13/03/25

ASSIGNMENT- I

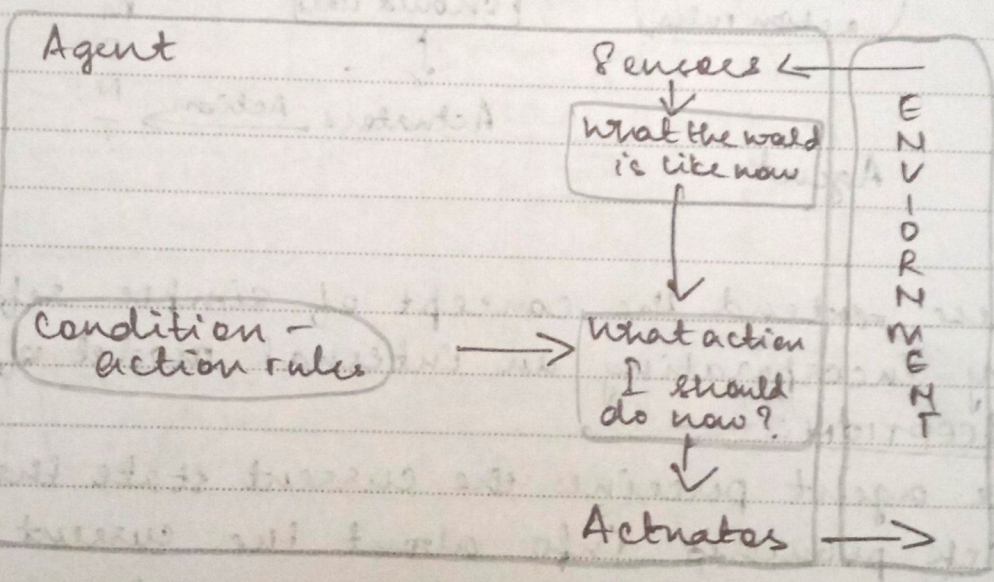
LEVEL-2

1. Compare and Contrast 4 types of agent programs? Explain in detail the properties of the task Environment.

Ans:- The 4 types of agent programs are:

- Simple reflex
- Model-based reflex
- Goal based agents
- Utility based agents

SIMPLE REFLEX AGENTS

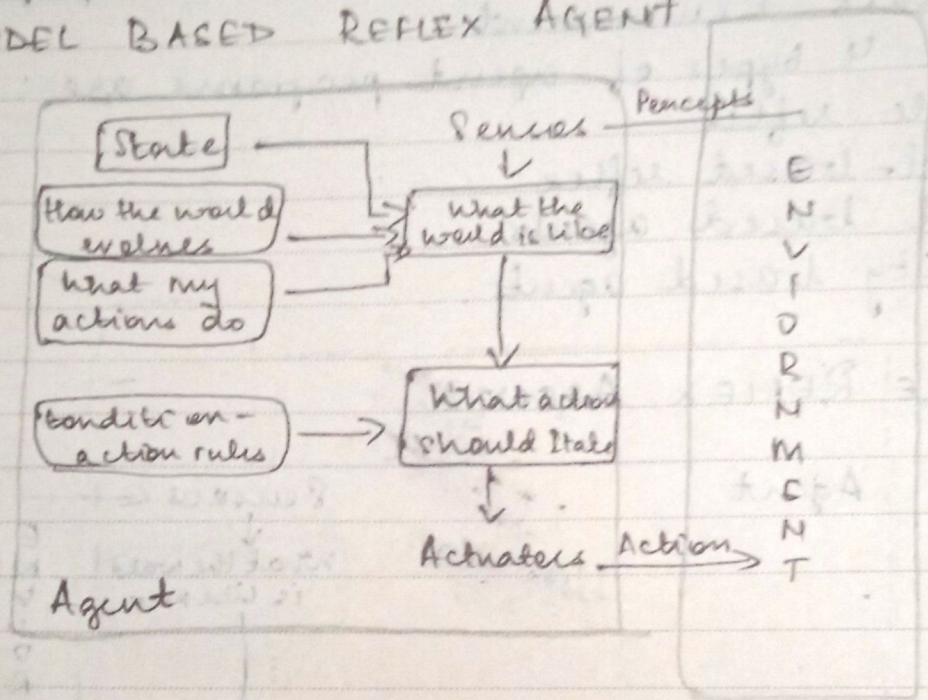


→ This is the simplest kind of reflex agent
→ These select actions on the basis of the current percept, ignoring the rest of percept history
→ They operate in a continuous loop of perception and action
→ Sensors capture information about the environment through sensors

eg: Reflex- Vaccum Agent:

If status = Dirty then return Suck
else if location = A then return Right
else if location = B then return left

→ MODEL BASED REFLEX AGENT



- these extend the concept of simple reflex agents by incorporating an internal model of environment
- PERCEPTION:

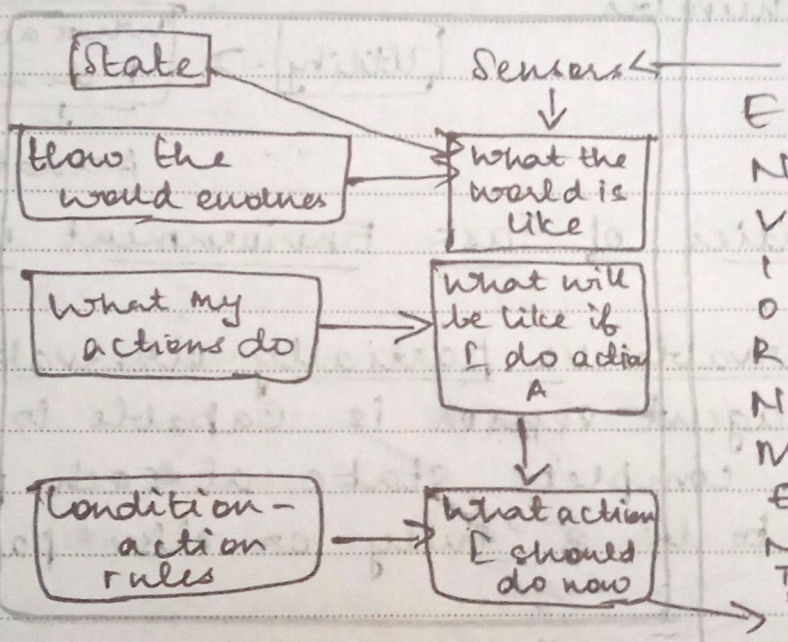
The agent perceives the current state through sensors, which provides info about the current state, like presence of obstacles, objects or others.

- MODELING THE ENVIORNMENT:
- the agent maintains an internal model of the environment, which includes info about the state of the world, the possible actions and the expected outcomes of those actions
- DECISION MAKING

the agent selects an action to perform based on its current perceptual input and internal model

- **ACTION EXECUTION:**
  The agent executes the selected action in environment, which causes changes to the state of the world
- **UPDATING THE MODEL:**
  After taking an action, the agent updates its internal model of the environment based on the new perceptual information it receives

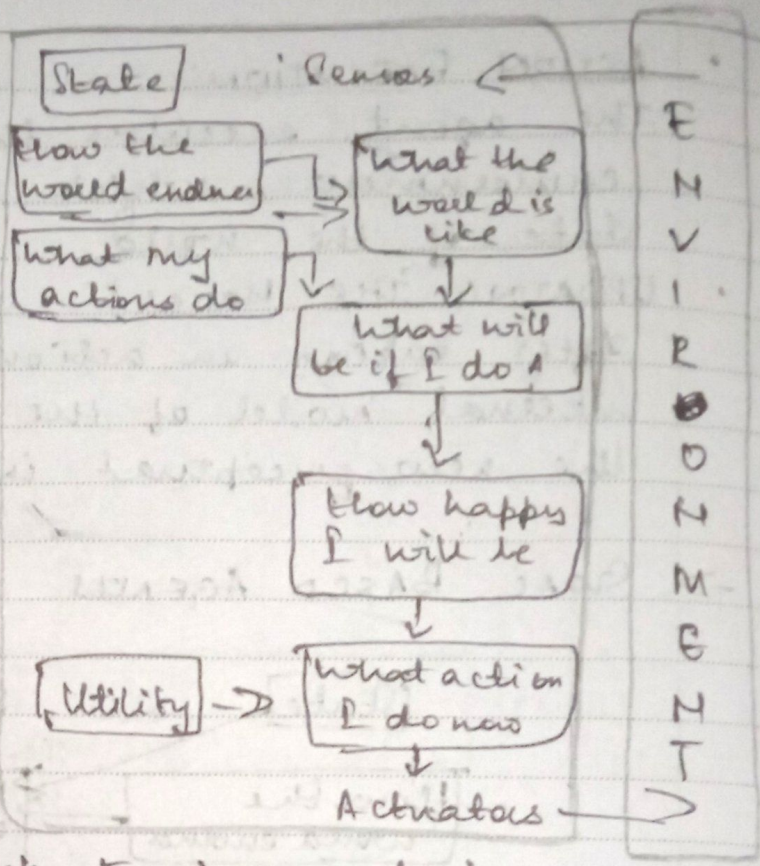
→ GOAL BASED AGENTS



- To reach their goals, these AI agents employ planning algorithms.
- The planning process often involves examining a tree of possibilities
- These consider the potential consequences of each action

→ UTILITY Based agents:

- These agents are similar to goal based but provide an extra component of ~~utility~~ utility.
- This is useful when there are multiple possible alternatives
- the utility function maps each state to a real number



The Properties of Task Enviornment include:

- Fully observable vs partially observable:
When an agent sensor is capable to sence or access the complete state at each point in time it is said to be a fully or else partially observable
- Deterministic vs stochastic:
When the uniqueness completely determines the next agent → Deterministic enviornment
When the enviornment is not unique and is random → Stochastic
- Episodic vs sequential:
When each of the agent's actions is divided into atomic incidents or episodes - Episodic
When the next action depends on the previous incidents → Sequential

- Static vs Dynamic:
An Idle environment with no change → static
An environment that keeps changing constantly - dynamic
- Discrete vs continuous:
If an environment consists of finite no of actions → discrete
The environment in which the actions are not discrete → continuous
- Known vs Unknown:
The output for all probable actions is given - known
The output is not given: Unknown.

2. Compare and Contrast the definitions of AI from the point of views:

Ans: Thinking Humanly:
"The exciting new effort to make comput-ers think... machines with minds, in the full and literal sense." - Haugeland, 1985.

Thinking Rationally:
The study of mental faculties through the use of computational models". - Charniak and McDermott, 1985.

Acting Humanly:
The art of creating machines that perform functions that require intelligence when performed by people." - Kurzweil, 1990

Acting Rationally:
"AI.... is concerned with intelligent be haviour in artifacts - Nilsson, 1998

2. Explain an Iterative deepening depth first search algorithm with an example:

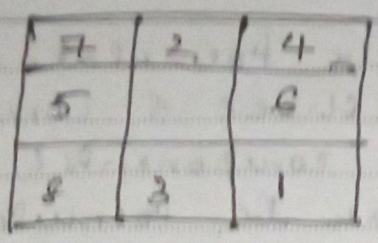
Ans: Iterative deepening search is a general strategy, often used in combination with depth first tree search that finds the best-depth limit. It does this by gradually increasing the limit - first 0, then 1, 2 and so on - until a goal is found.

Example: The iterative deepening search algorithm, which repeatedly applies depth-limited search with increasing limits. It terminates when a solution is found or if the depth limited search returns failure, meaning that no solution exists

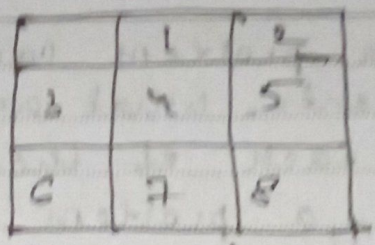
function Iterative Deepening Search (problem)
returns a solution, or failure
for depth = 0 to ∞ do
result ← Depth-limited-Search (problem, depth)
if result ≠ cutoff then return result.

4. Formulate a problem for a toy problem like vaccum world or 8-puzzle?

Ans: The 8-puzzle, an instance of which is consists of a 3×3 board with 8 numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space. The object is to reach a specified goal state.



Start



Goal

States:

- State: specifies the location of each of the eight tiles and the blank in one of the nine squares
- Initial state: Any state can be designated as the initial state. Note that any given goal can be reached from exactly half of the possible initial states
- Actions: The simplest formulation defines the actions as movements of the blank space left, Right, Up or down.
- Transition model: Given a state and action, this returns the resulting state; for eg: if we apply left to start state, the 5 and the blank switched
- Goal taste: this checks whether the state matches the goal configuration.
- Path cost: Each step costs 1, so the path cost is the number of steps in the path.

the 8- puzzle belongs to the family of sliding block puzzles, which are often used as test problems for new search algorithms in AI.

5. In AI a problem can be formulated using 5 components. What are these 5 components? Explain each of these components?

Ans: In AI, a problem can be formulated using 5 components which are:

→ Initial State:

- this is the starting point of the problem
- It represents the initial conditions or configuration from which the AI begins its search for a solution

eg: In a chess game, the initial state is the standard starting position of all pieces on the board.

→ Actions:

- The possible moves or transformations that can be applied to the current state to reach a new state
- Each action has a defined set of rules that determines how it modifies the current state

eg: In a route-finding problem, the action could be moving from 1 city to another.

→ Transition model:

- It defines the result of applying an action to a state
- It specifies how the system moves from 1 state to another based on the chosen action.

eg: In a maze-solving problem, if an agent moves "right", the transition model updates the position accordingly.

→ Goal State:

- The desired or target state that the AI system aims to reach.
- The AI searches for a sequence of actions that leads to this state

e.g. In a puzzle game, the goal state is when all pieces are arranged correctly.

→ Path Cost:

- A function that assigns a cost to each action or sequence of actions taken
- It helps in finding the most efficient or optimal solution rather than just any solution.