Name: Mohit S. Kerkar Div: D15C Roll No.

AI & DS - I

Assignment - 01

1 What is AI ? Considering the COVID-19 pandemic situation, how AI helped to survive.

Ans AI is Artificial Intelligence. It is a knowledge deriving and storing tool or mechanism used to gain, construct, provide information in ways to ~~build~~ bring about growth in numerous domains, namely, health, education, business, finance, etc. AI is the highest form of machine learning advancement.

COVID-19 brought numerous problems along with its outbreak such as Social Distancing, No proper vaccine facility, lack of providing education, problems related to businesses and closure of factories and industries.
AI helped during these crisis situations in the following ways.

→ Education: Helped in bringing about platforms for E-learning & Smart learning.

→ Finance & Business: Ease of carrying out transactions online

→ Helped in identifying areas where vaccine or hospital facilities were not available, so as to reach to those areas.

2 What are AI Agents terminology, explain with examples.

Ans AI agent is an entity that perceives its environment and takes actions to achieve specific goals.

P.T.O

Types of AI Agents

1. Simple Reflex Agent :-
Acts only based on current percepts. Eg: A thermostat controlling room temperature.

2 Model Based – Maintains internal memory.
Eg. A chess-playing AI remembering past moves.

3. Goal based Agent – Acts to achieve a specific goal.
Eg. A self driving car navigating to a destination

4. Utility Based Agent – Selects the best action based on utility.
Eg. AI recommending investment strategies

5 Learning Agent – Improves performance based on experience
Eg. AI powered personal assistants like Siri and Google Assistant

Example of working of an agent is as follows:
For Vacuum cleaner problem.

- Performance Measure : All rooms are well cleaned.
- Behaviour : Left, Right, Suck and no-op (Doing nothing)
- Percept : Location and status
- Agent function : Mapping (percept sequence, action).

Percept sequence	Action
[A, Clean]	Right
[B, Clean]	Left
[A, Dirty]	Suck

3. How AI technique is used to solve 8 puzzle problem.

Ans AI techniques are used to solve 8 puzzle problem by applying search algorithm & heuristic function.

1. Problem representation : The 8 puzzle is represented as a state space where each state is a 3×3 grid configuration.

Initial state :
1	2	3
4	0	6
7	5	8

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

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 $g(n)$ and $h(n)$ for new states.
   Insert new states into queue.
4. Repeat until goal is reached.

4. What is PEAS descriptor? Give PEAS descriptor for following
Taxi Driver
Medical Diagnosis system
A music composer.
An Aircraft autolander.
An essay evaluator
A robotic sentry gun for the keck Lab.

Ans PEAS stands for Performance, Environment, Actuators, Sensors

P → Criteria to evaluate Agent's success.

E → Surroundings / External area where agent operates.

A → Components that allow agent to take actions.

S → Components that help agent perceive its environment.

PEAS for Taxi Driver

P : Reaching Destination, fuel efficiency.

E : Roads, traffic, pedestrians.

A : Steering wheel, brakes, accelerator.

S : Camera, GPS, Speedometer.

PEAS for Medical Diagnosis System.

P : Accuracy of Diagnosis, speed of Diagonis

E : Medical records, test results, hospitals.

A : Recommeding treatments, sending alerts to patients and doctors, updating medical records.

S : Wearable sensors, medical imaging devices.

PEAS for music composer

P : User satisfaction, quality of composition.

E : Music production studio, streaming platforms.

A : Adjusting pitch and key of compositions, suggesting chord progressions and melodies

S : MIDI inputs, music databases, lyrics or text for ~~not~~ melody generation.

PEAS for Aircraft autolander.

P : Smooth and safer landing.

E : Weather, runway.

A : Gear, throttle, flaps.
S : Altitude sensor, GPS, Wind direction sensor

PEAS for Essay Evaluator.
P : Grading accuracy, feedback quality, accuracy in error checking.
E : Educational institutes, competitive exams.
A : Highlighting grammar and spelling errors, checking for plagarism.
S : Text input, NLP, AI based readability assessment tools.

PEAS for Robotic sentry gun for Keck lab.
P : Correctly identifying threats and target, speed of response
E : Keck Lab, facility, research centers.
A : Tracking, alerting security personnel.
S : Cameras, motion sensors, AI based threat recognition

5. Categorise a shopping bot for an offline bookstore according to each of the six dimensions

Ans
- Partially observable : The bot cannot fully observe customer preference or book placements.
- Stochastic : Customer behaviour and book availability are unpredictable.
- Sequential : Each interaction depends on previous queries & actions
- Dynamic : The backbone environment constantly changes.
- Discrete : The bot operates with a finite set of books, actions and interactions.
- Multiagent : The bot interacts with customers, employees, and inventory systems

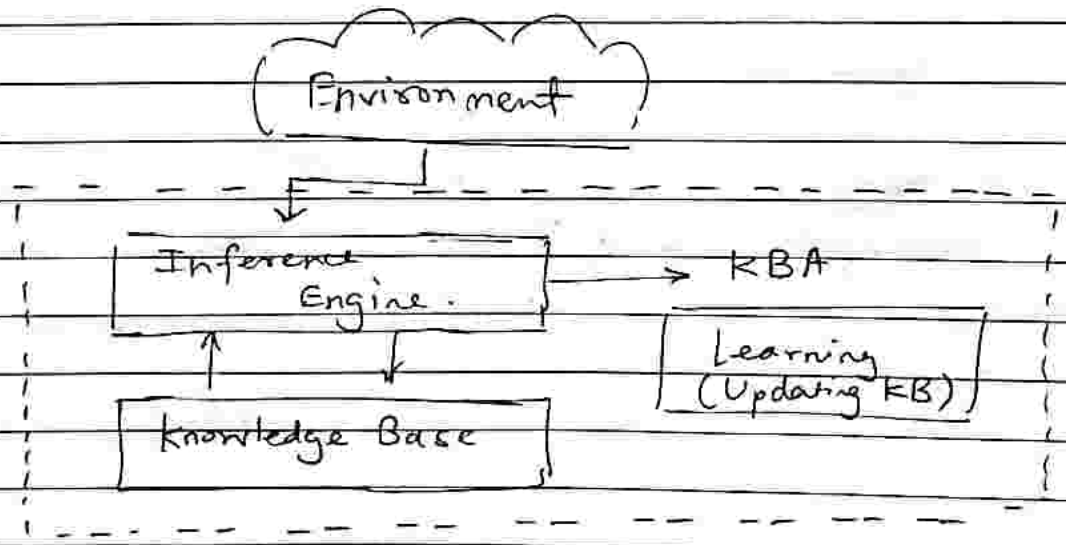
6. Differentiate Model based and Utility based agent.

Ans.

Model based agent	Utility based agent.
1. Uses an internal model of environment to make decisions.	1. Chooses actions based on utility function that measures performance.
2. Decisions are based on past and present percepts.	2. Selects action based on managing utility.
3. Can be goal based but does not necessarily optimize for best outcome.	3. Is goal based and searches for the most optimal solution.
4. Example: Robot vacuum using a map to navigate.	4. Example: A self driven car.

7. Explain the Architecture of a knowledge based agent and learning age

Ans. Architecture of knowledge based agent

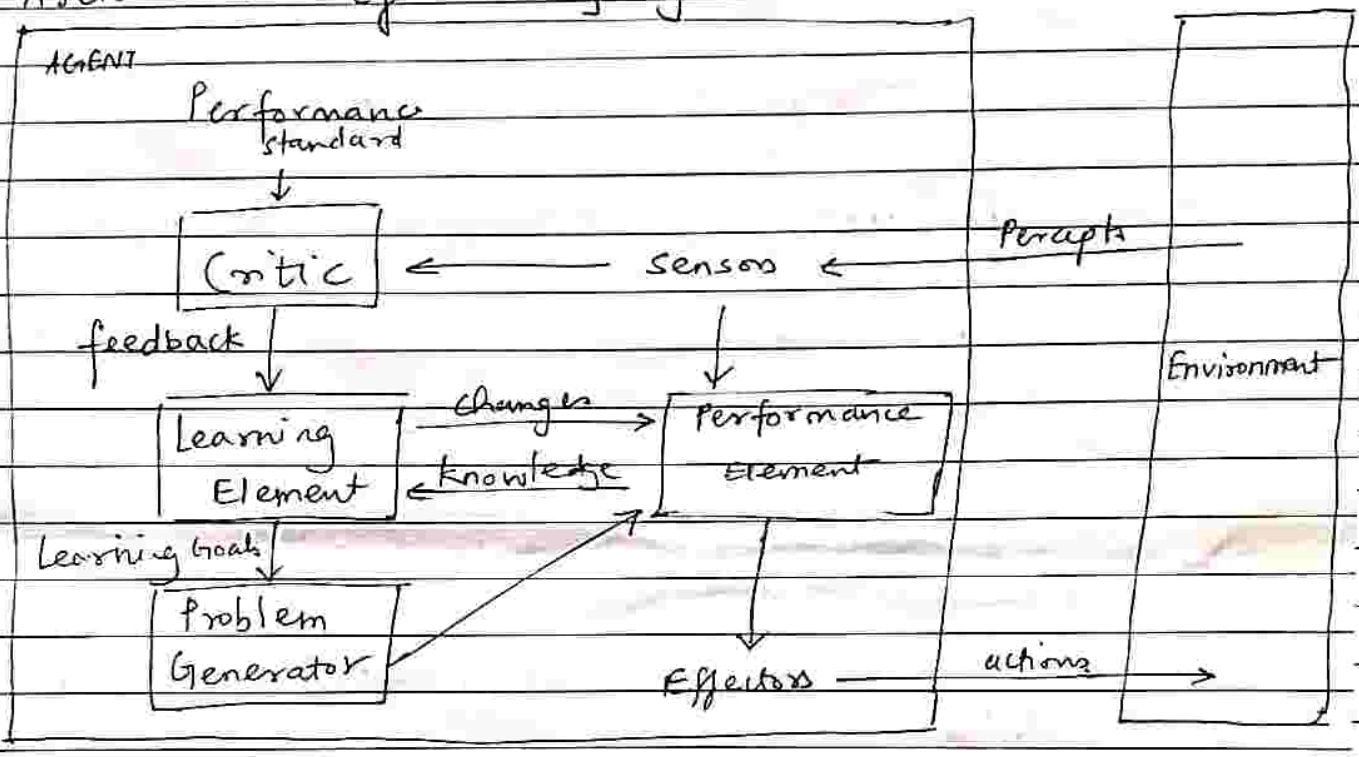


It uses stored knowledge to make decisions and consists of the following:

- Knowledge base: Stores facts, rules and logic

- Inference Engine: Uses reasoning to derive conclusion.
- Perception: Gather new information from environment.
- Action mechanism: Performs appropriate action based on reasoning.

Architecture of Learning Agent



It improves performance over time and consists of the following.

- Learning element: Updates knowledge based on experience
- Performance element: Decides actions based on current knowledge.
- Problem generator: Suggests new actions to improve learning.
- Critic: Provides feedback by evaluating action.

9. Convert the following to predicates

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

P.T.O

CarAvailable → Travels By Car (Anita).

7 CarAvailable → Travels By Bus (Anita).

b. Bus goes via Andheri and Goregaon.

goesVia (Bus, Andheri) ∧ GoesVia (Bus, Goregaon)

c. Car has puncture so is not available

Puncture (car)

Puncture (car) → 7 car available.

Will Anita travel via Goregaon? Use forward reasoning

From (c)

Puncture (car) is true

As Puncture (car) → 7 CarAvailable.

From (a).

7 CarAvailable, we use 7 CarAvailable → Travels By Bus (Anita)

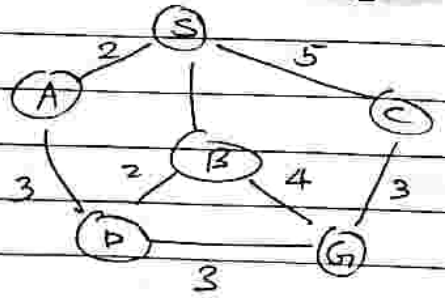
From (b)

Goes via (Bus, Goregaon)

Since Anita travels by bus she will follow this route

Thus, Anita will travel via Goregaon.

10. Find route from S to G using BFS.



1. Start at S
   Queue = [S].
2. Dequeue S and explore its neighboring nodes.
   Queue = [A, B, C]
3. Dequeue A and explore neighbour
   Queue = [B, C, D].
4. Dequeue B and explore neighbour
   Queue = [C, D, G].
5. Dequeue C and explore neighbour.
   Queue = [D, G].
6. Dequeue D.
   Queue = [G]
7. Dequeue G.

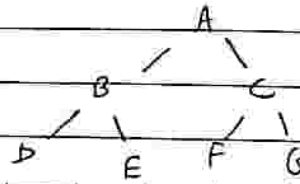
G is our destination node, BFS will stop here
Route from S to G : S → B → G.

11. What do you mean by depth limited search? Explain iterative deepening search? with examples.

Ans 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 deeping search combine DLS with BFS by incrementally increasing the depth limit.

Example:



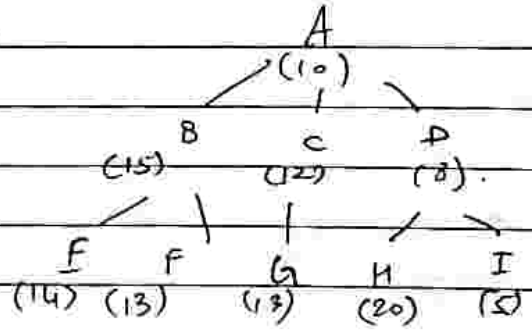
Goal = G.

- 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.

12. Explain Hill Climbing and its drawback in detail. with example.
Also state limitations of steepest ascent climbing.

Ans Hill Climbing is a local search optimization algorithm which moves forward towards a better neighboring solution until it reaches a peak

Example



Goal = G.

Steps.

1. Start at root node A (10).
2. Compare its children B, C, D
3. Move to child with highest value i.e B(15)
4. Repeat for B's children E and F.

4. Terminate at E(14).
The algorithm stops at E(14) not reaching the Goal G.

Drawbacks.
1. Local maxima: The algorithm greedily selects the best immediate child and can thus get stuck on a local maxima.
2. Plateau: If siblings have equal values, the algorithm cannot decide the next step and gets stuck.
3. Ridges: Narrow uphill paths require backtracking which Hill climbing algorithm does not support.

Limitations of steepest ascent hill climbing are as follows.
1. Computationally experience: 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 optimality: It only focuses on immediate improvements

13. Explain simulated annealing and write its algorithm.
Ans Simulated annealing is probabilistic optimization algorithm inspired by metallurgical process of annealing where materials are heated and cooled to reduce defects. It escapes the local optima by temporarily accepting worse solution with its probability.

Algorithm.
1. Initialize -
• Set an initial solution and define an initial temperature T.
2. Repeat until stopping condition
• Generate a new neighbour solution
• Compute changes in cost.

If new solution is better than accept it.
- If worse, accept it with probability
- Decrease temperature T.

3. Return best solution.

Example: Travelling salesman problem.

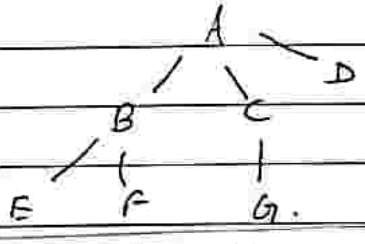
14. Explain A^* algorithm with an example.

Ans A^* is a best first search algorithm used in path finding 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.

Goal : G



Node	g(A, h).	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

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

$= 6$

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 neighbours of C.
5. $f(G) = 2+4+0 = 6$.
6. Goal reached at G with total cost = 6

Advantages:

- Efficient for finding the shortest path in weighted graphs.
- Balance exploration by considering both $g(n)$ and $h(n)$.

15. Explain Min-Max algorithm and draw game tree for Tic Tac Toe game.

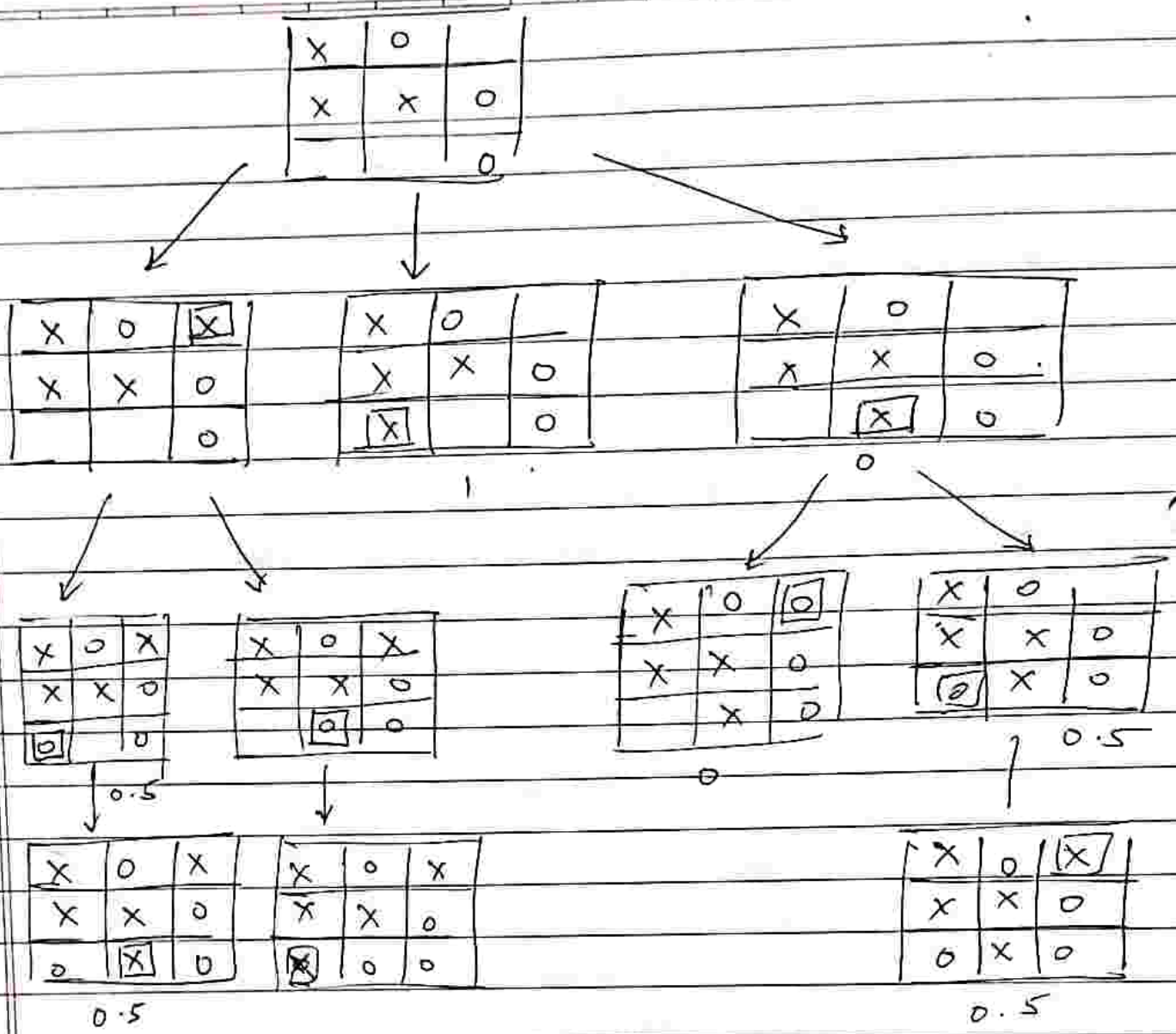
Ans The min max algorithm is a decision making algorithm used in 2 player games. It assumes

- One player (MAX) tries to maximize the 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.

Game tree for tic tac toe game is as follows:



16. Explain Alpha beta pruning algorithm for adversial search with example.

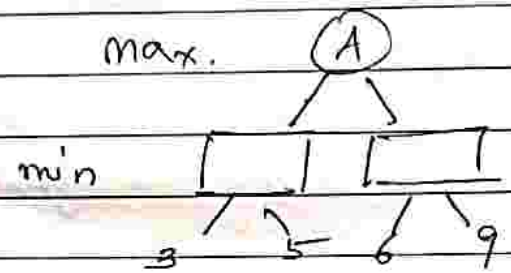
Ans Alpha Beta pruning is an optimization technique used in minimum algorithm to reduce the number of nodes evaluated in adversial search problems like game - playing AI (eg. chess, tic-tac-toe)

Alpha Beta pruning includes:

Alpha (α) : The best maximum score that the maximising 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 $\longrightarrow$ Update $\beta = 3$.
   - Check second child value = 5 $\longrightarrow$ β remains 3.
   - Min node returns 3 to MAX.
3. Right Min node (child of A)
   - Check first child value = 6 $\longrightarrow$ $\beta = 6$.
   - Here, $\alpha = 3$ at MAX node but $\beta(6) > \alpha(3)$ so no pruning
   - Explore 2nd child (9) $\longrightarrow$ Here pruning will occur
   ∴ MIN node already has a value ≤ 6 it will never choose 9 and so we prune the node with value 9.
4. Max value = 6

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

Ans The wumpus world environment is a simple grid-based environment used in AI to study intelligent agent behaviour

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 monster called WUMPUS.

PEAS one as follows:
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 perception received by 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 an 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 following crypto-arithmetic problem:

1. SEND + MORE = MONEY.

Ans Step 1: M must be 1. The sum of two four digit numbers cannot be more than 10,000.

```
  SEND
+ 10RE
------
 10NEY
```

Step 2: Now, S must be 8 because there is 1 carry over from column E-O-N. O must be 0 (if S=8) and there is a 1 carried or S=9 and there is no 1 carried or (if S=9 and there is 1 carried). But 1 is already taken so O must be 0.

```
  SEND
+ 10RE
------
 10NEY
```

Step 3: There cannot be a carry from column EON because any digits +0 < 10, unless there is a carry from the column NRE and E=9. But this cannot be the case because then N would be 0 and 0 is already taken. So E<9 and there is no carry from this column. Therefore, S=9 because 9+1 = 10.

Step 4:

Case 1: No carry : N+R = 10 + (N-1) = N+9.

R=9

But 9 is taken already so will not work.

Case 2: Carry : N+R+1 = 9.

R = 9-1 = 8 . This must be the solution of R.

Step 5:

Let us consider E = 5 or 6.

E = 5.

then D = 7, & Y = 3. So this part will work but look at the column N 8 E. There is a carry from the column DEY. N + 8 + 1 = 16. But then N = 7 and 7 is taken by D therefore E = 5

```
  9 5 N D.
+ 1 0 8 5
-----------
1 0 N 5 Y.
```

Now,

N + 8 + 1 = 15 , N = 6.

```
  9 5 6 D.
+ 1 0 8 5
-----------
1 0 6 5 Y.
```

Step 6:

The digits left are 7, 4, 3 & 2. We know there is a carry from column D5Y, so only pair that works is D = 7 & Y = 2.

```
  9 5 6 7
+ 1 0 8 5
-----------
1 0 6 5 2
```

19. Consider the following axioms.

All the people who are graduating are happy

All happy people are smiling

Someone is graduating.

Ans Representing these axioms in first order predicate logic.

We define the following predicates

- G(x) : x is graduating.
- H(x) : x is happy

• $S(x)$: x is smiling.

Translating axiom into predicate logic :

1. All people who are graduating are happy.

$\forall x \ (G(x) \rightarrow H(x))$.

2. All happy people are smiling.

$\forall x \ (H(x) \rightarrow S(x))$.

3. Someone is graduating

$\exists x \ G(x)$

Convert each formula to clause form :

1. Convert implication to clausal form.

$\forall x \ (G(x) \rightarrow H(x))$.

• Using implication removal.

$\forall x \ (\neg G(x) \vee H(x))$.

• In clause form.

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

2. $\forall x \ (H(x) \rightarrow S(x))$

• Using implication removal.

$\forall x \ (\neg H(x) \vee S(x))$.

• In clausal form.

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

3. $\exists x \ G(x)$

• In clausal form : $\{ G(x) \}$.

Prove " is someone smiling" using resolution.

1. Collect clauses.
(1) { ¬ G(x) , H(x) }.
(2) { ¬H(x), S(x) }.
(3) { G(x) }.

2. Apply resolution.
• Resolve (1) { ¬G(x) , H(x) } with (3).
{ G(a) }.
substituting x = a
{ ¬ G(a) , H(a) }.
∵ we have G(a) , resolving gives
{ H(a) }.
• Resolve (2) { ¬H(x) , S(x) } with { H(a) }.
Substituting x = a.
{ ¬H(a) , S(a) }
• Since, we have H(a) resolving gives.
{ S(a) }

Since, we derived S(a) , we conclude that- Someone (a) is ~~sin~~ smiling.

20. Explain modus ponen with suitable example.
Ans Modus ponen is a fundamental rule of inference in ~~propotional~~ propositional logic that allows us to deduce a conclusion from a conditional statement and its antecedant.
It follows the form.
1. P → Q (if P then Q)
2. P (P is true)
∴ Q (Q must be true).

Example

1. If it rains, the ground will be wet
   → P → Q.
2. It is raining → P
   ∴ Ground is wet → Q.

Q1. Explain forward chaining and backward chaining algorithm with the help of example.

Ans 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 : Diagnosing a disease.

Rules:

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

Facts.

: The patient has a fever.
. The patient has cough

Inferencence

1. Fever + cough → flu (rule 1 applies)
2. Conclusion; The patient might have flu.

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

Example : Diagonising a disease

Goal : Determine if a patient has flu.

Rules

1. (Fever $\wedge$ Cough) $\rightarrow$ Flu.
2. (Sore Throat $\wedge$ Fever) $\rightarrow$ Cold.

Process using backward chaining.

1. We want to prove flu.
2. Looking at rule 1 : (Fever $\wedge$ cough) $\rightarrow$ Flu, we need to check if patient has fever and cough.
3. We check our known facts.
   - Patients has fever.
   - Patient has cough.
4. Since, both ~~conclusions~~ conditions are met, we confirm flu is true.