INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SRI CITY

TERM-II EXAMINATION – SPRING 2024

Artificial Intelligence

CSE: UG4 (PC) **Date: 14-03-2024**

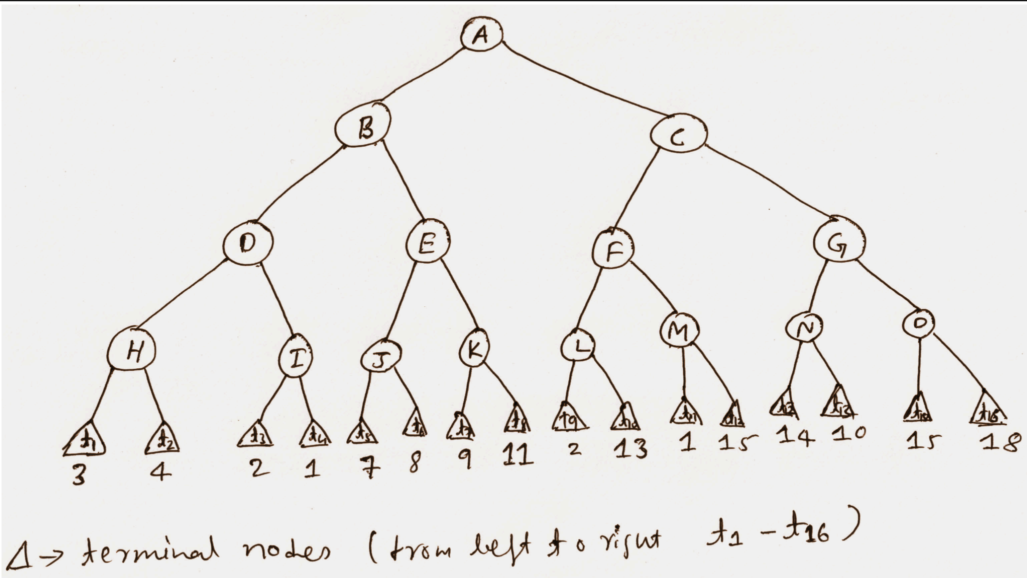
Duration: 90 Mins (03:30-05:00 PM) **Max. Marks: 25**

Instructions: **Roll No:**________________

1. This is a **closed-book exam**. You can use **a calculator** if necessary.
2. **Please Write/Draw legibly**! If we can't understand what you have written, we can't grade it.
3. **Don't use Pencils** for answering/drawing. The final answer **must** be in blue or black ink.
4. Mention the question number before the answer.
5. ***It is Mandatory to read all the notes/instructions given in the question and answer accordingly. Any irrelevant answers will not be evaluated and awarded with zero marks.***

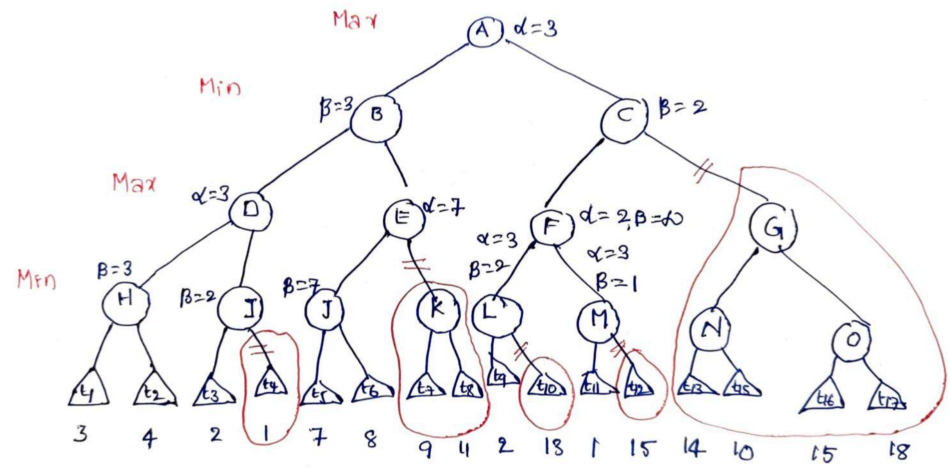
Section-A

1	**a.** Wumpus world is an episodic environment [True / False]. **False** **b.** Wumpus world is not a fully observable environment [True / False]. **True**	**[2 Mark]**
	c. Which of the following equivalences is true? (a) $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$ (b) $\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$ (c) $\neg(P \rightarrow Q) \Leftrightarrow P \wedge \neg Q$ (d) all of the above. **(d) All the above**	**[3 Marks]**
	d. The statement $(\neg p) \Rightarrow (\neg q)$ is logically equivalent to which of the statements below? I. $p \Rightarrow q$ II. $q \Rightarrow p$ III. $(\neg q) \vee p$ IV. $(\neg p) \vee q$ **II and III**	**[2 Marks]**
2	For a two-player game, a decision tree with utility values is given below. Find the best move for each player using the alpha-beta pruning technique. **Note**: *Write each step in the text mentioning the order of traversal with node name, determination of alpha-beta values based on the decision, and pruned node name due to the algorithm.* You may draw a single tree to show the alpha-beta values a	**[8 Marks]**



and the pruned nodes.

However, an explanation for each node is mandatory. Marks will be awarded based on the explanation and justification, the tree diagram will be used for cross reference only.



3 Solve the following Cryptarithmetic Problem. Show each step. **[4 Marks]**

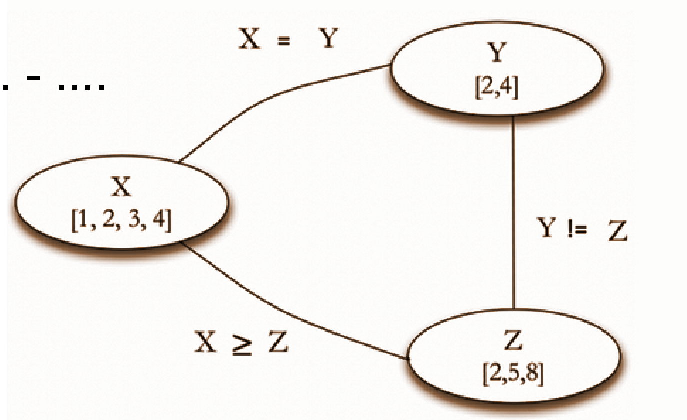
TWO
+TWO
FOUR

7 solutions.

ORWUTF = 483671

```
  TWO    734
+ TWO    734
 FOUR   1468
```

	ORWUTF = 506371 TWO 765 + TWO 765 FOUR 1530 ORWUTF = 623781 TWO 836 + TWO 836 FOUR 1672 ORWUTF = 624981 TWO 846 + TWO 846 FOUR 1692 ORWUTF = 746381 TWO 867 + TWO 867 FOUR 1734 ORWUTF = 862591 TWO 928 + TWO 928 FOUR 1856 ORWUTF = 863791 TWO 938 + TWO 938 FOUR 1876 Any of the above answers will get 4 Marks.	
4	Solve the following constraint graph using AC-3 Algorithm.	**[6 Marks]**



Variables: {X, Y, Z}
Domain values are given inside the node. for e.g. X={1,2,3,4}
Constraints are given at the edges of the graph. For e.g. {Y!=Z}

Note:

a) *In the First step, you have to specify all the variables and constraints and write all the arcs in the initial state of the queue in alphabetical order (like dictionary-based) only.* ***(1M)***
Indicative example: Queue={(A!=B);(A!=D);(A>E);(B=A),..}

b) *You must specify all the arcs in the queue w.r.t. previous steps, queue status, and domain sets for each variable at each step.* ***(5 M)***

c) Marks will be appropriately deducted for not following the above instructions.

d) No partial marks will be given for the remaining steps if note (a) is not followed in the first step.

e) No partial marks will be given for the problem if the queue and domain set are not written for each step.

Answer:

Evaluation Instructions:

1) *In the First step, students have to clearly specify all the variables and constraints and write all the arcs in the initial state of the queue in alphabetical order only.* ***(1 Mark)***

2) *You must clearly specify all the arcs in the queue w.r.t. previous steps, queue status and domain sets for each variable at each step.* ***(Each Step 0.25 Marks)***

3) No partial marks will be given for the remaining steps if step 1 is not correct.

4) Marks will be appropriately deducted for incorrect steps.

Key Answer:

Step 1: (1 Mark)

Variables:	**{X, Y, Z}**
Constraints:	**X=Y; X>=Z; Y!=Z;**
Arcs:	**X=Y; X>=Z; Y=X; Y!=Z; Z<=X; Z!=Y**

Queue (in Ascending order):
X=Y **X>=Z** **Y=X** **Y!=Z** **Z<=X** **Z!=Y**

-. - - - . - - .. - ....

Variables	Domain Values
X	{1,2,3,4}
Y	{2,4}
Z	{2,5,8}

Step 2: Removed (X=Y) (0.5 Marks)

Queue (in Ascending order):
~~**X=Y**~~ **X>=Z** **Y=X** **Y!=Z** **Z<=X** **Z!=Y**

Variables	Domain Values
X	{1,2,3,4}
Y	{2,4}
Z	{2,5,8}

Change in domain value of X= {2, 4}
Value 1 and 3 deleted
The queue is updated with all arcs (except with Y) in which X is on RHS {(Z<=X)}

Step 3: Removed (X>=Z) (0.5 Marks)

Queue (in Ascending order):
~~**X=Y**~~ ~~**X>=Z**~~ **Y=X** **Y!=Z** **Z<=X** **Z!=Y** **Z<=X**

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2,5,8}

Comments: No Change in the domain value of X

Step 4: Removed (Y=X) (0.5 Marks)

Queue (in Ascending order):
~~**X=Y**~~ ~~**X>=Z**~~ ~~**Y=X**~~ **Y!=Z** **Z<=X** **Z!=Y** **Z<=X**

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2,5,8}

Comments: No Change in the domain value of Y

Step 5: Removed (Y!=Z) (0.5 Marks)

Queue (in Ascending order):
~~**X=Y**~~

~~X>=Z~~
~~Y=X~~
~~Y!=Z~~
Z<=X
Z!=Y
Z<=X

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2,5,8}

Comments: No Change

Step 6: Removed (Z<=X) (0.5 Marks)

Queue (in Ascending order):
~~X=Y~~ ~~X>=Z~~ ~~Y=X~~ ~~Y!=Z~~ ~~Z<=X~~ **Z!=Y** **Z<=X**

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2,5,8}

Comments: Change in domain value of Z= {2}
Value 5 & 8 are deleted
The queue is updated with all arcs (except with X) in which Z is on RHS {(Y!=Z)}

Step 7: Removed (Z!=Y) (0.5 Marks)

Queue (in Ascending order):
~~X=Y~~ ~~X>=Z~~ ~~Y=X~~ ~~Y!=Z~~ ~~Z<=X~~

~~**Z!=Y**~~
Z<=X
Y!=Z

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2}

-. -   - - . - - .. - ....

Comments: No Change in domain value

Step 7: Removed (Z<=X) (0.5 Marks)

Queue (in Ascending order):

~~**X=Y**~~
~~**X>=Z**~~
~~**Y=X**~~
~~**Y!=Z**~~
~~**Z<=X**~~
~~**Z!=Y**~~
~~**Z<=X**~~
Y!=Z

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2}

Comments: No Change in domain value

Step 8: Removed Y!=Z (0.5 Marks)

Queue (in Ascending order):

~~**X=Y**~~
~~**X>=Z**~~
~~**Y=X**~~
~~**Y!=Z**~~
~~**Z<=X**~~
~~**Z!=Y**~~
~~**Z<=X**~~
~~**Y!=Z**~~

Variables	Domain Values
X	{2,4}
Y	{2,4}
Z	{2}

Comments: Change in domain value of Y={4}
Value 2 is deleted
The queue is updated with all arcs (except with Z) in which Y is on RHS {(X=Y)}

Step 9: Removed X=Y (0.5 Marks)

Queue (in Ascending order):
~~X=Y~~ ~~X>=Z~~ ~~Y=X~~ ~~Y!=Z~~ ~~Z<=X~~ ~~Z!=Y~~ ~~Y=X~~ ~~Z!=Y~~ ~~X=Y~~

-. - - - . - - .. - ....

Variables	Domain Values
X	{2,4}
Y	{4}
Z	{2}

Comments: Change in domain value of X={4}
Value 2 is deleted
The queue is updated with all arcs (except with Y) in which X is on RHS {(Z<=X)}

Step 10: Removed Z<=X (0.5 Marks)

Queue (in Ascending order):
~~X=Y~~ ~~X>=Z~~ ~~Y=X~~ ~~Y!=Z~~ ~~Z<=X~~ ~~Z!=Y~~ ~~Y=X~~ ~~Z!=Y~~ ~~X=Y~~

~~Z<=X~~

Variables	Domain Values
X	{4}
Y	{4}
Z	{2}

Comments: No Change in domain value

The queue is empty hence the algorithm stops. Final Domain Values are

X	{4}
Y	{4}
Z	{2}