Model Question Paper

Artificial Intelligence S7

Part A

1 What is a rational agent? Explain.

Answer:

A rational agent or rational being is a person or entity that always aims to perform optimal actions based on given premises and information. A rational agent can be anything that makes decisions, typically a person, firm, machine, or software.

2.Describe any two ways to represent states and the transitions between them in agent programs.

Answer:

There are two main ways to represent or design state transition, State transition diagram, and state transition table. In state transition diagram the states are shown in boxed texts, and the transition is represented by arrows. It is also called State Chart or Graph. It is useful in identifying valid transitions.

3. Differentiate between informed search and uninformed search.

Answer:

Parameters	Informed Search	Uninformed Search
Known as	It is also known as Heuristic Search.	It is also known as Blind Search.
Using Knowledge	It uses knowledge for the searching process.	It doesn't use knowledge for the searching process.
Performance	It finds a solution more quickly.	It finds solution slow as compared to an informed search.
Completion	It may or may not be complete.	It is always complete.

4. Define heuristic function? Give two examples.

Answer:

A heuristic function, is a function that calculates an approximate cost to a problem (or ranks alternatives). For example, the problem might be finding the shortest driving distance to a point. A heuristic cost would be the straight-line distance to the point. heuristic function, also simply called a heuristic, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow

Examples: pathfinding algorithm uses a heuristic to find the shortest path in a graph. ...

Traveling Salesman Problem (TSP) Gives a list of cities

5. What are the components of a Constraint Satisfaction Problem? Illustrate with an example.

Answer:

Constraint satisfaction problems (CSP) solve a wide variety of problems efficiently using a factored representation for each state: a set of variables, each of which has a value. A problem is solved when each variable has a value that satisfies all the constraints on the variable.

A constraint satisfaction problem consists of three components: X, D, and C.
X is a set of variables, {X1, ..., Xn}
D is a set of domains, {D1, ..., Dn}, one for each variable
C is a set of constraints that specify allowable combinations of values

Cryptarithmetic problems replace numbers with alphabets or other symbols to encrypt a message

For example, in this one where each letter must be a unique single digit:
TWO
+ TWO

_____
FOUR

6. Formulate the following problem as a CSP. Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.

Answer:

There are many choices here. Let's say we have K classes, L profs, M possible times and N possible rooms.

Formulation: Have three different variables for each class: which professor, which time, and which room. So, we'd have K variables with domain size L, K with domain size M, and K with domain size N. Constraints would have to be that profs can't be in two classes at the same time; that you cannot use the same room for two classes at the same time; that only appropriate professors are assigned to classes.

7. What is a knowledge based agent? How does it work?

Answer:

An intelligent agent needs knowledge about the real world for taking decisions and reasoning to act efficiently. Knowledge-based agents are those agents who have the capability of maintaining an internal state of knowledge, reason over that knowledge, update their knowledge after observations and take actions. These agents can represent the world with some formal representation and act intelligently.

Knowledge-based agents are composed of two main parts: Knowledge-base and Inference system.

8. Represent the following assertion in propositional logic: "A person who is radical (R) is electable (E) if he/she is conservative (C), but otherwise is not electable."

Answer:

R->(E<->C)

9 Describe the various forms of learning?

Answer:

Supervised: Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output.

Unsupervised: in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

Reinforced Learning: Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

10. State and explain Ockham's razor principle

Answer:

Ockham's Razor, also known as the principle of parsimony, is a principle in philosophy that states that, when presented with multiple explanations for a phenomenon, one should choose the explanation that makes the fewest number of assumptions. This principle is often used in science to guide the search for the most likely explanation for a given set of observations.The idea behind Ockham's razor is that, all else being equal, the simplest explanation is most likely to be correct.

Ockham's razor is not a scientific law or a guarantee that the simplest explanation is always correct. It is simply a heuristic that can be useful in guiding the search for the most likely explanation for a given phenomenon.

PART B

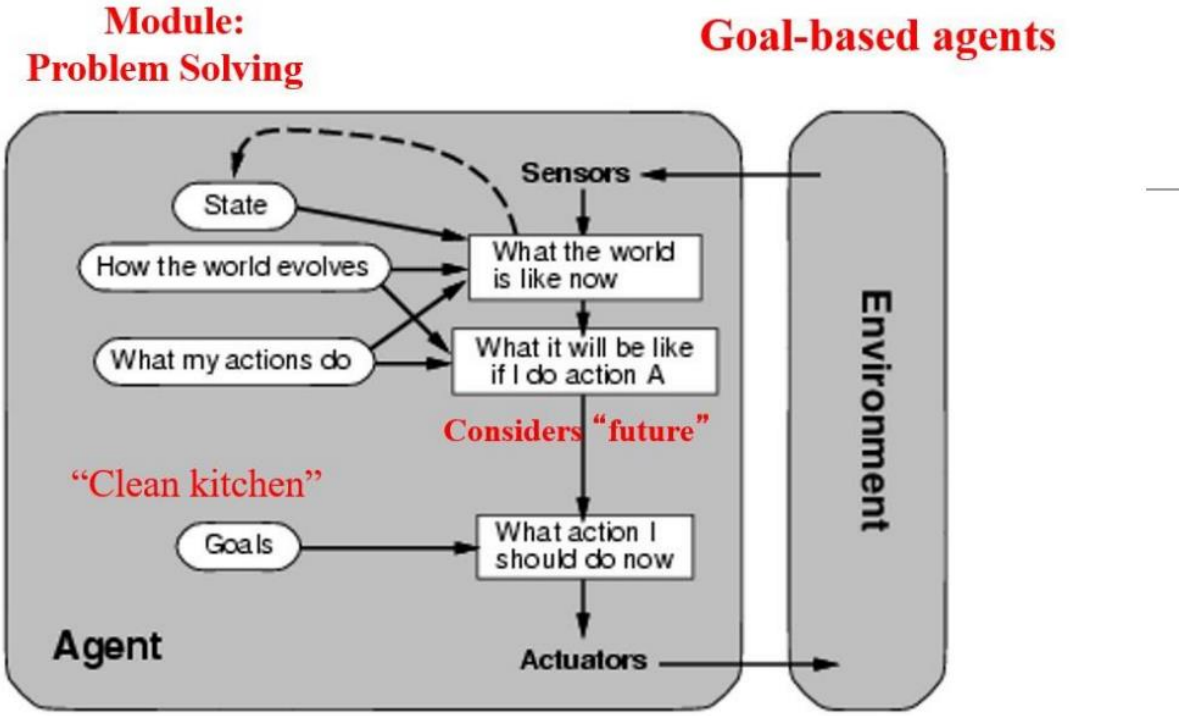
11 . (A) Explain the structure Goal-based agents and Utility-based agents with the help of diagrams.

(B) For the following activities, give a PEAS description of the task environment and characterize it in terms of the task environment properties. a) Playing soccer b) Bidding on an item at an auction.

Answer:

(A)Goal-Based Agents

These kinds of agents take decisions based on how far they are currently from their goal. Agents of this kind take future events into consideration. What sequence of actions can I take to achieve certain goals? Choose actions so as to (eventually) achieve a (given or computed) goal.

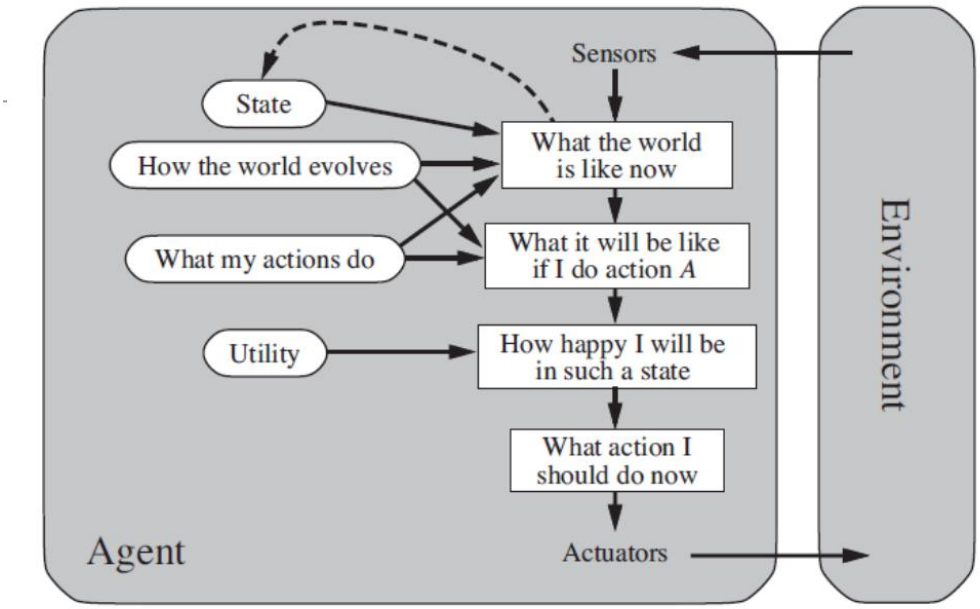


Agent keeps track of the world state as well as set of goals its trying to achieve: choose actions that will(eventually) lead to the goal. And it involves search and planning.

Utility-Based Agents

Goals alone are not enough to generate high-quality result they must provide a distinction between "happy" and "unhappy" states. Because "happy" does not sound very scientific, economists and computer scientists use the term utility instead.

An agent's utility function is essentially an internalization of the performance measure. If the internal utility function and the external performance measure are in agreement, then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.



(B) a) **Playing soccer**: Performance measure: Score as many goals as possible. Environment: A soccer field with artificial grass or natural grass. Actuators: The player's body, including their feet, legs, and torso. Sensors: The player's eyes, ears, and sense of touch.

Task environment properties:

- Physical: The soccer field, ball, and other players are physical entities that the player must interact with.
- Dynamic: The game is constantly changing, with players moving and the ball being passed or kicked.
- Stochastic: There is an element of chance in the game, as the ball can take unpredictable bounces and players may make mistakes.
- Discrete: The game is divided into discrete time periods, such as halves or quarters.

b) **Bidding on an item at an auction:** Performance measure: Successfully win the auction by placing the highest bid. Environment: An auction house or online auction platform. Actuators: The bidder's ability to place bids and communicate with the auctioneer. Sensors: The bidder's eyes and ears, as well as any information provided by the auctioneer or displayed on the auction platform.

Task environment properties:

- Physical: The item being auctioned is a physical entity that the bidder may be able to inspect before placing a bid.
- Dynamic: The auction is constantly changing, with other bidders placing bids and the price increasing.
- Stochastic: There is an element of chance in the outcome of the auction, as the bidder cannot control the actions of other bidders.
- Discrete: The auction is divided into discrete time periods, such as increments between bids.

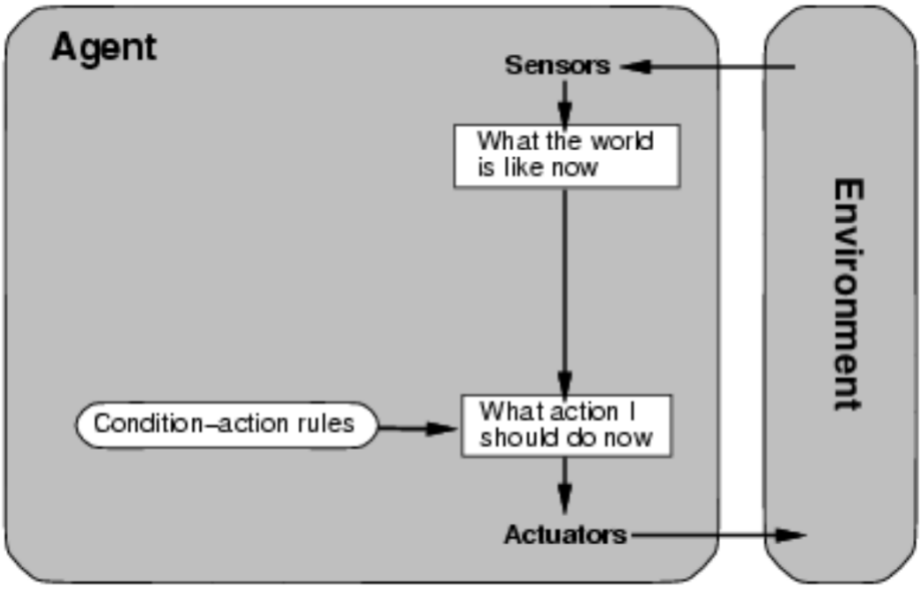
12. (A) Explain the structure Simple reflex agents and Model-based reflex agents with the help of diagrams.

(B) Discuss about any five applications of AI.

Answer:

(A)Simple Reflex Agent

Select actions on the basis of the current percept, ignoring the rest of the percept history. Agents do not have memory of past world states or percepts. So, actions depend solely on current percept. Actions are performed based on condition-action rules.



Simple reflex agents will work only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable.

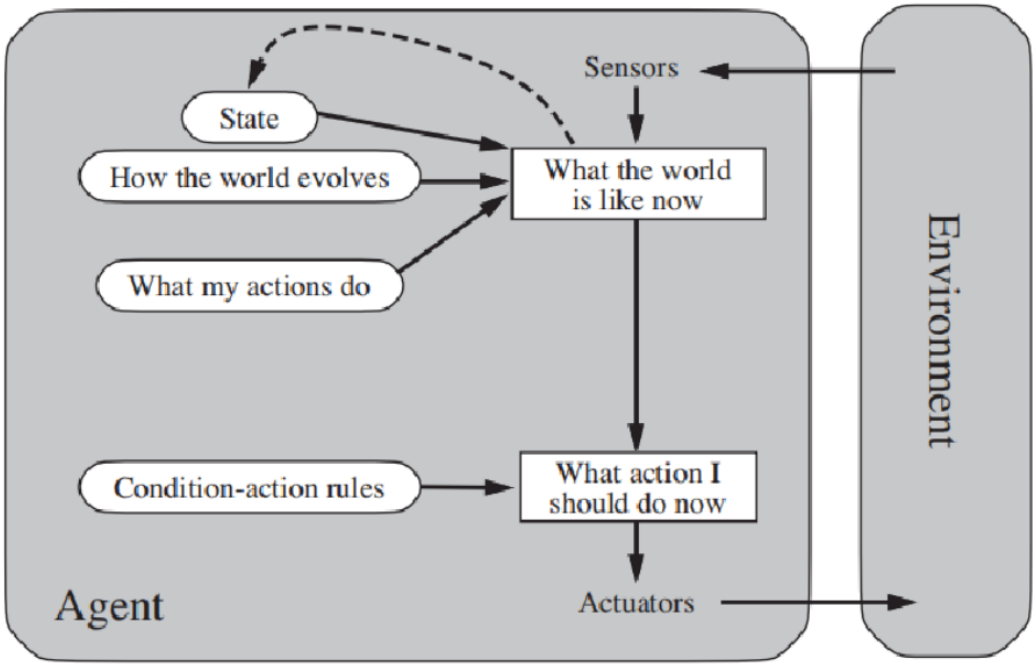
Model-Based Reflex Agent

It works by finding a rule whose condition matches the current situation.

Key difference with respect to simple reflex agents are:

- Agents have internal state, which is used to keep track of past states of the world.
- Agents have the ability to represent change in the World.

The current state is stored inside the agent which maintains some kind of structure describing the part of the world which cannot be seen. Knowledge about “how the world works” is called a model of the world. An agent that uses such a model is called a model-based agent.



Internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program

1. we need some information about how the world evolves independently of the agent
2. we need some information about how the agent's own actions affect the world

(B) Different applications of AI are:

1. 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.
2. Computer vision: AI algorithms can be used to analyze and understand visual data, such as images or video. This has applications in areas such as image classification, object detection, and facial recognition.
3. Robotics: AI algorithms can be used to control and coordinate the movements of robots, allowing them to perform tasks such as manufacturing, transportation, and search and rescue. 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.
4. Healthcare: AI algorithms can be used to analyze medical data and assist with diagnosis and treatment planning. For example, machine learning algorithms can be used to predict the likelihood of a patient developing a particular disease based on their medical history and other factors.
5. Finance: AI algorithms can be used to analyze financial data and make predictions about market trends and investment opportunities. For example, AI algorithms can be used to identify patterns in stock prices and make recommendations for investment portfolios. AI is helping shoppers to discover associated products with recommended size, color, or even brand.

13. (A) Explain Best First Search algorithm. How does it implement heuristic search?

(B) Describe any four uninformed search strategies.

Answer:

(A) Best First Search is a heuristic search algorithm that explores a search space by prioritizing the most promising nodes to be explored first, it is done with the help of a **heuristic function, h(n)**. The heuristic function is construed as a cost estimate, so the node with the lowest evaluation is expanded first. It uses a priority queue to store the nodes that have been explored, with the priority of a node being determined by a heuristic function.

The algorithm works by initially adding the start node to the priority queue and then repeatedly removing the node with the highest priority (lowest heuristic cost) from the queue and expanding it to its neighboring nodes. The process is repeated until the goal is reached or there are no more nodes to explore.

Best First Search implements heuristic search by using a heuristic function to guide the search towards the most promising nodes. The heuristic function, denoted by

h(n): h(n) = estimated cost of the cheapest path from the state at node n to a goal state.

If n is a goal node, then h(n)=0.

Best First Search is useful for searching large search spaces because it can find a solution more efficiently by prioritizing the most promising nodes. However, it can sometimes get stuck in a suboptimal solution if the heuristic function is not accurate.

(B) In **uninformed search**, also known as blind search, the algorithm does not have any prior knowledge about the structure of the search space or the location of the goal. Instead, it must explore the search space blindly, without any guidance about which paths are more promising. Some common uninformed search strategies in AI include:

I. **Breadth-first search:** This strategy involves exploring the search space by expanding the shallowest nodes (nodes at the lowest level of the search tree, i.e. the root node) first, then all the successors of the root node are expanded next, then their successors, and so on. It uses a queue to store the nodes that have been explored, with the oldest nodes being explored first.

   All the nodes are expanded at a given depth in the search tree before any nodes at the next level are expanded. Breadth-first search is an instance of the general graph-search algorithm in which the shallowest unexpanded node is chosen for expansion. This is achieved very simply by using a FIFO queue for the frontier new nodes go to the back of the queue, and old nodes, which are shallower than the new nodes, get expanded first the goal test is applied to each node when it is generated rather than when it is selected for expansion breadth-first search always has the shallowest path to every node on the frontier.

   The memory requirements are a bigger problem for breadth-first search than is the execution time.

II. **Depth-first search:** This strategy involves exploring the search space by expanding the deepest nodes (nodes at the highest level of the search tree) first. The search proceeds immediately to the deepest level of the search tree, where the nodes have no successors.

It uses a stack (LIFO, Last In First Out queue) to store the nodes that have been explored, with the most recent nodes being explored first. A LIFO queue means that the most recently generated node is chosen for expansion. This must be the deepest unexpanded node because it is one deeper than its parent which, in turn, was the deepest unexpanded node when it was selected.

III. **Uniform-cost search:** This strategy involves exploring the search space by expanding the nodes with the lowest cost first i.e., instead of expanding the shallowest node, uniform-cost search expands the node n with the lowest path cost g(n). the goal test is applied to a node when it is selected for expansion because the first goal node that is generated may be on a suboptimal path a test is added in case a better path is found to a node currently on the frontier. It uses a priority queue to store the nodes that have been explored, with the priority of a node being determined by its cost.

Uniform-cost search is useful in cases where the cost of moving between states is not constant, and it is important to find the path with the lowest total cost. It is also useful in cases where the search space is very large, as it is guaranteed to find the optimal solution if one exists.

IV. **Depth-limited search:** This strategy is similar to depth-first search, but with a limit on the maximum depth of the search tree that can be explored, i.e., if we set a limit to the depth of tree say n the all nodes at depth n are treated as if they have no successors. This can be useful in cases where the search space is very large and there is a need to bound the search in order to find a solution within a reasonable amount of time. It helps to avoid getting stuck in an infinite loop when searching a search space with loops.

It is depth-first search with a predefined maximum depth. However, it is usually not easy to define the suitable maximum depth if it is too small then no solution can be found, if it is too large then the same problems are suffered from. Anyway, the search is complete but still not optimal.

14. (A) Write and explain A* search algorithm.

(B) Explain the components of a well defined AI problem? Write the standard formulation of 8-puzzle problem.

Answer:

(A) **A* search** is an informed search algorithm that is used to find the shortest path between two points, called the start and the goal. It is an extension of Dijkstra's algorithm, which is used for the same purpose but without considering the cost of the path. A* search combines the advantages of Dijkstra's algorithm with the use of heuristics to speed up the search process.

The algorithm works by maintaining a set of nodes, called the open set, which represent the set of nodes that are candidates for exploration. At each step, the algorithm selects the node from the open set that has the lowest cost, called the f-value, which is calculated as the sum of the cost of the path from the start

node to the current node and the estimated cost of the path from the current node to the goal. This estimated cost is called the heuristic and it is a measure of the distance between the current node and the goal.

The algorithm also maintains a set of nodes, called the closed set, which represent the set of nodes that have already been explored and are not considered for further expansion. When a node is selected from the open set, it is added to the closed set and its neighbors are added to the open set. This process is repeated until the goal node is found or there are no more nodes in the open set.

One of the key features of A* search is its use of a heuristic function to guide the search process. This function estimates the distance between the current node and the goal and is used to prioritize the search. By using a heuristic function, A* search can often find the optimal solution much faster than Dijkstra's algorithm, which does not use a heuristic function.

Overall, A* search is a powerful and efficient algorithm for finding the shortest path between two points in a graph. It combines the strengths of Dijkstra's algorithm with the use of a heuristic function to guide the search process and find the optimal solution faster.

(B) A well-defined AI problem typically has the following components:

1. The **initial state** that the agent starts in
2. A description of the possible **actions** available to the agent.
    - Given a particular state s, ACTIONS(s) returns the set of actions that can be executed in s. We say that each of these actions is applicable in s.
    - For example, from the state In(Ernakulam), the applicable actions are {Go(Thrissur), Go(Palakkad), Go(Kozhikod)}.
3. **Transition model**: description of what each action does, specified by a function RESULT(s, a) that returns the state that results from doing action a in state s.
4. **Successor**: any state reachable from a given state by a single action
    - RESULT(In(Ernakulam),Go(Thrissur)) = In(Thrissur).
5. **state space**: the set of all states reachable from the initial state by any sequence of actions. forms a directed network or graph in which the nodes are states and the links between nodes are actions.
6. A **path** in the state space is a sequence of states connected by a sequence of actions
7. The **goal test**, which determines whether a given state is a goal state {In(Chennai)}
8. A **path cost** function that assigns a numeric cost to each path cost of a path can be described as the sum of the costs of the individual actions along the path.

The step cost of taking action a in state s to reach state s is denoted by c(s, a, s). A solution to a problem is an action sequence that leads from the initial state to a goal state. Solution quality is measured by the path cost function, and an optimal solution has the lowest path cost among all solutions

8-puzzle problem

The standard formulation of the 8-puzzle problem is as follows:

- Given a 3x3 grid with 8 numbered tiles and a blank space, the goal is to rearrange the tiles to a specific configuration using a set of allowed actions, such as moving a tile into the blank space.
- The state space of the problem is all the possible configurations of the tiles.
- The initial state is the starting configuration
- The goal test: This check weather the state matches the goal configuration.
- The goal state the specific configurations that the tiles must be rearranged to.
- Actions: Defines the actions as movements of blank space Left, Right, Up or Down.
- Transition Model: Given a state and action, this returns the resulting state.
- Path Cost: Each step cost 1, so the path cost is the number of steps in the path.

15. (A) Solve the following crypt arithmetic problem by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics.

$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$

(B) What is local consistency in CSP constraint propagation? Explain different types local consistencies.

Answer:

(A) Variables: {T, W, O, F, U, R, C1, C2, C3}

Domain of {T,W,O,F,U,R} = {0,1,2,3,4,5,6,7,8,9}

Domain of {C1, C2, C3} = {0,1}

Using Minimum Remaining Value Heuristic we choose variable with fewest legal values, Here we take C3 whose domain={0,1}

Since F cannot be 0, by forward checking we can eliminate 0 from C3. Since F is a carry eliminate other options too

So C3=1 F=1

T	W	O	F	U	R	C3	C2	C1
~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1,2,3,4,5,6,7,8,9	~~0~~,1	0,1	0,1

Now C2 and C1 are the MRV so we choose C2(D={0,1}) both have no issue in Forward checking

T	W	O	F	U	R	C3	C2	C1
0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1	0 1	0,1

Now C1 has Minimum Remaining Values so we choose C1(D={0,1)) both have no issue in Forward checking. Lets choose C1=0

T	W	O	F	U	R	C3	C2	C1
0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1	0 1	0 1

The variable O must be even number(T+T will always be even) and it should be less than 5 since

O+O=R+10X0

2O=R //if O is more than 5 then R become greater than 10 violating Constraint of C2=0. Lets Arbitarily choose O=4 So make R=8

T	W	O	F	U	R	C3	C2	C1
0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1	0 1	0 1

Propagating these constraints to T, we get 2T=14 , ie, T=7

T	W	O	F	U	R	C3	C2	C1
0,1,2,3, 4,5,6,7 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1	0 1	0 1

U must be an even no less than 9(2W=U)

Only variable U survive forward checking is 6

T	W	O	F	U	R	C3	C2	C1
0,1,2,3, 4,5,6,7 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1,2,3, 4,5,6,7, 8,9	0,1	0 1	0 1

Only variable left is W(2W=U) which is left with value 3

T	W	O	F	U	R	C3	C2	C1
~~0,1,2,3,~~ ~~4,5,6~~ 7 ~~8,9~~	~~0,1,2~~3, ~~4,5,6,7,~~ ~~8,9~~	~~0,1,2,3,~~ 4 ~~5,6,7,~~ ~~8,9~~	0 1,2,3, ~~4,5,6,7,~~ ~~8,9~~	~~0,1~~,2,3, ~~4,5~~,6 7, ~~8,9~~	~~0,1,2,3,~~ ~~4,5,6,7,~~ 8 ~~9~~	0,1	0 1	0 1

(B) CSPs an algorithm that can search or do a specific type of inference called constraint propagation using the constraints to reduce the number of legal values for a variable, which in turn can reduce the legal values for another variable, and so on. Constraint propagation may be intertwined with search, or it may be done as a preprocessing step, before search starts. Sometimes this preprocessing can solve the whole problem, so no search is required at all.

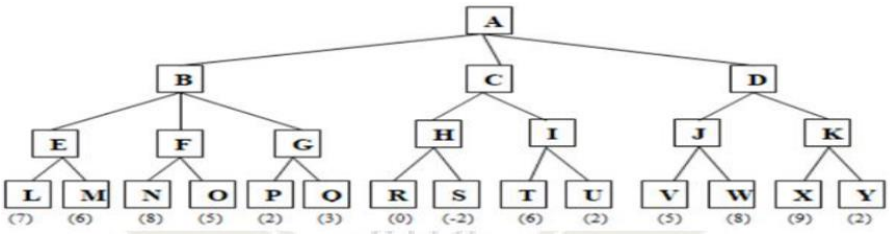
In Constraint Satisfaction Problems (CSPs), local consistency refers to the idea that the variables in a problem should be locally consistent with each other, meaning that no variable should take on a value that conflicts with the constraints of the problem. If we treat each variable as a node in a graph and each binary constraint as an arc, then the process of enforcing local consistency in each part of the graph causes inconsistent values to be eliminated throughout the graph.

There are several different types of local consistencies that can be used in CSP constraint propagation:

- Arc consistency: This means that for every value in the domain of a variable, there must be a corresponding value in the domain of another variable that satisfies the constraint between the two variables.
- Path consistency: This means that for any two variables in the problem, there must be a consistent path between them such that every constraint along the path is satisfied.
- Bound consistency: This means that for a given variable, the values in its domain must be within certain bounds set by the constraints of the problem.
- Range consistency: This means that for a given variable, the values in its domain must fall within a certain range defined by the constraints of the problem.
- Singleton consistency: This means that if a variable has only one value in its domain, it must be assigned that value.

16. (a) Illustrate the use of alpha-beta pruning in games.

(b) Consider the following game tree in which static evaluation score are all from the players point of view: static evaluation score range is (+10 to -10).



Suppose the first player is the maximizing player. What move should be chosen? Justify your answer.

Answer:

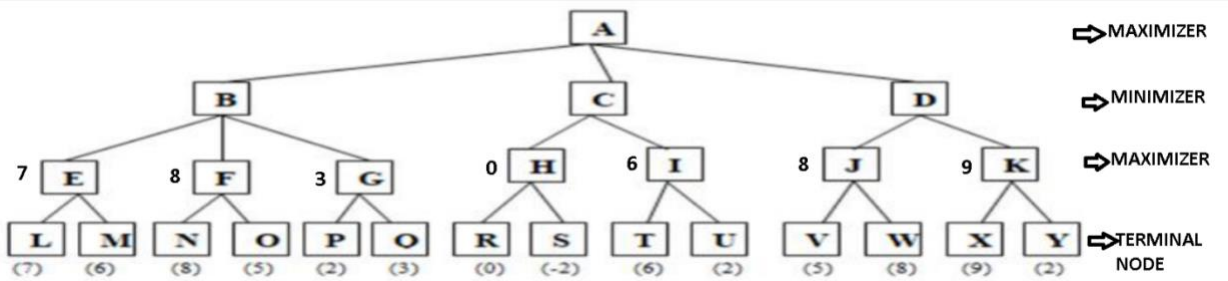
(A) Alpha-beta pruning is a technique used in two-player games, such as chess or Go, to reduce the number of nodes that need to be evaluated in the game tree. It is a refinement of the minimax algorithm, which is a search algorithm used to determine the best move for a player in a two-player, zero-sum game.

Here's an example of how alpha-beta pruning works in a simple game of tic-tac-toe:

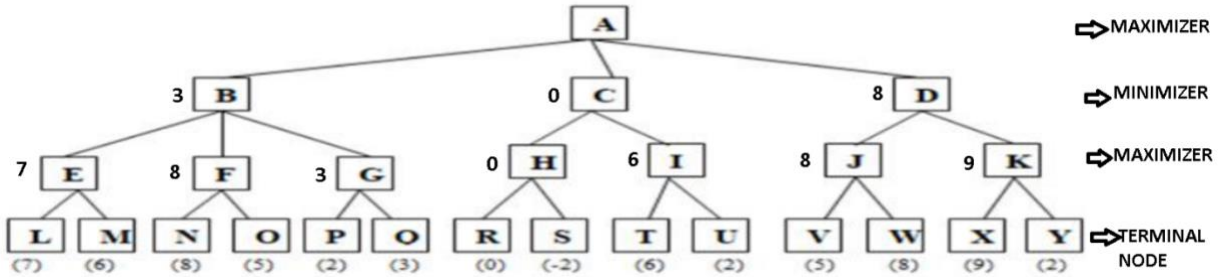
- The game tree for tic-tac-toe is generated, with the root node representing the current game state and the child nodes representing the possible moves that can be made from that state.
- The minimax algorithm is used to assign a score to each node in the game tree, based on the expected utility of that node for the current player.
- The alpha-beta pruning algorithm is used to prune the game tree, by comparing the scores of the child nodes with the scores of the parent nodes. If the score of a child node is less than the score of its parent (for the maximizing player) or greater than the score of its parent (for the minimizing player), that child node is pruned from the tree because it cannot be part of the optimal solution.
- The algorithm continues to prune the tree until it reaches a leaf node, which represents a terminal game state (i.e., a win or a draw).
- The optimal move for the current player is then determined by selecting the child node with the highest score (for the maximizing player) or the lowest score (for the minimizing player).

(B) The moves here can be chosen based on Min-Max Algorithm as show below(set the first player as maximizer player):

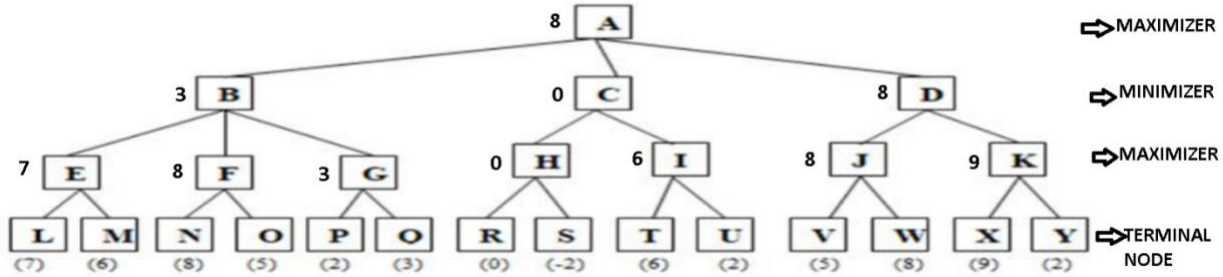
- Step 1:



- Step 2:



- Step 3:



Thus the moves that should be chosen are: A -> D -> J -> W

The score of first player(maximizer player) after these moves are 8.

17)

A) Convert the following sentences into first order logic: Everyone who loves all animals is loved by someone. Anyone who kills an animal is loved by no one. Jack loves all animals. Either Jack or Curiosity killed the cat, who is named Tuna. Did Curiosity kill the cat?

B) Give a resolution proof to answer the question "Did Curiosity kill the cat? "

Ans)

A)

1.∀x[(Loves(x, all animals) → LovedBy(x, someone))]

2.∀x[(Kills(x, animal) → ¬LovedBy(x, anyone))]

3.Loves(Jack, all animals)

4.[(Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)) ∧ Cat(Tuna)]

5.Kills(Curiosity, Tuna)

B)

To provide a resolution proof to answer the question "Did Curiosity kill the cat?", we would need to have a set of clauses representing the given information, and then apply the resolution rule to these clauses until we arrive at a contradiction or a tautology.

Here are the clauses representing the given information:

1.¬Kills(Curiosity, Tuna)

2.Kills(Curiosity, Tuna)

3.Cat(Tuna)

4.[(Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)) ∧ Cat(Tuna)]

We can then apply the resolution rule to these clauses as follows:

1.¬Kills(Curiosity, Tuna)

2.Kills(Curiosity, Tuna)

Resolving these two clauses results in the empty clause, which is a contradiction. Therefore, we can conclude that it is not the case that Curiosity killed the cat.

Alternatively, we could have resolved clauses 2 and 4 to obtain the clause "Kills(Jack, Tuna) ∧ Cat(Tuna)", and then resolved this clause with clause 3 to obtain the clause "Kills(Jack, Tuna)". This would show that it is Jack, not Curiosity, who killed the cat.

18)

A) Prove or find a counter example to the following assertion in propositional logic: If α |= (β ∧γ) then α |= β and α |= γ.

B) For each pair of atomic sentences, give the most general unifier if it exists: Older (Father (y), y), Older (Father (x), John).

Ans)

A)

The assertion "If α |= (β ∧ γ) then α |= β and α |= γ" is a tautology, which means that it is always true in propositional logic.

To prove this, we can use truth table to show that the assertion holds for all possible truth values of α, β, and γ.

Here is the truth table:

α	β	γ	(β ∧ γ)	α |= (β ∧ γ)	α |= β	α |= γ
T	T	T	T	T	T	T
T	T	F	F	F	F	T
T	F	T	F	F	F	T
T	F	F	F	F	F	T
F	T	T	F	F	F	F
F	T	F	F	F	F	F
F	F	T	F	F	F	F
F	F	F	F	F	F	F

As we can see from the truth table, the assertion "If α |= (β ∧ γ) then α |= β and α |= γ" holds for all possible truth values of α, β, and γ. Therefore, it is a tautology.

B)

unification steps:

1.Rename the variables in one of the sentences to avoid variable capture. We will rename the variables in the second sentence to avoid variable capture.

2.Unify the predicates. In this case, we can see that the predicates "Older" in both sentences are already the same, so we can proceed to the next step.

3.Unify the arguments. In this case, we can unify the first argument of the predicate "Older" in both sentences by substituting the variable x in the second sentence with the constant "John". This results in the substitution {x/John}.

4.Apply the substitution obtained in the previous step to the second argument in the second sentence. This results in the substitution {y/Father(John)}.

5.Combine the substitutions obtained in the previous steps to obtain the most general unifier {x/John, y/Father(John)}.

Therefore, the most general unifier for the pair of atomic sentences "Older(Father(y), y)" and "Older(Father(x), John)" is {x/John, y/Father(John)}.

19)

A) How is best hypothesis selected from alternatives?

B) Explain Univariate Linear Regression.

Ans)

A)

There are various methods for selecting the best hypothesis from a set of alternatives. Some common methods include:

1.Cross-validation: In this method, the data is divided into a training set and a test set. The hypotheses are learned on the training set, and then evaluated on the test set. The hypothesis that performs the best on the test set is selected as the best hypothesis.

2.Bias-variance tradeoff: In this method, the hypothesis that strikes the right balance between bias and variance is selected as the best hypothesis. A hypothesis with high bias is prone to underfitting the data, while a hypothesis with high variance is prone to overfitting the data.

3.Information criteria: There are various information criteria that can be used to select the best hypothesis from a set of alternatives, such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). These criteria balance the fit of the model to the data with the complexity of the model, and select the hypothesis that strikes the right balance between these two factors.

4.Grid search: In this method, a grid of possible values for the hyperparameters of the hypothesis is defined, and the hypotheses are trained and evaluated for each combination of hyperparameter values. The hypothesis that performs the best is selected as the best hypothesis.

B)

Univariate linear regression is a statistical method used to model the linear relationship between a dependent variable and a single independent variable.

In univariate linear regression, we are trying to model the relationship between the dependent variable y and the independent variable x as follows:

$y = \beta 0 + \beta 1 * x + \varepsilon$

where β0 and β1 are the model parameters, x is the independent variable, and ε is the error term.

The goal of univariate linear regression is to find the values of β0 and β1 that best fit the data. This can be done by minimizing the sum of squared errors between the predicted values of y (obtained using the model) and the actual values of y.

Once the values of β0 and β1 have been determined, the univariate linear regression model can be used to predict the value of y for a given value of x.

Univariate linear regression is a simple and widely used method for modeling the relationship between a dependent variable and a single independent variable. It is a useful tool for understanding the relationship between variables and for making predictions.

20)

A) Consider the following data set comprised of two binary input attributes (A1 and A2) and one binary output.

Example	A_1	A_2	Output y
x_1	1	1	1
x_2	1	1	1
x_3	1	0	0
x_4	0	0	1
x_5	0	1	0
x_6	0	1	0

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

B) Explain Linear classification with logistic regression

Ans)

A)

First find Entropy(S) whole dataset

Entropy(s)= -P(yes)$\log_2$ P(yes)- P(no) $\log_2$ P(no)

$$=-\frac{3}{6}log_2\frac{3}{6}-\frac{3}{6}log_2\frac{3}{6}=1$$

Gain(S, A1)= ES $-\{\frac{|S_{A1=1}|}{|S|}\,Entropy(S_{A=1})+\frac{|S_{A1=0}|}{|S|}\,Entropy(S_{A1=0})\}$

$S_{A1=1}$

X	A1	Y
X1	1	1
X2	1	1
X3	1	0

$|S_{A1=1}|=3$

$Entropy(S_{A1=1})=-P_1\log_2 P_1 - P_0 \log_2 P_0$

$$=-\frac{2}{3}log_2\frac{2}{3}-\frac{1}{3}log_2\frac{1}{3}=0.9183$$

$S_{A1=0}$

X	A1	Y
X4	0	1
X5	0	0
X6	0	0

$|S_{A1=0}|=3$

$$Entropy(S_{A1=0})=-\frac{1}{3}log_2\frac{1}{3}-\frac{2}{3}log_2\frac{2}{3}=0.9183$$

$$\text{So } Gain(S,A_1)=\text{ES}\ -\{\frac{|S_{A1=1}|}{|S|}\ Entropy(S_{A1=1})+\ \frac{|S_{A1=0}|}{|S|}\ Entropy(S_{A1=0})\}$$

$$=1-\left\{\frac{3}{6}\ X0.9183+\ \frac{3}{6}\ X\ 0.9183\right\}=0.0817\ \text{----(1)}$$

$S_{A2=1}$

X	A2	Y
X1	1	1
X2	1	1
X5	1	0
X6	1	0

$|S_{A2=1}|=4$

$Entropy(S_{A2=1})=-P_1\log_2 P_1 - P_0 \log_2 P_0$

$$=-\frac{2}{4}log_2\frac{2}{4}-\frac{2}{4}log_2\frac{2}{4}=1$$

$S_{A2=0}$

X	A2	Y
X3	0	0
X4	0	1

$|S_{A2=0}|=2$

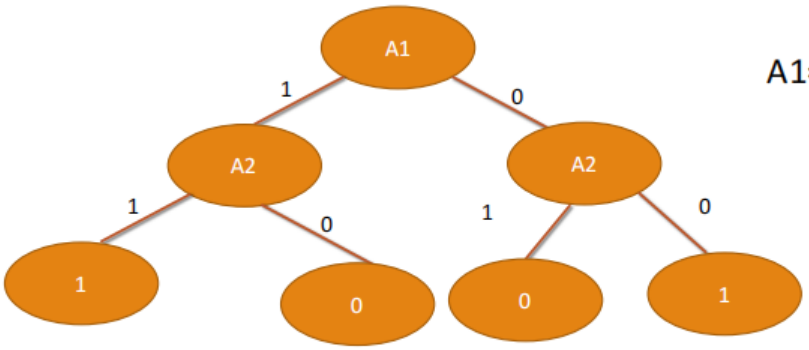
$$Entropy(S_{A1=0})=-\frac{1}{2}log_2\frac{1}{2}-\frac{1}{2}log_2\frac{1}{2}=1$$

$$\text{So } Gain(S,A_2)=\text{ES}\ -\{\frac{|S_{A2=1}|}{|S|}\ Entropy(S_{A2=1})+\ \frac{|S_{A2=0}|}{|S|}\ Entropy(S_{A2=0})\}$$

$$=1-\left\{\frac{4}{6}\ X\ 1+\ \frac{2}{6}\ X\ 1\right\}=0\ \text{---------(2)}$$

From (1) and (2) We found that A1 give more information gain.

So we make A1 as root node



A1=1

X	A2	Y
X1	1	1
X2	1	1
X3	0	0

A1=0

X	A2	Y
X4	0	1
X5	1	0
X6	1	0

B)

Linear classification is the process of using a linear function to separate a set of data points into two or more classes. Logistic regression is a type of linear classification method that is used to predict the probability that an instance belongs to a particular class.

In logistic regression, the goal is to learn a function that can predict the probability P(y|x) that an instance x belongs to a class y, given a set of feature values x. The function takes the form of a logistic function, which is defined as follows:

P(y|x) = 1 / (1 + e^-(wx + b))

where w is a vector of weights, x is a vector of feature values, b is a bias term, and e is the base of the natural logarithm.

The logistic function maps the input values to a value between 0 and 1, which can be interpreted as the probability that the instance belongs to the positive class. The function is plotted as an "S" shaped curve.

To train a logistic regression model, we need to learn the values of the weights w and the bias term b that result in the best fit to the training data. This can be done using an optimization algorithm such as gradient descent.

Once the model has been trained, we can use it to predict the probability that a new instance belongs to the positive class. If the probability is greater than 0.5, we can predict that the instance belongs to the positive class, and if it is less than 0.5, we can predict that the instance belongs to the negative class.

Logistic regression is a simple and effective method for linear classification, and is widely used in a variety of applications.