University of M'sila
Faculty of Maths and Computer Science
Computer Science Department

May 12, 2026
M1 IDO
Duration : 2h

Final Exam
Constraint Logic Programming

Exercise 1 (6 pts)

A university has 3 courses (Math, Computer Science, Physics) and 3 classrooms numbered 1 to 3 (capacity 30, 50, 80 students). Assign one classroom per course such that: each classroom hosts at most one course, Computer Science gets a room with capacity ≥ 50, and Physics does not use room 1.

تضم إحدى الجامعات ثلاثة تخصصات (الرياضيات، وعلوم الحاسوب، والفيزياء) وثلاث قاعات دراسية مرقمة من 1 إلى 3 (سعتها 30، 50، 80 طالبًا على التوالي). خصص قاعة دراسية لكل تخصص بحيث: تستضيف كل قاعة دراسية تخصصًا واحدًا على الأكثر، ويحصل تخصص علوم الحاسوب على قاعة تتسع لـ 50 طالبًا أو أكثر، ولا يستخدم تخصص الفيزياء القاعة رقم 1.

1- Give the CSP of this problem.
2- Write a Gnu-Prolog program to solve this problem.

Exercise 2 (8 pts)

Consider this CLP program.

```
schedule(T1, T2, T3) :-
    fd_domain([T1, T2, T3],1,5),
    T2 #>= T1 + 2,
    T3 #>= T2 + 1,
    T3 #=< 5,
    fd_labeling([T1, T2, T3]).
```

1- What problem does this program model? Describe it in concrete terms.
2- What are the variables and their domains?
3- Express each constraint in natural language. What do the constants 2 and 1 represent?
4- Is the constraint T3 #=< 5 redundant given the domain 1..5? Explain.
5- Enumerate all solutions manually, then state how many there are.

1- ما المشكلة التي يمثلها هذا البرنامج؟ صفها بعبارات محددة.
2- ما هي المتغيرات ومجالاتها؟
3- عبّر عن كل قيد بلغة طبيعية. ماذا يمثل الثابتان 2 و1؟
4- هل القيد T3 #=< 5 زائدٌ بالنظر إلى المجال 1..5؟ اشرح.
5- اذكر جميع الحلول يدويًا، ثم حدد عددها.

Exercise 3 (6 pts)

A project distributes a budget of 10 units across 3 departments (X, Y, Z), each receiving between 1 and 5 units. Department Y must receive strictly more than X. Department Z must receive at least as much as Y.
A student presented a program for this problem, but it contains some mistakes (bugs).

يوزع مشروع ميزانية قدرها 10 وحدات على 3 أقسام (X، Y، Z)، حيث يحصل كل قسم على ما بين وحدة واحدة و5 وحدات. يجب أن يحصل القسم Y على ميزانية أكبر من القسم X، بينما يجب أن يحصل القسم Z على ميزانية لا تقل عن ميزانية القسم Y.
قدّم أحد الطلاب برنامجًا لحل هذه المسألة، ولكنه يحتوي على بعض الأخطاء البرمجية.

```
allocate(X, Y, Z) :-
    fd_domain([X, Y, Z],0,5),
    X + Y + Z #= 9,
    Y #> X,
    Z #> Y,
    fd_labeling([X, Y]).
```

1- The domain starts at 0. Does this agree with the problem? What should the minimum be?
2- The total budget constraint is X + Y + Z #= 9. Is the constant correct? What effect does this error have on solutions?
3- The constraint Z #> Y, does this match the problem statement? What operator should be used?
4- In the `fd_labeling` command, are all variables included? What is the consequence of the omission?.
5- Write the corrected program, and give a solution that satisfies all constraints.

1- يبدأ المجال من الصفر. هل يتوافق هذا مع المسألة؟ ما هي القيمة الدنيا؟
2- قيد الميزانية الإجمالية هو X + Y + Z # = 9. هل الثابت صحيح؟ ما تأثير هذا الخطأ على الحلول؟
3- القيد Z #> Y ، هل يتطابق مع نص المسألة؟ ما هو المعامل الذي يجب استخدامه؟
4- في الأمر fd_labeling، هل جميع المتغيرات مُضمنة؟ ما هي نتيجة حذف أي منها؟
5- اكتب البرنامج المُصحح، وقدّم حلاً يُحقق جميع القيود.

ملاحظة: الإجابة يجب أن تكون باللغة الانجليزية، تم تعريب الأسئلة من أجل المساعدة على فهمها فقط.

University of M'sila
Faculty of Maths and Computer Science
Computer Science Department

January 19th, 2025
M1 AI
Duration : 1h30

Final Examen
Constraint Logic Programming
Solution

Exercise 1 (6 pts):

1- CSP Formulation **(3 pts)**

Variables: RoomMath, RoomCS, RoomPhys
Domains: {1, 2, 3} for each variable
Constraints: all_different, RoomCS ≥ 2, RoomPhys ≠ 1

2- Gnu-Prolog **(3 pts)**

```
solve_rooms(RoomMath, RoomCS, RoomPhys) :-
    % Step 1: Declare domains
    fd_domain ([RoomMath, RoomCS, RoomPhys,1,3),

    % Step 2: Constraints
    all_different([RoomMath, RoomCS, RoomPhys]), % one course per room
    RoomCS #>= 2,                                % CS needs large room
    RoomPhys #\= 1,                              % Physics not room 1

    % Step 3: Search
    fd_labeling([RoomMath, RoomCS, RoomPhys]).
```

Exercise 2 (8 pts):

1 — Problem **(1 pt)**

A task scheduling problem: three tasks (T1, T2, T3) must be scheduled in time slots (for example Hours) 1–5, with mandatory gaps between them — T2 cannot start until at least 2 slots after T1, and T3 must follow T2 by at least 1 slot, all finishing by slot 5.

2 — Variables & domains **(2 pts)**

Variables: `T1`, `T2`, `T3` (start times). Domain for each: {1, 2, 3, 4, 5}.

3 — Constraints **(2 pts)**

`T2 #>= T1 + 2` — T2 starts at least 2 slots after T1 (minimum gap of 2, e.g. a duration or cool-down)
`T3 #>= T2 + 1` — T3 starts at least 1 slot after T2 (minimum gap of 1)
`T3 #=< 5` — T3 must not exceed the planning horizon (slot 5)

4 — Redundancy **(1 pt)**

It is NOT redundant. The domain `1..5` only bounds each variable individually. Without `T3 #=< 5`, the combined effect of the precedence constraints could still allow T3=5, but it makes the intent explicit and helps the propagator prune T3's domain early without waiting for T1 and T2 to be fixed.

5 — Solutions 5 **(2 pts)**

T1=1, T2=3, T3=4
T1=1, T2=3, T3=5
T1=1, T2=4, T3=5
T1=2, T2=4, T3=5
4 solutions total.

Exercise 3 (6 pts)

1 - domain lower bound too low **(1 pt)**

Domain is `0..5`, allowing 0. The problem says each department receives **at least 1** unit.

2 - wrong budget total **(1 pt)**

The sum constraint uses `9` instead of **10**. This silently shifts all solutions downward, yielding allocations that do not match the actual budget.

3 - wrong comparison operator **(1 pt)**

`Z #> Y` means "strictly greater than", but the problem says Z must receive **at least as much** as Y (i.e., ≥). This eliminates valid solutions where Z = Y.

4 – `fd_labeling ( [X, Y] )` omits Z. **(1 pt)**

Z remains an uninstantiated constrained variable — the query returns a partial answer where Z has a domain but no concrete value.

5 – The corrected program **(1 pt)**

```
allocate(X, Y, Z) :-
    fd_domain([X, Y, Z],1,5),
    X + Y + Z #= 10,
    Y #> X,
    Z #>= Y,
    fd_labeling([X, Y, Z]).
```

Example solution: X=1, Y=4, Z=5 → 1+4+5=10 ✓ 4>1 ✓ 5≥4 ✓ **(1 pt)**