T.Y. BSC. (IT): Sem. V (Elective-I)

Artificial Intelligence

Mumbai University Examination Paper Solution: Nov-22

Q1. Attempt any three of the following [15]

Q1(a)Explain Artificial Inelligence with Turing Test approach. [5]

Ans.

- John McCarthy who has coined the word "Artificial Intelligence" in 1956, has defined AI as the "Science and engineering of making intelligent machines".
- Artificial Intelligence (AI) is relevant to any intellectual task where the machine needs to take some decision or choose the next action based on the current state of the system.
- In simple words, Artificial Intelligent System works like a Human Brain, where a machine or software shows intelligence while performing given tasks.
- It is also known as expert systems.
- To judge whether the system can act like a human, Sir Alan Turing had designed a test known as Turing test.
- In Turing test, a computer needs to interact with a human interrogator by answering his questions in written format.
- Computer passes the test if a human interrogator, cannot identify whether the written responses are from a person or a computer.
- For this test, the computer would need to possess the following capabilities:

  1. Natural Language Processing (NLP):

  This unit enables computer to interpret the English language and communicate successfully.

  2. Knowledge Representation:

This unit is used to store knowledge gathered by the system through input devices.

3.Automated Reasoning: This unit enables to analyze the knowledge stored in the system and makes new inferences to answer questions.

4. Machine Learning: This unit learns new knowledge by taking current input from the environment and adapts to new circumstances, thereby enhancing the knowledge base of the system.

To pass total Turing test, the computer will also need to have computer vision, which is required to perceive objects from the environment and Robotics, to manipulate those objects.

Q1(b)Describe the contribution of Philosophy and Mathematics to Artificial Intelligence.[5]

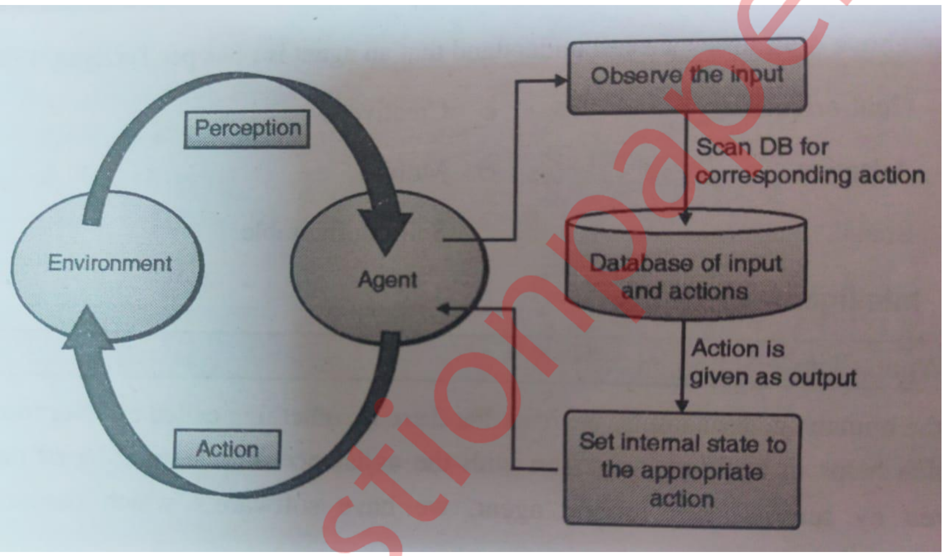
Ans.

- Philosophy has helped to shape our understanding of intelligence by providing us with a conceptual framework for thinking about what it means to be intelligent.
- For example, philosophical debates about the nature of consciousness have led to new insights into how we might create machines that are capable of experiencing subjective states.
- Similarly, philosophical discussions about the relationship between language and thought have helped to inform the development of natural language processing algorithms.
- Mathematics, on the other hand, has provided a framework for modeling and understanding complex systems.
- This has been particularly important in the development of machine learning algorithms, which are used to train AI systems to recognize patterns in data.

- The mathematical techniques used in machine learning, such as linear algebra and calculus, have also been applied to other areas of AI research, such as computer
- Overall, philosophy and mathematics have played an important role in the development of AI by providing conceptual frameworks and mathematical tools that have helped to shape our understanding of intelligence and create new AI technologies.

Q1(c)State the relationship between agents and environment. [5]

Ans


Perception
Environment
Agent
Action
Observe the input
Scan DB for corresponding action
Database of input and actions
Action is given as output
Set internal state to the appropriate action

- Agent is something that perceives its environment through sensors and acts upon that environment through effectors or actuators.
- From above image it can be observed how agent and environment interact with each other.
- Every time environment changes the agent first observes the environment through its sensors and get the input, then scans the database of input and actions for the

corresponding action for given input and lastly sets the internal state to the appropriate action.

- Let's understand this working with a real life example. Consider you are an agent and your surroundings is an environment.
- Now, take a situation where you are cooking in kitchen and by mistake you touch a hot pan. We will see what happens in this situation step by step.
- Your touch sensors take input from environment (i.e. you have touched some hot element), then it asks your brain if it knows "what action should be taken when you go near hot elements ?"
- Now the brain will inform your hands (actuators) that you should immediately take it away from the hot element otherwise it will burn. Once this signal reaches your hand you will take your hand away from the hot pan.

Q1(d)What is PEAS description? Explain with two examples. [5]

Ans

- PEAS stands for Performance Measure, Environment, Actuators, and Sensors. It is the short form used for performance issues grouped under Task Environment.
- You might have seen driverless/ self driving car videos of Audi/ Volvo/ Mercedes, etc. To develop such driverless cars we need to first define PEAS parameters.
- Performance Measure: It the objective function to judge the performance of the agent. For example, in case of pick and place robot, number of correct parts in a bin can be the performance measure
- Environment: It the real environment where the agent need to deliberate actions.

- Actuators: These are the tools, equipment or organs using which agent performs actions in the environment. This works as the output of the agent.
- Sensors: These are the tools, equipment or organs using which agent captures the state of the environment. This works as the input to the agent.
- Example:

A.**Medical diagnosis system**:

1. performance measure.:

System should make use of sterilized instruments to ensure the safety (healthiness) of the patient,

The automated system results should not be very costly otherwise overall expenses of the patient may increase, Lawsuits. Medical diagnosis system should be legal.

2. Environment:

Patient, Doctors, Hospital Environment.

3. Sensors: Screen, printer

4. Actuators:

Keyboard and mouse which is useful to make entry of symptoms, findings, patient's answers to given questions. Scanner to scan the reports, camera to click pictures of patients.

B.Soccer player Robot:

1. Performance Measures: Number of goals, speed, legal game.

2.Environment: Team players, opponent team players, playing ground, goal net.

3. Sensors: Camera, proximity sensors, infrared sensors.

4. Actuators: Joint angles, motors.

Q1(e) Explain following task environment [5]

- **Single agent vs. multi agent**
- **episodic vs. sequential**

Ans

1. Single agent vs. multi agent:

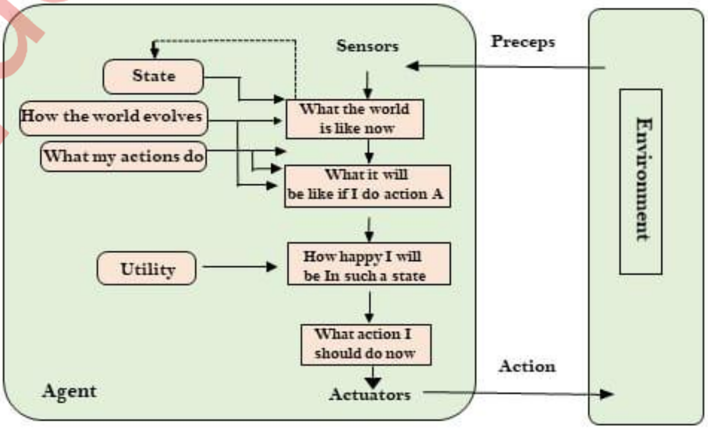
- A task environment is the set of circumstances under which a given agent or group of agents operate.
- In AI, there are two types of task environments: single-agent and multi-agent environments.
- In a single-agent environment, there is only one agent operating in the environment.
- The agent must make decisions based on the information it has available to it, and it must take into account the goals it is trying to achieve.
- Examples of single-agent environments include chess games, where the agent is the computer program that plays the game, and autonomous vehicles, where the agent is the vehicle itself.
- In a multi-agent environment, there are multiple agents operating in the environment, each with its own set of goals and constraints.
- The agents must interact with each other and coordinate their actions to achieve their individual goals.
- Examples of multi-agent environments include social networks, where agents are users of the network, and traffic systems, where agents are vehicles on the road.

2. Episodic vs. sequential:

- The task environment can also be classified as episodic or sequential.
- In an episodic environment, an agent's current decision is based only on the current state of the environment. Each decision is independent of previous decisions, and there is no need to take into account the history of previous states or decisions.
- Examples of episodic environments include playing a single game of chess or navigating a maze.
- In a sequential environment, an agent's current decision is based on the current state of the environment as well as the history of previous states and decisions.
- Each decision affects the future states of the environment, and the agent must take into account the long-term consequences of its decisions.
- Examples of sequential environments include playing a series of games of chess or navigating a changing environment such as a city.

Q1(f) Describe the structure of Utility based agent. [5]

- Agent is something that perceives its environment through sensors and acts upon that environment through effectors or actuators.


Sensors
Preceps
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Utility
How happy I will be In such a state
What action I should do now
Agent
Actuators
Action
Environment

- Utility function is used to map a state to a measure of utility of that state. We can define a measure for determining how advantageous a particular state is for an agent. To obtain this measure utility function can be used.
- The term utility is used to depict how "happy" the agent is to find out a generalized performance measure, various world states according to exactly how happy they would make an agent is compared.
- Take one example; you might have used Google maps to find out a route which can take you from source location to your destination location in least possible time. Same logic is followed by utility based automatic car driving agent.
- Goals utility based automatic car driving agent can be used to reach given location safely within least possible time and save fuel. So this car driving agent will check the possible routes and the traffic conditions on these routes and will select the route which can take the car at destination in least possible time safely and without consuming much fuel.

Q2 Attempt any three of the following [15]

Q2(a) Describe problem formulation of Vacuum World problem. [5]

Ans:

- Problem formulation is the process of deciding what states to be considered and what actions to be taken to achieve the goal. This is the first to be taken by any problem solving agent.
- State space: The state space of a problem is the set of all states reachable from the initial state by executing any sequence of actions.
- State is representation of all possible outcomes.

- The state space specifies the relation among various problem states thereby, forming a directed network or graph in which the nodes are states and the links between nodes represent actions.
- State Space Search: Searching in a given space of states pertaining to a problem under consideration is called a state space search.
- Path: A path is a sequence of states connected by a sequence of actions, in a given state space.
- **Problem formulation of Vacuum World problem:**
- **Initial State**:

  Identifying the dirty place to clean
- **Action:**

  Starts cleaning the place from left, right or corner.
- **Successor function:**

  Assigned place is cleaned.
- **Goal test:**

  Test whether the place is cleaned or not if any other dirt is there it clean that also and recheck whether the place is cleaned or not.
- **Path cost:**

  Whether the path choosen by agent is optimizing or not.

Q2 (b) Explain following terms: [5]

1. State space of the problem 2.Path in state space

3. Goal Test 4. Path Cost 5.Opimal Solution to problem

Ans:

- **State space of the problem:**

  The state space of a problem is the set of all states reachable from the initial state by executing any sequence of actions. State is representation of all possible outcomes.

- **Path in state space:**

  A path in state space is a sequence of states that lead from an initial state to a goal state in a problem-solving process. The path can be represented as a graph, where the nodes represent states and the edges represent transitions between states.

- **Goal Test:**

  It is a test to determine whether the current state is a goal state. In some problems the goal test can be carried out just by comparing current state with the defined goal state, called as explicit goal test. Whereas, in some of the problems, state cannot be defined explicitly but needs to be generated by carrying out some computations, it is called as implicit goal test.

- **Path Cost:**

  The path cost is the total cost of all the transitions or steps along a path in a problem-solving process. The cost can be defined based on various criteria, such as time, distance, energy, or resources required to perform each step. The path cost can be evaluated based on the sum of the costs of all the transitions along the path or the average cost per transition.

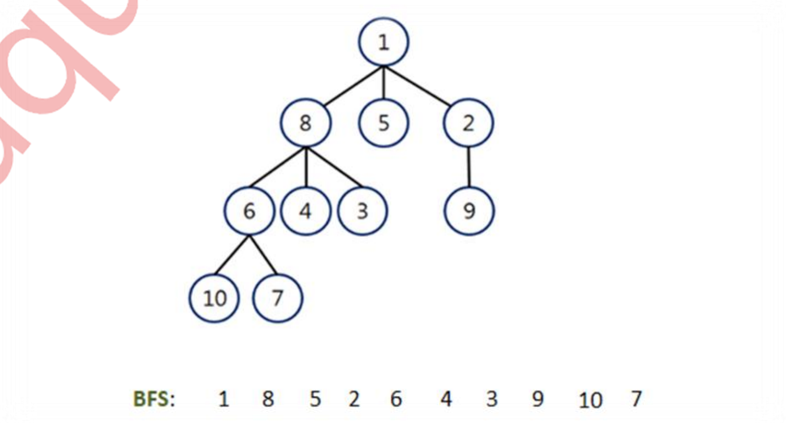
- **Optimal Solution to problem:**

An optimal solution to a problem is a solution that satisfies all the constraints of the problem and achieves the best possible outcome based on the criteria of the problem. The optimal solution can be defined based on various criteria, such as cost, time, quality, efficiency, or effectiveness.

Q2 (c) Give outline of Breadth First algorithm with respect to Artificial intelligence [5]

Ans:

- As the name suggest, in breadth first search algorithm tree is expanded breadth wise.
- The root node is expanded first, then all successor node of the root node are expanded then their successor node so on.
- In turn all the node at the depth in search tree are expanded first and then the search proceed for the next level node expansion.
- Thus, the Shallowest Unexpanded node will be selected for expansion.
- It starts from root node and visit all the nodes.
- It is implemented using queue which means first in first out.



Q2 (d) Write the local search algorithm, Explain the following concept [5]

1. Shoulder. 2. Global maximum 3.Local maximum

Ans:

- Local search algorithm is a type of heuristic search algorithm that starts with an initial solution and iteratively improves the solution by making small changes to it.
- The local search algorithm does not guarantee an optimal solution but rather a good solution in a reasonable amount of time.
- **The local search algorithm consists of the following steps:**

  1. Initialization:

  Start with an initial solution that satisfies the constraints of the problem.

  2. Evaluation:

  Evaluate the quality of the initial solution based on the objective function of the problem.

  3. Neighborhood generation:

  Generate a set of neighboring solutions by making small changes to the current solution.

  4. Selection:

  Select the best neighboring solution based on the objective function of the problem.

  5. Termination:

  If the stopping criterion is met, return the current solution as the output. Otherwise, go to step 3.

- **Shoulder:**

  In the context of local search algorithm, a shoulder is a flat region in the search space where many solutions have the same value. A shoulder can be thought of as a plateau where the search algorithm can get stuck and fail to explore other regions of the search space. This can lead to the algorithm converging to a sub-optimal solution.

- **local minimum:**

  A local minimum is a solution that is worse than all its neighboring solutions. It is a point in the search space where the search algorithm cannot improve the solution by making small changes. A local minimum can be a problem for local search algorithms because they can get stuck in this region and fail to explore other regions of the search space.

- **Global minimum** :

  A global minimum is the best possible solution in the search space. It is the solution that has the lowest value of the objective function. The global minimum is the optimal solution that the search algorithm is trying to find. The search algorithm may converge to a local minimum or get stuck in a shoulder, but it should eventually find the global minimum if given enough time and resources.

Q2 (e) Illustrate hill climbing method using 8 queen problem [5]

Ans:

- The 8 queen problem is a classic problem in which we need to place eight queens on an 8x8 chessboard such that no two queens attack each other. The problem can be solved using the hill climbing algorithm as follows:

  1. Initialization:

Start with an initial solution that satisfies the constraints of the problem. For example, we can place eight queens on the board such that each queen is in a different row and column.

2. Evaluation:

Evaluate the quality of the initial solution based on the objective function of the problem. In this case, the objective function is the number of pairs of queens that attack each other.

3. Neighborhood generation:

Generate a set of neighboring solutions by making small changes to the current solution. For example, we can move one queen to a different row in its column.

4. Selection:

Select the best neighboring solution based on the objective function of the problem. In this case, we want to select the neighboring solution that has the fewest number of pairs of attacking queens.

5. Termination:

If the stopping criterion is met, return the current solution as the output. Otherwise, go to step 3.

By repeating steps 3 to 5, we can continue to improve the solution until we reach a solution in which no two queens attack each other. However, the hill climbing algorithm can get stuck in a local maximum or a shoulder, which can lead to sub-optimal solutions.

Q2 (f) Explain the mechanism of genetic algorithm. [5]

Ans:

- GAs are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics.

- As such they represent an intelligent exploitation of a random search used to solve optimization problems.
- The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, especially those following the principles of "survival of the fittest" laid down by Charles Darwin.
- Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves towards better solutions.
- The solutions are represented in binary as strings of Os and Is, but other encodings are also possible. The evolution usually starts from a population of randomly generated individual and occurs in generations.
- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness). and modified to form a new population. The new population is then used in the next iteration of the algorithm.

Q3 Attempt any three of the following [15]

Q3 (a) Explain minimax algorithm in detail. [5]

Ans:

- The Minimax algorithm is a decision-making algorithm that is used in game theory, artificial intelligence, and decision theory. It is used to determine the best move for a player, assuming that the other player is also playing optimally. The algorithm works by recursively examining the possible moves that can be made by both players and assigning a score to each move. The score is based on the outcome of the game, assuming that both players are playing optimally.

- **The algorithm works as follows:**

1. The algorithm starts by examining the current state of the game and determining which player is currently playing.

2. The algorithm then generates a list of all possible moves that can be made by the current player.

3. For each possible move, the algorithm simulates the game by making the move and examining the resulting state of the game.

4. The algorithm then recursively calls itself, but with the roles of the players reversed. This means that the algorithm is now examining the possible moves that can be made by the other player.

5. The algorithm continues to recursively call itself until it reaches the end of the game. At this point, the algorithm assigns a score to the final state of the game. The score is based on whether the current player won, lost, or the game ended in a draw.

6. The algorithm then works its way back up the recursive tree, assigning a score to each possible move based on the score of the final state of the game that it leads to.

7. Finally, the algorithm chooses the move that leads to the highest score for the current player.

The Minimax algorithm is commonly used in games such as chess, checkers, and tic-tac-toe. It is also used in decision theory to determine the best course of action in situations where there are multiple possible outcomes.

Q3 (b) Describe the technique of Alpha- Beta pruning [5]

Ans:

- Alpha-Beta pruning is a search algorithm that is used in game theory and artificial intelligence to reduce the number of nodes that need to be evaluated in a search tree. The algorithm works by eliminating branches of the search tree that are guaranteed to be worse than previously examined branches.

The algorithm works as follows:

1. The algorithm starts by examining the root node of the search tree.

2. The algorithm then generates a list of all possible moves that can be made from the current node.

3. For each possible move, the algorithm simulates the game by making the move and examining the resulting state of the game.

4. The algorithm then recursively calls itself, but with the roles of the players reversed. This means that the algorithm is now examining the possible moves that can be made by the other player.

5. The algorithm evaluates the score of the current node based on the scores of its child nodes.

6. The algorithm maintains two values, alpha and beta. Alpha represents the best score that the maximizing player can achieve, while beta represents the best score that the minimizing player can achieve.

7. As the algorithm traverses the search tree, it updates the values of alpha and beta based on the scores of the nodes that it has examined.

8. If the algorithm encounters a node where the score is worse than the current value of alpha or beta, it prunes the branch of the search tree below that node. This is because the other player would never choose that branch, as there is already a better option available.

9. The algorithm continues to recursively call itself until it reaches the end of the game. At this point, the algorithm assigns a score to the final state of the game. The score is based on whether the current player won, lost, or the game ended in a draw.

10. The algorithm then works its way back up the recursive tree, assigning a score to each possible move based on the score of the final state of the game that it leads to.

11. Finally, the algorithm chooses the move that leads to the highest score for the current player.

Q3 (c) Write short note on Kriegspiel's partially observable chess [5]

Ans:

- Kriegspiel is a chess variant that is played by three players, each of whom controls one side of the board.
- It is a partially observable game, as each player can only see their own pieces and the squares that their pieces can move to.
- The game is played with a special board that has two screens, one for each of the non-moving players.
- The screens are placed on either side of the board, and each player has a set of pieces that are hidden from the other players.
- The game starts with each player placing their pieces on their side of the board.
- The game then proceeds with each player taking turns moving their pieces.

- When a player makes a move, they must announce the move to the other players, who then use their screens to check whether the move is legal or not.
- If a player's piece moves onto a square occupied by an opponent's piece, the player must announce that a capture has taken place, but they do not reveal which piece was captured. The captured piece is removed from the board, and the game continues.
- The game ends when one player checkmates the other player's king. The player who checkmates the king wins the game.
- Kriegspiel is a challenging game, as players must use deduction and logical reasoning to determine the positions of their opponent's pieces.
- It is often used as a training tool for chess players, as it helps to develop their ability to visualize the board and think several moves ahead.

Q3 (d) what is knowledge based agent? Explain its importance in problem solving technique [5]

Ans:

- A knowledge-based agent is an artificial intelligence system that uses a knowledge base to reason and make decisions.
- It is designed to solve problems by using its knowledge of the world to make inferences, draw conclusions, and take actions.
- The knowledge base is a collection of facts, rules, and heuristics that the agent uses to represent its knowledge of the world.
- The agent can use this knowledge to reason about the current situation, make predictions about the future, and plan actions to achieve its goals.

- Knowledge-based agents are important in problem-solving techniques because they can solve complex problems that require reasoning and decision-making.
- They can reason about the consequences of different actions and choose the one that is most likely to achieve their goals.
- They can also adapt to changing situations by updating their knowledge base with new information.
- Knowledge-based agents are used in a wide range of applications, such as expert systems, decision support systems, and intelligent tutoring systems.
- They are also used in robotics, natural language processing, and computer vision.
- The importance of knowledge-based agents in problem-solving techniques lies in their ability to automate decision-making processes.
- They can analyze large amounts of data, identify patterns, and make predictions based on this data.
- They can also learn from experience, improving their performance over time.

Q3 (e) Write short note on Wumpus world problem **[5]**

Ans:

- The Wumpus World problem is a classic artificial intelligence problem that involves navigating a maze-like environment to find gold while avoiding deadly pits and a monster called the Wumpus.
- The Wumpus World is represented as a grid of squares, each of which can contain a pit, gold, the Wumpus, or nothing at all.

- The agent, which is an intelligent agent, must navigate the environment to find the gold while avoiding the pits and the Wumpus.
- The agent has a set of sensors that allow it to perceive the environment. It can detect whether it is adjacent to a pit, the Wumpus, or gold.
- It can also detect whether it has bumped into a wall.
- The agent can take a set of actions to move around the environment.
- It can move forward, turn left or right, or shoot an arrow to kill the Wumpus.
- However, the agent has a limited supply of arrows, so it must use them wisely.
- The Wumpus World problem is important in artificial intelligence because it is a classic example of a problem that requires reasoning and decision-making in an uncertain environment.
- It requires the agent to use its knowledge of the environment to make inferences, plan actions, and take risks.
- The Wumpus World problem has been used as a benchmark for evaluating the performance of intelligent agents.
- .It has also been used to test different algorithms for solving problems in uncertain environments, such as Bayesian networks, decision trees, and reinforcement learning.
- Overall, the Wumpus World problem is a classic artificial intelligence problem that involves navigating a maze-like environment to find gold while avoiding deadly pits and a monster called the Wumpus.
- It is an important problem in artificial intelligence because it requires reasoning and decision-making in an uncertain environment.

Q3 (f) Explain Forward chaining algorithm for propositional definite clauses [5]

Ans:

- The Forward Chaining algorithm is a method for deriving the logical consequences of a set of propositional definite clauses.
- It is a type of inference algorithm that is used in artificial intelligence and automated reasoning.
- The algorithm starts with a set of known facts and uses inference rules to derive new facts.
- The algorithm continues to apply inference rules until no more new facts can be derived.
- The input to the algorithm is a set of propositional definite clauses, which are statements of the form "if A then B", where A and B are propositional atoms.
- The algorithm also takes as input a set of known facts, which are the initial premises.
- The algorithm works by repeatedly applying the Modus Ponens rule, which states that if we know that A implies B, and we know that A is true, then we can conclude that B is true.
- The algorithm starts with the known facts and applies the Modus Ponens rule to derive new facts.
- If a new fact is derived, it is added to the set of known facts. The algorithm continues to apply the Modus Ponens rule until no more new facts can be derived.
- The algorithm terminates when either a contradiction is derived, or when the goal is derived.
- A contradiction is derived when the algorithm derives a fact and its negation.
- The goal is derived when the algorithm derives a fact that matches the goal.

Q4 Attempt any three of the following [15]

Q4 (a) What is First order logic ? Explain syntax and semantics of first order Logic. [5]

Ans:

- Because of the inadequacy of PL discussed above there was a need for more expressive type of logic. Thus First-Order Logic (FOL) was developed.
- FOL is more expressive than PL, it can represent information using relations, variables and quantifiers, e.g., which was not possible with propositional logic.
- "Gorilla is Black" can be represented as :

  Gorilla(x) Black(x)
- "It is Sunday today" can be represented as :

  today(Sunday)
- First Order Logic(FOL)is also called as First Order Predicate Logic (FOPL).
- Since FOPL is much more expressive as a knowledge representation language than PL it is more commonly used in artificial intelligence.
- FOL symbol can be a constant term, a variable term or a function.
- Assuming that "X" is a domain of values, we can define a term with following rules:

  1. Constant term: It is a term with fixed value which belongs to the domain.

  2. Variable term: It is a term which can be assigned values in the domain.

  3. Function: Say "f" is a function of "n" arguments. If we assume that t1,12,,tn are terms then f(t1,12,.., tn) is also called as a term.
- All the terms are generated by applying the above three protocols.

- First order predicate logic makes use of propositional logic as a base logic, so the connectives used in PL and FOPL are common. Hence, it also supports conjunction, v disjunction, negation, implication and double implication.
- Ground Term: If a term does not have any variables it is called as a ground term. A sentence in which all the variables are quantified is called as a "well-formed formula".
- Every ground term is mapped with an object.
- Every condition (predicate) is mapped to a relation.
- A ground atom is considered as true if the predicate's relation holds between the terms objects.
- Rules in FOL: In predicate logic rule has two parts predecessor and successor. If the predecessor is evaluated to TRUE successor will be true. It uses the implication →symbol. Rule represents If-then types of sentences. Example: The sentence "If the bag is of blue colour, I will buy it." Will be represented as
- colour (bag, blue) → buy(bag).

Q4 (b) Write a short note on Universal and Existential quantifier with suitable example [5]

Ans:

Quantifiers:

Apart from these connectives FOPL makes use of quantifiers. As the name suggests they quantify the number of variables taking part in the relation or obeying the rule.

1. Universal Quantifier ' ∀ ':

Pronounced as "for all" and it is applicable to all the variables in the predicate

"∀x A" means A is true for every replacement of x.

Example: "Every Gorilla is Black" can be represented as:

"$\forall x$ (Gorilla(x)$\rightarrow$ Black(x))

2. Existential Quantifier ' ∃ '

Pronounced as "there exists".

"∃ x A" means A is true for at least one replacement of x.

Example: "There is a white dog" can be represented as,

∃ x (Dog(X) white(X))

Q4 (c) Explain the steps of knowledge Engineering process project in first order logic [5]

Ans:

Knowledge engineering is a process of knowledgebase construction.

It requires a knowledge engineer to investigate a particular domain, learn the important concepts in that domain, and create a formal representation and logical relations among objects in that domain.

Following is the general knowledge engineering process which can be applied to problem of any domain.

1. Identify the task:

This step is analogues to PEAS process while designing an agent While identifying task, the knowledge engineer must define the scope of knowledgebase and the range of questions that can be answered through the database**.**

2. Assemble the relevant knowledge:

Assembling the relevant knowledge of that particular domain is called the process of knowledge acquisition. In this process knowledge engineer learns how the domain actually works and can determine the scope of the knowledgebase as per the identified tasks.

3. Defining vocabulary:

Defining a complete vocabulary including predicates, functions and constants is a very important step of knowledge engineering. This process transforms the domain level concepts to logic level symbols. It should be exhaustive and precise.

4. Encoding of general knowledge about the domain :

In this step the knowledge engineer defines axioms for all the vocabulary terms by define meaning of each term.

5. Encode the problem :

In this step, the specific problem instance is encoded using the defined ontology. This step will be very easy if the ontology is defined properly. Encoding means writing atomic sentences about problem instances which are already part of ontology. It can be analogues to input data for a computer program.

6. Query the Knowledge base :

Once all the above steps are done, all input for the system is set and now is a time to generate some output from the system.

Q4 (d) Write a short note on unification process [5]

Ans:

- Unification process is a process of merging two or more entities into one. It is a common practice in various fields such as politics, science, and technology.
- The goal of unification is to create a stronger, more efficient, and more effective entity.

- In politics, unification can refer to the merging of two or more political parties to form a stronger political entity.
- This is often done to increase the chances of winning an election or to create a more unified front on a particular issue.
- In science, unification refers to the process of combining two or more theories into a single, more comprehensive theory.
- For example, the unification of electromagnetism and the weak nuclear force into the electroweak force.
- In technology, unification refers to the process of combining two or more technologies into a single, more efficient technology.
- For example, the unification of a camera and a phone into a single device. This can often result in a more streamlined and user-friendly experience**.**

Q4 (e) Explain Datalog used in first order definite clause [5]

Ans:

- Datalog is a logic programming language that is based on the first-order definite clause. It is used to create and manipulate relational databases.
- In Datalog, the facts and rules are represented using first-order definite clauses.
- A first-order definite clause is a rule that has a head and a body.
- The head is a single predicate that represents the conclusion of the rule, while the body is a conjunction of predicates that represents the conditions that must be met for the conclusion to be true.

- In Datalog, the predicates are used to represent the relations between the objects in the database.
- The rules are used to define the relationships between the predicates.
- The facts are used to represent the data that is stored in the database.
- Datalog is a declarative language, which means that the programmer only needs to specify what they want the program to do, rather than how to do it.
- The Datalog engine takes care of the implementation details.

Q4 (f) Describe Backward Channing algorithm for first order definite clauses [5]

Ans:

- Backward Chaining is an algorithm used in artificial intelligence and logic programming to prove the validity of a query by working backward from the conclusion to the premises.
- It is commonly used in first-order logic, where the premises are represented by a set of first-order definite clauses.
- The Backward Chaining algorithm starts with a goal or query that needs to be proven.
- It then searches for all the rules in the knowledge base that have the goal as their conclusion.
- It then tries to prove the premises of these rules by recursively applying the Backward Chaining algorithm to each premise.
- The algorithm continues in this manner until it either finds a fact that matches the query or it exhausts all the possible rules and fails to find a match.
- If it finds a fact that matches the query, it returns the fact as the answer. If it fails to find a match, it returns "false".

- The algorithm can be optimized by using a technique called memoization, which stores the results of previous queries to avoid redundant computations.
- Overall, the Backward Chaining algorithm is an algorithm used in artificial intelligence and logic programming to prove the validity of a query by working backward from the conclusion to the premises. It is commonly used in first-order logic, where the premises are represented by a set of first-order definite clauses.

Q5 Attempt any three of the following [15]

Q5 (a) Explain Planning domain definition language description for an Air cargo planning problem. [5]

Ans:

- Planning Domain Definition Language (PDDL) is a computer-readable language used to define a planning problem.
- In the context of an air cargo planning problem, PDDL would be used to describe the various states of the problem, the actions that can be taken, and the goals that need to be achieved.
- The initial state of the problem might include information about where the cargo is located, where the planes are located, and their capacities.
- For example, the initial state might describe that cargo A is located at airport X and cargo B is located at airport Y, while there are planes available at airports X and Y with different capacities.
- The actions that can be taken might include loading and unloading cargo, as well as moving the planes from one location to another.

- For example, an action could be defined as "Load(cargo, plane, airport)", which would indicate that a specific piece of cargo is being loaded onto a specific plane at a specific airport.
- Similarly, an action could be defined as "Fly(plane, from, to)", which would indicate that a specific plane is flying from one airport to another.
- The goals that need to be achieved might include delivering the cargo to a specific location within a certain timeframe.
- By using PDDL, it's easier for a computer to understand the problem and find a solution. PDDL provides a formal language for describing the problem, which can be used to generate a plan for achieving the specified goals.
- The plan might involve a sequence of actions that need to be taken, such as loading the cargo onto a plane, flying the plane to the destination, and unloading the cargo at the destination.

Q5 (b) Explain forward (progression) state space search algorithm with an example [5]

Ans:

- Forward state space search, also known as progression search, is a search algorithm used in AI to find a solution to a problem by exploring the state space of the problem.
- The algorithm starts from the initial state of the problem and applies the available actions to generate new states.
- It continues this process until it reaches a goal state or determines that no solution exists.
- Here is an example of forward state space search algorithm:

- Suppose we have a problem of finding a path from a starting city A to a destination city D. We have a map of the cities and the roads connecting them. The initial state of the problem is that we are at city A, and the goal state is to reach city D. The actions we can take are to move from one city to another along a road.
- The algorithm starts by expanding the initial state of the problem, which is the state of being in city A.
- It then applies the available actions to generate new states, which are the states of being in the cities that are connected to city A by a road.
- The algorithm then applies the available actions to each of these new states to generate further new states.
- In summary, the forward state space search algorithm starts from the initial state of the problem and applies the available actions to generate new states. It continues this process until it reaches a goal state or determines that no solution exists.

Q5 (c) Explain Hierarchical planning [5]

Ans:

- Hierarchical planning is a problem-solving technique that involves breaking down a complex problem into smaller sub-problems that are easier to solve.
- It is a top-down approach to planning, where the high-level goals are decomposed into subgoals, and each subgoal is further decomposed into smaller subgoals until the problem can be solved at the lowest level.
- The main idea behind hierarchical planning is to divide the problem into manageable sub-problems, and then solve each sub-problem separately.

- The sub-problems are organized in a hierarchy, where each level of the hierarchy represents a different level of abstraction.
- The higher levels of the hierarchy deal with the overall goals and constraints of the problem, while the lower levels deal with the details of the problem.
- Hierarchical planning is useful because it allows us to focus on the details of each sub-problem separately, without getting bogged down by the complexity of the overall problem.
- It also allows us to reuse solutions to sub-problems that have already been solved, which can save time and effort.

Q5 (d) Write short note on sensor less planning problem **[5]**

Ans:

- Sensorless planning is a type of planning problem where the agent does not have access to any sensors or other means of perceiving the environment.
- In other words, the agent has no way of directly observing the state of the world, and must instead rely on a model of the world to make decisions.
- This type of planning problem is particularly challenging because the agent must make decisions based on incomplete information.
- The agent must predict the outcomes of its actions and plan accordingly, without being able to directly observe the effects of its actions.
- One approach to solving sensorless planning problems is to use a model-based approach.
- The agent creates a model of the world, which includes information about the state of the world, the actions available to the agent, and the outcomes of those actions. The agent then uses this model to simulate the effects of its actions and plan accordingly.

- Another approach is to use a learning-based approach. The agent learns a model of the world through trial-and-error, by observing the outcomes of its actions and adjusting its model accordingly.
- This approach can be particularly useful in situations where the environment is complex and difficult to model accurately.
- Sensorless planning problems are common in many real-world applications, such as robotics, where sensors may be limited or unreliable.
- Solving these problems requires sophisticated planning algorithms that can reason about the effects of actions without direct observation of the environment.

Q5 (e) What are an events explain its importance [5]

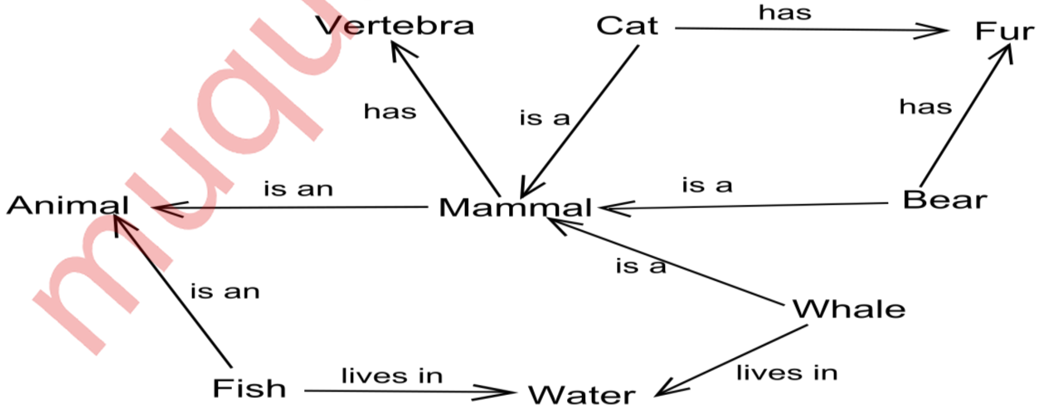
Ans:

- Events related to AI are crucial for advancing the field of artificial intelligence. These events bring together researchers, developers, and industry professionals from around the world to share their ideas, research, and latest developments in AI technology.
- They provide a platform for attendees to learn about the latest advancements in AI, including new algorithms, techniques, and applications.
- One of the primary benefits of AI events is that they promote innovation and collaboration.
- Researchers and developers from different organizations and backgrounds can come together to share their ideas and collaborate on new projects. This can lead to breakthroughs in AI technology that might not have been possible without these events.

- Another important benefit of AI events is that they provide an opportunity for attendees to learn about the latest trends and developments in AI.
- This can help attendees stay up-to-date with the latest advancements in the field, and can help them to identify new opportunities for research and development.
- In addition to promoting innovation and collaboration, AI events can also help to raise awareness about the potential benefits and risks of AI.
- These events provide a platform for discussions about the ethical and responsible development and deployment of AI systems. They can also help to identify and address potential risks associated with AI, such as bias, privacy concerns, and job displacement.
- Finally, AI events can help to create lasting connections and relationships between attendees.
- These connections can lead to future collaborations and partnerships, and can help to build a strong and supportive community around AI research and development.

Q5 (f) What is semantic network ? Show the semantic network representation with a suitable example **[5]**

Ans:



- A semantic net or semantic network is a knowledge representation technique used for propositional information, so it is also called a propositional net.
- In semantic networks the knowledge is represented as objects and relationships between objects.
- They are two dimensional representations of knowledge. It conveys meaning. Relationships provide the basic structure for organizing knowledge.
- It uses graphical notations to draw the networks. Mathematically a semantic net can be defined as a labeled directed graph.
- As nodes are associated with other nodes semantic nets are also referred to as associative nets.
- Semantic nets consist of nodes, links and link labels. Nodes of the graph denote objects while the links indicate relations among the objects.