The University of Melbourne
Department of Computer Science and Software Engineering
Semester 2 Assessment 2004

433-671 Constraint Programming

Time Allowed: 3 Hours
Reading Time: 15 minutes

Authorized materials: Books and calculators are not permitted.

Instructions to Invigilators: One 14 page script. Exam paper may leave the room.

Instructions to students:
This exam counts for 70% of your final grade. There are 7 pages and 6 questions for a total of 70 marks. Attempt to answer all of the questions. Values are indicated for each question and subquestion — be careful to allocate your time according to the value of each question.

Question 1 [9 marks]

The N queens problem is to place N queens on an N by N chessboard so that none is on the same horizontal, diagonal or vertical line.

R_1	R_2	R_3
B_{11}	B_{12}	B_{13}
B_{21}	B_{22}	B_{23}
B_{31}	B_{32}	B_{33}

(a) You can define a model for the N queens problem using $N \times N$ Boolean variables B_{ij} representing if there is a queen in the i^{th} row and j^{th} column position,

Give the Boolean model for the 3 queens problem. (Its a bit big, dont be afraid to use shorthand notation, e.g. $\wedge_{i=1}^{3} \wedge_{j=1}^{3} B_{ij} = B_{ji}$. This example of course is wrong!) [2 marks].

(b) You can define a model for the N queens problem using N integer variables R_i ranging from 1 to N representing the row in which the queen in the ith column is found.

Give an integer model for the 3 queens problem. [2 marks].

(c) Which of the two models would you expect to find solutions faster (when extended to the arbitrary N case—the case for 3 is rather uninteresting). Justify your answer. [2 marks].

(d) You can combine the models of parts (a) and (b) into one model.

Describe how would you do this, and the possible benefits that arise. [2 marks].

(e) In fact a much stronger model for the N queens problem can be constructed using `alldifferent` constraints.

Explain how. [1 marks].

Question 2 [13 marks]

(a) Explain how a solver that maintains node, arc and hyper-arc (or domain) consistency will treat the constraint

$$A = max(B, C) \wedge 2A = 3C \wedge A \geq 1 \wedge B \leq 6$$

given the initial domains are [0..10]. Give domains for A, B, and C after any changes, and explain why the change was made. [4 marks].

(b) Explain how a solver that maintains bounds consistency will treat the constraint

$$A = \max(B, C) \wedge 2A = 3C \wedge A \geq 1 \wedge B \leq 6$$

Give domains for A, B, and C after any changes, and explain why the change was made. You can assume the propagation rules for $A = \max(B, C)$ are

$$\begin{array}{l} A \leq \max(max_D B, max_D C) \\ A \geq \max(min_D B, min_D C) \\ B \leq max_D A \\ C \leq max_D A \end{array}$$

[4 marks].

(c) Give bounds propagation rules for the primitive constraint

$$X \leq Y + Z,$$

where X,Y and Z take integer values. [2 marks].

(d) Recall that a domain D is bounds consistent with respect to a constraint c if for each variable $v \in vars(c)$ there is a real solution θ of c where $v = min_D v$ and the other variables v' take values d' where $min_D v' \leq d' \leq max_D v'$. The propagation rules for $A = \max(B, C)$ above do not maintain bounds consitency. Consider the domain $D(A) = [7..10]$, $D(B) = [1..4]$, $D(C) = [5..8]$. Then there is no solution θ where C takes the value 5. Improve the bounds propagation rules for max to enforce bounds consistency. [3 marks].

Question 3 [10 marks]

Let P be the following SICStus `clpr` program: Consider the following program meant to define the Euclidean distance $D = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$ between two points $(X1, Y1)$ and $(X2, Y2)$:

```
dist(X1,Y1,X2,Y2,D) :-
   sqr(X1-X2,D1),
   sqr(Y1-Y2,D2),
   sqr(D, D1+D2),
   { D >= 0 }.
sqr(X,Y) :- { X = 0, Y = 0 }.
sqr(X,Y) :- { Y > 0, Y = X * X }.

nat(X) :- { X = 1 }.
nat(X) :- { X = X1 + 1 }, nat(X1).
```

(a) Give a simplified successful derivation for `dist(1,3,5,6,D)`. [3 marks.]

(b) Give a simplified finitely failed for `dist(1,X,5,6,0)`. [2 marks.]

(c) The goal {`D >= 100`}, `dist(1,3,5,6,D)` does not fail as expected since the solver delays non-linear primitive constraints until they are known to be linear. Add solver redundant constraints to the program so that the above goal finite fails. Ensure that all successful derivations for the original version are still successful in the new version. [3 marks].

(d) Even with the fixes for the previous question the goal `dist(1,3,5,7,D), nat(D)` runs forever. Add redundant constraints to the program to make it finitely fail. Again ensure that all successful derivations for the original program still succeed in the modified one. [2 marks].

Question 4 [11 marks]

A restaurant selling pasta can make its own pasta and or buy from outside sources. The production planning problem is to determine what is the optimal mix of inside production and buying to maximize profit.

Assume three products: spaghetti, fettucine and kluski, and an estimated demand given in the table below. To produce one unit of each they need a certain quantity of flour and eggs. Only 20 units of flour and 40 units of eggs can be stored in the restaurant. The costs to produce or buy each product are given in the table.

	spaghetti	fettucine	kluski
demand	100	200	300
flour	0.5	0.4	0.3
eggs	0.2	0.4	0.6
produce cost	0.6	0.8	0.3
buy cost	0.8	0.9	0.4

Assume you can only produce or buy whole units of each product.

(a) Write a mathematical model of the problem to determine the best strategy for the restaurant. Explain the meaning of each problem variable. [5 marks].

(b) Express the model to determine the best strategy for the restaurant as an OPL program.

Example OPL code for for finding a colouring of a graph of Europe with minimal colors has the form

```
enum Country {Belgium,France,Holland,Luxembourg};
enum Colors {red,blue,yellow};
var Colors color[Country];
var int used[Colors] in 0..1;
```

```
minimize sum (c in Colors) used[c]
subject to {
      forall (c in Colors & l in Country) color[l] = c => used[c] = 1;
      color[France] <> color[Belgium];
      color[France] <> color[Luxembourg];
      color[Luxembourg] <> color[Belgium];
      color[Belgium] <> color[Holland];
}
```

[4 marks].

(c) Which constraint solver: an integer linear programming solver or finite domain propagation solver would be best to tackle your model. Justify your choice. [2 marks].

Question 5 [11 marks]

A magic square of size n is an $n \times n$ matrix containing each of the numbers from 1 to n^2 such that the sum of each column and row and the two major diagonals each add up to the same number. An example of a magic square of size 3 is

2	7	6
9	5	1
4	3	8

(a) Write a SICstus Prolog program defining the predicate `magic(N,List)` which constraints *List* to be a list of N lists of length N such that they make up a magic square of size N. The goal `magic(3,[[A,B,C],[D,E,F],[G,H,I]])` should be able to succeed finding the solution `A = 2`, `B = 7`, `C = 6`, `D = 9`, `E = 5`, `F = 1`, `G = 4`, `H = 3`, `I = 8` illustrated above.

You can make use of the following predicate definitions which `transpose` a matrix represented as a list of lists, and `delfirstcol` which breaks a list of lists into the list of the first elements, and the list of the rest of the lists.

```
transpose([], []).
transpose(M, [C |T]) :-
   delfirstcol(M, C, M1),
   transpose(M1, T).
delfirstcol([],[],[]).
delfirstcol([[A]|T], [A|R], []):-
   delfirstcol(T, R, []).
delfirstcol([[A | B] | T], [A | R], [B | T1]):-
   delfirstcol(T, R, T1).
```

For example `transpose([[2,7,6],[9,5,1],[4,3,8]], T)` returns `T = [[2,9,4],[7,5,3],[4,3,8]]` while `delfirstcol([[2,7,6],[9,5,1],[4,3,8]], F, R)` returns `F = [2,9,4]`, `R = [[7,6],[5,1],[3,8]]`. [6 marks].

(b) Discuss possible labelling strategies for tackling the magic squares problem. Which would you expect to be best. Justify your answer. [3 marks].

(c) The sum of all the numbers from 1 to n^2 is $n^2(1+n^2)/2$. For the 3×3 case this is 45. So that means that the sum of each column and row (and also diagonal) must be $1/n$ of this or $n(1+n^2)/2$. For the the 3 case this is 15.

Add redundant constraints to your model taking into account this reasoning. Explain how you would expect this to affect the search. [2 marks].

Question 6 [16 marks]

A simple constraint domain $\mathcal{D}$ over a finite set of qualitiative intervals only allows primitive constraints of the forms shown in Figure **??** The six different constraints are: before,

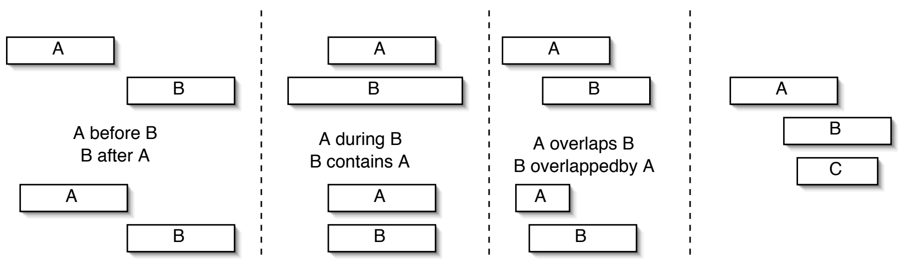

Figure 1: 6 different qualitative relations between intervals, and an example solution

after, during, contains, overlaps, and overlappedby. The diagram gives a two of examples for each.

- A before B and B after A hold if A finishes before (or at the same time) as B starts.
- A during B and B contains A hold if A starts after (or at the same time) as B starts and finishes before (or at the same time) as B finishes.
- A overlaps B and B overlappedby A hold if A starts strictly before B starts and finishes strictly before B finishes.

An example constraint is A overlaps $B \wedge B$ contains $C \wedge C$ overlappedby A. A valuation in $\mathcal{D}$ is a solution if under the valuation each primitive constraint is satisfied. Valuations for these domains can either be specified diagramatically or by pairs (start time, end time). For example the valuation shown in Figure **??** equates to the pairs $\{A \mapsto (0,8), B \mapsto (4,12), C \mapsto (6,10)\}$ and is a solution of the constraint above.

(a) Give a diagrammatic or pairs solution of each of the following constraints if one exists, or state it is unsatisfiable.

(*i*) A before $B \wedge A$ overlaps $C \wedge C$ during $D \wedge D$ before $B \wedge D$ after $A \wedge E$ contains A
(*ii*) A contains $E \wedge E$ before $B \wedge B$ overlappedby $A \wedge B$ overlaps $C \wedge C$ contains E

[2 marks].

(b) Define a function $compat(A\ R_1\ B, A\ R_2\ B)$ which returns *true* if $A\ R_1\ B \wedge A\ R_2\ B$ is satisfiable, and $false$ otherwise. [2 marks].

(c) A core to any constraint solving algorithm for qualitative intervals is a transitivity function for relations. That is given $A\ R_1\ B$ and $B\ R_2\ C$ then $tran(A\ R_1\ B, B\ R_2\ C)$ should return the set of compatible relations $A\ R\ C$, that is where $A\ R_1\ B \wedge B\ R_2\ C \wedge A\ R\ C$ is satisfiable. For example $tran(A \text{ before } B, B \text{ before } C) = \{A \text{ before } C\}$. while $tran(A \text{ contains } B, B \text{ before } C) = \{A \text{ contains } C, A \text{ before } C, A \text{ overlaps } C\}$. Give the values for $tran(A \text{ overlaps } B, B\ R\ C)$ for each R. [5 marks].

(d) Assuming you are given *compat* and *trans* functions defined in the previous parts, define as *complete* as possible constraint solving algorithm *qisolv* for a constraint in $\mathcal{D}$. [5 marks].

(e) Assuming each interval is represented by a pair of integer variables give SICstus `clpfd` code for each constraint, e.g. `overlaps(A,B)` should constrain the interval representing A to overlap B. You should give 6 predicate definitions. [3 marks].