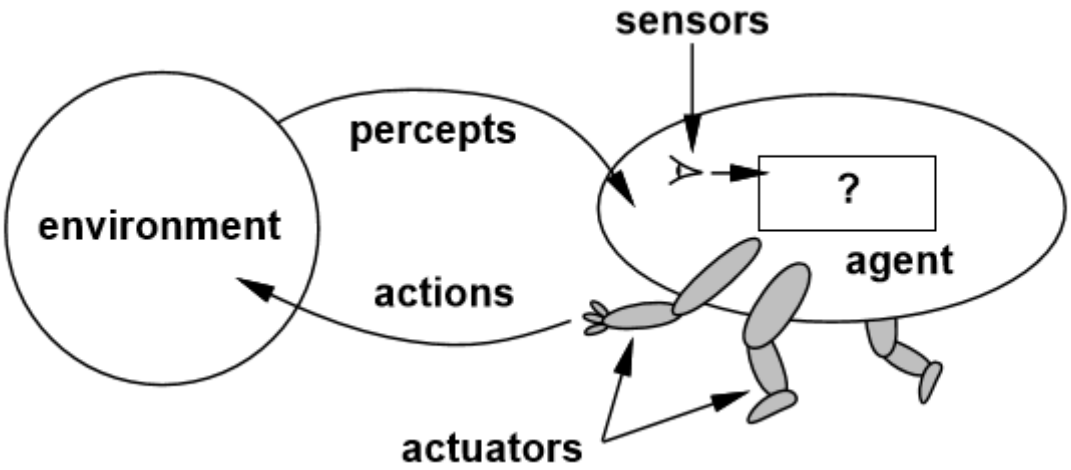
**1. Attempt *any three* of the following:** **15**

a. Elaborate artificial intelligence with suitable example along with its applications.
Ans: The art of creating machines that perform functions that require intelligence when performed by people."
The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning

b. Discuss the historical evolution of Artificial Intelligence.
1943: McCulloch & Pitts: model of neurons → Boolean circuit of the brain
♦ 1949: Donald Hebb-upgrading rule for modifying the connection strengths (Hebbian learning)
♦ 1950: Turing's Computing Machinery and Intelligence: introduces Turing Test, machine learning, genetic algorithms, and reinforcement learning.
The birth(1956): ♦McCarthy(1927-2011):make machine use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.

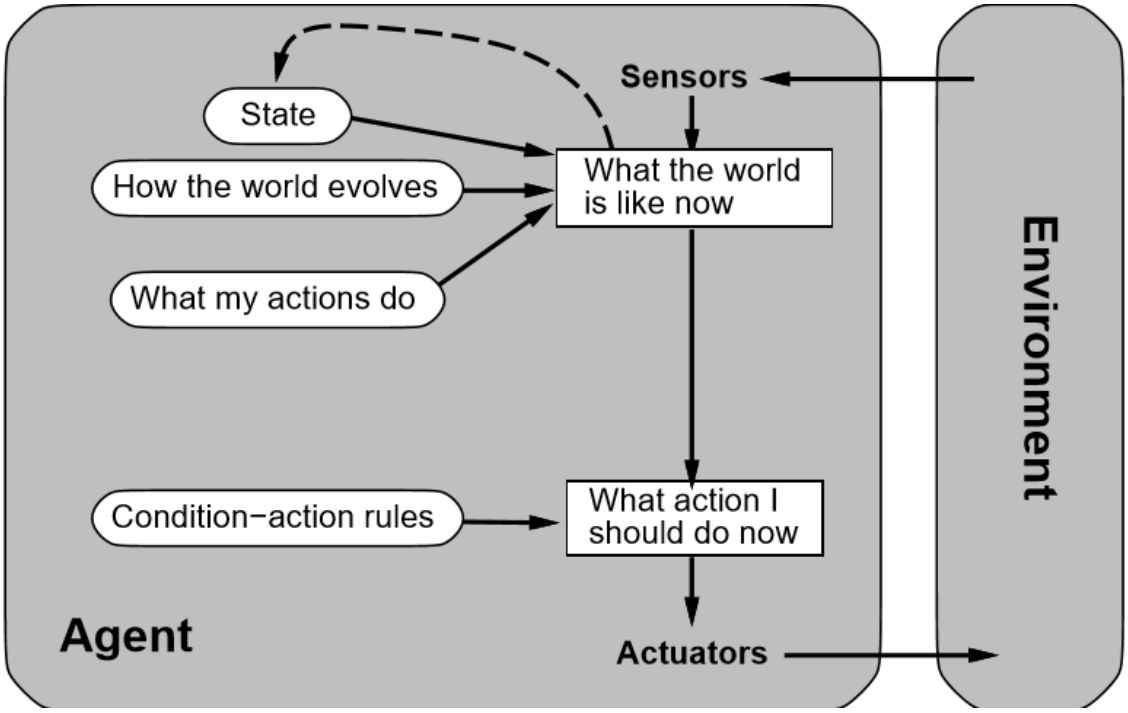
c. State the relationship between agents and environment.
An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. In agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived. the agent function for an artificial agent will be implemented by an agent program.




d. Explain the concept of Rationality.
A rational agent is one that does the right thing—conceptually speaking, every entry in the table for the agent function is filled out correctly. Doing the right thing is better than doing the wrong thing, but what does it mean to do the right thing'. Consider, for example, the vacuum-cleaner agent from the preceding section.
The performance measure that defines the criterion of success. • The agent's prior knowledge of the environment. • The actions that the agent can perform. • The agent's percept sequence to date.

e. Explain types of environments.
Fully observable:sensors give access to the complete state of the environment at each point in time. – partially obervable: noise, nosensors,
♦ Single/Multiagent Does an agent A(the taxi driver for example)have to treat an object B(another vehicle) as an agent, or can it be treated merely as an object behaving according to the laws of physics? –whether behavior is best described as maximizing a performance measure whose value depends on agent As behavior. –competitve/cooperative(chess,taxy,tenis)
♦ Deterministic/stochastic: the next state of the environment is completely determined by the current state and the action of the agent
♦ Uncertain:not fully observable or not deterministic.
♦ Nondeterministic: actions are characterized by their possible outcomes (no probabilities attached as in case of stochastic).

f. Explain reflex agents with state.




**2. Attempt *any three* of the following:** **15**

a. Write the procedure for tree search.

function Tree-Search(problem,strategy) returns a solution,or failure initialize the search tree using the initial state of problem
loop do
if there are no candidates for expansion then return failure
choose a leaf node for expansion according to strategy
if the node contains a goal state then return the corresponding solution
else expand the node and add the resulting nodes to the search tree
end

b. Explain the algorithm for breadth first search algorithm.

1. Create a variable called NODE-LIST and set it to initial state.
2. Until a goal state is found or NODE-LIST is empty do:

a. Remove the first element from NODE-LIST and call it E. if NODE-LIST was empty, quit.

b. For each way that each rule can match the state described in E do:
i. Apply the rule to generate a new state.
ii. If the new state is goal state, quit and return this state.
iii. Otherwise, add the new state to the end of NODE-LIST.

c. Give the outline of Uniform-cost search algorithm.

UNIFORM- COST-SEA RC (pro blens) returns a solution, or failure node 4— a node with STATE = proMern.INITIAL-STATE, PATH-COST = frontier 4— a priority queue ordered by PATH-COST, with node as the only element explored 4— an empty set loop do if EMPTY?( frontier) then return failure node q— Poet frontier) ftt chooses the lowest-cost node in frontier e/ if pro blen
2.GOAL - TEST (node.5 STATE) then return SOLUTION(node)
Edit node. STATE kJ explored
for each action in problem.ACTIONS(node.STATE)
do child 44— CHILD-Noon (problern, node, action) if child . STATE is not in explored or frontier then frontier INSERT( child, frontier) else if child.STATE is in frontier with higher PATH-COST then replace that frontier node with child.

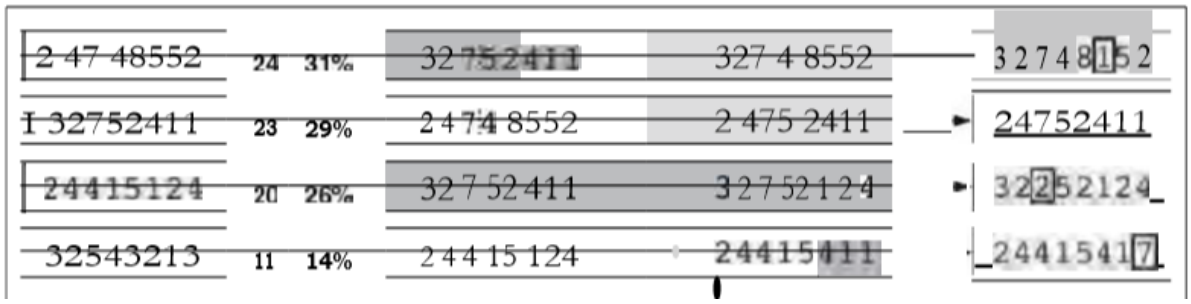
d. Explain A* algorithm for the shortest path.

A* search {pronounced "A-star search"). It evaluates nodes by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal: (n) = g(n) + h(n) . Since g(n) gives

the path cost from the start node to node n, and h(n) is the estimated cost of the cheapest path from r. to the goal, we have f (n) = estimated cost of the cheapest solution .

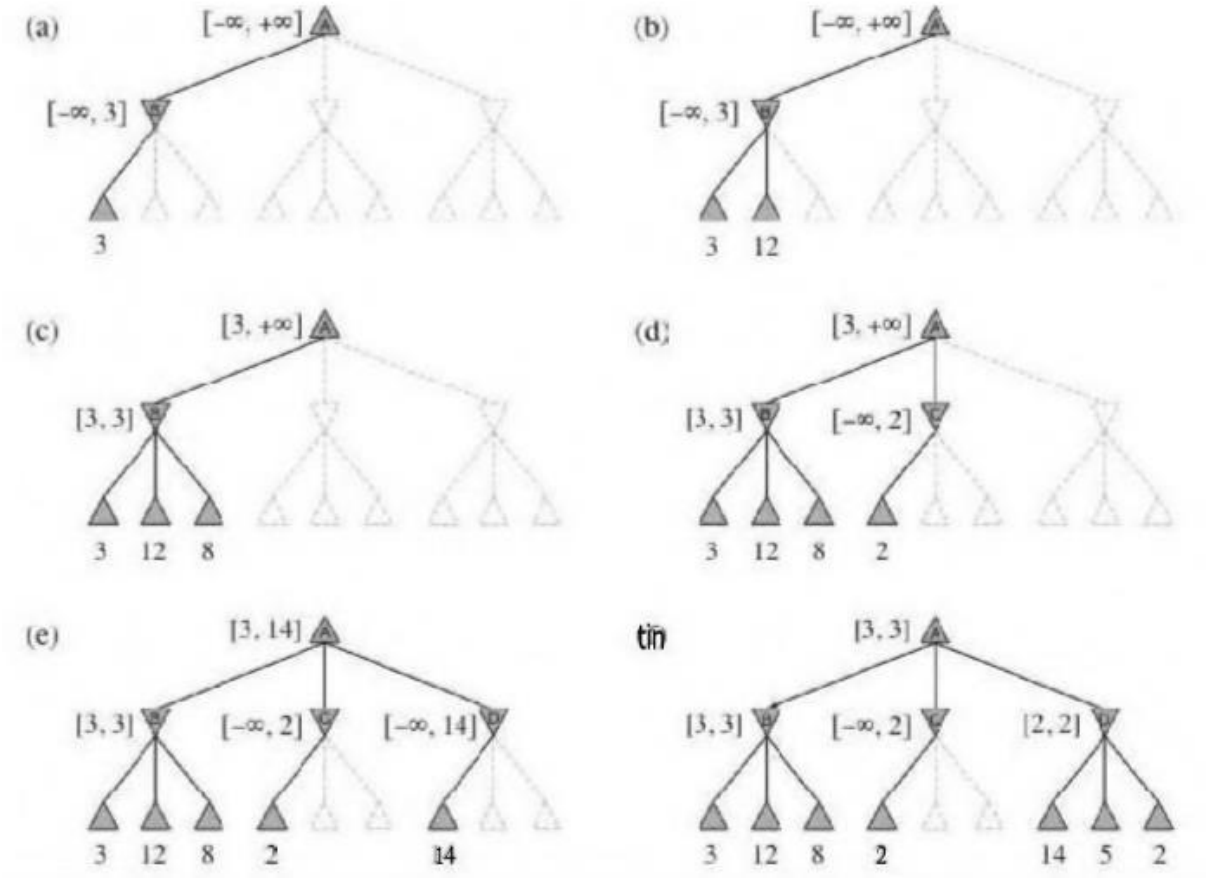
e. Give the outline of Hill climbing algorithm.
function Hill-Climbing(problem) returns a state that is a local maximum inputs: problem, a problem local variables: current,a node neighbor, a node current←Make-Node (Initial-State[problem])
loop do
neighbor←a highest-valued successor of current if Value[neighbor] ≤ Value[current] then return State[current]
current←neighbor
end

f. Explain the working mechanism of genetic algorithm.
A genetic algorithm (or GA) is a variant of stochastic beam search in which successor states are generated by combining two parent states rather than by modifying a single state. The analogy to natural selection is the same as in stochastic beam search, except that now we arc dealing with sexual rather than asexual reproduction.

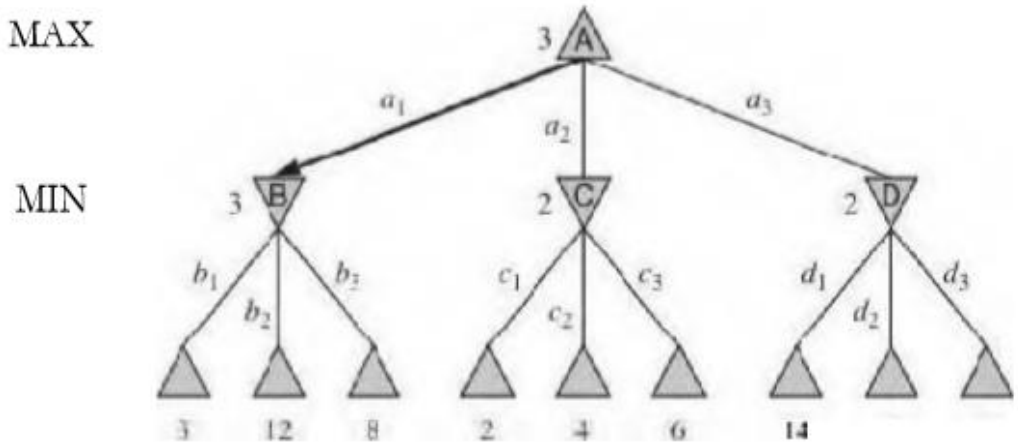



**3. Attempt *any three* of the following:** **15**

a. What is alpha-beta pruning? Explain the function of alpha beta pruning.
The problem with minimax search is that the number of game states it has to examine is exponential in the depth of the tree. Unfortunately, we can't eliminate the exponent, but it turns out we can effectively cut it in half. The trick is that it is possible to compute the correct minimax decision without looking at every node in the game tree.
Alpha—beta pruning can be applied to trees of any depth, and it is often possible to prune entire subtrees rather than just leaves.



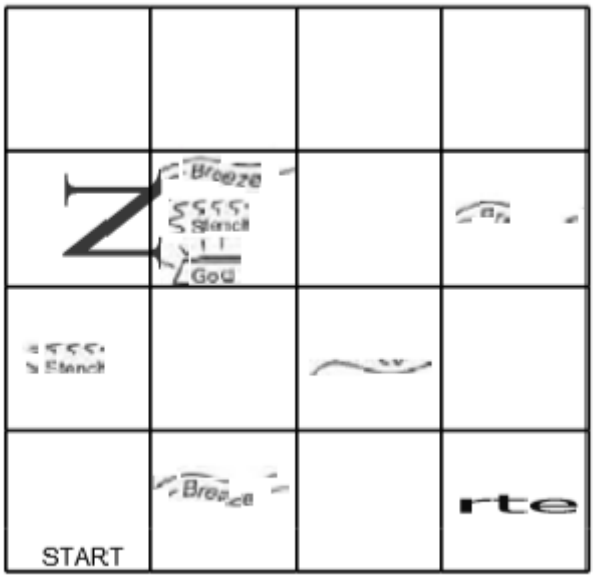

b. Give the outline of min-max algorithm.
The minimax algorithm computes the minimax decision from the current state. It uses a simple recursive computation of the minimax values of each successor state, directly implementing the defining equations. The recursion proceeds all the way down to the leaves of the tree, and then the minimax values are backed up through the tree as the recursion unwinds.




c. Write a note on card games.
Card games provide many examples of stochastic partial observability, where the missing information is generated randomly. For example in many games, cards arc dealt randomly at the beginning of the game, with each player receiving a hand that is not visible to the other players. Such games include bridge, whist, hearts, and some forms of poker. At first sight, it might seem that these card games are just like dice games: the cards are dealt randomly and determine the moves available to each player, but all the "dice" are rolled at the beginning! Even though this analogy turns out to be incorrect, it suggests an effective algorithm: consider all possible deals of the invisible cards; solve each one as if it were a fully observable game; and then choose the move that has the best outcome averaged over all the deals.

d. What is knowledge based agent? Explain its role and importance.
The central component of a knowledge-based agent is its knowledge base, or KB. A knowl edge base is a set of sentences. Each sentence is expressed in a language called a knowledge representation language and represents some assertion about the world, Sometimes we dignify a sentence with the name axiom, when the sentence is taken as given without being derived from other sentences. A knowledge-based agent can be built simply by TELLing it what it needs to know. Starting with an empty knowledge base, the agent designer can TELL sentences one by one until the agent knows how to operate in its environment.

e. Write a note on Wumpus world problem.




2 3 4

Wumpus agent exploring the environment shown in Figure. The first percept is :None, None, None, None, None], from which the agent can conclude that its neighboring squares, [1,2] and [2,1], are free of dangers—they are OK. Fig. shows the agent's state of knowledge at this point. A cautious agent will move only into a square that it knows to be OK. Let us suppose the agent decides to move forward to [2,1]. The agent perceives a breeze (denoted by "B") in [2,1], so there must be a pit in a neighboring square. The pit cannot be in [1,1], by the rules of the game, so there must be a pit in [2,2] or [3,1] or both. The notation "PT' in Figure indicates a possible pit in those squares. At this point, there is only one known square that is OK and that has not yet been visited. So the prudent agent will turn around, go back to [1,1], and then proceed to [1,2].

f. Give the outline of resolution algorithm.
In resolution algorithm. First, (KB A -,a) is converted into CNF. Then, the resolution rule is applied to the resulting clauses. Each pair that contains complementary literals is resolved to produce a new clause, which is added to the set if it is not already present. The process continues until one of two things happens: ■ there are no new clauses that can be added, in which case KB does not entail a; or, ■ two clauses resolve to yield the empty clause, in which case KB entails a.
function PL-REsoLuTioN(KB, a) returns true or false
inputs: KB. the knowledge base, a sentence in propositional logic
a:, the query, a sentence in propositional logic clauses 4— the set of clauses in the CNF representation of KB A -,cr new i }
loop do
for each pair of clauses C
in clauses do resolvents — PL-RESOLVE(Ci,Ci)
if resolvents contains the empty clause then return true new — new U resolvents
if new C clauses then return false clauses 4— clauses U new

**4. Attempt *any three* of the following:** **15**

a What are predicates? Explain its syntax and semantics.

- First-order logic is used to model the world in terms of
  - **objects** which are things with individual identities

  e.g., individual students, lecturers, companies, cars ...
  - **properties** of objects that distinguish them from other objects

  e.g., mortal, blue, oval, even, large, ...
  - **classes** of objects (often defined by properties)

  e.g., human, mammal, machine, ...
  - **relations** that hold among objects

  e.g., brother of, bigger than, outside, part of, has color, occurs after, owns, a member of, ...
  - **functions** which are a subset of the relations in which there is only one ``value'' for any given ``input''.

  e.g., father of, best friend, second half.
- **Predicates:** P(x[1], ..., x[n])
  - P: **predicate name**; (x[1], ..., x[n]): **argument list**
  - A special function with range = {T, F};
  - Examples: human(x), /* x is a human */
    father(x, y) /* x is the father of y */
  - When all arguments of a predicate is assigned values (said to be **instantiated**), the predicate becomes either true or false, i.e., it becomes a proposition. Ex. Father(Fred, Joe)

A predicate, like a membership function.

b What are Quantifiers? Explain the types with syntax and example.
Quantifiers let us do this First-order logic contains two standard quantifiers, called universal and existential.

- Universal quantification (for-all)
- Existential quantification (there-exist)

c Convert the following into predicate form:

i. Virat is software engineer.
ii. All vehicles have wheels
iii. Some-one speaks some language in this class.
iv. Everybody loves somebody sometime.
v. All software engineer develops software.

d Explain the process of knowledge engineering.

I. Identify the task
II. Assemble the relevant knowledge
III. Decide on a vocabulary of predicates, functions, and constants.
IV. Encode general knowledge about the domain
V. Encode a description of the specific problem instance
VI. Pose queries to the inference procedure and get answers
VII. Debug the knowledge base

e What is unification? Explain the process of unification.
Lified inference rules require finding substitutions that make different logical expressions look identical. This process is called unification and is a key component of all first-order inference algorithms. The UNIFY algorithm takes two sentences and returns a unifier for them if one exists.
Here are the results of unification with four different sentences that might be in the knowledge base: UNIFY (Knows (John x). Knows (John, Jane)) = Ix I Jane} UNIFY (Knows (John , x). Knows (1) , Bill)) = {x I Bill, y I John} UNIFY (Knows( John , x), Knows (11 , Mother (v))) = {g,r, John x I Illother (John)} UNIFY (Knows(John, x). Knows (x , El2zabeth)) = fail.

f Give the outline of simple forward chaining algorithm.

The first forward-chaining algorithm we consider is a simple one. is designed for ease of understanding rather than for efficiency of operation. There are three possible sources of inefficiency. First, the "inner loop" of the algorithm involves finding all possible unifiers such that the premise of a rule unifies with a suitable set of facts in the knowledge base. This is often called pattern matching and can be very expensive. Second, the algorithm rechecks every rule on every iteration to see whether its premises are satisfied, even if very few additions are made to the knowledge base on each iteration. Finally, the algorithm might generate many facts that are irrelevant to the goal. We address each of these issues in turn.

**5. Attempt *any three* of the following:** **15**

a. What is planning? Explain the need of planning.
A planning is sequence of action used to achieve a desire a goal.

b. Explain block world problem for the following start state and end state.

**Ans:** Assume suitable start and end states.


$Action(Move(b, x, y),$
PRECOND: $On.(b, x)$ A $Clear\ (b)$ A $Clear\ (y)$,
EFFECT: $On(b.a)$ A $Clear(x)$ A $Or a x$ A $Clear(y))$.

c. Write a note on planning graph.
These heuristics can be applied to any of the search techniques we have seen so far. Alternatively, we can search for a solution over the space formed by the planning graph, using an algorithm called GRAPHPLAN. A planning problem asks if we can reach a goal state from the initial state. Suppose we are given a tree of all possible actions from the initial state to successor states, and their successors, and so on. If we indexed this tree appropriately, we could answer the planning ques- tion "can we reach state G from state So" immediately, just by looking it up. Of course, the tree is of exponential size, so this approach is impractical. A planning graph is polynomial- size approximation to this tree that can be constructed quickly. The planning graph can't answer definitively whether G is reachable from So, but it can estimate how many steps it takes to reach G. The estimate is always correct when it reports the goal is not reachable, and it never overestimates the number of steps, so it is an admissible heuristic. A planning graph is a directed graph organized into levels: first a level So for the initial state, consisting of nodes representing each fluent that holds in So; then a level A0 consisting of nodes for each ground action that might be applicable in So; then alternating levels S, followed by A

d. What are events? Explain its importance.
Event calculus reifies fluents and events. The fluent At(Shankar, , Berkeley) is an object that refers to the fact of Shankar being in Berkeley, but does not by itself say anything about whether it is true. To assert that a fluent is actually true at some point in time we use the predicate T, as in T(At(Shankar. . Berkeley), t).

e. Write a note on semantic network.
A graphical notation of nodes and edges called existential graphs that he called "the logic of the future." The notation that semantic networks provide far certain kinds of sentences is often more

convenient, but if we strip away the "human interface" issues, the underlying concepts—objects, relations, quantification, and so on—are the same.

f. Write a note on Truth maintenance system.

One simple approach to truth maintenance is to keep track of the order in which sentences are told to the knowledge base by numbering them from P1 to P. When the call RETRAcT(K/3, P) is made, the system reverts to the state just before P was added, thereby removing both P, and any inferences that were derived from P. The sentences Pi+i through P r, can then be added again. This is simple, and it guarantees that the knowledge base will be consistent, but retracting .13 7. requires retracting and reasserting n — i sentences as well as undoing and redoing all the inferences drawn from those sentences.