DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE (AUTONOMOUS)

DEPARTMENT OF
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

II YEAR – IV SEMESTER

U23AIT41 ARTIFICIAL INTELLIGENCE

QUESTION BANK

ACADEMIC YEAR 2025-2026

U23AIT41 ARTIFICIAL INTELLIGENCE

UNIT I

INTELLIGENT AGENTS

Introduction to AI – Agents and Environments – Concept of rationality – Nature of environments – Structure of agents Problem solving agents – search algorithms – uninformed search strategies

PART A

1) What Are the Different Types of Intelligent Agents?

Simple reflex agent, Model-based reflex agent, Goal-based agent, Utility-based agent, Learning agent

2) What Are Intelligent Agents Used For?

IAs can serve as automated online assistants that perceive customers' needs to provide personalized customer service. These agents typically have a dialog system, an avatar, and an expert system that serves specialized functions. They can also optimize coordination between human groups online. Examples, as mentioned above, include Alexa and Siri. A smart vacuum cleaner that cleans an area by moving from one tile to another is also an IA.

3) What is An Intelligent Agent?

An intelligent agent (IA) is a computer software system that's capable of acting independently to achieve certain goals and responding to people or events that are happening around it. It is programmed using the field of artificial intelligence (AI) called ―machine learning (ML)‖ and equipped with sensors that allow it to observe and adapt to situations.

4) What Are the Characteristics of Agents?

. An agent is a computer software system whose main characteristics are situatedness, autonomy, adaptivity, and sociability.

5) Define Rational Agent.

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

6) What are PEAS? Explain with Example.

PEAS: Performance, Environment, Actuators, Sensors

Performance Measure	Environment	Actuators	Sensors
safe, fast, legal, comfortable trip, maximize profits	roads, other traffic, pedestrians, customers	steering, accelerator, brake, signal, horn, display	camera, sonar, speedometer, GPS, odometer, engine sensors, keyboard, accelerator

7) Explain Goal Based Agent .

Every IA has a set of goals to respond desirably to its situation. This kind of IA uses pre-programmed actions based on their possible outcomes to meet its objectives. It can perform a single or many activities, depending on its goal.

8) Explain the Structure of Intelligent Agents.

The IA structure consists of three main parts: architecture, agent function, and agent program.

1. Architecture: This refers to machinery or devices that consists of actuators and sensors. The intelligent agent executes on this machinery. Examples include a personal computer, a car, or a camera.
2. Agent function: This is a function in which actions are mapped from a certain percept sequence. Percept sequence refers to a history of what the intelligent agent has perceived.
3. Agent program: This is an implementation or execution of the agent function. The agent function is produced through the agent program's execution on the physical architecture.

9) State Weiss's Classification.

Logic-based agents – in which the decision about what action to perform is made via logical deduction;

Reactive agents – in which decision making is implemented in some form of direct mapping from situation to action;

Belief-desire-intention agents – in which decision making depends upon the manipulation of data structures representing the beliefs, desires, and intentions of the agent; and finally,

Layered architectures – in which decision making is realized via various software layers, each of which is more or less explicitly reasoning about the environment at different levels of abstraction.

10) Define 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.

11) Name Two Applications Use PEAS Representation.

Example of Agents with their PEAS representation

Agent	Performance measure	Environment	Actuators	Sensors
1. Medical Diagnose	○ Healthy patient ○ Minimized cost	○ Patient ○ Hospital ○ Staff	○ Tests ○ Treatments	Keyboard (Entry of symptoms)
2. Vacuum Cleaner	○ Cleanness ○ Efficiency ○ Battery life ○ Security	○ Room ○ Table ○ Wood floor ○ Carpet ○ Various obstacles	○ Wheels ○ Brushes ○ Vacuum Extractor	○ Camera ○ Dirt detection sensor ○ Cliff sensor ○ Bump Sensor ○ Infrared Wall Sensor

12) What Are The Properties of Environment.

Discrete / Continuous

Observable / Partially Observable

Static / Dynamic

Single agent / Multiple agents

Accessible / Inaccessible

Deterministic / Non-deterministic

Episodic / Non-episodic

13) What is The Role of Intelligent Agents?

An intelligent agent is a program that can make decisions or perform a service based on its environment, user input and experiences. These programs can be used to autonomously gather information on a regular, programmed schedule or when prompted by the user in real time.

14) What are the Five Fundamental Traits of Intelligent Function?

The top five characteristics of truly intelligent FS organisations, according to our research, are: living, enhanced, modular, liquid and human

15) Explain Sensors.

Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.

16) Explain Acuators.

Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.

17) Explain Effectors.

Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen.

18) What Are Four Main rules of Intelligent Agent?

- Rule 1: An AI agent must have the ability to perceive the environment.
- Rule 2: The observation must be used to make decisions.
- Rule 3: Decision should result in an action.
- Rule 4: The action taken by an AI agent must be a rational action.

19) What is Learning Agent?

A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt. A learning agent has mainly four conceptual components, which are:

Learning element: It is responsible for making improvements by learning from the environment

Critic: The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.

Performance element: It is responsible for selecting external action

Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.

20) How PEAS works in a Self Driving Car?

Performance: Safety, time, legal drive, comfort

Environment: Roads, other vehicles, road signs, pedestrian

Actuators: Steering, accelerator, brake, signal, horn

Sensors: Camera, GPS, speedometer, odometer, accelerometer, sonar.

PART B

1) State and Explain the Types of Environments in AI.

2) Explain the Types Of Intelligent Agents.

3) Explain Rationality With Example.

4) Explain All Four Characteristics of Intelligent Agent.

5) List The Features of Intelligent Agents.

UNIT II

PROBLEM SOLVING

Heuristic search strategies – heuristic functions Local search and optimization problems – local search in continuous space – search with nondeterministic actions – search in partially observable environments – online search agents and unknown environments

PART A

1. What is heuristic search

Heuristic search is class of method which is used in order to search a solution space for an optimal solution for a problem. The heuristic here uses some method to search the solution space while assessing where in the space the solution is most likely to be and focusing the search on that area

Heuristic Function

A heuristic function (algorithm) or simply a heuristic is a shortcut to solving a problem when there are no exact solutions for it or the time to obtain the solution is toolong

2. Define Problem Formulation ?

- An important aspect of intelligence is goal-based problem solving.
- The solution of many problems can be described by finding a sequence of actions that lead to a desirable goal.
- Each action changes the state and the aim is to find the sequence of actions and states that lead from the initial (start) state to a final (goal) state.

3. What are the four components to define a problem ? Define them . Answer :

A problem is defined by four items:

initial state : e.g., "at Arad"

successor function : S(x) = set of action-state pairs e.g., S(Arad) =

{[Arad -> Zerind;Zerind], ... }

goal test, can be

explicit, e.g., x = at Bucharest" implicit,

e.g., NoDirt(x)

path cost (additive) :

e.g., sum of distances, number of actions executed, etc. c(x; a; y) is the step cost, assumed to be >= 0

4. Define a graph and path.

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph

Path - sequence through state space

5. Discover what is optimal solution ?

An optimal solution is a feasible solution where the objective function reaches its maximum (or minimum) value – for example, the most profit or the least cost

explicit, e.g., x = at Bucharest" implicit,

e.g., NoDirt(x)

path cost (additive) :

e.g., sum of distances, number of actions executed, etc. c(x; a; y) is the

step cost, assumed to be >= 0

6. Define a graph and path.

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph

Path - sequence through state space

7. Discover what is optimal solution?:

An optimal solution is a feasible solution where the objective function reaches its maximum (or minimum) value – for example, the most profit or the least cost

8. Define Abstraction.

Data Abstraction is defined as the process of reducing the object to its essence so that only the necessary characteristics are exposed to the users

9. Rank and list criteria to measure the performance of search strategies.

criteria for measuring the performance of uninformed search strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No	No	Yes[a]	Yes[a,d]
Time	$O(b^{d+1})$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^{d+1})$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes[c]	Yes	No	No	Yes[c]	Yes[c,d]

Evaluation of search strategies,b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs >= E for positive E; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

10. Differentiate uninformed search and informed search.

UNINFORMED SEARCH STRATGES :

Uninformed Search Strategies have no additional information about states beyond that provided in the problem definition.

Informed (Heuristic) Search Strategies :

- Informed search strategy is one that uses problem-specific knowledge beyond the definition of the problem itself
- It can find solutions more efficiently than uninformed strategy.

11. Examine the breadth first search.

Breadth-first search

- Breadth-first search is a simple strategy in which the root node is expanded first, then all successors of the root node are expanded next, then their successors, and so on.
- In general, all the nodes are expanded at a given depth in the search tree before any nodes at the next level are expanded.
- Breath-first-search is implemented by calling TREE-SEARCH with an empty fringe that is a first-in-first-out(FIFO) queue, assuring that the nodes that are visited first will be expanded first.

Access the depth-limited search.

The problem of unbounded trees can be alleviated by supplying depth- first-search with a pre-determined depth limit l.

- That is nodes at depth l are treated as if they have no successors.
- This approach is called depth-limited-search.
- The depth limit solves the infinite path problem.

12. Summarize simulated annealing.

Simulated annealing is a method for solving unconstrained and bound- constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.

13. Summarize stochastic beam search. :

Stochastic Beam Search (SBS) is a simple search procedure that starts with a set of initial solutions and explores their neighborhood solutions using successor functions. It then keeps track of only the most promising (in terms of fitness) k solutions

14. Point out what is genetic algorithm.:

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution.

15. Compose what is best first search.

Best-first search

- Best-first search is an instance of general TREE-SEARCH or GRAPH-SEARCH algorithm in which a node is selected for expansion based on an evaluation function f(n).
- The node with lowest evaluation is selected for expansion, because the evaluation measures the distance to the goal.

16. Analyse the definition of greedy best-first search.

Greedy Best-first search

- Greedy best-first search tries to expand the node that is closest to the goal, on the grounds that this is likely to a solution quickly.
- It evaluates the nodes by using the heuristic function f(n) = h(n).

17. Point out and define node consistency , arc consistency and path consistency.

Node consistency :

Node consistency requires that every unary constraint on a variable is satisfied by all values in the domain of the variable, and vice versa. This condition can be trivially enforced by reducing the domain of each variable to the values that satisfy all unary constraints on that variable.

Arc consistency :

One method of constraint propagation is to enforce arc consistency

- Stronger than forward checking
- Fast

Arc refers to a directed arc in the constraint graph Path

consistency :

Path consistency is a property similar to arc consistency, but considers pairs of variables instead of only one. A pair of variables is path-

consistent with a third variable if each consistent evaluation of the pair can be extended to the other variable in such a way that all binary constraints are satisfied.

18. Define Admissible heuristic h (n).

In A* search, if it is optimal then, h(n) is an admissible heuristic which means h(n) never overestimates the cost to reach the goal.

19. What are the 2 types of memory bounded heuristic algorithms?

Recursive Best First Search(RBFS)

Memory bounded A*(MA*)

20. Differentiate BFS & DFS

BFS means breath wise search. Space complexity is more. Do not give optimal solution Queuing fn is same as that of queue operator

DFS means depth wise search. Space complexity is less Gives optimal solution Queuing fn is somewhat different from queue operator.

21. What is RBFS?

It keeps track of the f-value of the best alternative path available from any ancestor of the current node. RBFS remembers the f-value of the best leaf in the forgotten sub tree and therefore decide whether its worth re expanding the sub tree sometimes later.

PART B

1. Discuss about
   i. Greedy best-first search.
   ii. A*search
   iii. Memory bounded heuristic search.

2. Summarize the following uninformed
   i. Depth first search
   ii. Iterative deepening depth first search.
   iii. Bidirectional search.

3.
   i. Explain in detail about models for predicate logic.
   ii. Explain assertions and queries in first-order logic.

4. Relate first order logic with proposition logic and discuss in detail about the same

5.
   i. Compose what is uninformed search ?explain depth first search with example.
   ii. Compose the algorithm for recursive best first search.

6.

i. Explain the nature of heuristics with an example. What is the effect of heuristics accuracy on performance?

ii. Write a simple back tracking algorithm for constraint satisfaction problems.

7. Explain briefly about problem solving strategies.

8. Describe Alpha Beta pruning with Algorithm.

9. Explain stochastic games with examples

10. Show and explain optimization problems

UNIT III

GAME PLAYING AND CSP

Game theory – optimal decisions in games – alpha-beta search – monte-carlo tree search – stochastic games – partially observable games Constraint satisfaction problems – constraint propagation – backtracking search for CSP – local search for CSP – structure of CSP

PART A

1. What is game playing in AI?

General game playing (GGP) is the design of artificial intelligence programs to be able to play more than one ga me successfully.
For instance, a chess-playing computer program cannot play checkers.
General game playing is considered as a necessary milestone on the way to artificial general intelligence.
Gameplay is the specific way in which players interact with a game, and in particular with video games.
Gameplay is the pattern defined through the game rules, connection between player and the game, challenges and overcoming them, plot and player's connection with it.

2. What is game theory in artificial intelligence?

Game Theory is a branch of mathematics used to model the strategic interaction between different players

in a context with predefined rules and outcomes. Game Theory can be applied in different ambit of Artificial Intelli gence: Multi-agent AI systems.

3. Where is game theory used in AI?

Currently, game theory is being used in adversary training in GANs, multi-agent systems, and imitation and reinf orcement learning.

In the case of perfect information and symmetric games, many Machine Learning and Deep Learning techniques ar e applicable.

4. What are the characteristics of game theory?

game theory, branch of applied mathematics that provides tools for analyzing situations in which parties, called p layers,make decisions that are interdependent. This interdependence causes each player to consider the other player's poss ible decisions,or strategies, in formulating strategy.

5. What is optimal decision gaming?

arrow_back Artificial Intelligence. In a normal search problem, the optimal solution would be a sequence of actions leading to a goal state-a terminal state that is a win.
An optimal decision is a decision that leads to at least as good a known or expected outcome as all other availabl e decision options.

In practice, few people verify that their decisions are optimal, but instead use heuristics to make decisions that are "good enough"—that is, they engage in satisficing.

6. How are optimal decisions made?

To make an optimal decision, economists ask: ―What are the extra (marginal) costs and what are the extra (margi nal) benefits associated with the decision?‖

If the extra benefits are bigger than the extra costs, you shall go ahead with the decision, namely the decision is go od.

7. What is Alpha Beta algorithm in artificial intelligence?

Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minim ax algorithm.

It is also called as Alpha-Beta Algorithm. Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prune the tree leaves but also entire sub-tree.

8. What is alpha-beta pruning?

Alpha Beta Pruning is a method that optimizes the Minimax algorithm. The number of states to be visited by the minimax algorithm are exponential,

which shoots up the time complexity. Some of the branches of the decision tree are useless, and the same result ca n be achieved if they were never visited.

9. What is Monte Carlo tree search used for?

In computer science, Monte Carlo tree search (MCTS) is a heuristic search algorithm for some kinds of decision p rocesses, most notably those employed in software that plays board games. In that context MCTS is used to solve the game tr ee.

MCTS was combined with neural networks in 2016 for computer Go

10. What are the four steps of Monte Carlo tree search?

The process of Monte Carlo Tree Search can be broken down into four distinct steps, selection, expansion, simulation and backpropagation.

11. What are the advantages of Monte Carlo search? Advantages :

1 — MCTS is a simple algorithm to implement.

2 — Monte Carlo Tree Search is a heuristic algorithm. MCTS can operate effectively without any knowledge i n the particular domain,

apart from the rules and end conditions, and can find its own moves and learn from them by playing random playouts.

12. What are stochastic games in detail?

A stochastic game was introduced by Lloyd Shapley in the early 1950s. It is a dynamic game with probabilistic t ransitions played by one or more players.

The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain state.

Stochastic games (SG) - also called Markov games - extend Markov decision process (MDP) to the case where t here are multiple players in a common environment.

These agents perform a joint action that defines both the reward obtained by the agents and the new state of the en vironment.

13. What is partial observability in AI?

Partial observability means that an agent does not know the state of the world or that the agents act simultaneously.

In a partially observable system the observer may utilise a memory system in order to add information to the obser ver's understanding of the system.An example of a partially observable system would be a card game in which some of the cards are discarded into a pile face down

14. What is CSP in artificial intelligence?

A constraint satisfaction problem (CSP) consists of. a set of variables, a domain for each variable, and. a set of co nstraints.

15. What is CSP algorithm?

In general, a CSP is a problem composed of a finite set of variables, each of which has a finite domain of values, and a set of constraints.The task is to find an assignment of a value for each variable such that the assignments satisfy all the constraints. I n some problems, the goais to find all such assignments.

16. What is an example of CSP?

Some of the popular CSP problems include Sudoku, Cryptarithmetic, crosswords, n-Queen, etc**.**

17. What is constraint propagation?

Constraint propagation is the process of communicating the domain reduction of a decision variable to all of the constraints that are stated over this variable. This process can result in more domain reductions. These domain reductions, in turn, are communicated to the appropriate constraints.

18. How are constraints propagated in forward checking?

Forward checking detects the inconsistency earlier than simple backtracking and thus it allows branches of the sea rch treethat will lead to failure to be pruned earlier than with simple backtracking. This reduces the search tree and (hopef ully) the overall amount of work done.

19. What is backtracking search with example?

A depth-first search that chooses values for one variable at a time and backtracks when a variable has no legal v alues left to assign.

Backtracking algorithm repeatedly chooses an unassigned variable, and then tries all values in the domain of that variable in turn, trying to find a solution.

Examples where backtracking can be used to solve puzzles or problems include: Puzzles such as eight queens p uzzle, crosswords, verbal arithmetic,

Sudoku, and Peg Solitaire. Combinatorial optimization problems such as parsing and the knapsack problem**.**

20. Which search method is used in backtracking? depth-first search method

In order to find these solutions, a search tree named state-space tree is used. In a state-space tree, each branch is a variable,and each level represents a solution. A backtracking algorithm uses the depth-first search method.

21. What is local search for CSP?

A local search problem consists of a: CSP: a set of variables, domains for these variables, and constraints on th eir joint values. A node in the search space will be a complete assignment to all of the variables. local search is an incomplete met hod for finding a solution to a problem. It is based on iteratively improving an assignment of the variables until all constraints are satisfied.

In particular, local search algorithms typically modify the value of a variable in an assignment at each step.

PART B

1. What is the basic strategy in game playing in artificial

intelligence?

2. Why is game theory important in AI?

3. Define Minimax Algorithm in Game Theory? 4.How does Monte Carlo tree search work?

5.How CSP is formulated as a search problem? 6.What is backtracking search in CSP?

UNIT IV

LOGICAL AGENTS

Knowledge-based agents – propositional logic – propositional theorem proving – propositional model checking – agents based on propositional logic First-order logic – syntax and semantics – knowledge representation and engineering – inferences in first-order logic – forward chaining – backward chaining – resolution

PART A

1. **What is first-order logic?**

   First-order logic is also known as Predicate logic or First-order predicate logic. First-order logic is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects.

2. **What are the elements and symbols of First order logic?**

   Following are the basic elements of FOL syntax:

Constant	1, 2, A, John, Mumbai, cat,....
Variables	x, y, z, a, b,....
Predicates	Brother, Father, >,....
Function	sqrt, LeftLegOf, ...
Connectives	∧, ∨, ¬, ⇒, ⇔
Equality	==
Quantifier	∀, ∃

3. **What are the three families of First-order inference algorithms?**

   Forward chaining and its applications to deductive database and production systems
   Backward chaining and logic programming systems
   Resolution-based theorem- proving system

4. **What are the parts of knowledge in first-order logic?**

   Approval Publish requests approval from a manager of the knowledge base before moving the article to the published state.
   Approval retire requests approval from a manager of knowledge base before moving the article to the retired state.

5. **Define terms.**

   Term = logical expression that refers to an object.
   There are 2 kinds of terms:
   constant symbols: Table, Computer
   function symbols: LeftLeg(Pete), Sqrt(3), Plus(2,3) etc

6. **Define the first order definite clause.**

   They are called definite clause grammars because they represent a grammar as a set of definite clauses in first-order logic. The term DCG refers to the specific type of expression in Prolog and other similar languages; not all ways of expressing grammars using definite clauses are considered DCGs.

7. Define description logics.

A description logic is used to describe classes, properties, and individuals. One of the main ideas behind a description logic is to separate a terminological knowledge base that describes the terminology, which should remain constant as the domain being modeled changes, and
an assertional knowledge base that describes what is true in some domain at some point in time.

8. What is data-driven search? (forward chaining)

Forward chaining is the logical process of inferring unknown truths from known data and moving forward using determined conditions and rules to find a solution. The opposite of forward chaining is backward chaining.

9. Define complex sentences

Complex sentences are made by combining atomic sentences using connectives.

10. Show what is Skolemization?

Skolemization in Artificial Intelligence is a procedure used when there is a requirement of the reduction of any first-order formula to its Skolem normal form. This is usually done when there is a need for proving a theorem by using programming.

11. Define Purely Reactive Agents.

Reactive agents are software agents that carry out a simple task of retrieving pre-set behaviors similar to reflexes. It do not maintain the internal state, unlike deliberative agents. Finding a difference between reactive agents and deliberative agents can be indistinct though. It can simply be said that an agent that has no internal state is a reactive agent.

12. Explore some interesting properties of agents and perception.

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.

13. What are classes of agents?

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

14. What are logical formulae and logical deduction?

The phenomenon of deriving a conclusion from a single proposition or a set of given propositions, is known as logical deduction. The given propositions are also referred as the premises.

15. Define Agent Communication.

Agent communication is based on message passing, where agents communicate by formulating and sending individual messages to each other. The FIPA ACL specifies a standard message language by setting out the encoding, semantics and pragmatics of the messages.

16. Define Coherence.

Coherence arguments say that if an entity's preferences do not adhere to the axioms of expected utility theory, then that entity is susceptible to losing things that it values. This does not imply that advanced AI systems must adhere to these axioms ('be coherent'), or that they must be goal-directed.

17. Define belief-desire-intention (BDI) architectures

The belief–desire–intention software model (BDI) is a software model developed for programming intelligent agents. Superficially characterized by the implementation of an agent's beliefs, desires and intentions, it actually uses these concepts to solve a particular problem in agent programming.

18. What are the fields Used in protocol?

Routing Protocol.
Open Shortest Path First.
Transmission Control Protocol.
User Datagram Protocol.

19. What are the two types of control flow within layered architectures?

Open layers
Request flow.

20. State the advantage of horizontal layered architectures.

The advantage of horizontal layer architecture is that only n layers are required for mapping to n different types of behaviours. However, a mediator function is used to control the inconsistent actions between layer interactions.

PART B

1. Explain the inference process in first order logic, using suitable example Prolog Programming.
2. (i). Discuss backward chaining algorithm.(6)
   (ii).Explain the algorithm for computing more general unifiers.(7)
3. Explain the unification algorithm used for reasoning under predicate logic with an example.
4. Define Agent Communication. Write a short note on coordination, Dimensions of meaning and Message types.
5. Examine the Argumentation among Agents.

UNIT V

PROBABILISTIC REASONING

Acting under uncertainty – Bayesian inference – naïve Bayes models. Probabilistic reasoning – Bayesian networks – exact inference in BN – approximate inference in BN – causal networks.

PART A

1. ***Define uncertainty and list the causes of uncertainty.***

Uncertainty:

- The knowledge representation, A→B, means if A is true then B is true, but a situation where not sure about whether A is true or not then cannot express this statement, this situation is called uncertainty.
- So to represent uncertain knowledge, uncertain reasoning or probabilistic reasoning is used.

Causes of uncertainty:

1. Causes of uncertainty in the real world
2. Information occurred from unreliable sources.
3. Experimental Errors
4. Equipment fault
5. Temperature variation
6. Climate change.

2. ***Define Probabilistic reasoning.***

Probabilistic reasoning is a way of knowledge representation, the concept of probability is applied to indicate the uncertainty in knowledge.

3. ***List the Ways to solve problems with uncertain knowledge.***

- Bayes' rule
- Bayesian Statistics

4. ***Define Probability and the probability of occurrence.***

- Probability can be defined as a chance that an uncertain 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.
- Formula to find the probability of an uncertain event

$$\textbf{Probability of occurrence} = \frac{\text{Number of desired outcomes}}{\text{Total number of outcomes}}$$

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

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

5. ***Define the terms event, sample space, random variables, prior probability and posterior probability.***

- **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.

6. Define Conditional probability.

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

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

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

P(B)= Marginal probability of B.

7. Define Bayesian Inference.

- Bayesian inference is a probabilistic approach to machine learning that provides estimates of the probability of specific events.
- Bayesian inference is a statistical method for understanding the uncertainty inherent in prediction problems.
- Bayesian inference algorithm can be viewed as a Markov Chain Monte Carlo algorithm that uses prior probability distributions to optimize the likelihood function.
-

8. What are the Application of Bayes' theorem in Artificial intelligence?

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

9. Define Bayesian Network.

- "A Bayesian network is a probabilistic graphical model which represents a set of variables and their conditional dependencies using a directed acyclic graph."
- It is also called a Bayes network, belief network, decision network, or Bayesian model.
- Bayesian Network can be used for building models from data and experts opinions, and it consists of two parts:
  - Directed Acyclic Graph
  - Table of conditional probabilities

10. What are the Applications of Bayesian networks in AI?

1. Spam filtering
2. Bio monitoring
3. Information retrieval
4. Image processing
5. Gene regulatory network
6. Turbo code
7. Document classification

11. Define Bayesian Inference.

- Bayesian Network is to perform inference, which computes the marginal probability P(V=v) for each node V and each possible instantiation v.
- Inference can also be done on a Bayesian network when the values of some nodes are known (as evidence) and wish to compute the likelihood of values of other nodes.
- There are two types of inference on Bayesian networks: exact and approximate.
- Exact inference algorithms compute the exact values of each marginal or posterior probability, while approximate inference algorithms sacrifice some accuracy of the probabilities to report results quickly.

PART B

1. Explain the concept of uncertainty and acting under uncertainty with suitable example. Explain in detail about probabilistic reasoning.

2. Explain in detail about Bayesian inference and Naive Bayes Model or Naive Bayes Theorem or Bayes Rule.

3. Explain in detail about Bayesian Network.

4. Explain in detail about Bayesian Inference and its type Exact Inference with suitable example.

5. Explain approximate inference in Bayesian network (BN).