Name : Shubham B. Jha
Class : D15C
Roll No : 19

AIDS - Assignment - 1

1. What is AI? Considering the Covid-19 pandemic situation, how AI helped to survive & renovated our way of life with different applications?

=> Artificial Intelligence (AI) is the field of computer science that enables machines to ~~stimulate~~ simulate human intelligence, including learning, reasoning, and decision-making. AI encompasses technologies such as ML, NLP, CV to automate tasks.

AI's Role in COVID-19

1) Early Detection & Diagnosis - AI models (e.g BlueDot) detected outbreaks early; AI-assisted CT scans helped in quick diagnosis

2) Drug & Vaccine ~~Dete~~ Development - AI (e.g AlphaFold) accelerated drug discovery and vaccine research

3) Contact Tracing & Safety - AI powered apps (e.g Aarogya Setu) tracked virus spread; thermal cameras detected fever

4) Healthcare & Robotics - AI chatbots assisted with self-diagnosis; robots disinfected hospitals & delivered medicines.

5) Remote Work & Education - AI improved video conferencing, virtual assistants and adaptive learning platforms.

2.) What are AI agents terminology, explain with examples.

=> An AI agent perceives ~~its environment~~ its environment through sensors and acts using actuators to achieve goals.

Key terminologies include:

1. Agent - An entity that perceives and acts.
2. Percept - Input received by the agent (e.g camera detecting pedestrians)

3. Percept Sequence - The history of all past percepts
4.) Actuators - Components that allow the agent to act (e.g Car's steering & brakes).
5.) Sensors - Devices collecting data from the environment (eg, LiDAR in autonomous vehicles)
6.) Environment - The surroundings where the agent operates (e.g roads and traffic for a self-driving car).

Example : Self Driving Car
Sensors : Camera, LiDAR, GPS.
Percepts : Detects speed limit, traffic & obstacles.
Actuators : Accelerates brakes, steers
Environment : Roads, weather, pedestrians

3) How AI technique is used to solve 8 puzzle problem.
=> The 8-puzzle consists of a 3x3 grid with numbered tiles and one empty space. The goal is to reach a target arrangement by sliding tiles.
Various Informed and Uninformed search techniques are used e.g BFS, DFS, IDDFS, A*.
- For using uninformed searches, heuristics such as Misplaced tiles or Manhattan distance can be used.

Consider the following states

1/2/3
5/8/

initial
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 6 & 8 \\ 7 & 5 & \_ \end{bmatrix}$$

the goal state is
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & \_ \end{bmatrix}$$

The legal moves AI can make shif blank space to left, right, up and down.

Now using the manhattan distance heuristic the algo will exchange it so the blank space with 8
State change function

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 6 & - \\ 7 & 5 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & - & 6 \\ 7 & 5 & 8 \end{bmatrix} \rightarrow \begin{bmatrix} - & 2 & 3 \\ 4 & 5 & 6 \\ 7 & - & 8 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & - \end{bmatrix}$$ → End Goal State reached

Path cost could be 1 point per move.

Q.4) what is PEAS descriptor?
Give PEAS descriptor for following: i) Taxi Driver (ii) Medical diagnosis system (iii) A music composer, (iv) An aircraft autolander (v) An essay evaluator (vi) A robotic sentry gun for Keck lab

⇒ PEAS Descriptor: PEAS (Performance measure, Environment, Actuators, Sensors) is a framework used to define the components of an intelligent agent. It helps describe how an AI system interacts with its environment and performs tasks

(i) Taxi Driver Agent

P : safety, fuel efficiency, speed, comfort, traffic rules.

E : Roads, traffic, pedestrians, passengers, weather.

A : steering, accelerometer, brakes, indicators

S : GPS, cameras, speed sensors, proximity sensors.

(ii) Medical Diagnosis system

P: Accuracy, effectiveness

E: patient health records, medical symptoms, hospital database

A: Displaying diagnosis, suggesting treatments, prescribing medication

S: Patient input, lab, test results.

3.) AI Music Composer

P: Creativity, harmony, originality, satisfaction

E: Musical notes, user preferences, genre styles

A: Generating melodies, modifying pitch / tempo

S: Music database, user feedback, real time input (mood, style

4.) Autolander

P: Safe landing, smooth touchdown, weather adaption

E: Runway, weather conditions, wind speed, altitude.

A: Controlling thrust, flaps, landing gear, breaking system

S: Radar, altimeter, GPS, wind sensors, speedometer.

5. Essay Evaluator

P: Accuracy of grading, fairness, grammar / spelling correctness

E: Essays, grammatic rules, plagiarism database, rubrics

A: Assigning grades, providing feedback, suggesting improvements

S: Text input, word count, syntax & grammar checkers

6. Robotic Sentry Gun for Keck Lab

P: Intruder detection accuracy, response time, minimal false alarms.

E: Laboratory area, authorized personnel, intruders.

A: Rotating turret, firing warning shots, sounding alarm

S: Motion sensors, thermal cameras.

3

5) Categorize a shopping bot for an offline bookstore according to each of the six dimensions (fully / partially Observable, deterministic/stochastic, episodic / sequential, . .)

=>

Dimension	Category
Observability	Partially Observable
Deterministic/Stochastic	Stochastic
Episodic/Sequential	Sequential
Static/Dynamic	Dynamic
Discrete/Continuous	Discrete
Single/Multi-Agent	Multi-Agent

6) Differentiate Model based & Utility based Agent.

=>

Model based	Utility based
Uses an internal model of the world.	Chooses actions based on a utility function that maximizes performance
Uses stored knowledge (model) to simulate future states	Compares different actions and selects the one with the highest utility value.

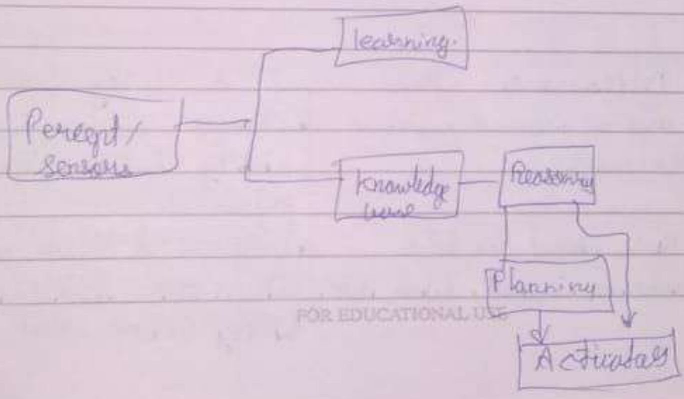
• Works towards achieving predefined goal	• Not just goal-driven but also optimizes for the best possible outcome
• Limited handling of uncertainity	Handles uncertainty by assigning utilities

7.) Explain the architecture of a knowledge based agent and Learning Agent

=) A Knowledge-Based Agent (KBA) makes decisions using a Knowledge Base (KB) and reasoning mechanisms.

Components:

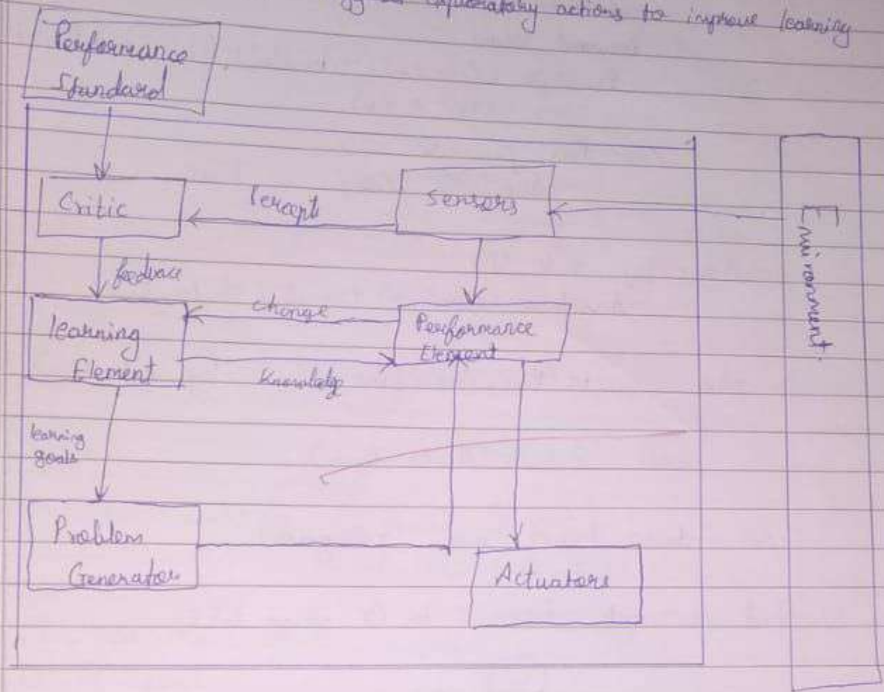
1.) Knowledge Base (KB) - Stores facts, rules and logical statements (e.g "if traffic is heavy, take an alternate route").
2. Inference Engine - Applies logical reasoning (e.g. modus ponens) to derive conclusions from the KB.
3. Perception (Sensors) - Collects input from the environment.
4. Action Execution (Actuators) - Performs actions based on decisions.
5. Updating Mechanism - Modifies the KB as new information is learned.



Learning Based Agent Architecture

Components

1. Learning Element - Updates the agents' knowledge based on feedback
2. Performance Element - Makes decisions and executes actions.
3. Critic - Evaluates the agents actions and provides feedback
4. Problem Generator - Suggests exploratory actions to improve learning



Q) Convert the following to predicates
a. Anita travels by Car if available otherwise travels by bus.
b. Bus goes via Andheri and Goregaon
c. Car has puncture so is not available.
Will Anita travel via Goregaon? Use forward reasoning.

⇒ (a) Available (car) → travels (Anita, Car) -- (i)
¬Available (Car) → travels (Anita, Bus) -- (ii)

(b) visits(Bus, Goregaon) ∧ visits(Bus, Andheri) - (iii)

(c) Puncture(Car) → (¬Available (Car)) - (iv)
Puncture (Car) - (v)
Forward reas.

⇒ Puncture ((Car) → (¬Available (Car)) ∧ Puncture (Car)
... (iv) & (v)
Car has puncture so,
¬Available (Car) is True

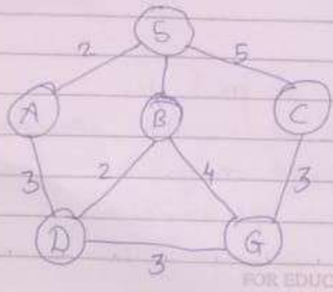
Now by rule (ii)
¬Available (Car) → travels (Anita, Bus)

also visits (Bus, Goregaon) by rule (iii)

and travels (Anita, Bus)

So Anita travel via Goregaon.

10. Find the route from S to G using BFS.



Sundaram

FOR EDUCATIONAL USE

5

Start from Sources
put in queue with cost (initial 0)
remove from queue calculate new distance (curr + edge weight)
~~distance~~ check if the current is the goal if not then
~~repeat~~ put in queue and repeat.

(S,0)

childrens ((A,2),(B,1),(C,5))

None of them is goal ~~push~~ add in queue.

(A,2),(B,1),(C,5)

childrens (D,3) ; ~~distance~~
dist : (D, 2+3)
Not goal ∴ Put in queue.

(B,1),(C,5),(D,5)

childrens [(D,2),(G,4)]
dist [(D,1+2) (G,1+4)]

D not Goal put in queue
G is goal.

∴ Distance found! G is 5

FOR EDUCATIONAL USE

Sundaram

12) Explain Hill Climbing and its drawbacks in detail with example. Also state limitations of steepest-ascent hill climbing.

=> Hill Climbing is a local search algorithm used to find an optimal solution by iteratively making small changes to the current state and choosing the best improvement.

Algorithm Steps:

1. Start with an initial solution (state)
2. Evaluate neighbouring states
3. Move to the neighbor with the highest value
4. Repeat until no better neighbors exists

Example: (Hill Climbing in Path Optimization): A robot trying to reach the highest hilltop (goal) uses hill climbing. It moves upwards step by step, choosing the steepest ascent ~~ue~~ until no higher step is available.

Drawbacks of Hill Climbing:

1. Local Maxima - May get stuck at a peak that is not the global maximum.
2. Plateau - A flat region with no improvement, leading to stagnation.
3. Ridges - A narrow path of improvements that the algorithm may fail to follow.
4. No Backtracking - Once a move is made, previous states are not reconsidered.

13. Explain simulated annealing and write its algorithm

→ A probabilistic search method inspired by the cooling process of metas.

Allows occasional downhill moves to escape local optima.

Algorithm:

i) Start with an initial solution, temperature T

(ii) Generate a neighbouring solution

(iii) If its better, accept it, otherwise accept it with probability $e^{(-\Delta E/T)}$

(iv) Decrease T gradually and repeat until T is very small

Use Case: Travelling salesman Problem (Optimized problems)

14. Explain A* Algorithm with an example

→ A* is a like best first search algorithm, used in pathfinding and graph traversal. It uses the following formulas:

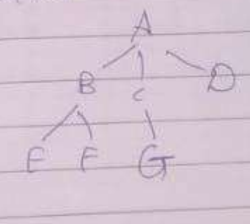
$f(n) = g(n) + h(n)$

$g(n)$ → cost to reach n from start

$h(n)$ → heuristic estimate of cost to reach from goal to n

$f(n)$ → total estimated cost

Eg: Goal G



Node	g(A, n)	h(n, G)
A	0	6
B	1	4
C	2	2
D	4	7
E	3	5
F	5	3
G	6	0

Steps :-

1. Start at root node A.

$f(A) = g(A) + h(A) = 0+6$
$= 6$

2. expand neighbours B, C, D

$f(B) = 1+4 = 5$
$f(C) = 2+2 = 4$
$f(D) = 4+7 = 11$

3. Choose lowest value that is f(C)
4. Expand neighbors of C. (G)

$f(G) = 6 + 0 = 6$

5. Goal reached at G with total cost 6

Advantages ->

efficient for finding shortest path in weighted graphs
balances exploration by considering both g(n) and h(n)

FOR EDUCATIONAL USE

15. Explain min-max algorithm and draw game tree for tic tac toe game.

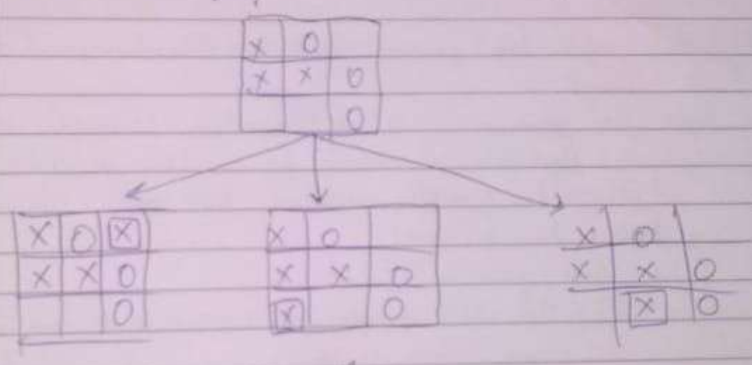
→ The min max algorithm is a decision making algorithm used in 2 player games. It assumes one player (MAX) tries to maximize its score & other player (MIN) tries to minimize the score.

Game tree represents all possible moves

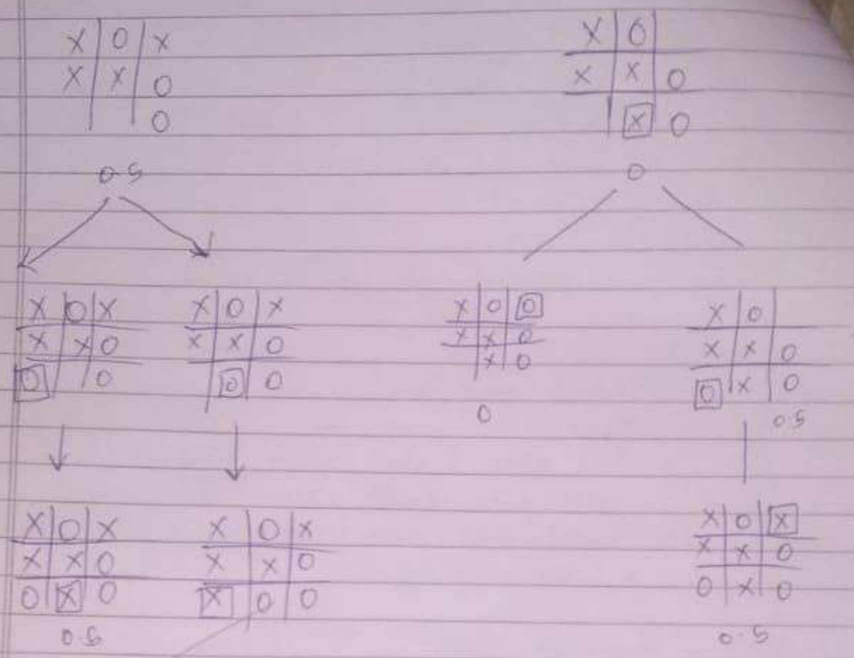
Algorithm:

1. Generate game tree
2. Assign scores
3. Max picks highest value from children
   MIN picks lowest value
4. Repeat until root node is evaluated starting a bottom up approach.

Game tree for the tic-tac-toe game



1



16] Explain alpha beta pruning algorithm for adversarial search with example

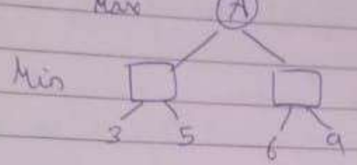
→ Alpha beta pruning is an optimization technique used in minimax algorithm to reduce the number of nodes evaluated in adversarial search problems like game-playing AI (eg chess, tic-tac-toe)

Alpha beta pruning includes

Alpha (α): The best maximum score that the maximizing player can guarantee so far.

Beta (β): The best minimum score that the minimizing player can guarantee so far.
The algorithm prunes branches that will not influence final decision

Example:



1. Start at root node A.
   $\alpha = -\infty$, $\beta = \infty$
2. · check left min node (child of A)
   · check first child value = 3 ⇒ update $\beta = 3$
   · check second child : value = 5 ⇒ β remains 3
   · Min node returns 3 to Max.
3. Right Min node (child of A)
   · check first child : value = 6 → $\beta = 6$.
   · Here $\alpha = 3$ at MAX node but $\beta(6) > \alpha(3)$ so no pruning.
   · Explore 2nd child (9) → Here pruning will occur.
   · MIN node already has a value ≤ 6 it will never choose 9 & so we prune the node with value 9.
4. Max value = 6

17) Explain WUMPUS world environment, giving its PEAS description. Explain how percept sequence is generated.

→ The Wumpus world environment is a simple grid-based environment, used in AI to study intelligent agent behavior.

In uncertain environments. It is a turn based environment where an agent must navigate a cave to find gold while avoiding hazards like pits and a monster called wumpus.

PEAS:

P: the agent is rewarded for grabbing gold and exiting safely. Penalty is imposed for falling into pits and getting eaten by wumpus.
E: 4x4 grid world containing the agent, wumpus, pits, gold.
A: The agent can move forward, left, Right, shoot, climb.
S: Agent perceives stench (near wumpus), breeze (near a pit), glitter (near gold), bump and scream.

Percept Sequence generation:
It is the history of all perceptions received by the agent. At each time step, the agent At each time step the agent perceives information based on its current location and surroundings.

Example percept sequence:

1. Agent starts at (1, 1):
   - No breeze, no stench, no glitter → safe square
2. Agent moves to (2, 1)
   Breeze detected → A pit is nearby but not in current square)
3. Agent moves to (1, 2):
   Stench detected → wumpus is in adjacent cell
4. Agent moves to (2;2):
   Glitter detected → gold is here.
5. Agent moves back to (1, 1) and climbs out.

18) Solve the crypt Arithmetic SEND + MORE = MONEY

⇒ Step 1 : M must be 1 ∵ sum of 2 '4' digit number cannot be greater than equal to 20000

```
  S E N D
+ 1 O R E
---------
1 O N E Y
```

Step 2 : Assume E + O = 10 + N
then E + O generates carry & ∴ S has to be 8
otherwise if S is 9 & carry is 1 and 9+1+1 = 11 but
O cannot be 1 ∵ M = 1.

S = 9, M = 1, O = 0

```
 S E N D          8 E N D
 M O R E    ⇒     1 O R E
M O N E Y       ---------
                1 O N E Y
```

Step 3 but then E + O cannot generate carry, so initial assumption of carry is wrong and so S is 9.

```
  9 E N D
+ 1 O R E
---------
1 O N E Y
```

Step 4 Now E + O = N ~~so but~~
⇒ E = N but that is a contradiction
So N+R must generate a carry! & E+1+O = N
∴ N = E+1
Also N+R + K1 = 10 + E
R + K1 = 10 - 1 = 9

K cannot be 0 otherwise R = 9 contrad

$\therefore$ R+1 = 9

$\therefore$ R = 8

$$\begin{array}{cccc} & S & E & N & D \\ & M & O & R & E \\ \hline M & O & N & E & Y \end{array} \Rightarrow \begin{array}{ccccc} & 9 & E & N & D \\ & 1 & 0 & 8 & E \\ \hline 1 & 0 & N & E & Y \end{array}$$

$\therefore$ D+E = 10+Y

also N+8+1 = 10+E

but N = E+1, lets assume E=5, then N=6

$$\begin{array}{ccccc} & 9 & 5 & 6 & D \\ & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & Y \end{array}$$

So only requirement is D should be a number such that when added to 5 generate a query

available space { ~~0~~, ~~1~~, 2, 3, 4, ~~5~~, ~~6~~, 7, ~~8~~, ~~9~~ }

2,3,4 cannot generate carry $\therefore$ D = 7

$\therefore$

$$\begin{array}{ccccc} & 9 & 5 & 6 & 7 \\ & 1 & 0 & 8 & 5 \\ \hline 1 & 0 & 6 & 5 & 2 \end{array}$$

is the solution!

19.) Consider the following axioms: All people who are graduating are happy, All happy people are smiling, Someone is graduating

→ (1) Represent in predicate logic

$G(x)$: x is graduating
$H(x)$: x is happy
$S(x)$: x is smiling

Translating axioms in predicate logic

1. All people who are graduating are Happy
$\forall x\, (G(x) \to H(x))$

2. All happy people are smiling
$\forall x\, (H(x) \to S(x))$

3. Someone is graduating
$\exists x\, G(x)$

2. Convert each formula to clause form

1. $\forall x\, (G(x) \to H(x))$
Using implication removal
$\forall x\, (\neg G(x) \lor H(x))$
In clause form
$\{\neg G(x), H(x)\}$

2. $\forall x\, (H(x) \to S(x))$
Using implication removal
$\forall x\, (\neg H(x) \lor S(x))$
In clause form
$\{\neg H(x), S(x)\}$

3. $\exists x\, G(x)$
In clause form : $\{G(v)\}$

③ Prove "is someone smiling" using resolution

1. Collect clauses

(1) $\{\neg G(x), H(x)\}$

(2) $\{\neg H(x), S(x)\}$

(3) $\{G(a)\}$

(4) Assume contradiction $(\neg S(a))$

2. Apply resolution

Resolve (1) $\{\neg G(H), H(x)\}$ with (3)

$\{G(x)\}$

Substitute $x = a \Rightarrow \{G(a)\}$

$\{\neg G(a), H(a)\}$

∵ we have $G(a)$, resolving gives

$\{H(a)\}$

Resolve (2) $\{\neg H(x), S(x)\}$ with $\{H(a)\}$

Substitute $x = a$

$\{\neg H(a), S(a)\}$ & $\{H(a)\}$

Now left with $\{S(a)\}$

we can resolve it with $\{\neg S(a)\}$

∴ We are left with

$\{\phi\}$

∴ Our assumption of contradiction is wrong.

Explain modus ponen with suitable example

Modus ponen is a fundamental rule of inference in propositional logic that allows us to deduce a conclusion from a conditional statement and its antecedent.

It follows the form

1. $P \Rightarrow Q$ (if P then Q)
2. P (P is true)

$\therefore$ Q (Q must be true)

Example:

1. If it rains, the ground will be wet : $P \rightarrow Q$
2. It is raining : P

$\therefore$ Ground is wet $\rightarrow Q$.

2. Explain forward ~~chaining~~ ~~and~~ chaining & backward chaining algorithm with the help of example.

→ forward chaining :- It starts with given facts and applies inference rules to derive new facts until the goal is reached. It is a data driven approach because it begins with known data and works forward to reach a conclusion.

Example : Diagonosing a disease

Rules :

1. If a person has a fever and cough they might have flu
2. If a person has a sore throat and fever, they might have cold

Facts :

- The patient has a fever. • The patient has cough.

- Inference : i) Fever + cough → flu (rule 1 applies)

ii) Conclusion The patient might have flue.

Backward chaining : It starts with goal and works backwards by checking what facts are needed to support it. It is a goal driven approach.

Example : Diagosing a disease

Goal : Determine if patient has flue

Rules

1. (Fever ∧ cough) → flu
2. (Sore Throat ∧ fever) → cold

Process using backward chaining

1. We want to prove flu.
2. looking at rule 1 (Fever ∧ cough) → Flue

we need to check if patient has fever and cough.

3. we check our known facts
   - Patient has fever.
   - Patient has cough.
4. Since both conditions are met, we confirm flue is true.

Sundaram

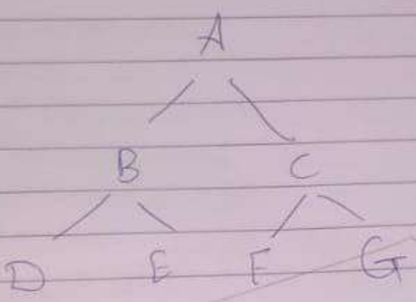
FOR EDUCATIONAL USE

(11) What do you mean by depth limited search? Explain Iterative Deepening search with example

→ Depth limited search (DLS) is an uninformed search algorithm that modifies DFS by introducing a depth limit L preventing exploration beyond the defined level. This prevents infinite loops in graphs but risks missing goals beyond L.

Iterative Deepening Search (IDS) combines DLS with BFS by iterating for after incrementally increasing the depth limit.

Example



Goal G

Initially the depth limit is 0 for iteration 1.

Nodes Visited = A

Goal not Found

Iteration 2, Limit = 1

Nodes visited = A -> B -> C

Goal not found

Iteration 3, limit 2

Nodes visited = A -> B -> D -> E -> C -> F -> G

Goal G is found.