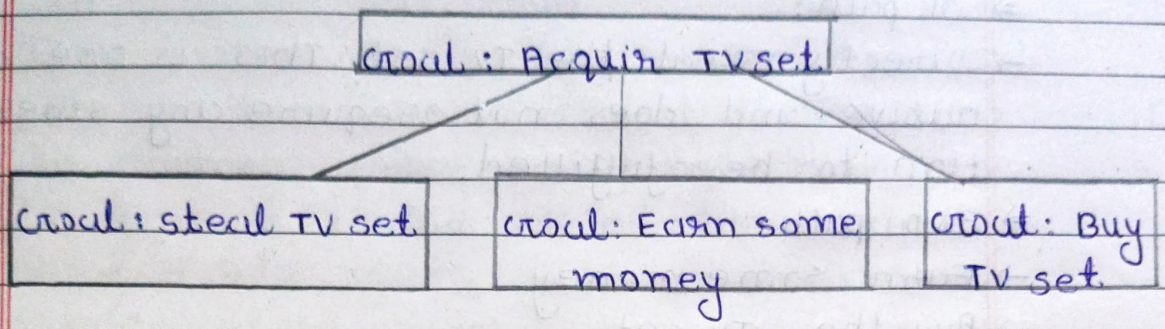
Assignment

1. What is an AND-OR graph? How is it useful? Briefly describe what the following AND-OR graph illustration



⇒ What is an AND-OR graph? How is it useful?

→ An AND-OR graph is a graphical representation used in problem-solving and artificial intelligence to model complex problem that can be taken down into sub-problems. It consist of two type of nodes:

⇒ OR node: Represent situations where any one of the sub-goal needs to the achived to each the main goal.

⇒ AND nodes: Represent situations where all the sub-goals must be achived together to reach the main goal.

⇒ Usefulness:

→ And-or graphs helps in representing probem that can be solved through multiple path.

→ They are used in field like automated planning game theory, expert system, and decision making.

→ They provide a structured way to represent goals and alternative solutions, making problem-solving more efficient.

→ The given graph illustrates the goal: "Acquire TV set."
→ This goal can be achived in the following way.
⇒ OR path:
→ Directly steal the TV set. This is one alternative and does not require any other condition to be fulfilled.
⇒ AND path:
→ Earn some money
→ Buy the TV set.
→ Both this action must be completed to acquire the TV legally.

2 write and explain 8 step of A* algorithm. Mention the limitation of A* algorithm.
⇒ A* search algorithm :
→ The A* algorithm is widly used pathfinding and graph traversal algorithm. It is often used in artificial intelligence and game development to find the shortest path from a start node to goal node.
→ g(n) : the cost from the start node to the current node n.
→ h(n) : a heuristic estimation of the cost from n to the goal.
→ f(n) = g(n) + h(n) : the total estimated cost of the cheapest solution through n.

⇒ Steps of the the A* algorithm :
1. Initialize the open list and add the start node.
2. Initialize the closed list as empty.

3. Repeat the following steps until the goal is reached or the open list is empty.
a. select the node n from the open list that has the lowest f(n) value.
b If n is the goal node, return the path.
c Move n from the open list to the closed list.
d for each neighbor m of node n:
(i) if m is in the closed list, skip it.
(ii) calculate g(m) = g(n) + cost(n, m).
(iii) calculate h(m)
(iv) calculate f(m) = g(m) + h(m)
(v) If m is not in the open list, add it.
(vi) If m is already in the open list with a higher f(m), update it with the lower value.
4. If the open list is empty and the goal was not reached, return failer.

⇒ Limitation of A* Algorithm:
1. High memory consumption
2. Performance depends on heuristic.
3. May explore unnecessary path
4. Not suited for Real-time application.
5. Inconsistent Heuristic cause suboptimal path.

3. Explain the analogy behined simulated annealing. How does this algorithm differ from simple hill climbing?
⇒ Simulated Annealing - Analogy:
→ The concept of simulated Annealing comes from metallugy, a process in which a material is heated to high temprature and the slowly

cooled to remove defects and increase strength.
→ when a material is heated its atoms moves freely.
→ As it cools slowly, atoms settle into a minimum energy configuration.
→ If cooled too quickly, the atoms may settle in local minimum, resulting in a weak structure.
→ In Artificial intelligence, simulated annealing applies the same idea to optimization problem
→ explore solutions that may seem worse temporarily.
→ over time, it reduce the probability of accepting worse solutions.

⇒ steps of simulated Annealing :
1. start with an initial solution and an initial high temprature.
2. Repeat until system cools :
   a. slightly changes the current solution to generate a new solution.
   b. Calculate the difference in cost between the new and current solution.
   c. If the new solution is better, accept it
   d. If the new solution is worse, accept it with a certain probability based on the temprature.
   e. Gradually reduce the temprature.

Aspect	simulated Annealing	simple hill climbing
→ search behave.	can move to worse solution temporarily.	only move to better or equal solution.
→ Escape from local optima.	yes, due to probabilitic acceptace.	No, gets stuck in local optima
→ Uus of probability.	→ Accepts worse solution with decreasing probability as it cool.	Does not accept worse solution.
→ Final solution.	More likely to find the global optimum.	→ often ends up in a local optimum
→ cooling schedule	Temprature decreses over time to refine the search.	No seach concept

4. Illustrate MinMax algorithm on the game of Tic-Tac-Toe. Explain how alpha-beta cut-offs improve efficiency of search in a game tree. what is quiescence?

→ The minimax algorithm is a recursive decision making algorithm used in two-player games like Tic-Tac-Toe. It assumes:

→ one player is the maximizer

→ The other player is minimizer.

+1 → win for max

-1 → win for min.

0 → Draw

⇒ Alpha - Beta cut-offs:

→ Alpha-Beta pruning improves the efficiency of the minimax algorithm by pruning branches that cannot affect the final decision

→ Alpha: The best already explored value along the path to the maximizer.

→ Beta: The best already explored value along the path to maximizer.

⇒ Quiescence:

→ Refers to the stable state of a game position where no dramatic changes are about happen.

→ In standard minimax a cutoff depth is used to limit the depth of search.

→ If a cutoff occurs at an unstable node, it may lead to poor evaluation.

5 How does minimax differ from expectimax?

Feature	Minimax	Expectimax
→ opponent type	Assume optimal adversary	Assume random behavior based on probability
→ Game Type	Deterministic, adversarial	stochastic
→ Node type	Max and min node.	Max and chance nodes
→ Decision strategy	Min player choose worst-case for max	chance node calculates average case
→ Applications	chess, Tic-Tac-Toe	Backgammon, Dice game, poker

6. What is meant by zero-sum game? Name two such game.

→ A zero-sum game is a type of competitive game where the total gain of one player is exactly equal to the total loss of the other player.

→ If player A gains +1, player B must lose -1

⇒ characteristics:

→ compititive in nature.

→ Involves perfect conflict between player.

→ Minimax algorithm is often used on determine optimal strategy.

→ Example games: Tic-Tac-Toe, chess

7. Explain case based reasoning? List the perference heuristic for CBR.

→ CBR is a problem-solving paradigm that uses past experiences to understand and solve new problems.

⇒ steps in CBR cycle:

1. Retrieve: Find the most similar case from the case database.
2. Reuse: Apply the solution from the past case to the current problem.
3. Revise: Adapt and test the proposed solution to ensure it fits new context.
4. Retain: Save the new experience as a part of a new cases in the case base for future.

=> Preference Heuristics:
-> Similarity Heuristic.
-> Recency Heuristic.
-> Frequency Heuristic.
-> Sucess Heuristic
-> Typicality Heuristic

8. what is episodic memory and semantic memory? How are they useful for commonsense reasoning?

=> Episodic memory:
-> Refers to the memory of personal experience and specific event that occurred at a particular time and place.

=> semantic memory:
-> semantic memory refers to the memory of general facts, concepts, and knowledge that are not tied to personal experiences

=> Use in commonsense Reasoning:
-> commonsense reasoning is the ability of AI to make human-like assumptions and decisions in everyday situation. Both type of memory play a crucial role in this:

-> Episodic memory : contextual knowledge.
-> semantic memory: general world knowledge

9. Give difference between "fuzzy logic" and "crisp logic". Given an example that clearly differentitate between the two.

Aspect	crisp logic	Fuzzy logic
-> Nature	Binary	Multivalued.
-> values	only 0 or 1	between 0 and 1
-> Decision Boundry.	sharp and well defined.	Vague and flexible.
-> Type of thinking	Human-like strict logic	Human-like approximate reasoning
-> Application	Digital system, programming language.	washing machine, AC, control system.
-> complexity Handling.	cannot handle vague or imprecise data	can handle data like vague, imprecise
-> Example.	if temp ≥ 30 -> Hot Else -> Not hot & if temp = 29.9°c <- not if temp = 30° <- Hot	At 28°c -> 0.3 Hot 30°c -> 0.6 Hot 35° -> 0.9 Hot

to Explain nonmonotonic reasoning.

-> Nonmonotonic reasoning refers to a type of logical reasoning in which adding new information can reduce the set of conclusion. This is, previous conclusion may need to be withdrow in light of new evidence.

=> characteristics :

-> Dynamic : change over time.

-> Flexible : Allows system to revise beliefs.

-> Models human thinking : Human often change their ebelife when they learn somting new.

-> Tinitial Knowledge
Bird can fly
"Tweety is a bird"
-> Tweety can fly.

-> New information added
"Tweety is pegnuin"
-> Tweety can not fly.

Q1 Explain forward and backward reasoning with a suitable example

⇒ Forward Reasoning (Forward chaining):

→ Forward reasoning is a data-driven reasoning approach where the system start from known fact and applies inference rules to extract more facts until the goal is reached.

→ used in : Expert systems, production system

→ Example :

R1 : If A and B, then C

R2 : If C then D

R3 : If D, then E

⇒ Backward Reasoning (Backward chaining):

→ Backward reasoning is a goal-driven approach. It starts from the goal and works backward to find the fact that supports it.

→ used in: Problem-solving, theorem providing goal to prove F from above lines

Q2 consider the following facts :

→ solve only like easy courses.

→ science course are hard.

→ All the courses in the HaveFun department are easy.

→ BK301 is HaveFun department course.

→ use resolution to answer the question "what course whould steve like?"

⇒ Given facts (converted in predicate logic)
→ Likes (steve, x)
→ Easy (x)
→ Hard (x)
→ Science (x)
→ HaveFun (x)
⇒ Resolution steps :
→ HaveFun (BK301)
→ ¬ HaveFun (x) ∨ Easy (x)
Substitute = BK301
¬ HaveFun (BK301) ∨ Easy (BK301)
→ ¬ Likes (steve, x) ∨ Easy (x)
¬ Likes (steve, BK301) ∨ Easy (BK301)

13 State the Bayes theorem. Illustrate how a Bayesian Network can be used to represent causality relationship among attribute
→ It is a fundamental concept in probability theory that describes the probability of a hypothesis given observed evidence.

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)}$$

⇒ Bayesian network:
→ A Bayesian network is graphical model that represents a set of variables and their conditional dependencies using a directed acyclic graph (DAG)
→ Nodes : Represent random variable.
→ Edge : Represent direct dependencies
→ Each node has a conditional probability table that quantifies the effect its parents.

14. Explain cut and Fail predicate with example

⇒ Cut (!) predicate in prolog :

→ The cut operator in prolog used to control backtracking. It commits prolog to the choice made at the point of the cut and prevents it from trying alternative rule on clauses

→ Example:

```
grade(Marks, pass) :- Marks >= 50, !.
grade(Marks, fail).
```

⇒ Fail predicate in prolog:

→ The fail predicate always fails when executed. It is used to focus backtracking or to singal that no solution exist for a clause.

→ Example:

```
show_numbers :- write(1), nl, fail.
```

15 Explain the different issues in knowledge representation.

→ Knowledge representation is a critical area of artificial intelligence that deals with how knowledge can be represented symbolically and manipulated in an automated in an automated way by reasoning program.

⇒ Representational Adequacy.

→ The system should be able to repeat all kinds of knowledge required for a given task.

=> Inferential Adequacy:
-> The system must be derive new knowledge from existing knowledge using inference rule.
=> Inferential Efficiency:
-> Inference mechanisms must be computationally efficient.
=> Acquisitional Efficiency:
-> The system should allow for easy acquisition of knowlege, euther manually by the user or automatically via learning.
=> Transparency or human understandability:
-> The representation should be understandable and interpretable by human.
=> consistency and completeness:
-> The knowlege base should be logically consistent and complete enough to answer queries.
=> Handling Uncertainty:
-> Real-world information is often incomplete, uncertain, or ambiguos.
=> Expressiveness and scalability:
-> The KR language must be expressive enough to capture complex relationship.

16 write aprolog program to find minimum num. from the given input list.

-> domains:

```
list = integer*
num = integer*
```

predicates

minimum (list, num)

clauses

minimum ([x], x).

minimum ([H|T], min):-

minimum (T, Tempmin),

~~minimum (T~~

min is min (H, Tempmin).

=) output:

minimum ([21, 20, 19, 22, 28], x)

x = 19 (19 is the minimum).