Indian Institute of Information Technology, Sri City, Chittoor

Name of the Exam: **AI Set: 4** Duration: 90 mins Max Marks for Written Exam: 30

Read the Instructions before proceeding:

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. Submit the answer for each question **separately** (**as two different PDFs**) via **google classroom**
5. Duration: **70 mins for written** exam + **20 mins for objective** questions. The objective questions will be distributed separately (as quiz) via google classroom.

Question 1: *[Total: 15 Marks] has two parts A and B*

Our institute IIITS has invited some experts to deliver lectures on various fields of computer science for a one day workshop. You have been asked to create the schedule during the workshop.Unfortunately, there are only four time slots available **(3pm - 6pm).** Each lecture is to be scheduled for **1 Hour**.
List of the Speakers from area of:

- Artificial Intelligence (AI)
- Computer Organization (CO)
- Machine Learning (ML)
- Theoretical Computer Science (TC)
- Data Mining (DM)
- Algorithms Design (AD)
- Soft Computing (SC)

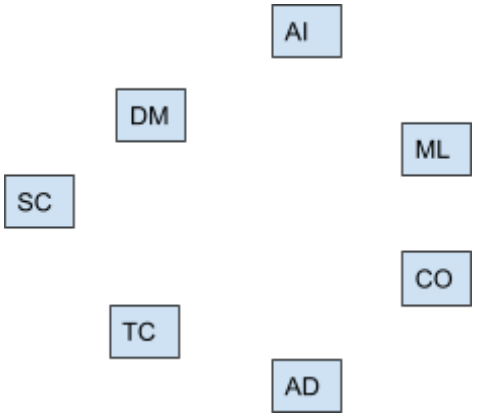
Lectures **can happen parallely** if they don't violate any of the below constraints:
1. The **ML** expert can only be assigned to the 6 pm slot as he has some work and will join late.
2. The Mechanical Engineering students want to attend the Experts: **SC**, **AI**, and **CO**.
(Hint: This corresponds to the constraint **SC ≠ AI ≠ CO**)
3. The Electrical Engineering students want to attend the Experts: **TC**, **AD**, and **CO**.
4. The Faculty Members want to attend the lectures of: **DM** and **AD**.
5. The visiting faculty want to attend the lectures of speakers: **AI** and **TC**.
6. The Computer Science students want to see the British speakers: **ML, AI**, and **CO**.
7. Finally, you decide that you will be happy if and only if you get to attend both **AI** and **AD** lectures. (Yes, even if you belong to one or more of the groups above.)

Domain of the problem

Variable	Domain values
AI	3 4 5 6
CO	3 4 5 6
ML	6
TC	3 4 5 6
DM	3 4 5 6
AD	3 4 5 6
SC	3 4 5 6

Part A:
Draw the **constraint graph** for the above problem.

- Represent all binary constraints as **edges**
- Represent all ternary(and above) constraints by introducing **square-box nodes** as necessary
- You MUST use the **below layout** (i.e don't change where the nodes are located and just add the constraints as edges, etc.)



Part B:
Show the final tree resulting from a **backtracking-search traversal.** Use the following

- Use **minimum remaining value (MRV)** heuristic for unassigned-variable selection
- Use **degree heuristic** for tie-breaking, when MRV alone is not sufficient
  (Hint: Using the constraint graph above will help you with degree heuristic computation)
- Use **Forward checking (FC)** during the traversal.

Hints:

1. We have purposefully excluded Least Constraining Value heuristic. LCV can be very difficult+confusing to compute when you are drawing the traversal manually. So, don't do LCV.
2. We have not fully tested whether a solution exists for all possible constraints. Hence, we are asking you to show only the traversal. Please remember that a solution **may or may not exist**.In either case, you should show the traversal tree.
3. We recommend **landscape orientation** for both the constraint graph and the traversal tree.
4. Use as many rough sheets as necessary for trying things out. The **final submitted version** of the tree should be clean (without any rough work or scribbles).

Question 2: *[Total: 15 Marks] has two parts Part A and part B*

Part A Prove by Resolution (Refutation proof) [8 Marks]
You can show the proof as text or a diagram(tree). Either way we expect a clean legible proof. An example resolution-step as text could look like this:

F11: A(x) ∨ B(y) resolves with F12: ¬A(x) ∨ C(y) to produce F13: B(y) ∨ C(y) by substitution {x/C1, y/C2}

As shown above, even though it is redundant, we want you to specify the fact-number along with the fact. This will help us give you marks, even when you make minor mistakes. If you insist on using a diagram, please show the **path** and **substitutions** clearly. You may use landscape layout, if you decide that you need more space for the diagram. **Please don't change the fact number when you convert facts to CNF**.

Important: You must always start your resolution from Fact 1 (F1)

a) Prove by resolution **Hardworker (<*your-name*>).** (use your first-name here) *[2 Marks]*

Knowledge Base
F1: □x UndergradIIITSStudent (x) => StudentIIITS(x)
F2: □x StudentIIITS(x) => Hardworker(x)
F3: UndergradIIITSStudent(<your-name>)

b) Prove by resolution **Older(Jim, Don)** *[2 Marks]*

Knowledge Base
F1: □x □y (Superior(x,y) ∧ Alive(x)) => Older(x,y)
F2: □x □y Officer(x,y) => Superior(x,y)
F3: Alive(Jim)
F4: Officer(Jim, Don)

c) Prove by resolution **GymMates (John, Jake)** *[2 Marks]*

Knowledge Base
F1: □x □y □x (GymInstructor(z,x) ∧ GymInstructor(z,y)) => GymMates(x,y)
F2: □x □y Trainer(x,y) => GymInstructor(x,y)
F3: Trainer(Stan, Jake)
F4: Trainer(Stan, John)

d) Prove by resolution **Expensive(Souffle)** *[2 Marks]*

Knowledge Base
F1: □x Loves(x, John) => Food(x)
F2: FrenchRecipe(Souffle)
F3: Loves(Souffle, John)
F4: □x Food(x) ∧ FrenchRecipe(x) => Expensive(x)

Part B: Rete Algorithm [7 Marks]

a) Please construct the Rete Network corresponding to the given rule-base [3 Marks]

a) Please show the final state of the alpha storage table(s) and beta storage table(s) after the facts F1 to F4 has been added (in the same order). i.e., the state of the table after the forward chaining cycle stops. [4 Marks]

Instructions (read very carefully):

- The Rete network diagram will not fit in a single A4 sheet, if you draw it in portrait mode. Instead, you MUST use **landscape** orientation only.
- For both alpha and beta storages you should show time-steps. For example, (say) one of your rules contains a pattern '(mother ?x ?y)' then typically you will have an alpha storage for that pattern with two columns: ?x and ?y. Instead, **please use three columns where the first column is 'T' for time-step**. (say) The very first fact we are processing is '(mother Sara Nura)'. Then the first row in the corresponding alpha storage table is (T1, Sara, Nura).
- When a rule fires, it might assert **a new fact** into the KB. Please **show the new facts/assertions along with time steps**. Example: (say) there is a rule R1: mother(?x, ?y) => parent(?x, ?y). After processing the first fact 'mother(Sara, Nura)' in time-step T1, we get a new assertion 'parent(Sara, Nura)' . (say) we already have 5 facts in the KB. Then the new assertion should be recorded as ***F6: parent(Sara, Nura) (By R1 in T1)***
- All new facts are added/processed only after all the other facts in the queue are processed. For example, that fact Missile(M1) will fire rule Missile(X) => Weapon(X) and leads to the addition of a fact Weapon(M1). This will be F6 aka fact number 6 i.e. **do not process the new facts before the other** 5 facts.

Rules
R1: American(x) ∧ Weapon(x) ∧ Sells(x,y,z) ∧ Hostile(z) => Criminal(x)
R2: Missile(x) ∧ Owns(Nono, x) => Sells(West, x, Nono)
R3: Missile(x) => Weapon(x)
R4: Enemy(x, America) => Hostile(x)
Facts
F1: American(West)
F2: Owns(Nono, M1)
F3: Missile(M1)
F4: Enemy(Nono, America)