1) 2021

(a) What is Artificial Intelligence?

⇒ Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and learn like humans.

(b) What is meant by Knowledge Inference?

⇒ Knowledge Inference refers to the process of drawing conclusions or making deductions from existing knowledge or data.

It involves using reasoning, logic, and rules to infer new information, relationships, or insights that are not explicitly stated in the original data.

* In other word knowledge inference is the ability to drive new knowledge from existing knowledge.

(c) What is contradictory sentence?

⇒ A contradictory sentence is a statement that asserts two or more incompitible or opposite ideas, making it impossible for both to be true at the same time.

Example

"I am going to the party, and I am not going to the party."

The sentence is contradictory because it states two opposing idea's simultaneously.

(D) write one Advantage of DFS?

→ One Advantage of DFS (Depth first search) is:

It can detect cycles in a graph.

(E) Translate the sentence - "Mangoes are sweet" into a formula in predicate logic.

⇒ Let $M(x)$ be the predicate "x is mango"
Let $S(x)$ be the predicate "x is sweet"

The sentence "Mangoes are sweet" can be ~~treated/~~translated to:

$$\forall x \, (M(x) \rightarrow S(x))$$

This formula reads: "For all x, if x is mango, then x is sweet."

(F) Construct the truth table for the expression $(A \wedge (A \vee B))$

→

A	B	A ∨ B	A ∧ (A ∨ B)
T	T	T	T
T	F	T	T
F	T	T	F
F	F	F	F

T = True
F = False
$\wedge$ = Conjuction (AND)
$\vee$ = Disjunction (OR)

(G) write name of some Intelligent Systems.

→ 1. Siri
2. Alexa
3. Google Assistant
4. IBM Watson
5. Alpha GO
6. Self Driving cars
7. Expert systems
8. Chat bots

(h) Write one difference between a frame and a script.

⇒ A frame represents a static concept or object, whereas a script represents a dynamic sequence of events or actions.

2022

(1) (a) Why do we need Artificial Intelligence?

⇒ We need AI for various things like, Automation, Efficiency, Innovation, and improved decision-making.

(b) What is constraint satisfaction problem (CSP)?

⇒ A constraint satisfaction problem is a problem where:

1. There are variables with specific domains (possible values).
2. There are constraints that restrict the values of the variables.
3. The goal is to find an assignment of the values to variables that satisfies all the constraints.

Example: Scheduling, Resource Allocation, and Logic Puzzles.

(c) Which domain study Artificial Intelligence?

⇒ 1. Computer science
2. Mathematics
3. Engineering
4. Cognitive Psychology
5. Neuroscience
6. Philosophy
7. Linguistics

(d) What is meant by semantics Net?

→ A semantic net is a knowledge representation technique that uses a network of nodes and ares to represent relationships between concepts, obtects, and ideas.

(e) Constructs a truth table for the expression $(P \wedge (\sim P \vee q))$

→

T → True
F → false
∧ → AND
∨ → OR
~ → NOT

P	q	~P	~P ∨ q
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

P ∧ (~P ∨ q)
T
F
F
F

(F) Explain Turing testing in AI?

→ The Turing Test is a method for determining whether a machine, such as a computer program, is capable of thinking like a human being.

(G) Translate the sentence "Every man respect his parent" into a formula in first order predicate Logic (FOPL).

→ Let:

M(x) = x is a man
R(x,y) = x respect y
P(x,y) = y is a parent of x

The sentence "Every man respects his parent" can be ~~written~~ translated into FOPL as:

$\forall x (M(x) \rightarrow \exists y (P(x,y) \wedge R(x,y)))$

(a) What do we ~~want~~ meant by universal and existential quantifier?

In first-order predicate logic (FOPL), quantifiers are used to specify the scope of a variable.

1. Universal Quantifiers ($\forall$): "for all" or "For every". It asserts that a property is true for all elements in the domain.

   Example: $\forall x P(x)$ means "For all x, $P(x)$ is true"

2. Existential Quantifiers ($\exists$): "There exists" or "For some". It asserts that a property is true for at least one element in the domain.

Example: $\exists x P(x)$ means "There exists an x such that $P(x)$ is true".

2022-23

1 (a) What is constraint satisfaction problem (CSP)?

(b) Give some real world applications of AI.

→ Virtual Assistants, Healthcare, Natural Language Processing, Image processing, predict, Outcome,

(c) What are the different domains/subset of AI

⇒ Machine Learning (ML),
Natural Language Processing (NLP),
Computer vision (CV),
Robotics,
Expert systems,
Neural Network,
Deep Learning,
Cognitive computing,
Human computer Interaction,
Speech Recognition.

(d) Explain Turing test in AI.

(e) What is meant by ~~Semantics~~ Net?

(f) What do you mean by Heuristic search Technique?

⇒ Heuristic search Techniques are methods used to find the optimal solution to a problem by guiding the search process with heuristics, which are rules of thumb or educated guesses.

(g) What is predicate logic in AI?

⇒ Predicate logic in AI is a ~~formal~~ system for representing and reasoning about knowledge using predicates, variables, and quantifiers.

(h) What do you mean by Backtracking in AI?

⇒ Backtracking in AI is a problem-solving strategy that involves:

1. Exploring a solution spaces: ~~Symmetry~~
   ⇒ systematically trying different possibilities.
2. Making a choice: selecting a possible ~~solution~~ solution or path

3. Evaluating the choice: checking if the chosen path leads to a solution.

4. Backtracking: if the chosen path fails, reverting to a previous choice and trying an alternative path.

2023-24

(1) (a) what is artificial Intelligence?

(b) what do you meant by universal and existential quantifier?

⇒ In predicate logic, quantifiers are used to specify the scope of variables.

Universal quantifier (∀):

- Means "for all" or "for every".
- Indicates that a statement is true for all values and variables

Existential quantifiers (∃):

- Means "there exists" or "for some".
- Indicates that a statement is true for at least one value of a variable.

(c) what is contradictory sentence?

(d) what is meant by semantics Net?

(e) Translate the sentence → "mangoes are sweet" into a formula in predicate logic.

⇒ Let M(x) = "x is a mango"

S(x) = "x is sweet"

So, the sentence can be translate as:

$\forall x (M(x) \rightarrow S(x))$

"for all x, if x is a mango, then x is sweet."

(f) What is constraint satisfaction problem (CSP)?

(g) Write name of some Intelligent systems.

(h) Write one difference between frame and a script.

⇒ Frame: Represents a static concept or object focusing on its attributes and properties.

Script: Represents a dynamic sequence of events or actions, focusing on the temporal relationship between them.

In other words, frames capture the "what", while script capture the "how" and "when".

5×2

2021

(a) How do you measure exactly 2 gallons of water from a well using a 4-gallon Jug and a 3 gallon Jug? (Neither has any ~~mark~~ measuring mark on it).

⇒ To do that we have to follow few steps to solve the problem.

Step-1: Fill the 4-gallon Jug from the well, so it contain exactly 4 galons of water.

Step-2: Pour water from the 4-gallon Jug into 3 gallon Jug until the 3-gallon Jug is full, leaving 1 gallon remaining in the 4-gallon Jug.

Step-3: Empty the 3-gallon jug completely.

Step-4: Pour the remaining 1 gallon from the 4-gallon Jug into the 3-gallon Jug.

Step-5: Fill the 4-gallon Jug again from the well.

Step-6: Pour water from the 4-gallon Jug into the 3-gallon Jug until the 3-gallon Jug is full, which will require 2 gallons (since the 3 gallon Jug is already contains 1 gallon).

At the End we have exactly 2 gallons of water left in the 4-gallon Jug.

(6) What is FOPL? write the following expressions in FOPL: *

i) All employees earning $1400 or more per year pay taxes

ii) Some employees are sick today

iii) No employee earns more than the president.

⇒ FOPL stands for first-order predicate logic. It's a formal system for representing and reasoning about logical statements using predicates, variables, and quantifiers.

Here are the expressions written in FOPL:

i) All employees earning $1400 or more per year pay taxes:

$\forall x$ (Employee(x) $\wedge$ Salary(x) $\geq$ 1400 $\rightarrow$ PaysTaxes(x))

Read as: "for all x, if x is an employee and x's salary is greater than or equal to 1400, then x pays taxes."

ii) Some employee are sick today:

$\exists x$ (Employee(x) $\wedge$ Sick(x))

Read as: "There exists an x such that x is an employee and x is sick."

iii) No employee earns more than president.

$\neg \exists x$ (Employee(x) $\wedge$ Salary(x) > Salary(President))

Read as: "It is not the case that there exist an x such that x is an employee and x's salary is greater than the president's salary

Alternatively

$\forall x$ (Employee(x) $\rightarrow$ Salary(x) $\leq$ Salary(President))

© Write down the importance of control strategy, ~~and its importance~~ in production systems.

⇒ Control strategy plays a vital role in production systems, and its importance can be summerized as follows:

1. Improved Productivity: A well-designed control strategy helps optimize production processes, reducing downtime, and increasing overall productivity.

2. Enhanced Quality Control: Control strategies enable real-time monitoring and control of production processes, ensuring that products meet quality standards

3. Increased Efficiency: By automating and optimizing production processes, control strategies help reduce energy consumption, minimize waste, and lower production costs.

4. Reduced Risk: control strategies help identify and mitigate potential risks, such as equipment failures, raw material shortages, or changes in demand.

5. Flexibility and Adaptability: A robust control strategy enables production systems to adapt ~~to changes~~ quickly to changes in demand, production schedules, or equipment availability.

1 (d) What is resolution? write down the algorithm of resolution.

⇒ Resolution is a rule of inference in logical reasoning that allows us to deduce a conclusion from two or more premises. It's fundamental concept in artificial intelligence, logic, and computer science.

Definition

Resolution is a process of combining two or more clauses (sets of literals) to produce a new clauses.

Algorithm

1. Select two clauses: Choose two clauses, C1 and C2, from the knowledge base or premises.
2. Check for complementary literals: Look for a literal in C1 that is complementary to a literal in C2. Two literals are complementary if one is the negation of the other (e.g., P and ¬P).
3. Resolve the clauses: If a complementary pair of literals is found, remove both literals from their respective clauses. The resulting clause is the resolvent.
4. Simplify the resolvent: If the resolvent contains duplicate literals, remove the duplicates

5. Add the resolvent to the knowledge base:

Add the simplified resolvent to the knowledge base or promises.

6. Repeat the process: Repeat the steps 1-5 until new resolvents can be generated.

Example

Suppose we have the following clauses:

C1: P V Q
C2: ¬Q V R

We can resolve these clauses by noticing that Q and ¬Q are complementary literals.

Resolvent: P V R

The resolvent is added to the knowledge base, and the process is repeated until no new resolvent can be generated.

2022

@ What do you meant by search in AI

⇒ In Artificial Intelligence (AI), search is refers to the process of finding a solution or a goal state within a problem space or graph. The problem space is represented as a set of states, and the goal is to find a path from a Initial state to a goal state.

Distinguish between informed and uninformed search technique in AI.

(1) Uninformed search Techniques.
also known as blind sear, do not use any additional information about the problem space. They rely solely on the definition of the problem and the goal state.

Examples
- Breadth-first-search (BFS)
- Depth-first search (DFS)
- Uniform cost search (UCS)

(2) Informed search Techniques
Informed search techniques, also known as ~~heuristic~~ heuristic search, use additional information about the problem space, such as heuristics or evaluation functions. This information guids the search towards the goal state more efficiently

Example
- Best-First search (BFS)
- A* search (A-star search)
- Greedy search

differences

1) Informed search techniques use heuristics or evaluation functions to guide the search, while uninformed techniques do not.

2) Informed search techniques are generally more efficient than uninformed search techniques, especially for large problem spaces.

3) ~~Optimality~~ Informed search techniques can guarantee optimality (finding the best solution) if the heuristic function is admissible and consistent. uninformed search techniques may not guarantee optimality.

4) ~~Computational~~ Informed search techniques often have higher computational complexity than uninformed search techniques due to the additional calculations required for heuristic evaluation.

(5) What are the limitations of Hill climbing algorithm? write down the difference between simple Hill climbing and steepest Hill climbing?

Limitations of Hill climbing Algorithm

1. Local optima: Hill climbing algorithms can get stuck in local optima, which are solutions that are better than their neighbors but not the global optimum.
2. Premature convergence: Hill climbing algorithms can converge prematurely, which means they may stop improving the solution too early.
3. Sensitivity to initial solution: The performance of hill climbing algorithms can be highly sensitive to the initial solution.
4. Not suitable for complex problems: Hill climbing algorithms are not suitable for complex problems with many local optima.
5. No guarantee of optimality: Hill climbing algorithms do not guarantee optimality, even if they converge to a solution.

Difference between simple Hill climbing and steepest Hill climbing.

1. Exploration strategy: simple Hill climbing explores the neighborhood of the current solution randomly, while steepest Hill climbing explores the entire neighborhood and select the best neighbor.

2. Selection of next solution: simple Hill climbing selects the next solution based on a random neighbor, while steepest Hill climbing selects the next solution based on the best neighbor.

3. Convergence Rate: steepest Hill climbing converges faster than simple Hill climbing, especially for problems with a large number of local optima.

4. Computational complexity: Steepest Hill climbing has a higher computational complexity than simple Hill climbing, especially for large neighborhoods.

(f) What do you meant by knowledge interface?

⇒ A knowledge interface refers to the interaction between a human expert and a computer system where the expert provides knowledge to the system, and the system represents, stores, and uses that knowledge to solve problems or make decisions.

The knowledge interface is responsible for:

1. Knowledge acquisition: Gathering knowledge from human experts or other sources
2. Knowledge representation: Storing and organizing the acquired knowledge in a format that the computer can understand.
3. Knowledge retrieval: Accessing and retrieving ~~and~~ the stored knowledge as needed.

Techniques of Knowledge Representation in AI:

There are several techniques used to represent knowledge in AI systems:

1. Semantic networks: A graphical representation of knowledge using nodes and edges to represents concepts and relationships.
2. Frames: A knowledge representation techniques that uses a structured framework to organize knowledge about objects, concepts, and ~~systems~~ solutions.
3. Rules: A knowledge ~~Re~~ Representation technique that uses if-then statement to represent knowledge about relationships between concepts.

4. Decision Tree: A knowledge representation technique that uses a tree like structure to represent knowledge about decision making processes.

5. Ontologies: A formal representation of knowledge that defines a set of concepts, relationships, and rules to reason about a specific domain.

6. Pre positional and first-order Logic: Formal languages used to represent knowledge using logical statements and predicates.

7. Associative networks: A knowledge representation techniques that uses a network of associations between concepts to represent knowledge.

8. Conceptual Graphs: A knowledge representation technique that uses a graphical structure to represent knowledge about concepts and relationships.

The knowledge representation techniques enable AI systems to store, organize, and use knowledge to solve problems, make decisions, and learn from experience.

(d) Write FOPL for the following expressions:

i) Some people who lives in India.

ii) Not all students like both mathematics and science.

iii) Only one student failed in mathematics.

→ i) Some people who live in India:

$\exists x (\text{person}(x) \wedge \text{Lives in}(x, \text{India}))$

Read as: "There exists a person x such that x is a person and x lives in India."

ii) Not all students like both mathematics and science

$\neg \forall x (\text{Students}(x) \rightarrow \text{Likes}(x, \text{mathematics}) \wedge \text{Likes}(x, \text{science}))$

Read as: "It is not that case that for all x, if x is a student, then x likes mathematics and x likes science."

iii) only one student failed in mathematics:

$\exists x (\text{Student}(x) \wedge \text{failed}(x, \text{mathematics}) \wedge \forall y (\text{student}(y) \wedge \text{failed}(y, \text{mathematics}) \rightarrow y = x))$

Read as: "There exists a student x such that x failed mathematics, and for all y, if y is a student and y failed mathematics, then y is equal to x."

2022-23

@ Which algorithm is used in tic-tac-toe? How does Tic-Tac-Toe AI Algorithm works? what is the time complexity in AI?

⇒ The algorithm used in tic-tac-toe is the Minimax Algorithm.

The minimax algorithm is a recursive algorithm used for decision making in games, like Tic-Tac-Toe, Chess, Go. It ~~contains~~ considers all possible moves, their outcomes, and the opponent's possible responses.

Working of the Algorithm

1. Game Tree: The algorithm creates a game tree, which represents all possible moves and their outcomes.
2. Minimax function: The algorithm applies the Minimax function to each node in the game tree. The Minimax function evaluates the best move for the current player (MAX) and the best response for the opponent (MIN)
3. Evaluation Function: The algorithm uses an evaluation function to assign a score to each node in the game tree. The score represents the desirability of the move.
4. Recursion: The algorithm recursively applies the Minimax function to each ~~move~~ node in the game tree, exploring all possible moves and their outcomes.

5. Best Move selection: The algorithm selects the best move by choosing the node with the highest score (for MAX) or the lowest score (for MIN).

Time complexity

The time complexity of the minimax algorithm in Tic-Tac-Toe is $O(b^d)$,

(b) What are the limitations of hill climbing algorithm? (2022-b)

How do you use the A* Algorithm?

→ A* (pronounced "A-star") is a popular pathfinding algorithm used to find the shortest path between two points in a weighted graph or network.

Steps

1. Define start and goal nodes: Identify the starting point and the destination.
2. Choose a heuristic function: Select a function that estimates the distance from a node to the goal node.
3. Initialize priority queue: Create a priority queue with the start node.
4. Evaluate nodes: Dequeue a node, calculate its cost, and enqueue its neighbors.

5. Repeat 2 continue until the goal node is reached

6. Reconstruct the path: Trace back the nodes from the goal node ~~the~~ to the start node.

Time complexity: $O(b^d)$

Space complexity: $O(b^d)$

(c) What do you mean by ~~knowledge Infe~~ Inferential knowledge

⇒ Inferential knowledge refers to the knowledge that is derived or inferred from existing knowledge using reasoning, deduction, or inference. It involves making logical connections between pieces of information to arrive at new conclusions or insights.

Inferential knowledge is obtained by:

1. Deductive reasoning: using logical rules to derive conclusions from premises.
2. Inductive reasoning: making generalizations or drawing conclusions based on specific instance or observations.
3. Abductive reasoning: making educated guesses or hypotheses based on incomplete information.

~~(c)~~ What are the various techniques of knowledge representation in AI? (2022-C

(d) which algorithm is better between BFS and DFS & Explain why?

→ It depends on the type of problem we're trying to solve, but typically BFS is often considered better in AI, especially for problems that require finding the shortest solution or for exploring all possibilities in an optimal manner.

Here's why:

1. optimal solution for unweighted problems:
   - BFS guarantees the shortest path in an unweighted graph. This is particularly useful in AI applications like pathfinding. Since BFS explores level by level, the first time it reaches the goal, it does so with the minimum number of steps.
   - DFS, on the other hand, does not guarantee the shortest path. It might explore a long path before finding the solution, which could result in a sub-optimal solution in such cases.
2. completeness:
   - BFS is complete, meaning it will always find a solution if one exists. It is crucial in AI, where we want to ensure that if a solution exists, the algorithm will find it.

- DFS can fail to find a solution if the search is deep or infinite.

3. Level-wise search
   - In many AI problems, we are interested in the structure of the solution space, such as finding solutions in a layered manner or evaluating possible moves in a game. BFS is naturally works well for these types of problems, like in puzzles, where each move represents a different depth level.

4. Guaranteed to find a Goal
   - In AI problems like search and game playing, where we aim to reach a particular goal (e.g., solving a puzzle or reaching a target state), BFS will always find the goal state at the shallowest depth level. This makes it the preferred choice when you need to ensure the goal is reached efficiently.

2023-24 -

(a) (2023-a)

(b) (2021-b)

(c) (2022-c)

(d) (2021-d)

10x1

2021

@ what is Turing test?

The Turing test, proposed by Alan Turing evaluates a machine's ability to exhibit intelligent behavior indistinguishable from a human. A human Judge interacts with a machine and a human via text. If the Judge can't reliably distinguish the machine from the human, the machine passes the test

@ write down the algorithm to build tic-tac-toe?

⇒ To build an AI for Tic-Tac-Toe, use the Minimax algorithm. Here's the algorithm:

Step-1: Represent the Tic-Tac-Toe board as 3x3 grid/matrix (9 cells)

Step-2: Define players: AI (e.g., 'X' and opponent (e.g., 'O').

Step-3: Implement the minimax function:

- check if game is over (win, lose, or draw):
  - Return +1 if AI wins, -1 if opponent wins, 0 for a draw
- If not terminal, for each empty cell:
  - simulate a move for the current player (AI maximizes, opponent minimizes).

- Recursively call minimax for the next state.
- Undo the move after evaluation.

- AI chooses the move with the highest score; opponent chooses the lowest.

Step-4: Optionally, add alpha-beta pruning to optimize by skipping unpromising branches.

Step-5: Initialize the game loop: AI computes the best move using minimax, updates the board, and alternates turns with the opponent.

(a) How many possibilities are there in tic-tac-toe

→ The total numbers of possible tic-tac-toe games (sequences of moves) is approximately 255,168 (considering valid game sequences not just board states). For unique board states (ignoring rotations/symmetries), there are 5,478 possible configurations after accounting for all valid move sequences.

(b) How a problem is solved using means end analysis?

⇒ Means-End analysis (MEA) is a problem solving technique in AI that reduces the difference between the current state and goal state.

Process

- Identify the current state and goal state.
- Determine the main differences between them.
- Select an operator (action) to reduce the largest difference.
- If the operator cannot be applied directly set a subgoal to enable it and recursively apply MEA.
- Repeat until goal is reached.

Example: In a blocks world, the stack block A on B, MEA identifies the difference (A is not on B), applies operators like "move A to B" if clear, or sets subgoals like "clear B" if blocked.

(b) Write down the A* Algorithm in AI.

→ A* is a heuristic-based search Algorithm to find the shortest path in a weighted graph.

Algorithm

- Initialize an open list (priority queue) with the start node, with cost $f(n) = g(n) + h(n)$, where $g(n)$ is the cost from start to node n, and $h(n)$ is a estimated cost to the goal.
- Initialize a closed list (empty).
- While the open list is not empty:
  - Pop the node with the lowest ($f(n)$).
  - If it's the goal, reconstruct and return the path.
  - Add the node to closed list.
  - For each neighbor:
    - If in closed list, skip.
    - Compute g(neighbor) via current node.
    - If neighbor is not in open list or has a lower g(neighbor), update g(neighbor), f(neighbor), and it's parent.
    - Add neighbor to open list.
- If open list empties, no path exists.

(b) write one Application of A* Algorithm.

⇒ A* is used in path finding for video games,

e.g., finding the shortest route for a character to move around obstacles.

2022

(a) what are the usage of Bfs and Dfs in AI explain with examples?

⇒ Bfs (Breadth first Search)

Usage: Explores all node level by level, ensuring the shortest path in an weighted graphs. used in puzzle solving (e.g., sliding tile puzzle) and GPS navigation.

Example: In 8 puzzle, Bfs finds the minimum moves to reach the goal state by exploring all possible board configurations level by level.

Dfs (Depth first Search)

Usage: Explores as far as possible along a branch before backtracking. used in maze-solving and topological sorting

Example: In a maze, Dfs explore one path fully, backtracking when it hits a dead end, until find the exit.

Q) which algorithm is better between BFS and DFS?

→ BFS is better for finding the shortest path in unweighted graph and when the solution is likely near and the start (shallow depth) It guarantees optimality but uses more memory (store all nodes at the current level).

- DFS is better for tasks requiring deep exploration (e.g., puzzles with deep solutions) or when memory is limited, as it only stores only the current path, however, It may get stuck in infinite loops in unbounded graphs and doesn't guarantee the shortest path.

Choice

Depends on the problem use BFS for shortest paths, DFS for memory efficiency or Deep solutions.

Q) what is chronological backtracking?

⇒ Chronological backtracking is a search technique that upon encountering a failure (e.g., constraint violation), backtracks to the most recent decision point and tries an alternative choice. It's used in constraint satisfaction problems (e.g., Sudoku) but can be inefficient as it doesn't prioritize the source of failure.

(b) How AI and game theory are related?

→ AI and game theory, AI uses game theory to model strategic interactions in competitive scenarios (e.g., chess, poker). Game theory provides frameworks like, zero-sum games, where AI algorithms (e.g., Minimax) optimize decisions assuming rational opponents.

(b) Write down the properties of min-max algorithm.

→
- Evaluates game states to find the best move in a two-player zero-sum game.
- Assumes the opponent plays optimally (minimizes the AI's score).
- Uses a recursive tree search, alternating between maximizing (AI's turn) and minimizing (opponent's turn) scores.
- Guaranteed to find the optimal move if the tree is fully explored.
- Time complexity: $O(b^d)$, where b is the branching factor, d is depth.
- Requires a heuristic evaluation function for incomplete searches.

(b) Compare between the Min-max and alpha-beta pruning algorithm.

→ Min-Max:

- Explores the entire game tree, evaluating all possible moves.
- Computationally expensive especially for deep trees.
- Always finds the optimal solution.

Alpha-Beta Pruning:

- Optimizes Minimax by pruning branches that won't affect the final decision
- Uses two bounds: Alpha (best score for maximizer) and beta (best score for minimizer).
- Skips branches where beta ≤ alpha, reducing computation.
- Same optimal result as minimax but faster (effective branching factor reduced).

Key Differences

Alpha-beta is more efficient, exploring fewer nodes, while minimax is exhaustive

2022-23

@ what do you mean by game theory?

⇒ Game theory studies strategic decision-making among rational agents in competitive or cooperative settings. In AI, it models scenarios like games (e.g., chess) or auctions, where agents optimize outcomes based on others' actions.

@ compare between the min-max and alpha-beta pruning algorithm. (2022-b)

@ Prove the following expressions by means of resolution

i) man (Marcus)

ii) Pompeian (Marcus)

iii) ¬Pompeian (Marcus) (x1) ∨ Roman(x1)

iv) Ruler (caesar)

v) ¬Roman (x2) ∨ loyality (x2, caesar) ∨ hate (x2, caeser)

vi) ~~loyalito~~ loyalto (x3, f1(x3))

vii) ¬man (x4) ∨ ¬(Ruler(y1) ∨ ¬try assassinate(x4, y1) ∨ loyalto (x4, y1)

viii) try toassasinate (Marcus, caesar)

~~Goal~~

To prove expression via resolution convert them to clauses and resolve until deriving the empty clause or the desired conclusion.

Goal: Prove consistency or derive a conclusion (e.g., hate(Marcus, Caesar)).

Step-1: Convert to clauses (already in CNF).

Step-2: Substitute constants:

> From ii and iii: ¬ Pompeian (Marcus) ∨ Roman (Marcus), with Pompeian (Marcus)

¬ Roman (Marcus).

> From v: ¬ Roman (Marcus) ∨ loyalto (Marcus, Caesar) ∨ hate (Marcus, Caesar).

> From vii: ¬ man (Marcus) ∨ ¬ Ruler (Caesar) ∨ ¬ tryassassinate (Marcus, Caesar) ∨ loyalto (Marcus, Caesar).

Step-3: Resolve:

> From i, ii, iii: Marcus is Roman

> From v and Roman (Marcus):

loyalto (Marcus, Caesar) ∨ hate (Marcus, Caesar).

> From vii, i, iv, viii: loyalto (Marcus, Caesar).

: Resolve loyalto(Marcus, Caesar) with ¬V's disjunction: hate(Marcus, Caesar).

Conclusion: hate(Marcus, Caesar) is derived assuming no contradictory loyalto function in vi.

(b) Define production system in AI.

A production system in an AI framework consisting of:

- A set of rules (condition-action pairs).
- A working memory (current state).
- A control mechanism to select and apply rules.

→ Used in expert systems and problem-solving (e.g., theorem proving).

(b) Explain the water jug problem in AI.

→ Given a 4-gallon jug (A) and a 3-gallon jug (B), measure exactly 2 gallons in Jug (A).

State: (a, b), where a = gallons in A,
b = gallons in B.

Initial state: (0, 0)
Goal state: (2, _)

Operators:

- Fill A: (a,b) → (4,b)
- Fill B: (a,b) → (a,3)
- Empty A: (a,b) → (0,b)
- Empty B: (a,b) → (a,0).
- Pour A to B: (a-b) → (a - min(a,3-b), b + min(a, 3-b)).
- Pour B to A: (a,b) → (a + min(b,4-a), b - min(b, 4-a)).

Solution Path

- (0,0) → Fill A → (4,0)
- (4,0) → Pour A to B → (1,3)
- (1,3) → Empty B → (1,0)
- (1,0) → Pour A to B → (0,1)
- (0,1) → Fill A → (4,1)
- (4,1) → Pour A to B → (2,3)

Result: (2,3) achieves 2 gallons in A.

(b) write down the important requirements for control strategies.

- Completeness: Ensure a solution is found if one exists.
- Efficiency: Minimize computational resources (time and memory).
- Optimality: Find the best solution (e.g., shortest path) when multiple solutions exists.

2023-24

(a) what is Turing test? Write down the algorithm to build tic-tac-toe in AI? How many possibilities are there in the tic-tac-toe? (2021 a)

(b) How AI and game theory are related? *(2022-b)

(b) write down the properties of min-max algorithm. (2022-b)

(c) compare between the min-max and alpha-beta priming algorithm. (2022-b)

Internal Suggestions

① Goals of AI?

→ The primary goal of AI is to perform tasks requiring human Intelligence, such as, Problem solving, reasoning, learning, perception, and decision-making.

② What are Agents in AI?

→ An AI agent is an entity that perceives it's environment through sensors, processes information, and takes action to achieve specific goals.
Example: A self driving car, a chatbot, or a robotic vacuum cleaner.

③ Types of Agents:

→ Simple Reflex Agent: Acts based on current percepts using condition-action rules (e.g., thermostat).

Model-Based Reflex Agent: maintains an internal model of the world to handle partially observable environments.

Goal-Based Agent: makes decisions based on goals and future consequences.

Utility-Based Agent: chooses actions to maximize a utility function (e.g., happiness or efficiency).

Learning Agent: Improves performance over time by learning from experience.

(4) Turing test.

→ Turing test was proposed by Alan Turing to evaluate a machine's ability to exhibit intelligent behavior indistinguishable from a human.

- A human evaluator interacts with a machine and a human via text. If the evaluator cannot reliably distinguish the machine from the human, the machine passes the test.

(5) Difference Between informed and uninformed search:

→ Uninformed search: No additional information about the problem beyond the state space and goal (eg., Breath-first search, Depth first search). Relies on exploring all possibilities systematically.

Informed search: uses heuristic knowledge about the problem to guide the search toward the goal more efficiently (e.g., A* search, Greedy, Best first search).

6) Difference Between BFS and DFS.

→ BFS: Explores all nodes at the current depth level before moving to the next level. Uses a queue.

DFS: Explores as far as possible along a branch before backtracking. (Uses a stack or recursion).

- BFS is level by level: DFS is branch by branch.

7) Advantages and disadvantages of BFS and DFS.

→ BFS:

- Advantages: Guarantees the shortest path in unweighted graphs, complete if the state space is finite.
- Disadvantages: High memory usages (store all nodes at the current level) slow for deep graphs.

DFS:

Advantages: Low memory usage (only store the current path), effective for deep solutions.

Disadvantages: May get stuck in the infinite loops, does not guarantee the shortest path.

(8) Heuristic search:

A search strategy that uses a heuristic function to estimate the cost or distance to the goal, guiding the search toward more promising paths.

Example: A* search, Greedy, Best-first search, It's faster than uninformed search but relies on the quality of the heuristic.

(9) Limitation of Hill climbing:

- Local Maxima: Gets stuck at a local peak instead of the global optimum.
- Plateaus: struggles on flat regions where no progress is made.
- Ridges: May oscillate between points without reaching the goal.
- No Backtracking: So it may miss better solutions.

(10) water jug problem. (2022-23 b)

11) Control Strategy Rules

- Rules that govern how an AI system selects and applies actions or rules to solve a problem.

Types:

- Forward chaining: starts with known facts and applies rules to reach the goal (data-driven).
- Backward chaining: starts with the goal and works backward to find supporting facts (goal-driven).

Characteristics: must be complete, consistent and efficient to ensure the system progresses toward the solution.

Example: In a production system, control strategies prioritize which rule to apply when multiple rules are applicable.