**Solution IAT-1**
**Dr P. N. Singh, Professor(CSE)**

| Sub: | **ARTIFICIALINTELLIGENCEANDMACHINELEARNING** | Sub Code: | **18CS71** | Branch: | **CSE** |
|---|---|---|---|---|---|

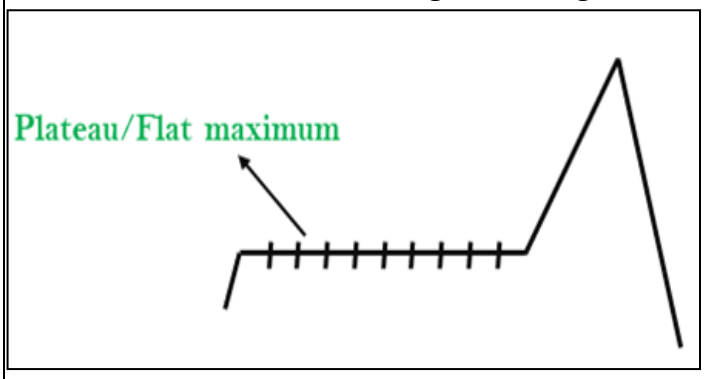
| | **Dr P. N. Singh, Professor(CSE)** | MARKS | CO | RBT |
|---|---|---|---|---|
| **1** | a)Describe simple hill-climbing algorithm.<br>b)Discuss drawbacks of hill climbing algorithms and methods to overcome them.<br>c) Explain the terms local minima and plateau in hill climbing algorithm<br>Ans:<br>a)<br>Algorithm for Simple Hill Climbing:<br>**Step 1:** Evaluate the initial state, if it is goal state then return success and Stop.<br>**Step 2:** Loop Until a solution is found or there is no new operator left to apply.<br>**Step 3:** Select and apply an operator to the current state.<br>**Step 4:** Check new state:<br>If it is goal state, then return success and quit.<br>Else if it is better than the current state then assign new state as a current state.<br>Else if not better than the current state, then return to step2.<br>**Step 5:** Exit.<br><br>b)Drawbacks of Hill- Climbing Algorithm and methods to overcome it<br>• **Local Maximum:** A local maximum is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.<br>**Solution:** Backtracking technique can be a solution of the local maximum in state space landscape. Create a list of the promising path so that the algorithm can backtrack the search space and explore other paths as well.<br><br>• **Plateau:** A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorithm does not find any best direction to move. A hill-climbing search might be lost in the plateau area.<br>**Solution:** The solution for the plateau is to take big steps or very little steps while searching, to solve the problem. Randomly select a state which is far away from the current state so it is possible that the algorithm could find non-plateau region.<br><br>• **Ridges:** A ridge is a special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope, and cannot be reached in a single move.<br>• **Solution:** With the use of bidirectional search, or by moving in different directions, we can improve this problem. | [4+3+3] | CO1 | L1 |

c) Local minima and plateau in hill climbing algorithm

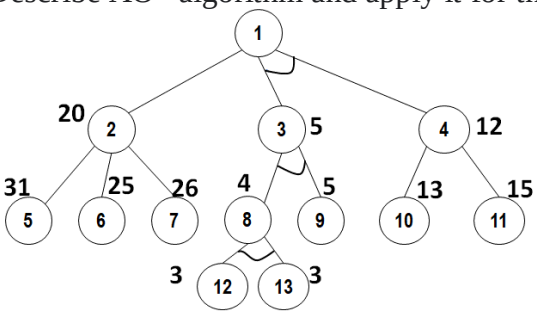
**Local minima / Local Maximul**

Hill climbing will follow the graph from vertex to vertex, always locally increasing (or decreasing) the value of f(x), until a local maximum or local minimum is reached.
This is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.

**Plateau:** A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorithm does not find any best direction to move. A hill-climbing search might be lost in the plateau area.




| | | | | |
|---|---|---|---|---|
| 2. | Describe AO* algorithm and apply it for the following graph and find the final path | [5+5] | CO2 | L3 |




Ans:

AO* Algorithm:

- AO* algorithm is a best first search algorithm.
- AO* algorithm uses the concept of AND-OR graphs to decompose any complex problem given into smaller set of problems which are further solved.
- The AO* algorithm is a knowledge-based search technique, meaning the start state and the goal state is already defined , and the best path is found using heuristics. The time complexity of the algorithm is significantly reduced due to the informed search technique.Compared to the A* algorithm , AO* algorithm is very efficient in searching the AND-OR trees very efficiently.

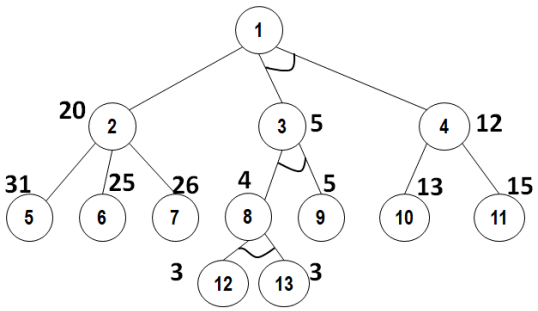
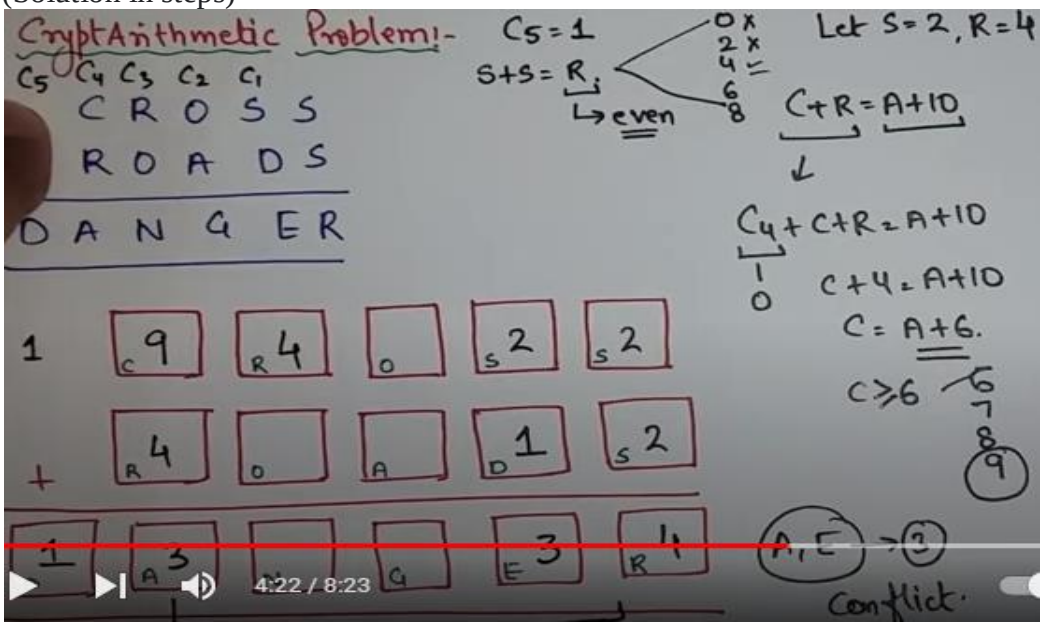
**Working of AO algorithm:**

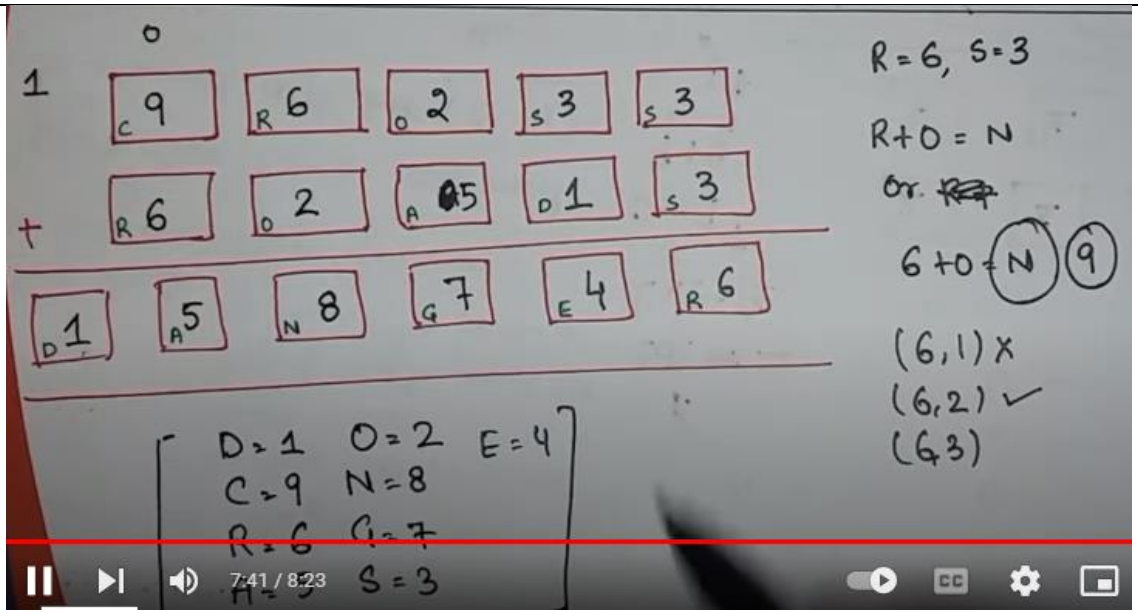
The AO* algorithm works on the formula given below :

**f(n) = g(n) + h(n)**

where,

g(n): The actual cost of traversal from initial state to the current state.

| | | | | |
|---|---|---|---|---|
| | h(n): The estimated cost of traversal from the current state to the goal state.<br>f(n): The actual cost of traversal from the initial state to the goal state<br><br>Now given graph is:<br>  <br><br>Where 1,3,4 and 3,8,9 and 8, 12,13 have AND arcs so, both sides weights to be considered with edge cost for estimating cost and it will be more.<br>Now, left to 1 there is no AND arc and we can go for computing the estimation cost.<br>1-2-6 will be the final path with 25 + 1 + 1 = 27 (Edge cost is considered as 1) | | | |
| **3** | Using constraint satisfaction, solve the cryptarithmetic problem<br>C R O S S<br>+ R O A D S<br>----------<br>D A N G E R<br><br>Ans: (Solution in steps)<br>  <br><br>It produces conflict | 10 | CO2 | L3 |




| | | | | | |
|---|---|---|---|---|---|
| **4** | a) List all production rules for the water jug problem and present a solution<br>b) List all the task domains of AI | 7+3 | CO2 | L2 |

Ans (a) : Production rules for the water jug problem and solution

Problem statement: There are *two jugs, a 4-liter one and a 3-liter one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 liters of water into a 4-liter jug."*

Production rules:

| 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 to fill the four gallon jug |
| 8. | (x,y) | If (x+y)<7 | (x-[3-y],y) | Pour some water from the 4 gallon jug to fill the 3 gallon jug. |
| 9. | (x,y) | If (x+y)<4 | (x+y,0) | Pour all water from 3 gallon jug to the 4 gallon jug |
| 10. | (x,y) | if (x+y)<3 | (0, x+y) | Pour all water from the 4 gallon jug to the 3 gallon jug |

Solution by rules followed:

| S.No. | 4 gallon jug contents | 3 gallon jug contents | Rule followed |
|---|---|---|---|
| 1. | 0 gallon | 0 gallon | Initial state |
| 2. | 0 gallon | 3 gallons | Rule no.2 |
| 3. | 3 gallons | 0 gallon | Rule no. 9 |
| 4. | 3 gallons | 3 gallons | Rule no. 2 |
| 5. | 4 gallons | 2 gallons | Rule no. 7 |
| 6. | 0 gallon | 2 gallons | Rule no. 5 |
| 7. | 2 gallons | 0 gallon | Rule no. 9 |

c) Task domains of AI

**Artificial Intelligence tasks are divided into three groups:**
**Mundane Tasks, Formal Tasks and Expert Tasks.**
**Mundane Tasks:**
Perception
Vision
Speech
Natural Languages
Understanding
Generation
Translation
Common sense reasoning
Robot Control
**Formal Tasks**
Games: chess, checkers, etc
Mathematics: Geometry, logic, Proving properties of programs
**Expert Tasks:**
Engineering ( Design, Fault finding, Manufacturing planning)
Scientific Analysis
Medical Diagnosis
Financial Analysis

**5** a) Write various knowledge representation issues. 6+4 CO1 L1
b) What are the properties of a good system for knowledge representation?

Ans:

a) various knowledge representation issues

The main objective of knowledge representation is to draw the conclusions from the knowledge, but there are many issues associated with the use of knowledge representation techniques.

**1. Important attributes**

**2. Relationships among attributes**
**3. Choosing the granularity of representation**
**4. Representing sets of objects.**
**5. Finding the right structure as needed**

b) Properties of a good system for knowledge representation

A good system for the representation of knowledge in a particular domain should possess the following four properties:

Representational Adequacy — the ability to represent all of the kinds of knowledge that are needed in that domain.

- Inferential Adequacy — the ability to manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old.
- Inferential Efficiency — the ability to incorporate into the knowledge structure additional information that can be used to focus the attention of the inference mecha- nisms in the most promising directions.
- Acquisitional Efficiency — the ability to acquire new information easily. The simplest case involves direct insertion, by a person, of new knowledge into the database. Ideally, the program itself would be able to control knowledge acquisition.

Unfortunately, no single system that optimizes all of the capabilities for all kinds of knowledge has yet been found. As a result, multiple techniques for knowledge representation exist. Many programs rely on more than one technique. In the chapters that follow, the most important of these techniques are described in detail. But in this section, we provide a simple, example-based introduction to the important ideas.

**6** — 5+5 — CO2 — L3

a) Explain concept learning task with an example
b) Describe Find-S algorithm and apply it on the dataset to arrive at a maximally specific hypothesis. The target attribute is "smart".

| *hair* | *body* | *likesSimon* | *pose* | *smile* | *smart* |
|---|---|---|---|---|---|
| blond | thin | yes | arrogant | toothy | no |
| brown | thin | no | natural | pleasant | yes |
| blond | plump | yes | goofy | pleasant | no |
| black | thin | no | arrogant | none | no |
| blond | plump | no | natural | toothy | yes |

Ans:

a)Concept Learning task:

In Machine Learning, concept learning can be termed as "*a problem of searching through a predefined space of potential hypothesis for the hypothesis that best fits the training examples" – Tom Mitchell.*

Most of human learning is based on past instances or experiences. For example, we are able to identify any type of vehicle based on a certain set of features like make, model, etc., that are defined over a large set of features.

These special features differentiate the set of cars, trucks, etc from the larger set of vehicles. These features that define the set of cars, trucks, etc are known as concepts. Similar to this, machines can also learn from concepts to identify whether an object belongs to a specific category or not. Any algorithm that supports concept learning requires the following:

- Training Data
- Target Concept
- Actual Data Objects

**General Hypothesis**

Hypothesis, in general, is an explanation for something. The general hypothesis basically states the general relationship between the major variables. For example, a general hypothesis for ordering food would be *I want a burger.*

G = { '?', '?', '?', …..'?'}

**Specific Hypothesis**

The specific hypothesis fills in all the important details about the variables given in the general hypothesis. The more specific details into the example given above would be *I want a cheeseburger with a chicken pepperoni filling with a lot of lettuce.*
S = {'Φ','Φ','Φ', ……,'Φ'}

**b) Find-S Algorithm**

1. Initialize 'h' to the most specific hypothesis.
2. The Find-S algorithm only considers the positive examples and eliminates negative examples. For each positive example, the algorithm checks for each attribute in the example. If the attribute value is the same as the hypothesis value, the algorithm moves on without any changes. But if the attribute value is different than the hypothesis value, the algorithm changes it to '?'.
   The process starts with initializing 'h' with the most specific hypothesis, generally, it is the first positive example in the data set.

1. We check for each positive example. If the example is negative, we will move on to the next example but if it is a positive example we will consider it for the next step.
2. We will check if each attribute in the example is equal to the hypothesis value.
3. If the value matches, then no changes are made.
4. If the value does not match, the value is changed to '?'.
5. We do this until we reach the last positive example in the data set.

Given dataset

| *hair* | *body* | *likesSimon* | *pose* | *smile* | *smart* |
|---|---|---|---|---|---|
| blond | thin | yes | arrogant | toothy | no |
| brown | thin | no | natural | pleasant | yes |
| blond | plump | yes | goofy | pleasant | no |
| black | thin | no | arrogant | none | no |
| blond | plump | no | natural | toothy | yes |
| **?** | **?** | **no** | **natural** | **?** | |

**Straight forward logic: Not matching then ?**
**Matching then replace with same**

2nd and 5th row having yes of target Attribute (Positive)
We will check the value of other attributes for similarity:
So final answer according to Find-S algorithm:
S = [ '?' , '?' , ' no' , 'normal' , '?' ]