ARTIFICIAL INTELLIGENCE

1. WHAT IS ARTIFICIAL INTELLIGENCE AND WHY DO WE NEED IT?

Artificial intelligence is computer science technology that emphasizes creating intelligent machine that can mimic human behavior. Here Intelligent machines can be defined as the machine that can behave like a human, think like a human, and also capable of decision making. It is made up of two words, "Artificial" and "Intelligence," which means the "man-made thinking ability."

With artificial intelligence, we do not need to pre-program the machine to perform a task; instead, we can create a machine with the programmed algorithms, and it can work on its own.

In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a chatbot that is fed examples of text can learn to generate lifelike exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples. New, rapidly improving generative AI techniques can create realistic text, images, music and other media.

AI is important for its potential to change how we live, work and play. It has been effectively used in business to automate tasks done by humans, including customer service work, lead generation, fraud detection and quality control. In a number of areas, AI can perform tasks much better than humans. Particularly when it comes to repetitive, detail-oriented tasks, such as analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors. Because of the massive data sets it can process, AI can also give enterprises insights into their operations they might not have been aware of. The rapidly expanding population of generative AI tools will be important in fields ranging from education and marketing to product design.

2. EXPLAIN HOW ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, AND DEEP LEARNING DIFFER FROM EACH OTHER?

The difference between AI, ML, and Deep Learning is given in the below table:

Artificial Intelligence	Machine Learning	Deep Learning
The term Artificial intelligence was first coined in the year 1956 by John McCarthy.	The term ML was first coined in the year **1959 by Arthur Samuel**.	The term DL was first coined in the year **2000 Igor Aizenberg**.
It is a technology that is used to create intelligent machines that can mimic human behavior.	It is a subset of AI that learns from past data and experiences.	It is the subset of machine learning and AI that is inspired by the human brain cells, called neurons, and imitates the working of the human brain.
AI completely deals with structured, semi-structured data.	ML deals with structured and semi-structured data.	Deep learning deals with structured and unstructured data.
It requires a huge amount of data to work.	It can work with less amount of data compared to deep learning and AI.	It requires a huge amount of the data compared to the ML.
The goal of AI is to enable the machine to think without any human intervention.	The goal of ML is to enable the machine to learn from past experiences.	The goal of deep learning is to solve the complex problems as the human brain does, using various algorithms.

3. DISCUSS THE CHARECTERISTICS OF AI PROBLEM.

A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions. Some of the main key features of a problem are given below.

Is the problem decomposable into set of sub problems?

Can the solution step be ignored or undone?

Is the problem universally predictable?

Is a good solution to the problem obvious without comparison to all the possible solutions?

Is the desire solution a state of world or a path to a state?

Is a large amount of knowledge absolutely required to solve the problem?

Will the solution of the problem required interaction between the computer and the person?

The above characteristics of a problem are called as 7-problem characteristics under which the solution must take place.

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, entertainment, finance, education, etc. AI is making our daily life more comfortable and fast.

4. APPLICATIONS OF ARTIFICIAL INTELLIGENCE

The following are some sectors which have the application of Artificial Intelligence:

1. AI in Astronomy

- Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc.

2. AI in Healthcare

- In the last, five to ten years, AI becoming more advantageous for the healthcare industry and going to have a significant impact on this industry.
- Healthcare Industries are applying AI to make a better and faster diagnosis than humans. AI can help doctors with diagnoses and can inform when patients are worsening so that medical help can reach to the patient before hospitalization.

3. AI in Gaming

- AI can be used for gaming purpose. The AI machines can play strategic games like chess, where the machine needs to think of a large number of possible places.

4. AI in Finance

- AI and finance industries are the best matches for each other. The finance industry is implementing automation, chatbot, adaptive intelligence, algorithm trading, and machine learning into financial processes.

5. AI in Data Security

- The security of data is crucial for every company and cyber-attacks are growing very rapidly in the digital world. AI can be used to make your data more safe and secure. Some examples such as AEG bot, AI2 Platform,are used to determine software bug and cyber-attacks in a better way.

6. AI in Social Media

- Social Media sites such as Facebook, Twitter, and Snapchat contain billions of user profiles, which need to be stored and managed in a very efficient way. AI can organize and manage massive amounts of data. AI can analyze lots of data to identify the latest trends, hashtag, and requirement of different users.

7. AI in Travel & Transport

- AI is becoming highly demanding for travel industries. AI is capable of doing various travel related works such as from making travel arrangement to suggesting the hotels, flights, and best routes to the customers. Travel industries are using AI-powered chatbots which can make human-like interaction with customers for better and fast response.

8. AI in Automotive Industry

- Some Automotive industries are using AI to provide virtual assistant to their user for better performance. Such as Tesla has introduced TeslaBot, an intelligent virtual assistant.
- Various Industries are currently working for developing self-driven cars which can make your journey more safe and secure.

9. AI in Robotics:

- Artificial Intelligence has a remarkable role in Robotics. Usually, general robots are programmed such that they can perform some repetitive task, but with the help of AI, we can create intelligent robots which can perform tasks with their own experiences without pre-programmed.
- Humanoid Robots are best examples for AI in robotics, recently the intelligent Humanoid robot named as Erica and Sophia has been developed which can talk and behave like humans.

10. AI in Entertainment

- We are currently using some AI based applications in our daily life with some entertainment services such as Netflix or Amazon. With the help of ML/AI algorithms, these services show the recommendations for programs or shows.

11. AI in Agriculture

- Agriculture is an area which requires various resources, labor, money, and time for best result. Now a day's agriculture is becoming digital, and AI is emerging in this field. Agriculture is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis. AI in agriculture can be very helpful for farmers.

12. AI in E-commerce

- AI is providing a competitive edge to the e-commerce industry, and it is becoming more demanding in the e-commerce business. AI is helping shoppers to discover associated products with recommended size, color, or even brand.

13. AI in education:

- AI can automate grading so that the tutor can have more time to teach. AI chatbot can communicate with students as a teaching assistant.
- AI in the future can be work as a personal virtual tutor for students, which will be accessible easily at any time and any place.

5. DISCUSS UNINFORMED BREADTH-FIRST SEARCH IN DETAIL

Breadth- First -Search: Consider the state space of a problem that takes the form of a tree. Now, if we search the goal along each breadth of the tree, starting from the root and continuing up to the largest depth, we call it breadth first search.

• **Algorithm:**

1. Create a variable called NODE-LIST and set it to initial state

2. Until a goal state is found or NODE-LIST is empty do

a. Remove the first element from NODE-LIST and call it E. If NODE-LIST was empty, quit

b. For each way that each rule can match the state described in E do:

i. Apply the rule to generate a new state

ii. If the new state is a goal state, quit and return this state

iii. Otherwise, add the new state to the end of NODE-LIST

BFS illustrated:

Step 1: Initially fringe contains only one node corresponding to the source state A.

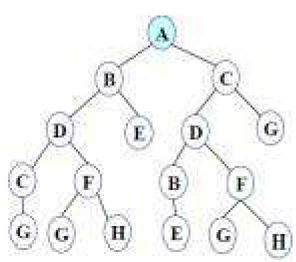

Figure 1

FRINGE: A

Step 2: A is removed from fringe. The node is expanded, and its children B and C are generated. They are placed at the back of fringe.

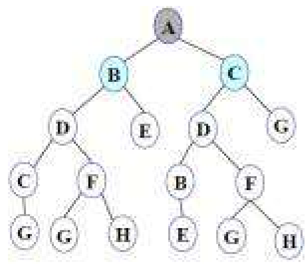

Figure 2

FRINGE: B C

Step 3: Node B is removed from fringe and is expanded. Its children D, E are generated and put at the back of fringe.

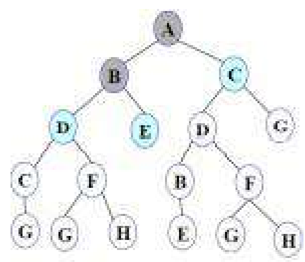

Figure 3

FRINGE: C D E

Step 4: Node C is removed from fringe and is expanded. Its children D and G are added to the back of fringe.

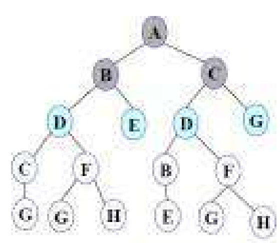

Figure 4

FRINGE: D E D G

Step 5: Node D is removed from fringe. Its children C and F are generated and added to the back of fringe.

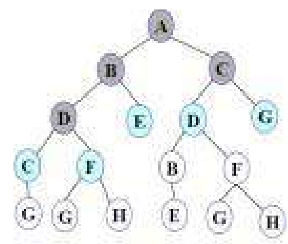

Figure 5

FRINGE: E D G C F

Step 6: Node E is removed from fringe. It has no children.

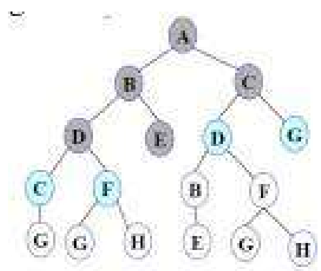

Figure 6

FRINGE: D G C F

Step 7: D is expanded; B and F are put in OPEN.

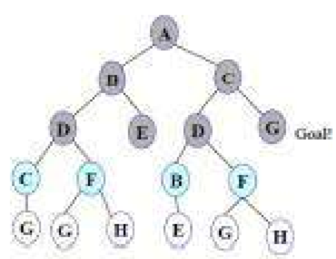

Figure 7

FRINGE: G C F B F

Step 8: G is selected for expansion. It is found to be a goal node. So the algorithm returns the path A C G by following the parent pointers of the node corresponding to G. The algorithm terminates.

Breadth first search is:

One of the simplest search strategies

- Complete. If there is a solution, BFS is guaranteed to find it.
- If there are multiple solutions, then a minimal solution will be found
- The algorithm is optimal (i.e., admissible) if all operators have the same cost. Otherwise,
- breadth first search finds a solution with the shortest path length.

Time complexity : O(bd•)

Space complexity : O(bd•)

Optimality :Yes•

b - branching factor(maximum no of successors of any node),

d – Depth of the shallowest goal node

Maximum length of any path (m) in search space

Advantages:

Finds the path of minimal length to the goal.

Disadvantages:

Requires the generation and storage of a tree whose size is exponential the depth of the shallowest goal node.

The breadth first search algorithm cannot be effectively used unless the search space is quite small.

6. COMPARE AND CONTRAST BREADTH FIRST SEARCH AND DEPTH FIRST SEARCH

Breadth-First Search:

BFS, Breadth-First Search, is a vertex-based technique for finding the shortest path in the graph. It uses a Queue data structure that follows first in first out. In BFS, one vertex is selected at a time when it is visited and marked then its adjacent are visited and stored in the queue. It is slower than DFS.

Example:

Input:

```
    A
   / \
  B   C
 /   / \
D   E   F
```

Output:

A, B, C, D, E, F

Depth First Search:

DFS, Depth First Search, is an edge-based technique. It uses the Stack data structure and performs two stages, first visited vertices are pushed into the stack, and second if there are no vertices then visited vertices are popped.

Example:

Input:

```
    A
   / \
  B   D
 /   / \
C   E   F
```

BFS vs DFS

S.No.	Parameters	BFS	DFS
1.	Stands for	BFS stands for Breadth First Search.	DFS stands for Depth First Search.
2.	Data Structure	BFS(Breadth First Search) uses Queue data structure for finding the shortest path.	DFS(Depth First Search) uses Stack data structure.

3.	**Definition**	BFS is a traversal approach in which we first walk through all nodes on the same level before moving on to the next level.	DFS is also a traversal approach in which the traverse begins at the root node and proceeds through the nodes as far as possible until we reach the node with no unvisited nearby nodes.
4.	**Technique**	BFS can be used to find a single source shortest path in an unweighted graph because, in BFS, we reach a vertex with a minimum number of edges from a source vertex.	In DFS, we might traverse through more edges to reach a destination vertex from a source.
5.	**Conceptual Difference**	BFS builds the tree level by level.	DFS builds the tree sub-tree by sub-tree.
6.	**Approach used**	It works on the concept of FIFO (First In First Out).	It works on the concept of LIFO (Last In First Out).
7.	**Suitable for**	BFS is more suitable for searching vertices closer to the given source.	DFS is more suitable when there are solutions away from source.
8.	**Suitability for Decision-Trees**	BFS considers all neighbors first and therefore not suitable for decision-making trees used in games or puzzles.	DFS is more suitable for game or puzzle problems. We make a decision, and the then explore all paths through this decision. And if this decision leads to win situation, we stop.
9.	**Time Complexity**	The Time complexity of BFS is O(V + E) when Adjacency List is used and O(V^2) when Adjacency Matrix is used, where V stands for vertices and E stands for edges.	The Time complexity of DFS is also O(V + E) when Adjacency List is used and O(V^2) when Adjacency Matrix is used, where V stands for vertices and E stands for edges.
10.	**Visiting of Siblings/ Children**	Here, siblings are visited before the children.	Here, children are visited before the siblings.

11.	**Removal of Traversed Nodes**	Nodes that are traversed several times are deleted from the queue.	The visited nodes are added to the stack and then removed when there are no more nodes to visit.
12.	**Backtracking**	In BFS there is no concept of backtracking.	DFS algorithm is a recursive algorithm that uses the idea of backtracking
13.	**Applications**	BFS is used in various applications such as bipartite graphs, shortest paths, etc.	DFS is used in various applications such as acyclic graphs and topological order etc.
14.	**Memory**	BFS requires more memory.	DFS requires less memory.
15.	**Optimality**	BFS is optimal for finding the shortest path.	DFS is not optimal for finding the shortest path.
16.	**Space complexity**	In BFS, the space complexity is more critical as compared to time complexity.	DFS has lesser space complexity because at a time it needs to store only a single path from the root to the leaf node.
17.	**Speed**	BFS is slow as compared to DFS.	DFS is fast as compared to BFS.
18,	**Tapping in loops**	In BFS, there is no problem of trapping into infinite loops.	In DFS, we may be trapped in infinite loops.
19.	**When to use?**	When the target is close to the source, BFS performs better.	When the target is far from the source, DFS is preferable.

7. EXPLAIN PROBABLISTIC REASONING AND BAYES THOEREM IN DETAIL.

Probabilistic reasoning Causes of uncertainty: Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with logic to handle the uncertainty.

Following are some leading causes of uncertainty to occur in the real world. Information occurred from unreliable sources.

- Experimental Errors
- Equipment fault
- Temperature variation
- Climate change.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is the result of someone's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as "It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0 and 1 that represent ideal uncertainties.

$0 \leq P(A) \leq 1$, where $P(A)$ is the probability of an event A. $P(A) = 0$, indicates total uncertainty in an event A. $P(A) =1$, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

$P(\neg A)$ = probability of a not happening event.

$P(\neg A) + P(A) = 1$.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.
Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability: Conditional probability is a probability of occurring an event when another event has already happened.

Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

Where P(A⋀B)= Joint probability of a and B

P(B)= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of P(A⋀B) by P(B).

Bayes' theorem: Bayes' theorem is also known as Bayes' rule, Bayes' law, or Bayesian reasoning, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.

Bayes' theorem was named after the British mathematician Thomas Bayes.

The Bayesian inference is an application of Bayes' theorem, which is fundamental to Bayesian statistics

It is a way to calculate the value of P(B|A) with the knowledge of P(A|B).

Bayes' theorem allows updating the probability prediction of an event by observing new information of the real world.

Bayes' theorem can be derived using product rule and conditional probability of event A with known event B:

As from product rule we can write:

1. $P(A \wedge B) = P(A|B)\, P(B)$

or Similarly, the probability of event B with known event A:

2. 1. $P(A \wedge B) = P(B|A)\, P(A)$

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

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

The above equation is called as Bayes' rule or Bayes' theorem.

This equation is basic of most modern AI systems for probabilistic inference.

It shows the simple relationship between joint and conditional probabilities.

Here, 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.

8. EXPLAIN ABOUT PROBABILISTIC REASONING

Probabilistic reasoning

Causes of uncertainty:

Following are some leading causes of uncertainty to occur in the real world.

- Information occurred from unreliable sources.
- Experimental Errors
- Equipment fault
- Temperature variation
- Climate change.

Probabilistic reasoning is a way of knowledge representation where we apply the concept of probability

to indicate the uncertainty in knowledge. In probabilistic reasoning, we combine probability theory with

logic to handle the uncertainty.

We use probability in probabilistic reasoning because it provides a way to handle the uncertainty that is

the result of someone's laziness and ignorance.

In the real world, there are lots of scenarios, where the certainty of something is not confirmed, such as
"It will rain today," "behavior of someone for some situations," "A match between two teams or two players." These are probable sentences for which we can assume that it will happen but not sure about it, so here we use probabilistic reasoning.

Probability: Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur. The value of probability always remains between 0

and 1 that represent ideal uncertainties.

$0 \leq P(A) \leq 1$, where $P(A)$ is the probability of an event A.

$P(A) = 0$, indicates total uncertainty in an event A.

$P(A) =1$, indicates total certainty in an event A.

We can find the probability of an uncertain event by using the below formula.

$P(\neg A)$ = probability of a not happening event.

$P(\neg A) + P(A) = 1$.

Event: Each possible outcome of a variable is called an event.

Sample space: The collection of all possible events is called sample space.

Random variables: Random variables are used to represent the events and objects in the real world.

Prior probability: The prior probability of an event is probability computed before observing new information.

Posterior Probability: The probability that is calculated after all evidence or information has taken into account. It is a combination of prior probability and new information.

Conditional probability: Conditional probability is a probability of occurring an event when another event has already happened.
Let's suppose, we want to calculate the event A when event B has already occurred, "the probability of A under the conditions of B", it can be written as:

Where $P(A \wedge B)$= Joint probability of a and B

$P(B)$= Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as:

It can be explained by using the below Venn diagram, where B is occurred event, so sample space will be reduced to set B, and now we can only calculate event A when event B is already occurred by dividing the probability of $P(A \wedge B)$ by $P(B)$.

Example:

In a class, there are 70% of the students who like English and 40% of the students who likes English and mathematics, and then what is the percent of students those who like English also like mathematics?

Solution:

Let, A is an event that a student likes Mathematics

B is an event that a student likes English.

Hence, 57% are the students who like English also like Mathematics.

Why Reason Probabilistically?

- In many problem domains it isn't possible to create complete, consistent models of the world.
- Therefore agents (and people) must act in uncertain worlds (which the real world is).
- Want an agent to make rational decisions even when there is not enough information to prove that an action will work.
- Some of the reasons for reasoning under uncertainty:
- True uncertainty. E.g., flipping a coin.

o Theoretical ignorance. There is no complete theory which is known about the problem domain. E.g., medical diagnosis.

o Laziness. The space of relevant factors is very large, and would require too much work to list the complete set of antecedents and consequents. Furthermore, it would be too hard to use the enormous rules that resulted.

o Practical ignorance. Uncertain about a particular individual in the domain because all of the information necessary for that individual has not been collected.

9. DISCUSS THE A* SEARCH ALGORITHM. HOW DOES IT INCORPORATE BOTH COST AND HEURISTIC INFORMATION IN ITS SEARCH STRATEGY?

What is A* Search Algorithm?

A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.

Why A* Search Algorithm?

Informally speaking, A* Search algorithms, unlike other traversal techniques, it has "brains". What it means is that it is really a smart algorithm which separates it from the other conventional algorithms. This fact is cleared in detail in below sections.

And it is also worth mentioning that many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).

Explanation:

Consider a square grid having many obstacles and we are given a starting cell and a target cell. We want to reach the target cell (if possible) from the starting cell as quickly as possible. Here A* Search Algorithm comes to the rescue.

What A* Search Algorithm does is that at each step it picks the node according to a value-'f' which is a parameter equal to the sum of two other parameters – 'g' and 'h'. At each step it picks the node/cell having the lowest 'f', and process that node/cell.

We define 'g' and 'h' as simply as possible below

g = the movement cost to move from the starting point to a given square on the grid, following the path generated to get there.

h = the estimated movement cost to move from that given square on the grid to the final destination. This is often referred to as the heuristic, which is nothing but a kind of smart guess. We really don't know the actual distance until we find the path, because all sorts of things can be in the way (walls, water, etc.). There can be many ways to calculate this 'h' which are discussed in the later sections.

Algorithm :

We create two lists – Open List and Closed List (just like Dijkstra Algorithm)

// A* Search Algorithm

1. Initialize the open list
2. Initialize the closed list

   put the starting node on the open

   list (you can leave its f at zero)

3. while the open list is not empty

a) find the node with the least f on

the open list, call it "q"

b) pop q off the open list

c) generate q's 8 successors and set their

parents to q

d) for each successor

i) if successor is the goal, stop search

ii) else, compute both g and h for successor

successor.g = q.g + distance between

successor and q

successor.h = distance from goal to

successor (This can be done using many ways, we will discuss three heuristics-

Manhattan, Diagonal and Euclidean Heuristics)

successor.f = successor.g + successor.h

iii) if a node with the same position as

successor is in the OPEN list which has a

lower f than successor, skip this successor

iv) if a node with the same position as

successor is in the CLOSED list which has

a lower f than successor, skip this successor

otherwise, add the node to the open list

end (for loop)

v) push q on the closed list

end (while loop)

10. EXPLAIN THE RELATIONSHIP BETWEEN AN "AGENT" AND ITS "ENVIRONMENT."

An AI system is composed of an agent and its environment. The agents act in their environment. The environment may contain other agents.

What are Agent and Environment?

An **agent** is anything that can perceive its environment through **sensors** and acts upon that environment through **effector**.

A human agent has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effector.

A robotic agent replaces cameras and infrared range finders for the sensors, and various motors and actuators for effector.

A software agent has encoded bit strings as its programs and actions.

Agent Terminology :

Performance Measure of Agent − It is the criteria, which determines how successful an agent is.

Behavior of Agent − It is the action that agent performs after any given sequence of percepts.

Percept − It is agent's perceptual inputs at a given instance.

Percept Sequence − It is the history of all that an agent has perceived till date.

Agent Function − It is a map from the precept sequence to an action.

Rationality:

Rationality is nothing but status of being reasonable, sensible, and having good sense of judgment.

Rationality is concerned with expected actions and results depending upon what the agent has perceived. Performing actions with the aim of obtaining useful information is an important part of rationality.

What is Ideal Rational Agent?

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

Its percept sequence

Its built-in knowledge base

Rationality of an agent depends on the following −

The performance measures, which determine the degree of success.

Agent’s Percept Sequence till now.

The agent’s prior knowledge about the environment.

The actions that the agent can carry out.

A rational agent always performs right action, where the right action means the action that causes the agent to be most successful in the given percept sequence. The problem the agent solves is characterized by Performance Measure, Environment, Actuators, and Sensors (PEAS).

The Structure of Intelligent Agents

Agent’s structure can be viewed as −

Agent = Architecture + Agent Program

Architecture = the machinery that an agent executes on.

Agent Program = an implementation of an agent function.

Simple Reflex Agents

They choose actions only based on the current percept.

They are rational only if a correct decision is made only on the basis of current precept.

Their environment is completely observable.

Condition-Action Rule − It is a rule that maps a state (condition) to an action.

Model Based Reflex Agents

They use a model of the world to choose their actions. They maintain an internal state.

Model − knowledge about “how the things happen in the world”.

Internal State − It is a representation of unobserved aspects of current state depending on percept history.

Updating the state requires the information about −

How the world evolves.

How the agent’s actions affect the world.

Goal Based Agents

They choose their actions in order to achieve goals. Goal-based approach is more flexible than reflex agent since the knowledge supporting a decision is explicitly modeled, thereby allowing for modifications.

Goal − It is the description of desirable situations.

Utility Based Agents

They choose actions based on a preference (utility) for each state.

Goals are inadequate when −

There are conflicting goals, out of which only few can be achieved.

Goals have some uncertainty of being achieved and you need to weigh likelihood of success against the importance of a goal.

The Nature of Environments

Some programs operate in the entirely artificial environment confined to keyboard input, database, computer file systems and character output on a screen.

In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a very detailed, complex environment. The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the real as well as an artificial environment.

The most famous artificial environment is the Turing Test environment, in which one real and other artificial agents are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

Turing Test

The success of an intelligent behavior of a system can be measured with Turing Test.

Two persons and a machine to be evaluated participate in the test. Out of the two persons, one plays the role of the tester. Each of them sits in different rooms. The tester is unaware of who is machine and who is a human. He interrogates the questions by typing and sending them to both intelligences, to which he receives typed responses.

This test aims at fooling the tester. If the tester fails to determine machine’s response from the human response, then the machine is said to be intelligent.

Properties of Environment:

The environment has multifold properties −

Discrete / Continuous − If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).

Observable / Partially Observable − If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.

Static / Dynamic − If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.

Single agent / Multiple agents − The environment may contain other agents which may be of the same or different kind as that of the agent.

Accessible / Inaccessible − If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.

Deterministic / Non-deterministic − If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.

Episodic / Non-episodic − In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

11. EXPLAIN FORWARD AND BACKWARD CHAINING IN AI.

Forward Chaining and backward chaining in AI

In artificial intelligence, forward and backward chaining is one of the important topics, but before understanding forward and backward chaining lets first understand that from where these two terms came.

Inference engine:

The inference engine is the component of the intelligent system in artificial intelligence, which applies logical rules to the knowledge base to infer new information from known facts. The first inference engine was part of the expert system. Inference engine commonly proceeds in two modes, which are:

- Forward chaining
- Backward chaining

Horn Clause and Definite clause:

Horn clause and definite clause are the forms of sentences, which enables knowledge base to use a more restricted and efficient inference algorithm. Logical inference algorithms use forward and backward chaining approaches, which require KB in the form of the first-order definite clause.

Definite clause: A clause which is a disjunction of literals with exactly one positive literal is known as a definite clause or strict horn clause.

Horn clause: A clause which is a disjunction of literals with at most one positive literal is known as horn clause. Hence all the definite clauses are horn clauses.

Example: ($\neg$ p V $\neg$ q V k). It has only one positive literal k.

It is equivalent to p $\wedge$ q $\rightarrow$ k.

A. Forward Chaining

Forward chaining is also known as a forward deduction or forward reasoning method when using an inference engine. Forward chaining is a form of reasoning which start with atomic sentences in the knowledge base and applies inference rules (Modus Ponens) in the forward direction to extract more data until a goal is reached.

The Forward-chaining algorithm starts from known facts, triggers all rules whose premises are satisfied, and add their conclusion to the known facts. This process repeats until the problem is solved.

Properties of Forward-Chaining:

It is a down-up approach, as it moves from bottom to top.

It is a process of making a conclusion based on known facts or data, by starting from the initial state and reaches the goal state.

Forward-chaining approach is also called as data-driven as we reach to the goal using available data.

Forward -chaining approach is commonly used in the expert system, such as CLIPS, business, and production rule systems.

B. Backward Chaining:

Backward-chaining is also known as a backward deduction or backward reasoning method when using an inference engine. A backward chaining algorithm is a form of reasoning, which starts with the goal and works backward, chaining through rules to find known facts that support the goal.

Properties of backward chaining:

It is known as a top-down approach.

Backward-chaining is based on modus ponens inference rule.

In backward chaining, the goal is broken into sub-goal or sub-goals to prove the facts true.

It is called a goal-driven approach, as a list of goals decides which rules are selected and used.

Backward -chaining algorithm is used in game theory, automated theorem proving tools, inference engines, proof assistants, and various AI applications.

The backward-chaining method mostly used a depth-first search strategy for proof.

Forward Chaining		Backward Chaining
1.	When based on available data a decision is taken then the process is called as Forward chaining.	Backward chaining starts from the goal and works backward to determine what facts must be asserted so that the goal can be achieved.
2.	Forward chaining is known as data-driven technique because we reaches to the goal using the available data.	Backward chaining is known as goal-driven technique because we start from the goal and reaches the initial state in order to extract the facts.
3.	It is a bottom-up approach.	It is a top-down approach.
4.	It applies the Breadth-First Strategy.	It applies the Depth-First Strategy.
5.	Its goal is to get the conclusion.	Its goal is to get the possible facts or the required data.
6.	Slow as it has to use all the rules.	Fast as it has to use only a few rules.

7.	It operates in forward direction i.e it works from initial state to final decision.	It operates in backward direction i.e it works from goal to reach initial state.
8.	Forward chaining is used for the planning, monitoring, control, and interpretation application.	It is used in automated inference engines, theorem proofs, proof assistants and other artificial intelligence applications.