## AI Question Solve-2022

**1.**

**(a) What is an intelligent agent? Explain the learning agent model elaborately.**

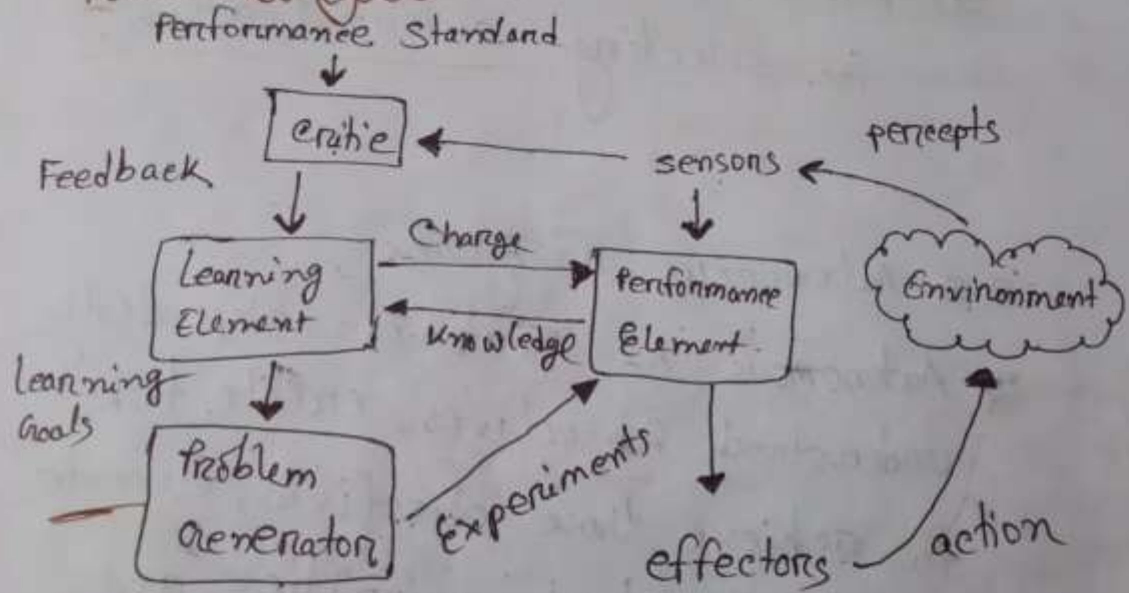
Ans.

• Intellegent Agent: (কোনো কিছুকে) (Final-2022)

An intellegent agent perceives its environment via sensors and acts upon that environment with its effectors.

PEC 2E PG (আগান)

### Learning Agents: (Final-2022)




→ learning capabilities

→ starts with basic knowledge.

**(b) What is PEAS? For each of the following activities, give a PEAS description of the task environment: (i) Automated Car Driving (ii) Medical diagnosis system.**

Ans.

- PEAS: (Final-2022)

⇒ PEAS stants for Performance, Environment, Actuators, sensors. They help define the task environment for an intelegent agent. Hence, PEAS is an important representation system for defining an Artificial Intellegence model.

- Automated driving Car. (Final-2022)

→ Performance: maintain safety, reach destination, maximize profits (fuel, tire wear), obey laws, provide passenger comfort.

→ Actuators: steer, accelerate, brake, horn, speak/display ....

→ Environment: streets, freeways, traffic, pedestrains, (পথচারী) weather, customers, ....

→ Sensors: video, sonar, গাড়ির গতিবেগ speedometer, odometer, দূরত্ব পরিমাপযন্ত্র engine sensors, keyboard input, microphone, GPS ....

- PEAS description for Medical Diagnosis System: (Final-2022)

→ Performance: Accuracy in diagnosing medical condition, Timely and relevant recommendations, sensitivity and specificity of diagnoses.

→ Environment: Patient health data, symptoms, and test results, Various medical condition and diseases, medical literature and research, medical imaging data and laboratory results.

→ Actuators: Diagnostic reports for health professionals, Recomendations for further tests or treatments, Patient education materials.

→ Sensors: Patient helth records, symptoms, history, Diagnostic test results, medical imaging, Real time patient monitoring data, medical literature and research database

**c) What is a rational agent? Why should a rational agent be autonomous?**

Ans.

What is rational Agent? Why should a rational agent be autonomous. (Final-2022)

⇒ Retional Agent is a theoretical entity (সত্ত that considers realistic models of how people think, with preferences for advantageous outcomes and an ~~ability~~ ability to learn.

Rational agents can operate independently of human intervention (হস্তক্ষে This can lead to increased efficienky and reduced human error.

That's why, Rational agents ~~can~~ ~~operate~~ should be autonomous.

**2.**

**(a) Describe the terms complete and optimal with regards to evaluating search strategies. Is either depth-first-search or breadth-first-search complete? Is either of them optimal?**

Ans.

Ques. 01

Describe the terms complete and Optimal with regards to evaluating search strategies. Is either depth-first search or breadth-first search complete? Is either of them Optimal? [final-2022]

Solve:

We have many searching techniques that we have discussed as uniformed but to evaluate such strategies we talk about the following four:

Complete:

complete is a guarantee of finding of a solution whenever one exists.
This means that if a solution exists and if our searching techique can

find that solution and the technique is said to be complete.

## Time Complexity:

Time complexity is how long does it take to find a solution and this is usually measured in terms of the number of nodes that the searching technique expands.

## Space Complexity:

Space complexity is the measurement of the maximum size of the nodes list during the search.

## Optimal: (সন্তোষজনক/ সর্বোত্তম)

Optimality states, If a solution is found, is it guaranteed to be an optimal one? is it the one with minimum cost?

Breadth First search is complete but Depth-first-search is ~~not~~ complete when the search tree is finite.
On the other hand, BFS is optimal but DFS is not optimal.

$\boxed{\text{Ques. 2}}$ Is either DFS or BFS complete or optimal? justify your answer. When will DFS outperform BFS? (midterm - 2022)

$\boxed{\text{Solve:}}$ ① Completeness

• BFS: Breadth first search is complete. Because it systematically explores the search space level by level. So, if a solution exists, BFS will find the shortest path to it.

• DFS:

DFS is not complete when search tree is infinite. But DFS is complete when search tree is finite. DFS will eventually ~~fd~~ find it when search tree is finite.

(ছাড়িয়ে যাওয়া / অতিক্রম করা)

• DFS outperform BFS: (midterm-22)

DFS will outperform on thickly connected graph with high branching factor, ~~increases~~. ~~BFS. BFS will outp~~ because BFS experiences an exponential blowup in nodes expanded as the branching factor increases. BFS

• ছোটো বড় প্রশ্ন:

• BFS will outperform DFS:

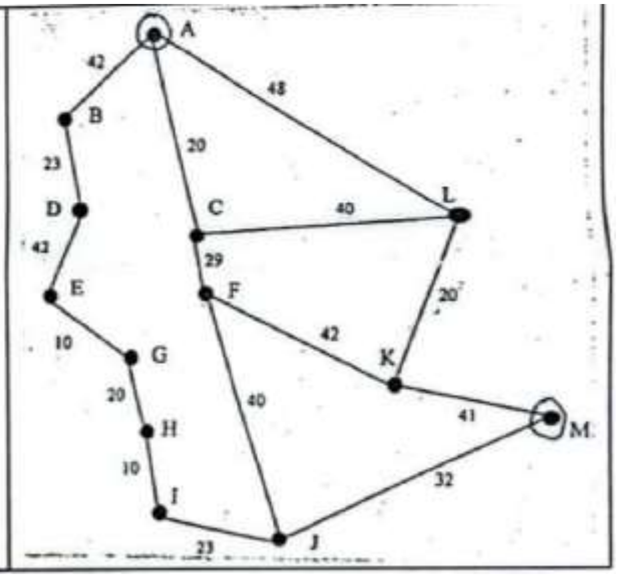
BFS will outperform DFS on space graph with low branching factor, where DFS can get stuck following long, irrelevant chains.

**(b) For the following map, using A* algorithm, find a route from town A to town M. Show the search for your solution, showing the order in which the nodes are expanded and cost at each node. Assume previously visited states will not be revisited. The straight-line distances between any town and town M are shown in the table below. Use it as a measure of the straight-line distance heuristic.**

heuristic.

**Straight Line Distance to M**

| | |
|---|---|
| A | 51 |
| B | 50 |
| C | 32 |
| D | 28 |
| E | 42 |
| F | 14 |
| G | 33 |
| H | 43 |
| I | 50 |
| J | 32 |
| K | 41 |
| L | 56 |
| M | 0 |




Ans.

$\boxed{\text{Ques. 2}}$ Final - 2022

(b) Initialization:

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

$$= 0 + 51$$

$$= 51$$

Iteration-1:

$A \to B$ $\quad$ $A \to C$ $\quad$ $A \to L$

$42+50=92$ $\quad$ $20+32=52$ $\quad$ $48+56=104$

Iteration-2:

$A \to C \to F$ $\quad$ $A \to C \to L$ $\quad$ $A \to B$ $\quad$ $A \to L$

$20+29+14=63$ $\quad$ $20+40+56=116$ $\quad$ $4+50=92$ $\quad$ $48+56=104$

Iteration-3:

$A \to C \to F \to K$ $\quad$ $A \to C \to F \to J$ $\quad$ $A \to C \to L$ $\quad$ $A \to B$ $\quad$ $A \to L$

$20+29+42+41$ $\quad$ $20+29+40+32$ $\quad$ $=116$ $\quad$ $=92$ $\quad$ $=104$

$=132$ $\quad$ $=121$

Iteration-4:

$A \to B \to D \to E \to G \to H \to I \to J$

$42+23+42+10+20+10+23+32$

$= 202$

$A \to C \to F \to K$

$= 132$

$A \to C \to F \to J$

$= 121$

$A \to C \to L$

$= 116$

$A \to L$

$= 104$

Iteration-5:

$A \to L \to K$

$= 48+20+41$

$= 109$

$A \to L \to C$

$48+40+32$

$= 120$

$A \to B \to D \to E \to G \to H \to I \to J$

$= 202$

$A \to C \to F \to K$

$= 132$

$A \to C \to F \to J$

$= 121$

$A \to C \to L$

$= 116$

Iteration-6:

$\cancel{A \to C \to L \to K}$

$\cancel{20+40+20+41}$

$\cancel{=121}$

## Iteration-6:

$A \to L \to K \to F$

$= 48+20+42+14$

$= 124$

$A \to L \to K \to M$

$48+20+41+0$

$= 109$

$A \to L \to C$

$= 120$

$A \to B \to D \to E \to G \to H \to I \to J$

$= 202$

$A \to C \to F \to K$

$= 132$

$A \to C \to F \to J$

$= 121$

$A \to C \to L$

$= 116$

$\therefore$ The cost effective path : $A \to L \to K \to M$

Minimum cost value $= 109$.

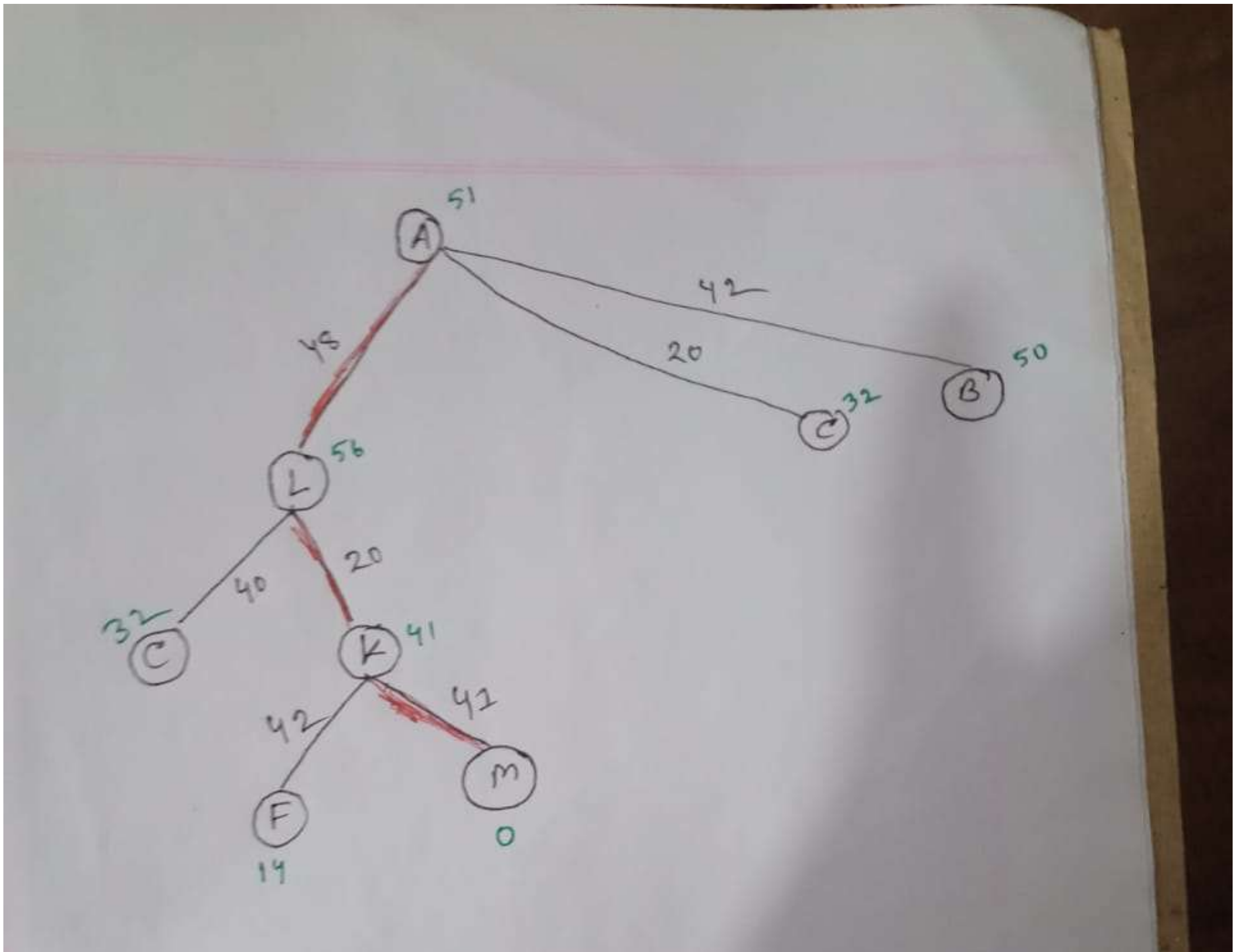


Fig: Route

**(c) Derive the worst-case time and space complexity of depth-first-search and breadth-first-search algorithms?**

Ans.

- DFS worst-case time complexity: O(V + E)
- DFS worst-case space complexity: O(V)
- BFS worst-case time complexity: O(V + E)
- BFS worst-case space complexity: O(V)

**3.**

**(a) Discuss A* algorithm. Give one example whether it is suitable to apply.**

Ans.

**A* Algorithm:**

The A* algorithm is a popular and widely used search algorithm in artificial intelligence and computer science. It is employed for pathfinding and graph traversal, particularly in situations where the search space is represented as a graph. A* is an informed search algorithm that combines the benefits of both Dijkstra's algorithm and greedy best-first search.

**Key Components of A* Algorithm:**

**i. Cost Function:**

A* uses a cost function that takes into account both the cost to reach a node (g(n)) and an estimated cost to reach the goal from that node (h(n)). The function f(n) = g(n) + h(n) is minimized during the search.

**ii. Open and Closed Sets:**

The algorithm maintains two sets: open set and closed set. The open set contains nodes to be evaluated, while the closed set contains nodes that have already been evaluated.

**iii. Node Expansion:**

A* expands nodes in a way that minimizes the cost function f(n). It selects nodes from the open set based on the lowest f(n) value.

**iv. Heuristic Function:**

A* relies on a heuristic function (h(n)) to estimate the cost from the current node to the goal. The heuristic guides the search by prioritizing nodes that are likely to lead to the optimal solution.

**Example of Application:**

**Scenario:** Pathfinding in a Grid-based Environment

**Use Case:** A* algorithm can be applied to find the shortest path from a start point to a destination point on a grid. Each grid cell represents a node, and the cost function includes the actual cost to reach a cell (distance) and the estimated cost to reach the destination from that cell (heuristic).

**Example:**

Consider a grid where certain cells are obstacles, and the goal is to find the shortest path from the top-left corner to the bottom-right corner.

**Start: S, Goal: G, Obstacle: X**

S - - - - -

- X X - - -

- - S - X -

- X X X X -

- - - - - G

In this example, A* would explore grid cells while considering both the distance traveled and the estimated remaining distance to the goal. It intelligently navigates around obstacles and selects the path with the minimum overall cost.

**Suitability:**

- Suitable: A* is suitable for scenarios where finding the optimal path is essential, and the search space can be represented as a graph or grid. It is commonly used in robotics, video games, and route planning applications where efficiency is crucial.

**(b) Consider the water jug problem as stated here: "You are given two jugs, a 4-gallon one and a 3- gallon one. Neither has 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?" Represent this as a problem in State Space Search and state its Production Rules. Show at least one solution to this problem.**

Ans.

**State Representation and Initial State** – we will represent a state of the problem as a tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug. Note $0 \leq x \leq 4$, and $0 \leq y \leq 3$. Our initial state: (0,0)

**Goal Predicate** – state = (2,y) where $0 \leq y \leq 3$.

**Operators** – we must define a set of operators that will take us from one state to another:

| | | | |
|---|---|---|---|
| 1. Fill 4-gal jug | (x,y)<br>x < 4 | → | (4,y) |
| 2. Fill 3-gal jug | (x,y)<br>y < 3 | → | (x,3) |
| 3. Empty 4-gal jug on ground | (x,y) | → | (0,y) |

$x > 0$

| | | | |
|---|---|---|---|
| 4. Empty 3-gal jug on ground | $(x,y)$<br>$y > 0$ | $\rightarrow$ | $(x,0)$ |
| 5. Pour water from 3-gal jug to fill 4-gal jug | $(x,y)$<br>$0 < x+y \geq 4$ and $y > 0$ | $\rightarrow$ | $(4, y - (4 - x))$ |
| 6. Pour water from 4-gal jug to fill 3-gal-jug | $(x,y)$<br>$0 < x+y \geq 3$ and $x > 0$ | $\rightarrow$ | $(x - (3-y), 3)$ |
| 7. Pour all of water from 3-gal jug into 4-gal jug | $(x,y)$<br>$0 < x+y \leq 4$ and $y \geq 0$ | $\rightarrow$ | $(x+y, 0)$ |
| 8. Pour all of water from 4-gal jug into 3-gal jug | $(x,y)$<br>$0 < x+y \leq 3$ and $x \geq 0$ | $\rightarrow$ | $(0, x+y)$ |

Through Graph Search, the following solution is found :

| **Gals in 4-gal jug** | **Gals in 3-gal jug** | **Rule Applied** |
|---|---|---|
| 0 | 0 | |
| | | 1. Fill 4 |
| 4 | 0 | |
| | | 6. Pour 4 into 3 to fill |
| 1 | 3 | |
| | | 4. Empty 3 |
| 1 | 0 | |
| | | 8. Pour all of 4 into 3 |
| 0 | 1 | |
| | | 1. Fill 4 |
| 4 | 1 | |
| | | 6. Pour into 3 |
| 2 | 3 | |

**(c) What are the properties of an artificial agent?**

Ans.

**Properties of an artificial agent:**

**Autonomy**: An AI virtual agent is capable of performing tasks independently without requiring constant human intervention or input.

**Perception:** The agent function senses and interprets the environment they operate in through various sensors, such as cameras or microphones.

**Reactivity:** An AI agent can assess the environment and respond accordingly to achieve its goals.

**Reasoning and decision-making:** AI agents are intelligent tools that can analyze data and make decisions to achieve goals. They use reasoning techniques and algorithms to process information and take appropriate actions.

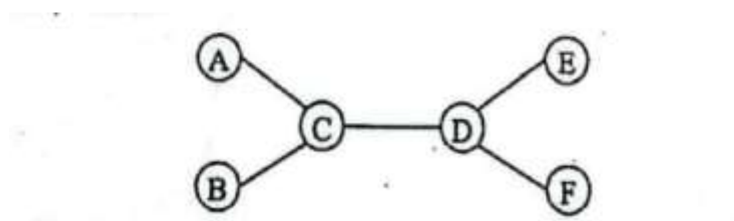
**Learning**: They can learn and enhance their performance through machine, deep, and reinforcement learning elements and techniques.

**Communication:** AI agents can communicate with other agents or humans using different methods, like understanding and responding to natural language, recognizing speech, and exchanging messages through text.

**Goal-oriented:** They are designed to achieve specific goals, which can be pre-defined or learned through interactions with the environment.

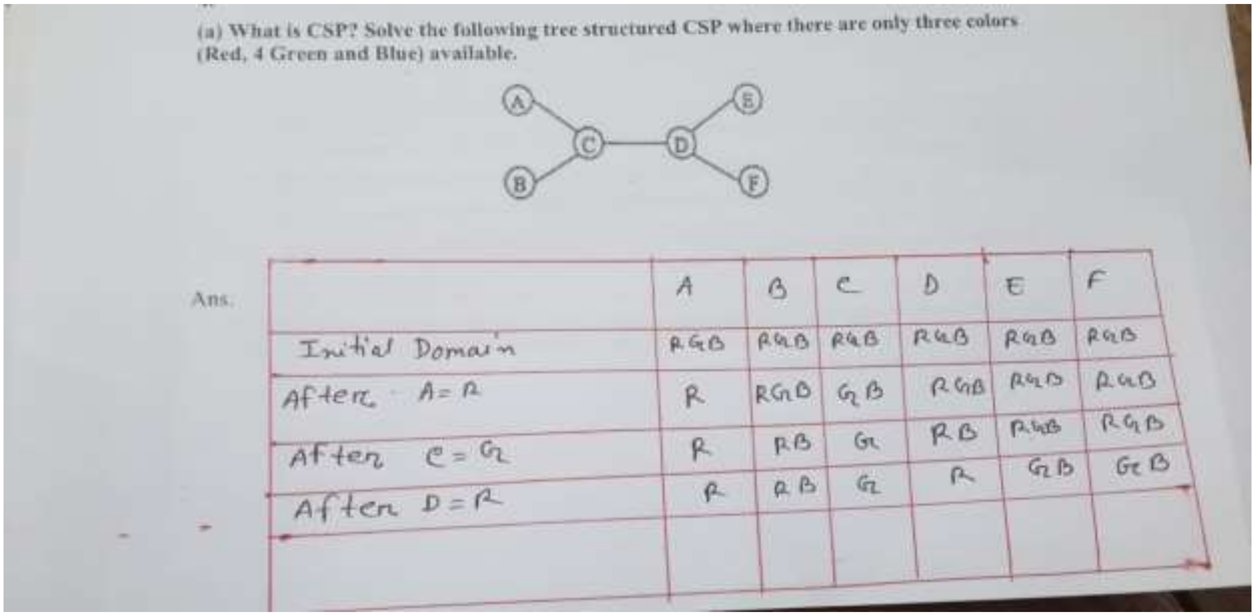
**4.**

**(a) What is CSP? Solve the following tree structured CSP where there are only three colors (Red,  Green and Blue) available.**




Ans.

CSP means **Contraint satisfaction Problem**.



(a) What is CSP? Solve the following tree structured CSP where there are only three colors (Red, 4 Green and Blue) available.



Ans.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Initial Domain | RGB | RGB | RGB | RGB | RGB | RGB |
| After A=R | R | RGB | GB | RGB | RGB | RGB |
| After C=G | R | RB | G | RB | RGB | RGB |
| After D=R | R | RB | G | R | GB | GB |
| | | | | | | |

**(b) Develop a local search algorithm for solving N-Queen problems using min-conflict heuristic.**

Ans.

Min-Conflicts solves the *N*-Queens Problem by randomly selecting a column from the chess board for queen reassignment. The algorithm searches each potential move for the number of conflicts (number of attacking queens), shown in each square. The algorithm moves the queen to the square with the minimum number of conflicts, breaking ties randomly. Note that the number of conflicts is generated by each new direction that a queen can attack from. If two queens would attack from the same direction (row, or diagonal) then the conflict is only counted once. Also note that if a queen is in a position in which a move would put it in greater conflict than its current position, it does not make a move. It follows that if a queen is in a state of minimum conflict, it does not have to move.

This algorithm's run time for solving *N*-Queens is independent of problem size. This algorithm will even solve the *million-queens problem* on average of 50 steps. This discovery and observations led to a great amount of research in 1990 and began research on local search problems and the distinctions between easy and hard problems. *N*-Queens is easy for local search because solutions are densely distributed throughout the state space. It is also effective for hard problems. For example, it has been used to schedule observations for the Hubble Space Telescope, reducing the time taken to schedule a week of observations from three weeks to around 10 minutes.[5]

**(c)What is uncertainty? What are the types of uncertainty? Explain with suitable examples.**

Ans.

Artificial intelligence (AI) uncertainty is when there's not enough information or ambiguity in data or decision-making. It is a fundamental concept in AI, as real-world data is often noisy and incomplete. AI systems must account for uncertainty to make informed decisions.

AI deals with uncertainty by using models and methods that assign probabilities to different outcomes. Managing uncertainty is important for AI applications like self-driving cars and medical diagnosis, where safety and accuracy are key.

Uncertainty in artificial intelligence (AI) refers to the lack of complete information or the presence of variability in data and models. Understanding and modeling uncertainty is crucial for making informed decisions and improving the robustness of AI systems. There are several types of uncertainty in AI, including:

**Types of uncertainty:**

i. **Aleatoric Uncertainty:** This type of uncertainty arises from the inherent randomness or variability in data. It is often referred to as "data uncertainty." For example, in a classification task, aleatoric uncertainty may arise from variations in sensor measurements or noisy labels.
ii. **Epistemic Uncertainty:** Epistemic uncertainty is related to the lack of knowledge or information about a model. It represents uncertainty that can potentially be reduced with more data or better modeling techniques. It is also known as "model uncertainty" and arises from model limitations, such as simplifications or assumptions.
iii. **Parameter Uncertainty:** This type of uncertainty is specific to probabilistic models, such as Bayesian neural networks. It reflects uncertainty about the values of model parameters and is characterized by probability distributions over those parameters.
iv. **Uncertainty in Decision-Making:** Uncertainty in AI systems can affect the decision-making process. For instance, in reinforcement learning, agents often need to make decisions in environments with uncertain outcomes, leading to decision-making uncertainty.

v. **Uncertainty in Natural Language Understanding:** In natural language processing (NLP), understanding and generating human language can be inherently uncertain due to language ambiguity, polysemy (multiple meanings), and context-dependent interpretations.
vi. **Uncertainty in Probabilistic Inference:** Bayesian methods and probabilistic graphical models are commonly used in AI to model uncertainty. Uncertainty can arise from the process of probabilistic inference itself, affecting the reliability of model predictions.
vii. **Uncertainty in Reinforcement Learning:** In reinforcement learning, uncertainty may arise from the stochasticity of the environment or the exploration-exploitation trade-off. Agents must make decisions under uncertainty about the outcomes of their actions.
viii. **Uncertainty in Autonomous Systems:** Autonomous systems, such as self-driving cars or drones, must navigate uncertain and dynamic environments. This uncertainty can pertain to the movement of other objects, sensor measurements, and control actions.
ix. **Uncertainty in Safety-Critical Systems:** In applications where safety is paramount, such as healthcare or autonomous vehicles, managing uncertainty is critical. Failure to account for uncertainty can lead to dangerous consequences.
x. **Uncertainty in Transfer Learning:** When transferring a pre-trained AI model to a new domain or task, uncertainty can arise due to domain shift or differences in data distributions. Understanding this uncertainty is vital for adapting the model effectively.
xi. **Uncertainty in Human-AI Interaction:** When AI systems interact with humans, there can be uncertainty in understanding and responding to human input, as well as uncertainty in predicting human behavior and preferences.

Addressing and quantifying these various types of uncertainty is an ongoing research area in AI, and techniques such as probabilistic modeling, Bayesian inference, and Monte Carlo methods are commonly used to manage and mitigate uncertainty in AI systems.

**SECTION-B (30 Marks)**

**5.**

**a) Why does a hill-climbing search often fail to find the global optimal solution? What is the random-restart hill-climbing search? Briefly explain how you can perform hill-climbing search for the map coloring problem?**

Ans.

**Hill-Climbing Search and Global Optimal Solution:**

Hill-climbing search is a local search algorithm that starts with an arbitrary solution to a problem and iteratively moves to a neighboring solution that improves upon the current one. However, hill climbing often fails to find the global optimal solution due to the following reasons:

**Local Optima:**The algorithm may get stuck in a local optimum, where there is no immediate neighboring solution that is better than the current one, even if a better solution exists elsewhere in the search space.

**Plateaus:**In regions where the objective function is relatively flat (plateaus), hill climbing may struggle to make progress, as there is no clear direction to move toward a better solution.

**Ridges:** If the search space has ridges, where the optimal solution is along a narrow path, hill climbing may have difficulty traversing such paths efficiently.

**Random-Restart Hill-Climbing Search:**

To address the issue of getting stuck in local optima, a random-restart hill-climbing search involves running the hill-climbing algorithm multiple times, each time starting from a different initial solution. This increases the chances of reaching different parts of the search space and finding a better overall solution.

**Hill-Climbing Search for the Map Coloring Problem:**

The map coloring problem involves assigning colors to regions on a map such that no two adjacent regions have the same color. Here's a brief overview of how hill climbing can be applied to this problem:

- **State Representation**: Represent a state as a set of assignments where each region is assigned a color.
- **Objective Function:** Define an objective function that measures the quality of a state, such as the number of adjacent regions with the same color.
- **Neighbors**: Define the neighbors of a state by considering possible changes to the color assignments for one or more regions.
- **Hill-Climbing Steps**:

- Start with an initial assignment of colors.

- Evaluate the objective function for the current state.

- Iterate:

- Select a neighboring state with a better objective function value.

- If no such state is found, the algorithm terminates.

- Update the current state to the selected neighbor.

- **Random-Restart:** Repeat the hill-climbing process from different initial assignments to increase the chances of finding a globally optimal solution.

Keep in mind that hill climbing is not guaranteed to find the globally optimal solution due to its local nature. The random-restart strategy aims to mitigate this limitation by exploring different parts of the solution space.

**(b) Translate the following sentences into first order logic (FOL) sentences:**

**i) Every member of the Roofers club is either a skjer or a mountain climber or both.**

**ii) Some women are more knowledgeable than others except for herself.**

**iii) Every fruit other than apricots and pineapples is bad and spoiled.**

**iv) Apricots and pineapples are bad fruits and spoiled.**

Ans.

i. $\forall x(\text{RoofersClubMember}(x) \rightarrow (\text{Skier}(x) \vee \text{MountainClimber}(x)))$

ii. $\exists x \exists y(\text{Woman}(x) \wedge \text{Woman}(y) \wedge \text{Knowledgeable}(x) \wedge \text{Knowledgeable}(y) \wedge x \neq y \wedge \forall z(\text{Woman}(z) \rightarrow \text{Knowledgeable}(z) \geq \text{Knowledgeable}(x)))$

iii. $\forall x(\text{Fruit}(x) \wedge \neg(\text{Apricot}(x) \vee \text{Pineapple}(x)) \rightarrow (\text{Bad}(x) \wedge \text{Spoiled}(x)))$

iv. $\forall x(\text{Apricot}(x) \vee \text{Pineapple}(x) \rightarrow (\text{Bad}(x) \wedge \text{Spoiled}(x)))$

**(c)Write down the differences between first order logic (FOL) and propositional logic.**

Ans.

## Propositional. Logic *vs* **First Order**

| | | |
|---|---|---|
| *Ontology* | Facts (P, Q) | Objects,<br>Properties,<br>Relations |
| *Syntax* | Atomic sentences<br>Connectives | Variables & quantification<br>Sentences have structure: terms<br>father-of(mother-of(X))) |
| *Semantics* | Truth Tables | Interpretations<br>(Much more complicated) |
| *Inference* | Efficient SAT algorithms | Unification<br>Forward, Backward chaining<br>Prolog, theorem proving |

45

**6.**

**(a)**

**Express the following sentences in propositional logic.**

**"The person is a toddler; if the person is a toddler, then the person is a child; if the person is a child and male then the person is a boy; if the person is an infant then the person is a child; if the person is a child and female then the person is a girl; the person is female." Also prove that the person is a girl using the rule of inferences.**

Ans.

Let's represent the given information in propositional logic:

Let $T$ represent "The person is a toddler."

Let $C$ represent "The person is a child."

Let $M$ represent "The person is male."

Let $B$ represent "The person is a boy."

Let $I$ represent "The person is an infant."

Let $F$ represent "The person is female."

Let $G$ represent "The person is a girl."

Now, let's express the given sentences in propositional logic:

Sentence 1: $T$

Sentence 2: $T \Rightarrow C$

Sentence 3: $(C \wedge M) \Rightarrow B$

Sentence 4: $I \Rightarrow C$

Sentence 5: $(C \wedge \mathrm{F}) \Rightarrow G$

Sentence 6: $F$

Now, to prove that the person is a girl using the rules of inference, we can apply modus ponens to sentences 5 and 6:

From sentence 5: If $(C \wedge \mathrm{F}) \Rightarrow G$ ) then we can say , $(C \wedge \neg M) \Rightarrow G$

From sentence 6: $F$

Applying modus ponens: $(C \wedge \neg M) \Rightarrow G$

Thus, $G$ is true.

Therefore, we have proven that if the person is a child and female $C \wedge \neg M$), then the person is a girl ($G$). Since the person is female ($F$), we can conclude that the person is a girl ($G$) using the rules of inference.

**(b) How can you build a knowledge-based agent using a declarative approach? Convert the logic A=> (B V C) into CNF.**

Ans.

**Building a knowledge-based agent using a declarative approach:**

Building a knowledge-based agent using a declarative approach involves representing knowledge in a declarative form, often using logic. One common representation is propositional logic. Here are the steps to build a knowledge-based agent using a declarative approach:

- **Define the Knowledge Base (KB):** Identify the relevant information and express it using logical statements. In propositional logic, these statements involve propositions or variables.
- **Choose a Logical Language:** Choose a logical language that suits the problem domain. Propositional logic is often used for simplicity, while first-order logic allows for more expressive representations.
- **Formulate Rules and Facts**: Express the rules and facts of the problem in logical form. Rules are typically represented as implications, and facts as atomic propositions.
- **Use Inference Mechanisms:** Implement inference mechanisms that allow the agent to derive new information from the existing knowledge. Common methods include forward chaining, backward chaining, and resolution.

Now, let's convert the logical statement $A \Rightarrow (B \vee C)$ into Conjunctive Normal Form (CNF):

1. **Eliminate Implication:**
   $\neg A \vee (B \vee C)$
2. **Apply Distribution:**
   $(\neg A \vee B \vee C)$

The resulting CNF expression is $(\neg A \vee B \vee C)$. This is a disjunction (OR) of three literals, and it is in CNF, which is a conjunction (AND) of clauses.

**(C ) Find predicate, function and term from the following sentences:**
**(i) Brother (John, Richard)**
**(ii)>(Length(LeftlegOf(Richrad)), (Length(LeftlegOf(John)))**
Ans.

**Sentence (i):**

Predicate: Brother

Functions: No functions in here

Terms: John, Richard

**Sentence (ii):**

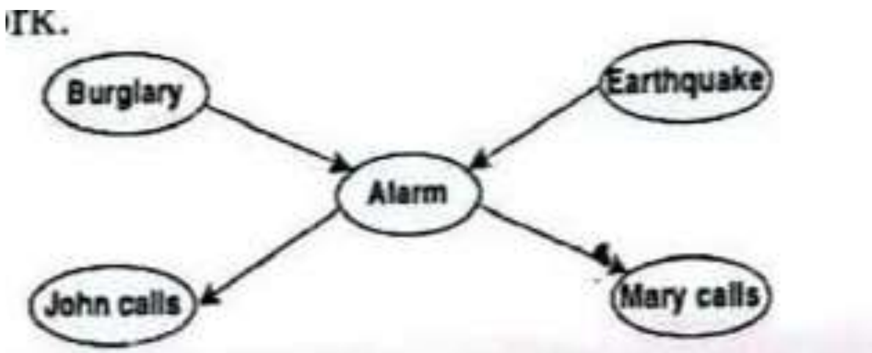
Predicates: > (Greater than)

Functions: Length, LeftLegOf

Terms: John, Richard

**7.**

**(a) Derive the product formula for Bayesian networks. Show the steps to calculate the product formula of the following Bayesian network.**




Ans.

If P[a1,a2, a3,….., an] is the JPD of the following variables from a1 to an, then there are several ways of calculating the Joint Probability Distribution as a combination of various terms such as,

P[a1,a2, a3,….., an] = P[a1 | a2, a3,….., an] * P[a2, a3,….., an]

= P[a1 | a2, a3,….., an] * P[a2 | a3,….., an]….P[$a_{n-1}$|$a_n$] * P[$a_n$]

Generalizing the above equation, we can write the Joint Probability Distribution as,

**P($X_i$|$X_{i-1}$,………, $X_n$) = P($X_i$ |Parents($X_i$ ))**

**The steps to calculate the product formula of the following Bayesian network:**

P[ Burglary, Earthquake, Alarm, John calls, Mary calls]

=P(Mary calls | Alarm) * P( Alarm | Burglary, Earthquake) * P(John calls | Alarm) * P(Burglary) * P(Earthquake)

**(b) Consider a scenario, where 5% of the populations have some sort of cancer (cr) disease. Among the cancerous population, 40% of them have chronic cough (cc) as one of their symptoms. There are many other reasons for chronic cough, and among all chronic cough patients, 9 out 10 people are non-cancerous. If you have a symptom of chronic cough, what will be the probability that you will be diagnosed with cancer?**

Ans.

• we know from, Bayes' Rule,

$$P(+cn|+ce) = \frac{P(+ce|+cn)\,P(+cn)}{P(+ce|+cn)\,P(+cn) + P(+ce|-cn)\,P(-cn)}$$

$$= \frac{0.4 \times 0.05}{0.4 \times 0.05 + 0.9 \times 0.95} = 0.02286$$

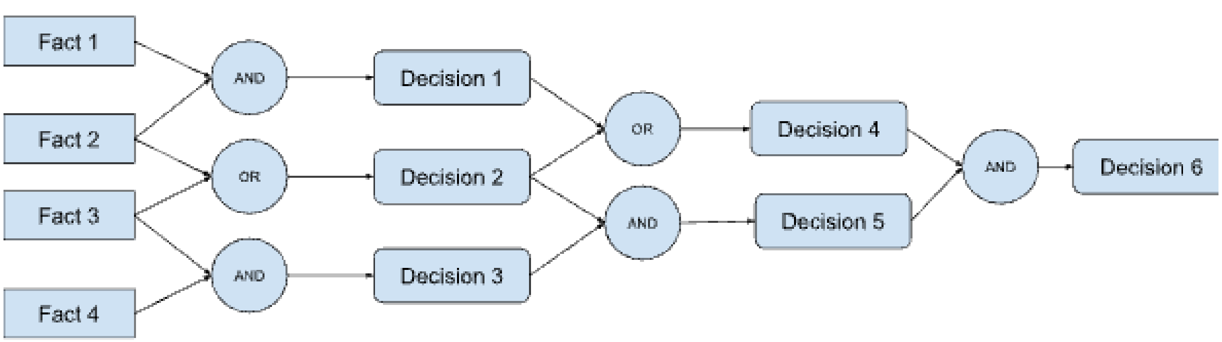
Here,
$P(+cn) = 0.05$
$P(+ce|+cn) = 0.4$
$P(+ce|-cn) = 0.9$

**( c) How does backward chain work? Explain.**
Ans.

**Forward Chaining:**

Forward chaining is also known as a forward deduction or forward reasoning method when using an inference engine. The forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied and adds their conclusion to the known facts. This process repeats until the problem is solved.

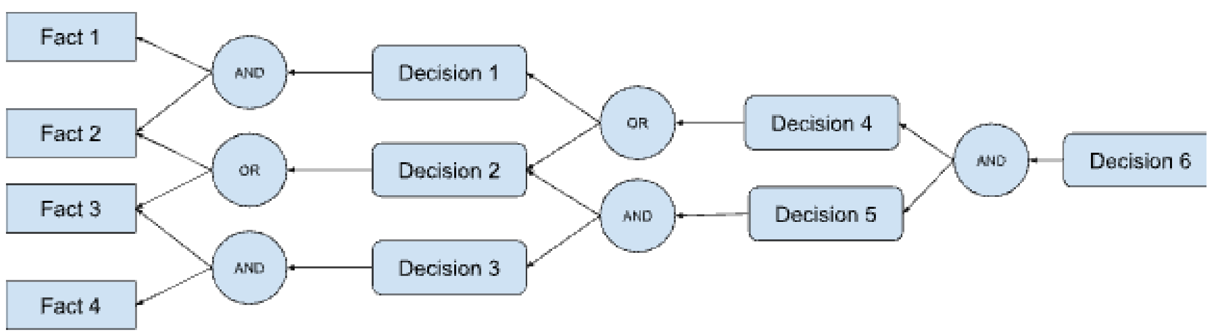
In this type of chaining, the inference engine starts by evaluating existing facts, derivations, and conditions before deducing new information. An endpoint, or goal, is achieved through the manipulation of knowledge that exists in the knowledge base.



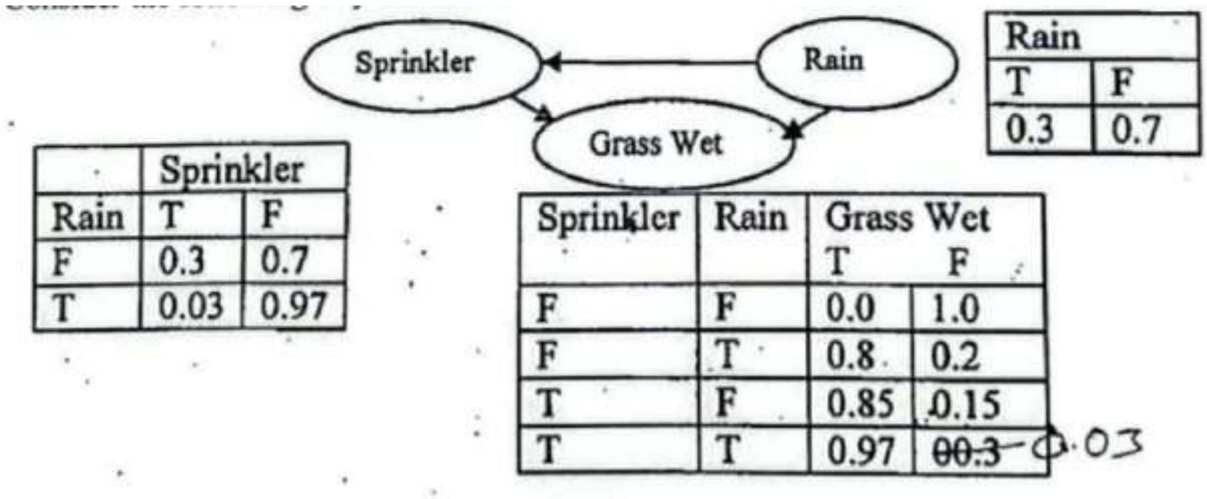
**Backward Chaining:**

Backward chaining is also known as a backward deduction or backward reasoning method when using an inference engine. In this, the inference engine knows the final decision or goal. The system starts from the goal and works backward to determine what facts must be asserted so that the goal can be achieved.

For example, it starts directly with the conclusion (hypothesis) and validates it by backtracking through a sequence of facts. Backward chaining can be used in debugging, diagnostics and prescription applications.




**8.**
**(a) Consider the following Bayesian network.**




| Rain | |
|---|---|
| T | F |
| 0.3 | 0.7 |

| | Sprinkler | |
|---|---|---|
| Rain | T | F |
| F | 0.3 | 0.7 |
| T | 0.03 | 0.97 |

| Sprinkler | Rain | Grass Wet | |
|---|---|---|---|
| | | T | F |
| F | F | 0.0 | 1.0 |
| F | T | 0.8 | 0.2 |
| T | F | 0.85 | 0.15 |
| T | T | 0.97 | ~~0.3~~ 0.03 |

**What is the probability that it is raining given that the grass is wet?**

Ans.

Solve:

The formula for the JPD is given by,

P[Grass wet = T, Rain = T, Sprinkler = F]

= P[Grass wet = T | Rain = T, Sprinkler = F] · P(Rain = T)

· P(Sprinkler = F | Rain = T)

= 0.8 * 0.3 * 0.97

= 0.2328

**(b) What are the key components in first order logic? Consider the following knowledge base:**

**Consider the following knowledge base:**

**A=>B**

**C∧D=> A**

**Q∧C =>D**

**P∧A=> C**

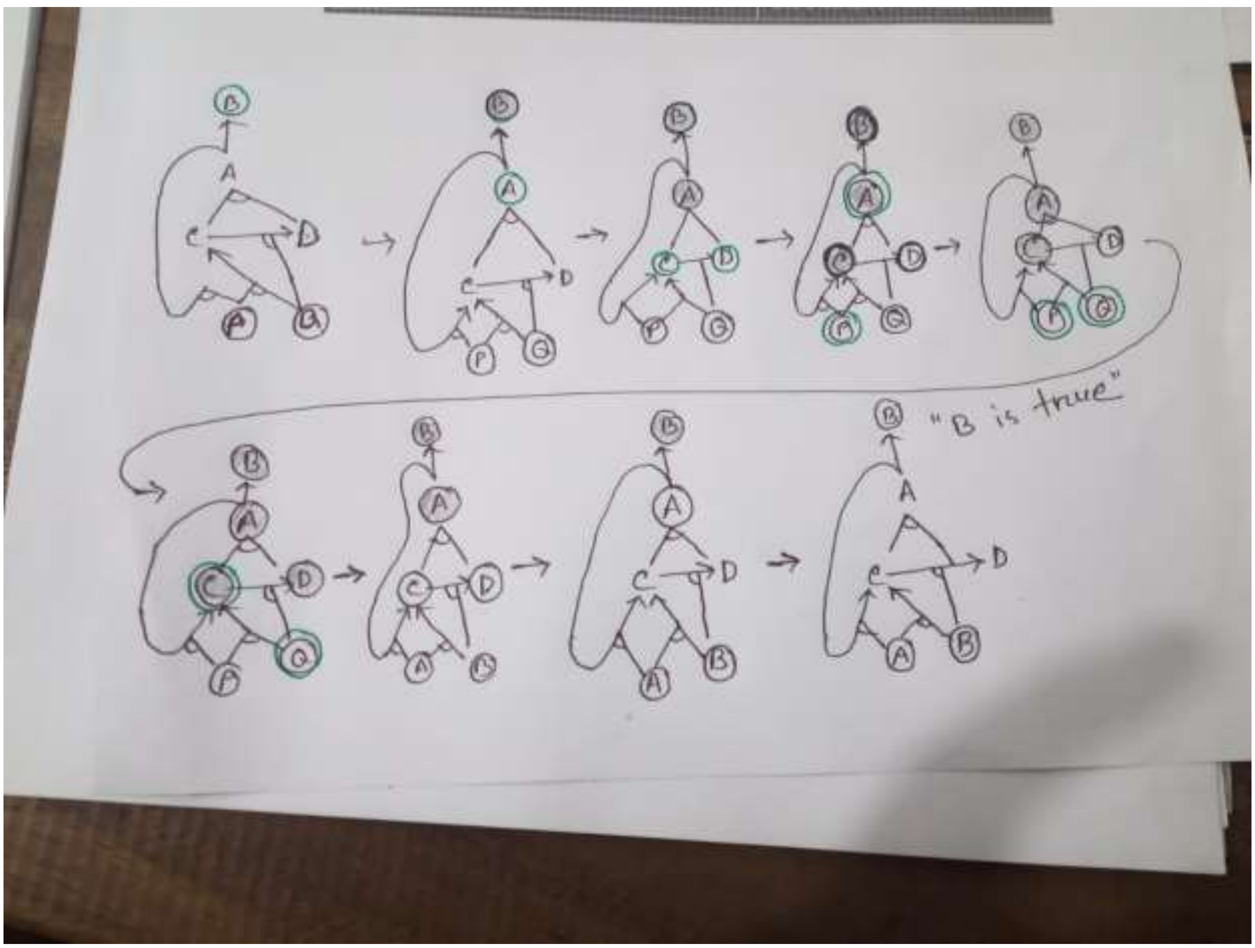
**P∧Q =>C**

**P**

**Q**

**Using backward chaining, find out whether the query B is true.**

Ans.




**( c) What are the main differences between a probabilistic agent and logical agent?**

Ans.

| Feature | Logical Agent | Probabilistic Agent |
| --- | --- | --- |
| **Representation of Knowledge** | Symbolic logic (propositional or first-order) | Probabilities |
| **Handling Uncertainty** | Assumes a deterministic world | Explicitly models and handles uncertainty |
| **Inference and Decision Making** | Logical inference based on rules | Probabilistic inference, decision based on probabilities and expected utility |
| **Representation of Actions** | Symbolic expressions for actions | Actions with associated probabilities |
| **Domain of Application** | Expert systems, rule-based systems | Decision-making under uncertainty, Bayesian networks, statistical learning |
| **Flexibility** | May struggle in dynamic or uncertain environments | More flexible in handling dynamic and uncertain environments |