DISC-45

05/05

Page No.
Date

Assignment-1

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

Ans. Artificial Intelligence (AI) refers to simulation of human intelligence in machines that are peogrammed to think & learn like humans.

AI has helped during the COVID-19 pandemic in various ways:

1) Healthcare & Diagnosis:
AI peedicted vieus speed, assisted in diagnostics using X-ray, CT scans & sped up deug/ vaccine discovery.

2) Data & Peediction:
AI tracked vieus speed helping with resource allocation & govetnment planning.

3) Public health safety:
Contact tracing apps, powered by AI, helped track exposure & reduce vieus transmission.

4) Public awareness & mental health:
AI analyzed social media to gauge public sentiment, guiding mental health support & awareness campaigns.

5) Education & Remote Learning:
AI powered platforms allowed personalized online learning & remote education during schol closures

Date

2. What are AI agents terminologies, explain with example.

AI Agent terminologies:

1) Agent: An entity that percieves its environments & take actions to achieve goals.
   Eg. A chatbot answering user queries.

2) Environment: Everything the user interacts with.
   E.g. Roads, vehicles & traffic signals for a self-driving car.

3) Perception: The process of collecting data from the environment through sensors.
   E.g. Cameras in self driving cars, detecting obstacl

4) Actuators: Mechanisms that allow the agent to act on the enviroment. for more movement
   Eg. Robots arms/ wheels

5) Actions: The behaviour that agent performs based on i perception.
   Eg. Virtual assistant responding to a query.

3. How AI techniques is used to solve 8 puzzle problem?

Soln: Initial State:
1	2	3
4	0	6
7	5	8

Goal State:
1	2	3
4	5	6
7	8	0

Misplaced tiles : $h(n)=2$

Steps to solve 8-puzzle problem by A*:

1) Initialize a priority queue.
2) Insert the initial state with $F(n) = g(n) + h(n)$
3) While queue not empty :
   Remove state with lowest $F(n)$
   If state = goal, return solution.
   Generate valid moves (up, down, left, right)
   compute $g(n)$ & $h(n)$ for new states.
   Insert new states into queue.
4. Repeat until goal is reached.

Step 1 :
1	2	3
4	0	6
7	5	8

($h=2$)

Step 2 :
1	2	3
4	5	6
7	0	8

($h=1$)

Step 3 : Goal
1	2	3
4	5	6
7	8	0

($h=0$)

4. What is PEAS descriptor ? Give PEAS descriptor for following :

Soln: PEAS stands for Performance measure, environment, actuators & sensors.

P → Criteria to evaluate the agent's success.

E → (Environment) → surrounding where agent operates

A → (Actuators) → common components that allow agent to take actions

S → (Sensors) components that help agent to perceive environment.

1. Taxi Deiver:
P - Reaching destination, fuel efficiency.
E - Roads, traffic, pedestrians.
A - Stearing wheel, accelerator, brakers.
S - Cameras, GPS, Speedometer.

2. Medical Diagnosis System:
P - Accuracy of Diagnosis, patient satisfaction.
E - medical records, test results, symptoms.
A - Peescription generation, Report generation.
S - patient data input, lab test results.

3. Music composer:
P - Quality of composition, user satisfication.
E - Musical genres, musical theory constraints.
A - Music synthesis, instruments, generating musical notes
S - composition structure, user feedback.

4. Aiecraft Autolander:
P - Smooth & safe landing
E - Weather, runway condition, aie traffic.
A - Flaps, landing gear, throttle.
S - Altitude sensor, GPS, wind direction sensor.

5. Essay Evaluate:
P - Accurate grading, feedback quality.
E - Grammer rules, submitted essays.
A - displaying grades & feedback
S - Text input, spelling & grammer checkers.

6. Robotic sentry gun for tech lab:
P- Correctly identifying threats.
E - Intruders, authorized personnel.
A - Camera movement, alarm system, firing mechanism
S - motion sensors, facial recognition.

5. Categorize a shopping bot for an offline bookstore according to each of the six dimensions.
1. Partially observable
— bot may not have full visibility of store's inventory
2. Stochastic
— outcomes uncertain due to customer behaviour, stock availability
3. Sequential
— Bot will suggest books based on previous customer queries
4. Dynamic
— Bookstore environment changes like books, getting sold
5. Discrete
— Bots process finite no. of actions.
6. Multi agent
— Bot interacts with customers & store employees each have their own goals.

6. Differentiate between model-based & utility-based agents.

Model-based	Utility-based
1. Uses internal model of environment to make decisions	Chooses actions based on utility fn's that measures performances.

2) Decisions based on past & present percepts.	Selects action based on maximizing utility.
3) Can be goal-based but doesn't necessarily optimize for best performance.	optimizes for highest possible utility ensuring performance.
4) Ex. robot vaccum using a map to navigate.	Ex. Self driving car, choosing safest & fastest route.

7. Explain the architecture of knowledge-based agent & learning agent.

Sol^n:

Knowledge-based agent:

A knowledge-based agent (KBA) uses stored knowledge to make decisions. It consists of

1) knowledge Base → stores facts, rules & logic.
2) Inteference Enginge → Uses reasoning to derive conclusions
3) Perception (Sensors) → Gathers new information from enviroment.
4) Action mechanism → Performs appropriate actions based on reasoning.

Eg. AI in medical diagnosis uses past cases to diagnose diseases

Learning-agent: A learning agent improves its performance over time. It consists of:

- Learning Element → Updates knowledge based on experience
- Performance Element → Decides actions based on current knowledge.
- Critic → Provides feedback by evaluating actions
- Problem generator → suggests new actions to improve learning

Eg.: A self-driving car learns to from traffic patterns & adjusts driving behaviour.

88 Convert the Following to predicates.

a) Anita travels by car if available otherwise travels by bus.

Car Available → Travels By Car (Anita)

→ Car Available → Travels By Bus (Anita)

b) Bus goes via Andheri & Goregaon.

Goes via (Bus, Andheri) ∧ goes via (Bus, Goregaon)

c) Car has puncture so is not available.

Puncture (car)

puncture (car) → ¬ Car available.

Will Anita travel via goregaon ? Use forward reasoning

From (c) :

Puncture (car) is true

From (a)

¬ Car available, we use ¬ Car available → Travels by bus (Anita)

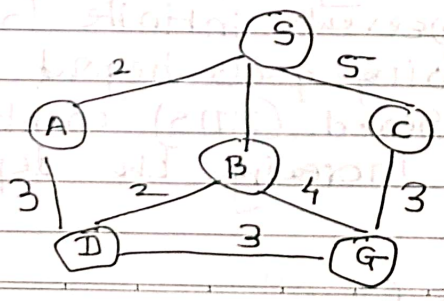
From (b)

goes via (Bus, goregaon)

Since Anita travels by bus, she follows this route

Thus, Anita will travel via Goregaon.

89 Find route from S to G wing BFS.



Teacher's Sign.: _____

To Find route From S to G using BFS, we systematically explore all nodes level by level sta from source node (S) until we reach destination nod

1) start at S
   queue = [S]
2) From S, we go to it neighbours: A, B, C.
   queue = [A, B, C]
3) Dequeue A & explore its neighbourss:
   queue = [B, C, D]
4) Dequeue B & explore it neighbues.
   queue = [C, D, G]
5) Dequeue C & queue neighbours
   queue = [D, G]
6) Dequeue D
   queque = [G]
7) Dequeue G
   As G is our destination, BFS stops here.

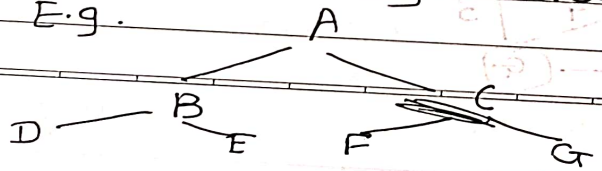
Bouk from S to G : S → B → G

10) What do you mean by depth limited seaech ? Explain iterative deepening seatch with example.

Sol^n : Depth Limited Seatch (DLS) is an uninformed search algorithm that modifies DFS by introducing a depth limit L, preventing exploration beyond the predefined level. This prevents infinite loops in infinite graphs but risks missing goals beyond L.

Iterative Deepening Seatch (IDS) combines DLS with BFS by incrementally increasing the depth limit.

E.g.



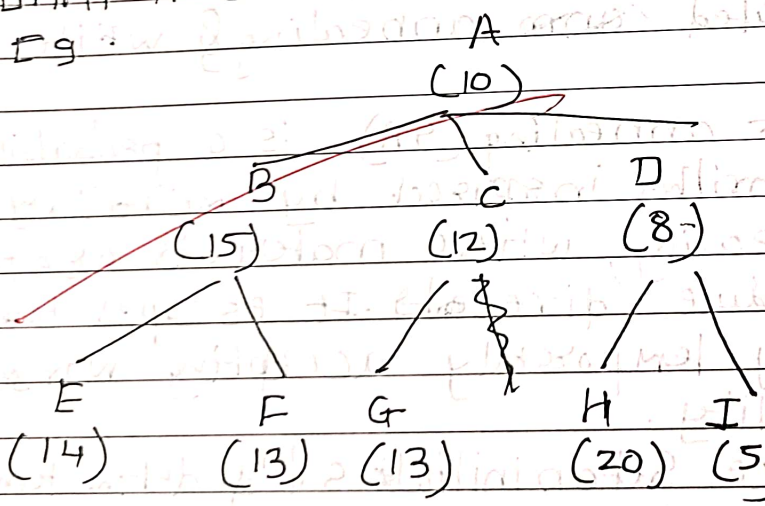
Teacher's Sign.: _____

Iteration 1 : Depth limit = 0
Nodes visited : A
Result : Goal not found.
Iteration 2 : L=1
Nodes visited A → B → C
goal not found.
Iteration 3 : L=2
Nodes visited : A → B → ~~C~~ D → E → C → F → G
goal G found at L=2.

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

Sol^n: Hill climbing is a total search optimization algorithm which moves toward better neighboring solutions until it searches a peak.

Eg:



Steps: Start at root node A(10)
Compare its children B, C & D
move to child with highest value i.e B (15)
Repeat for B's children E & F
Terminate at E (14)

The algo stops at E(14) not reaching the goal (G).

Teacher's Sign.

Drawbacks:

1) Local Maxima: The algorithm greedily selects the best immediate child & can thus get stuck in a local maxima.
2) Plateaus - If siblings have equal values, the algorithm can't decide the next step & gets stuck.
3) Ridges - Narrow uphill paths require backtracking with hill climbing algorithm does not support.

Limitations of steepest- Ascent Hill Climbing:

1) Computationally Expensive: Evaluates all neighbours before selecting the best.
2) Can get stuck - It can still get stuck in local maxima plateaus or ridges.
3) No global optiminality - It only focuses on immediate improvements.

13) Explain simulated ~~annn~~ annealing & write its algorithm.

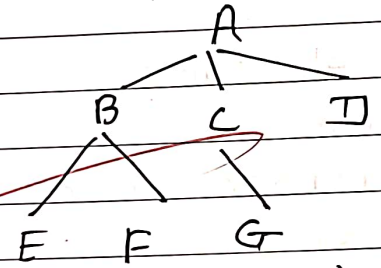
Soln: Simulated ~~e fate~~ annealing (SA) is a probabilistic optimization algorithm inspired by metallurgical process of annealing, where materials are heated & cooled to reduce direcets. It escapes the local optima by temporarily accepting worse soln with a probability.

Algo: 1) Initialize: Set an initial soln & define temp. T.
2) Repeat until stopping condition.
- Generate a new neighbour condition.
- Compute change in cost ($\Delta E = E_{new} - E_{current}$]
- If new soln is better i.e $\Delta E > 0$, accept it.
- If, worse, accept it with probability $P = e^{-\Delta E/T}$

Teacher's Sign.:

• Decrease temp. T (coding schedule)
3. Return best solution
Eg. Travelling salesman Problem:
Swap two cities in a route accept a longer route easly (high T) but reject it later (low T)

Explain A* algorithm with an example.
A* is a best first search algo used in path finding & graph traversal. It uses the following formula
$f(n) = g(n) + h(n)$
$g(n) \rightarrow$ cost to reach node n from start.
$h(n) \rightarrow$ heuristic estimates of cost to reach from goal to n.
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 C ($f(C) = 4$)

4) Expand neighbours of C : G

$f(G) = 2+4+0 = 6$

5) Goal reached at a with total cost 6

advantages : 1) Efficient for finding shortest paths in weighted graphs.

2) balances exploration by considering both $g(n)$ & $h(n)$.

15) Explain min-max algorithm & draw the game tree for Tic Tac Toe game.

Min-max algorithm is a decision rule used for finding the best possible move in two-player turn-based games such as Tic-Tac-Toe, chess & checkers

Algorithm:

1) Define Evaluation Function: Return +1 if X wins, -1 if O wins, 0 for a draw.

2) Check for terminate state:

If the board is a win/loss/draw state, return score.

3) Generate all possible moves.

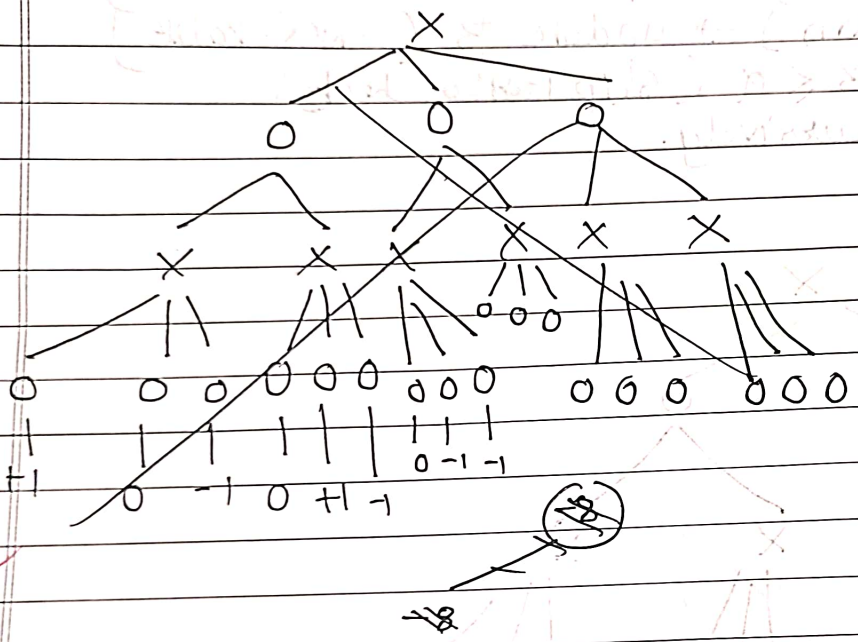
4) Apply Recursion (1) If its max's turn (X): call minmax on possible moves & choose the maximum score.

(2) If it's min's turn (O): Call minmax on possible moves & choose the minimum score.

5) Backpropagates Scores: Return the best score up the game tree.

6) Pick the best move (At root level): max chooses the move with highest score.

Game tree for Tic-Tac-Toe :



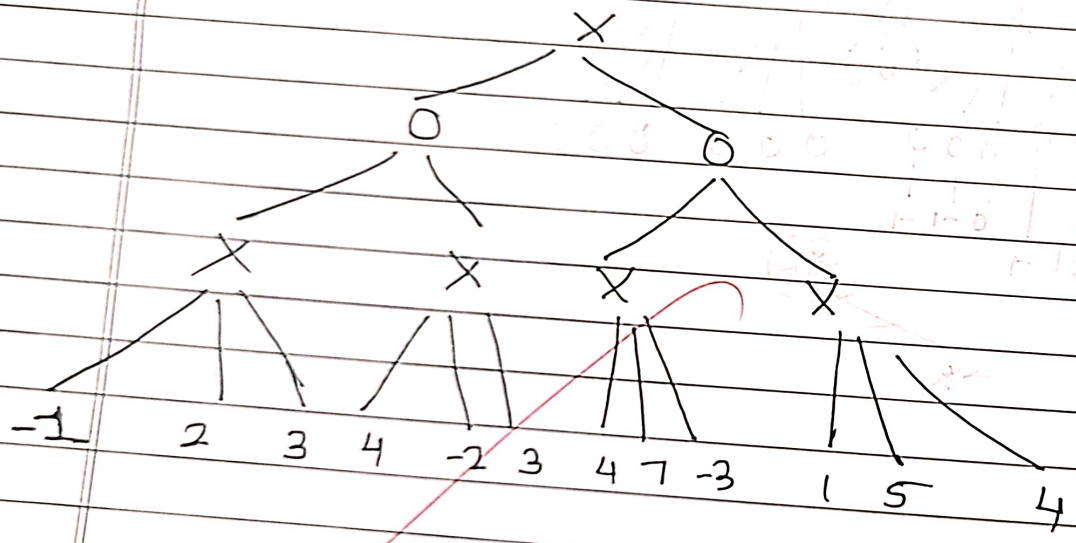
Teacher's Sign.: _____

16) Explain Alpha beta pruning algorithms for adversarial search with ~~explain~~ example.

Alpha-beta pruning is the optimized version of minimax algorith by skipping unnecessary branches reducing computation.

Algorithm: 1) Initialize $\alpha = -\infty$, $\beta = +\infty$.
2) Max (X's turn) → update α (highest value)
3) Min (O's turn) → update β (lowest value)
4) Prune if $\beta \leq \alpha$ (skip further checks)
5) Repeat recursively.

Eg.



* max selects the highest, min selects the lowest.
* If min finds the move worse than existing, we prune the branch.

Advantages: 1) Reduces time complexity from $O(b^d)$ to $O(b^{d/2})$
2) Faster pruning
3) Same optimal move as minmax.

Teacher's Sign.: _____

Page No.
Date

1) Explain WUMPUS World environment giving its PEAS & also the percept sequence generation.

* PEAS:

① P - • Maximize Rewards.
+1000 for collecting gold & exiting grid.
• minimize penalties:
-1000 for falling into a pit or being eaten by WUMPUS.
-1 point for each action taken.
-10 points for using an arrow.

②⑤ E - • Grid Layout (A*)
containing pits, WUMPUS, gold, wall, breeze
• Partially observable - agent can't see entire grid & must rely on sensory input.

③ A - • Move left, right, forward.
Grab - to collect gold, shoot (to eliminate WUMPUS)

④ S - Breeze: indicates pit is adjacent.
stench: indicates WUMPUS in adjacent cell.
glitter: indicates gold
bump: indicates a wall has been encountered.

* Percept Sequence: 1) Initial pos^n: Agent starts at (1,1)
2) Movement: As agent moves from one cell to other, it uses sensors to gather info about surrounding.
3) Creating percept sequence:
Each time the agent moves, it records its percepts as sequence. Eg. After moving to (1,2): [None, Breeze, None]
After moving to (2,1): [Stench, breeze, None]
These indicate no stench/glitter & ~~yes~~ yes to nearby dangers respectively. It continues as agent explores more
4) Decision Making: The agent uses these percepts sequences to make decisions about its next actions based on logical reasoning & inference from observations

18. Solve following cryptarithmatic problem.

SEND + MORE = MONEY

To solve this problem, we need to assign a unique digit (0-9) to each letter, such that the given equation holds true.

1) Set up the equation:

```
  S E N D
  M O R E
M O N E Y
```

2) Since M is the leading digit it must be 1.

$\therefore M = 1$

3) $D + E = Y$ (if carry = 0)

or $D + E = 10$ (if carry = 1)

Tens place: $N + R + \text{carry} = E$

Hundred's place

$E + O + \text{carry} = N$

Thousand's place: $S + 1 = 1 + \text{carry}$

$S + \text{carry} = 0$

$\therefore S = 9$ since there's no carry.

4) Try $E = 5$

If $D = 7$ then

$Y = 7 + 5 = 12$ (invalid)

$O = 2$ then

$Y = 7$

5) $Y = 7$, Assume $N = 8$

$N + R = E$

$8 + R = 5$ (impossible)

$\therefore \boxed{N = 6}$

final: $S = 9$, $E = 5$, $N = 6$, $D = 7$, $O = 0$, $R = 8$, $Y = 2$

```
  SEND          9 5 6 7
+ MORE    ---->  1 0 8 6
-------         -------
 MONEY    ---->  1 0 6 5 2
```

Page No.
Date

1) Explain Modus Ponen with example.

Modus ponen is a fundamental rule of interference in logic. It states that if $P \rightarrow Q$ is true P is true, then Q must also be true formula : $\frac{P \rightarrow Q, \; P}{Q}$

Eg. if it is raining, then it is soggy $(P \rightarrow Q)$
if it raining (P)
∴ it is soggy (Q)

2) Explain forward & backward chaining algo with example.

- Forward chaining : It is data driven, inference algo that starts with known facts & applies inference rules to derive new facts with until the goal is reached.

Fact : A, B
Rule : $A \rightarrow C$, $B \rightarrow D$, $C \wedge D \rightarrow I$
Goal I
Start with A & B
apply $A \rightarrow C$ to derive C
$B \rightarrow D$ to derive D
$C \wedge D \rightarrow I$ to derive I
The goal I is reached.

- Backward chaining : BC is a goal-driven, starts with goal & works backward to find the facts that support it start with goal.

Fact : A, B . Rule : $A \rightarrow C, B \rightarrow D, C \wedge D \rightarrow I$, goal I
Find the rule $C \wedge D \rightarrow I$
if C & D are true: use $A \rightarrow C$ since A is true, C is true
use $B \rightarrow D$ since B is true, D is true
C & D is true, I is true.
∴ conclusion : I is reached.

Page No.
Date

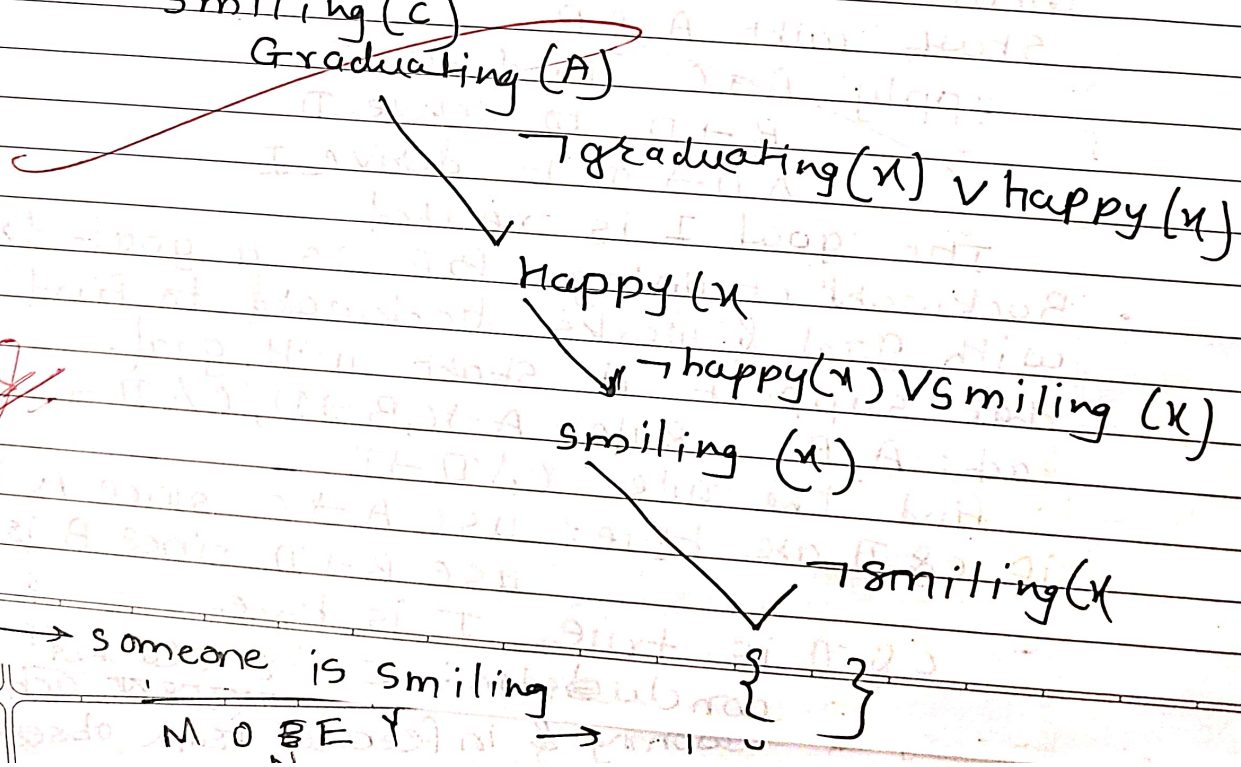
19) Consider the axioms:
All people who are graduating are happy
All happy people are smiling
Someone is graduating.
Explain 1: Represent these axioms in First Predicate logic:

1) $\forall x$ (Graduating $(x) \rightarrow$ Happy (x))
2) $\forall x$ (Happy $(x) \rightarrow$ smiling (x))
3) $\exists x$ (Graduating (x))

$C_1 = \{ \neg$ Graduating (x), Happy $(x)\}$
$C_2 = \{ \neg$ Happy (y), Smiling $(y)\}$
C_3 = Graduating (C)

Resolution between C_3 & C_1
Substitute C for x in L_1
From graduating (C)
Happy (C)

Resolving with C_2
Smiling (C)



→ someone is smiling

MOBEY →