CMRIT

Internal Assessment Test 1 – Nov.2021								
Sub:	Artificial Intelligence and Machine Learning			**Sub Code:**	18CS71	**Branch:**	CSE	
Date:	11/11/21	**Duration:**	90 mins	**Max Marks:** 50	*Sem / Sec:*	VII A,B,C		OBE
	Answer any FIVE FULL Questions					MARKS	CO	RBT
1 (a)	**The water jug problem states: You are provided with two jugs, first one with 4- gallon capacity And second one with 3 gallons of capacity. Neither have any measuring mark on it. How can we get exactly two gallons of water in a 4 gallon jug?**					[05]	CO1	L2

i) Write down the production rules for the above problem

ii) Write any one solution for the above problem

Answer:

Production rules for solving the water jug problem

Here, let ***x*** denote the 4-gallon jug and ***y*** denote the 3-gallon jug.

S.No.	Initial State	Condition	Final state	Description of action taken
1.	(x,y)	If x<4	(4,y)	Fill the 4 gallon jug completely
2.	(x,y)	if y<3	(x,3)	Fill the 3 gallon jug completely
3.	(x,y)	If x>0	(x-d,y)	Pour some part from the 4 gallon jug
4.	(x,y)	If y>0	(x,y-d)	Pour some part from the 3 gallon jug
5.	(x,y)	If x>0	(0,y)	Empty the 4 gallon jug
6.	(x,y)	If y>0	(x,0)	Empty the 3 gallon jug
7.	(x,y)	If (x+y)<7	(4, y-[4-x])	Pour some water from the 3 gallon jug
8.	(x,y)	If (x+y)<7	(x-[3-y],y)	Pour some water from the 4 gallon jug
9.	(x,y)	If (x+y)<4	(x+y,0)	Pour all water from 3 gallon jug to the
10.	(x,y)	if (x+y)<3	(0, x+y)	Pour all water from the 4 gallon jug to

The listed production rules contain all the actions that could be performed by the agent in transferring the contents of jugs. But, to solve the water jug problem in a minimum number of moves, following set of rules in the given sequence should be performed:

Solution of water jug problem according to the production rules:

S.No.	4 gallon jug contents	3 gallon jug contents
1.	0 gallon	0 gallon
2.	0 gallon	3 gallons
3.	3 gallons	0 gallon
4.	3 gallons	3 gallons
5.	4 gallons	2 gallons
6.	0 gallon	2 gallons
7.	2 gallons	0 gallon

(b)	**Explain steepest Hill Climbing technique with an algorithm.** **Comment on its drawbacks and how to overcome these drawbacks**	[05]	CO1	L1

Answer:

The steepest-Ascent algorithm is a variation of the simple hill-climbing algorithm. This algorithm examines all the neighbouring nodes of the current state and selects one neighbour node which is closest to the goal state. This algorithm consumes more time as it searches for multiple neighbours.

Algorithm:

1. Evaluate the initial state. If it is also a goal state, then return it and quit. Otherwise, continue with the initial state as the current state.
2. Loop until a solution is found or until a complete iteration produces no change to current state:
   (a) Let *SUCC* be a state such that any possible successor of the current state will be better than *SUCC*.

(b) For each operator that applies to the current state do:

(i) Apply the operator and generate a new state.

(ii) Evaluate the new state. If it is a goal state, then return it and quit. If not, compare it to *SUCC*. If it is better, then set *SUCC* to this state. If it is not better, leave *SUCC* alone.

(c) If the SUCC is better than current state, then set current state to SUCC.

Problems faced in Hill Climbing Algorithm

Local maximum: The hill climbing algorithm always finds a state which is the best but it ends in a local maximum because neighboring states have worse values compared to the current state and hill climbing algorithms tend to terminate as it follows a greedy approach.

To overcome such problems, backtracking technique can be used where the algorithm needs to remember the values of every state it visited.

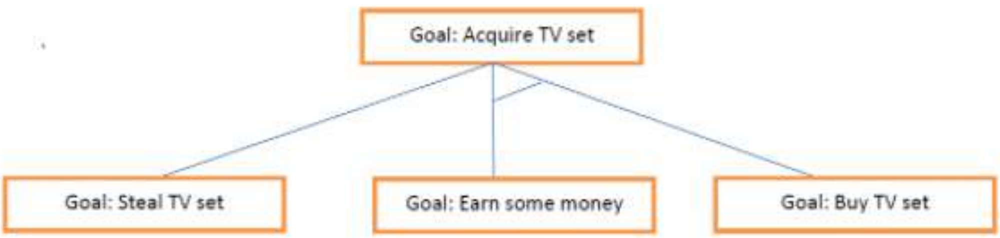
Plateau: In this region, all neighbors seem to contain the same value which makes it difficult to choose a proper direction.

To overcome such issues, the algorithm can follow a stochastic process where it chooses a random state far from the current state. That solution can also lead an agent to fall into a non-plateau region.

Ridge: In this type of state, the algorithm tends to terminate itself; it resembles a peak but the movement tends to be possibly downward in all directions.

To overcome such issues, we can apply several evaluation techniques such as travelling in all possible directions at a time.

2 (a)	**Explain simulated annealing** Answer: Simulated annealing is a probabilistic technique for approximating the global optimum of a given function. • Simulated annealing is a variation of hill climbing in which at the beginning of the process some downhill moves may be made. • The idea is to do enough exploration of the whole space early on so that the final solution is relatively insensitive to the starting state. • This should lower the chances of getting caught at local maxima, a plateau, or a ridge. • Annealing is a thermal process for obtaining low energy states of a solid in a heat bath. **The process contains two steps:** • Increase the temperature of the heat bath to a maximum value at which the solid melts. • Decrease carefully the temperature of the heat bath until the particles arrange themselves in the ground state of the solid. Ground state is a minimum energy state of the solid. • The ground state of the solid is obtained only if the maximum temperature is high enough and the cooling is done slowly. To do enough exploration of the whole space early on, so that the final solution is relatively insensitive to the starting state. Lowering the chances of getting caught at a local maximum, or plateau, or a ridge. **Physical Annealing** Physical substances are melted and then gradually cooled until some solid state is reached. The goal is to produce a minimal-energystate. Annealing schedule: if the temperature is lowered sufficiently slowly, then the goal will be attained. Nevertheless, there is some probability for a transition to a higher energy state: $e^{-\Delta E/kT}$. **Algorithm:** 1.Evaluate the initial state. 2.Loop until a solution is found or there are no new operators left to be applied: −Set T according to an annealing schedule −Selects and applies a new operator −Evaluate the new state: goal →quit ΔE= Val(current state) −Val(new state) ΔE	[05]	CO1	L1

2 (b)	**Explain problem reduction with respect to AND-OR graphs** Answer: When a problem can be divided into a set of sub problems, where each sub problem can be solved separately and a combination of these will be a solution, AND-OR graphs or AND - OR trees are used for representing the solution. The decomposition of the problem or problem reduction generates AND **arcs**. ***AND-OR GRAPHS*** The AND-OR GRAPH (or tree) is useful for representing the solution of problems that can solved by decomposing them into a set of smaller problems, all of which must then be solved. This decomposition, or reduction, generates arcs that we call AND arcs. One AND arc may point to any number of successor nodes, all of which must be solved in order for the arc to point to a solution. Just as in an OR graph, several arcs may emerge from a single node, indicating a variety of ways in which the original problem might be solved. This is why the structure is called not simply an AND-graph but rather an AND-OR graph (which also happens to be an AND-OR tree) ***EXAMPLE FOR AND-OR GRAPH***  A SIMPLE AND-OR GRAPH	[05]	CO1	L1
3 (a)	**Write an algorithm for : Best-first search with an example** The Best first search uses the concept of a Priority queue and heuristic search. To search the graph space, the BFS method uses two lists for tracking the traversal. An 'Open' list which keeps track of the current 'immediate' nodes available for traversal and 'CLOSED' list that keeps track of the nodes already traversed. Answer: 1. Start with *OPEN* containing just the initial state. 2. Until a goal is found or there are no nodes left on *OPEN* do: (a) Pick them best node on *OPEN.* (b) Generate its successors. (c) For each successor do: (i) If it has not been generated before, evaluate it, add it to *OPEN*, and record its parent. (ii) If it has been generated before, change the parent if this new path is better than the previous one. In that case, update the cost of getting to this node and to any successors that this node may already, have.	[05]	CO1	L1

3 (b) **Discuss about constraint satisfaction and solve the below crypt arithmetic problems:**

CROSS + ROADS =DANGER

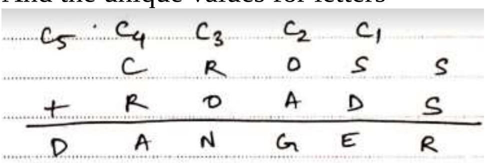
Answer:

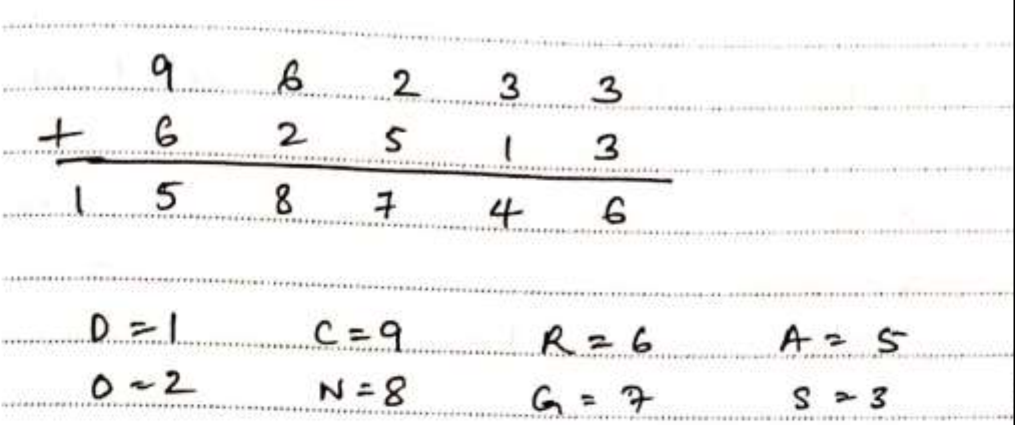
```
   C R O S S
+  R OA D S
 --------------
 DA N G E R
---------------
```

Constraint satisfaction is a search procedure that operates in a space of constraint sets. The initial state contains the constraints that are originally given in the problem description. A Goal State is any state that has been constrained "enough," where "enough" must be defined for each problem.

- Consider the cryptarithmetic problem given above. The goal state is a problem state in which all letters have been assigned a digit in such a way that all the initial constraints are satisfied.
- The solution process proceeds in cycles. At each cycle, two significant things are done (corresponding to steps 1 and 4 of this algorithm):
  - ✓ Constraints are propagated by using rules that correspond to the properties of arithmetic.
  - ✓ A value is guessed for some letter whose value is not yet determined.
- In the first step, it does not usually matter a great deal what order the propagation is done in, since all available propagations will be performed before the step ends.
- In the second step, though, the order in which guesses are tried may have a substantial impact on the degree of search that is necessary.
- A few useful heuristics can help to select the best guess to try first.
- Reasonable approach for this problem would be to store all the constraints in one central database and also to record at each node the changes that must be undone during backtracking.

And the unique values for letters





C=9, R=6, 0=2, S=3, A=5, D=1, N=8, G=7, E=4

4 (a)	**Explain production systems with components and characteristics. List the requirements of good control strategies.** Answer: A production system consists of **four basic components:** a. A set of rules of the form Ci → Ai where Ci is the condition part and Ai is the action part. The condition determines when a given rule is applied, and the action determines what happens when it is applied. b. One or more knowledge databases that contain whatever information is relevant for the given problem. Some parts of the database may be permanent, while others may temporary and only exist during the solution of the current problem. The information in the databases may be structured in any appropriate manner. c. A control strategy that determines the order in which the rules are applied to the database, and provides a way of resolving any conflicts that can arise when several rules match at once. d. A rule applier which is the computational system that implements the control strategy and applies the rules. **Control Strategy should cause Motion** Each rule or strategy applied should cause the motion because if there will be no motion than such control strategy will never lead to a solution. Motion states about the change of state and if a state will not change then there be no movement from an initial state and we would never solve the problem. **Control strategy should be Systematic** Though the strategy applied should create the motion but if do not follow some systematic strategy than we are likely to reach the same state number of times before reaching the solution which increases the number of steps. Taking care of only first strategy we may go through particular useless sequences of operators several times. Control Strategy should be systematic implies a need for global motion (over the course of several steps) as well as for local motion (over the course of single step). Widely used Control Strategies are Breadth-First Search, Depth-First Search, Generate and Test, Hill-Climbing, Best-first search, Problem Reduction	[10]	CO1	L1,L1

5 (a) **What do mean by version space? Discuss the limitations of FindS algorithm over Candidate Elimination Algorithm.** [05] CO 1 L1

Version Space :

Version space have set of hypotheses consistent with the all the training examples.

Definition The version space, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

Limitations of FindS over CEA:

1. Find-S is sensitive to noise that is (almost always) present in training examples.
2. There is no guarantee that h returned by Find-S is the only h that fits the data.
3. Several maximally specific hypotheses may exist that fits the data but, Find-S will output only one.

- Candidate Elimination Algorithm (CEA) addresses several of the limitations of Find-S Algorithm.
- Candidate Elimination Algorithm outputs the set of all hypotheses consistent with the training examples.

(b) **Find a maximally specific hypothesis for the training instances given below.** [05] CO1 L1

Example	Citations	Size	InLibrary	Price	Editions	Buy
1	Some	Small	No	Affordable	One	No
2	Many	Big	No	Expensive	Many	Yes
3	Many	Medium	No	Expensive	Few	Yes
4	Many	Small	No	Affordable	Many	Yes

Answer:

h1=h0={Ø,Ø,Ø,Ø,Ø}
h2={Many,Big,No,Expensive,Many}
h3={Many,?,No, Expensive,?}
h4={ Many,?,No, ?,?}

6 (a) **Apply both FindS and Candidate Elimination Algorithms for the given training examples.** [10] CO1 L1,L1

Size	Color	Shape	Class / Label
Big	Red	Circle	No
Small	Red	Triangle	No
Small	Red	Circle	Yes
Big	Blue	Circle	No
Small	Blue	Circle	Yes

Answer:

Applying FindS Algorithm:

h2=h1= h0={Ø,Ø,Ø}

h4=h3=<Small,Red,Circle}

h5={Small,?,Circle}

Applying Candidate Elimination algorithm:

S0={ Ø,Ø,Ø }

G0={?,?,?}

S1={Ø,Ø,Ø}

G1={<Small,?,?>, <?,Blue,?>, <?,?,Triangle>}

S2={Ø,Ø,Ø}

G2={<Small, Blue,?>, <Small, ?, Circle>, <?,Blue,?>, <Big, ?,Triangle>, <?,Blue,Triangle>}

S3={Small, Red, Circle}

G3={ Small,?,Circle }

S3=S4={ Small, Red, Circle }

G3=G4={ Small,?,Circle }

S5={ Small,?,Circle }

G5={ Small,?,Circle }

S5= G5={ Small,?,Circle }