Algorithm	• Depth First Search (DFS) • Breadth First Search (BFS) • Branch and Bound

Compare A* and AO * algorithms with each other.

A* algorithm	AO* algorithm
Not designed for handling changes in the environment	Designed for handling changes in the environment
Less suited for frequent environmental changes	More suited for frequent environmental changes
Requires a complete restart of the search after an environmental change	Doesn't require a complete restart of the search after an environmental change
Suitable for static environments	Suitable for dynamic environments
Considers one path at a time	Explores multiple paths simultaneously
Requires less computational resources	Requires more computational resources
Uses less memory	Uses more memory
Requires a consistent heuristic	Doesn't require a consistent heuristic

Why sometimes unnecessary backward propagation occurs in AND OR graph

1. **Redundant Evaluation:** Node with multiple parents, is evaluated from multiple paths
2. **Inefficient Caching Mechanisms:** Inefficient caching mechanism causes redundant computation
3. **Failure to Exploit Structure:** Failure to exploit structure of AND-OR graph causes failure to reuse common sub problems
4. **Lack of Pruning:** Lack of effective pruning mechanism causes redundant propagation
5. **Inefficient Algorithms:** Inefficient algorithms perform redundant computations
6. **Partially explored branch:** Partially explored branch causes partial backward propagation
7. **Cycles:** Presence of cycles causes infinite loop, consuming more resources
8. **Misaligned heuristic:** Misaligned heuristic causes exploration of irrelevant branches

Explain in detail Baysian Theory and its use in AI.

Bayes' Theorem relates the conditional probability and marginal probabilities of two random events.

Proof:

According to product rule: $P(A \wedge B) = P(A|B)\,P(B)$

Similarly: $P(A \wedge B) = P(B|A)\,P(A)$

Equating right hand side of both the equations, we will get:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \quad \text{....(a)}$$

P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the **likelihood**, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

$$P(A_i|B) = \frac{P(A_i) * P(B|A_i)}{\sum_{i=1}^{k} P(A_i) * P(B|A_i)}$$

Where $A_1, A_2, A_3, \ldots\ldots, A_n$ is a set of mutually exclusive and exhaustive events.

Following are some applications of Bayes' theorem:

- It is used to calculate the next step of the robot when the already executed step is given.
- Bayes' theorem is helpful in weather forecasting.
- It can solve the Monty Hall problem.

What is the difference between minimax and alpha-beta pruning in game playing?

Feature	Minimax	Alpha-Beta Pruning
Purpose	Find the best possible move	Improve efficiency of Minimax
Approach	Recursively evaluate all possible moves	Set bounds (alpha & beta) to prune irrelevant branches
Strengths	Guarantees optimal move	Significantly faster than Minimax

Weaknesses	Can be computationally expensive	May miss optimal move due to pruning
Search space	Explores all possible paths	Focuses on promising paths within bounds
Decision-making	Systematic exploration of all options	Efficiently identifies best move within bounds
Suitable for	Games with smaller branching factors	Games where efficiency is crucial

What is the purpose of a semantic network in knowledge representation?

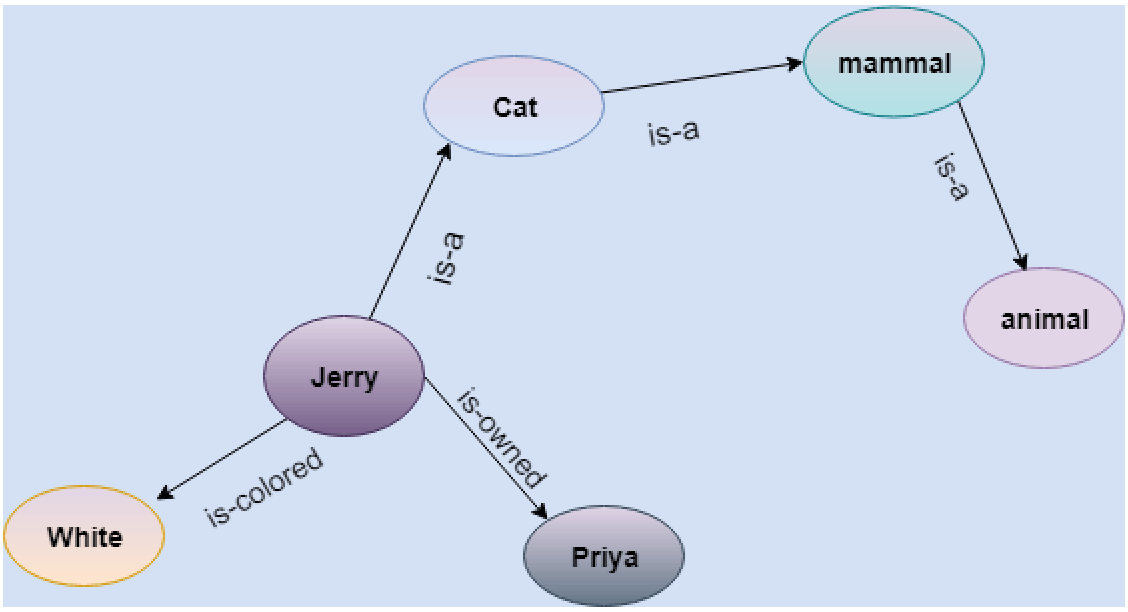
In Semantic networks, we can represent our knowledge in the form of graphical networks. This network consists of nodes representing objects and arcs which describe the relationship between those objects.

This representation consists of mainly two types of relations:

a. IS-A relation (Inheritance)
b. Kind-of-relation

For example, consider the statements:

a. Jerry is a cat.
b. Jerry is a mammal
c. Jerry is owned by Priya.
d. Jerry is brown colored.
e. All Mammals are animal.



Advantages of Semantic network:

1. Semantic networks are a natural representation of knowledge.
2. Semantic networks convey meaning in a transparent manner.
3. These networks are simple and easily understandable.

What is the purpose of STRIPS partial order planning?

1. Efficiency: Unlike traditional total-order planners, which explore all possible action sequences, STRIPS allows for partial ordering of actions. This means actions that don't depend on each other can be scheduled independently, reducing the search space and finding solutions faster.

2. Interleaving actions: STRIPS allows for interleaving actions from different branches of the plan, enabling parallel execution when possible. This can significantly shorten the overall plan execution time.

3. Expressive power: STRIPS incorporates preconditions and postconditions for actions, providing a richer description of state changes and allowing for reasoning about their effects on the environment. This enables more robust and adaptable plans that can handle unexpected situations.

4. Focus on critical actions: STRIPS prioritizes the execution of actions that are critical for achieving the goal, while postponing less crucial ones. This allows for flexibility and adaptation to changing circumstances.

5. Applicable to real-world problems: STRIPS is well-suited for real-world planning problems where the order of actions might not be crucial and flexibility in execution is desired. This makes it applicable in robotics, autonomous systems, and various planning tasks.

How do probabilistic reasoning systems differ from deterministic reasoning systems?

Feature	Probabilistic Reasoning System	Deterministic Reasoning System
Nature of Uncertainty	Handles uncertainty explicitly	No inherent consideration of uncertainty
Representation of Knowledge	Uses probabilities to express likelihood of events	Represents knowledge through fixed rules
Handling Incomplete Information	Assigns probabilities to different possible outcomes	Struggles with incomplete or uncertain information
Decision Making	Considers likelihood of outcomes	Follows a clear set of rules
Applications	Commonly used in machine learning	Suitable for problems with well-defined rules

Flexibility	Flexible in adapting to changing conditions	Rigid and less adaptable to changing conditions
Examples	Bayesian networks	Traditional algorithms, rule-based systems

Explain the Application of AI in the following fields:

Environmental Science

- **Monitoring and prediction:** AI models analyse data from sensors, satellites, and drones to monitor air quality, water pollution, deforestation, and climate change. They can predict environmental events like floods, droughts, and wildfires, allowing for proactive mitigation.
- **Resource management:** AI optimizes resource utilization in agriculture, water management, and energy production, contributing to sustainability and conservation.
- **Conservation efforts:** AI models analyse wildlife populations, track illegal activities like poaching, and assist in habitat restoration and species protection.

Robotics

- **Automation and precision:** Robots powered by AI can perform complex tasks in manufacturing, healthcare, and logistics with high accuracy and efficiency.
- **Autonomous navigation and exploration:** AI-powered robots can navigate in unstructured environments, explore hazardous areas, and perform tasks like search and rescue operations.
- **Human-robot interaction:** AI enables robots to understand human gestures, language, and emotions, leading to more natural and collaborative interactions.

Aerospace

- **Flight control and optimization:** AI algorithms optimize flight paths, improve fuel efficiency, and assist pilots in making critical decisions during flight.
- **Space exploration:** AI-powered robots explore planets and moons, analyse data from space missions, and assist in designing and building spacecraft.
- **Satellite image analysis:** AI analyses satellite data to monitor weather patterns, track deforestation, and detect changes in the Earth's surface.

Medical Science

- **Medical diagnosis and treatment:** AI models analyse medical images, scans, and patient data to aid in diagnosis, predict disease progression, and recommend personalized treatment plans.
- **Drug discovery and development:** AI accelerates the drug discovery process by analysing large datasets of molecules and identifying potential candidates for further development.
- **Robotic surgery:** AI-powered surgical robots assist surgeons in performing complex procedures with greater precision and minimal invasiveness.

What is the mathematical formula for calculating Gini impurity and information gain?

Gini Impurity is a score that evaluates how accurate a split is among the classified groups. The Gini Impurity evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes.

$$\text{Gini Impurity} = 1 - \sum p_i^2$$

Here,

- p_i is the proportion of elements in the set that belongs to the i^{th} category.

Information gain represents the difference in entropy before and after a split on a given attribute. The attribute with the highest information gain will produce the best split as it's doing the best job at classifying the training data according to its target classification. Information gain is usually represented with the following formula:

$$\text{Information Gain}(S,a) = \text{Entropy}(S) - \sum \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where:

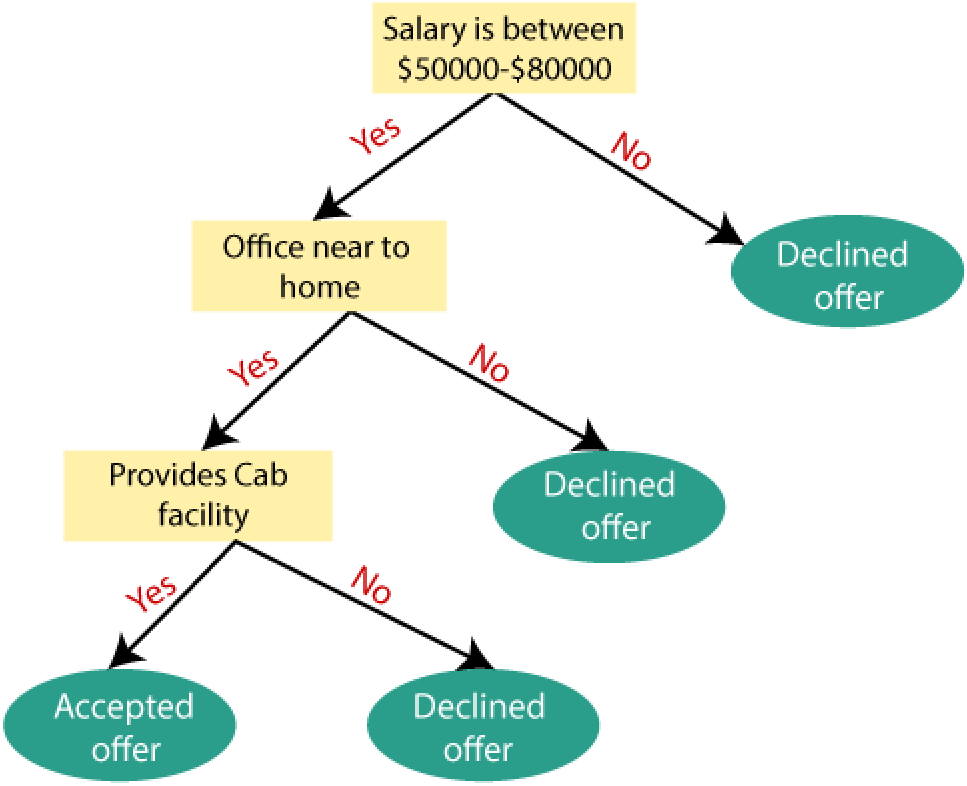
- *a* represents a specific attribute or class label
- *Entropy(S)* is the entropy of dataset, S
- |Sv|/ |S| represents the proportion of the values in S_v to the number of values in dataset, S
- *Entropy*(S_v) is the entropy of dataset, S_v

Define the concept of inductive learning and give an example of a decision tree.

Inductive Learning Algorithm (ILA) is an iterative and inductive machine learning algorithm that is used for generating a set of classification rules, which produces rules of the form "IF-THEN", for a set of examples, producing rules at each iteration and appending to the set of rules.

Decision Tree: It is a graphical representation for getting all the possible solutions to a problem based on given conditions.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or not. Decision Tree for this problem is given below:



How is inductive learning used in environmental science?

- **Climate change:** Inductive learning models can analyze climate data and predict future climate trends
- **Biodiversity loss:** By analysing data on species distribution, researchers can predict the likelihood of species extinction
- **Pollution forecasting:** Models trained on air quality data can predict pollution levels in the future
- **Disease outbreaks:** Analyzing data on disease vectors can help predict the spread of infectious diseases
- **Sustainable agriculture:** Models trained on data about soil conditions can recommend optimal strategies for sustainable agricultural.

- **Conservation planning:** Inductive learning models can be used to identify areas of high ecological value