The University of Melbourne
Semester 2 Assessment 2008

Department: Computer Science and Software Engineering

Subject Number: 433-433 and 433-633

Subject Title: Constraint Programming

Exam Duration: 3 hours

Reading Time: 15 minutes

This paper has 5 pages, including this front page.

Authorized materials:
No calculators or computers are allowed. No books or notes are permitted.

Instructions to Students:
This examination counts for 70% of your final grade. There are 5 pages and 5 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. Make sure that your answers are *readable*. Any unreadable parts will have to be considered wrong. For programming questions, marks will generally be allocated not only for correctness of code, but also for simplicity and elegance. You will not be penalised for minor syntactic errors (you may use standard mathematical notation such as $\forall$, $\exists$, and $\sum$ to specify constraints in your models).

Question 1 [13 marks]

(a). You are given the constraint

$$y \neq z + 3 \wedge 2x = 3z \wedge x \geq 2 \wedge z \leq 7 \wedge y = x + z$$

with initial domains for all variables being 1..10. Show the variable domains after successively applying first node, then arc, then domain consistency, explaining why each change is made. [4 marks]

(b). You are given the constraint

$$w = x + 2y + 3z$$

with initial domains for all variables being 1..10. Show the variable domains after applying bounds consistency, explaining how the changes are made. [4 marks]

(c). A bounds propagation rule is an inequality between a variable and (some function of) the upper and lower bounds of other variables. For example, the bounds propation rules for the primitive constraint $x < y$ are $x < ub(y)$ and $y > lb(x)$. Give the bounds propagation rules for the primitive constraint

$$x = \min(y, z)$$

[3 marks]

(d). Describe the advantages and disadvantages of bounds consistency over domain consistency. [2 marks]

Question 2 [13 marks]

A magic square is a square matrix of different numbers with the property that the rows, columns, and major diagonals all sum to the same value. In this question, we consider 3x3 magic squares, such as:

8	3	4
1	5	9
6	7	2

(a). Give the variable declarations and constraints for a MiniZinc model for a 3x3 magic square where each element in the square has domain 1..9. [4 marks]

(b). The elements of an $n \times n$ magic square sum to $n^2(n^2+1)/2$. It follows that each row, column, (and major diagonal) must therefore sum to $n(n^2+1)/2$. Give constraints to add to your MiniZinc model which would take advantage of this fact. How would you expect this extra information to affect the search for a solution? [3 marks]

(c). Magic squares have several kinds of symmetry. For example, mirroring or rotating a magic square also gives you a magic square. Give symmetry breaking constraints to add to your MiniZinc model which would take advantage of this fact. How would you expect this extra information to affect the search for a solution? [4 marks]

(d). Discuss possible search strategies for this problem. Explain which you would expect to do best and why. [2 marks]

Question 3 [12 marks]

(a). Add slack variables to convert the following linear programming problem into simplex form:

$$\text{Minimize } x - y \text{ subject to } \begin{array}{rcl} x & \geq & 1 \\ x & \leq & 3 \\ 2y & \leq & x + 3 \\ y & \geq & 0 \end{array}$$

[2 marks]

(b). What does it mean for a problem to be in basic feasible solved (BFS) form? Your explanation should include an explanation of the basis, basic variables, parametric variables, and the objective function. [3 marks]

(c). *Describe* how the above simplex form can be converted into BFS. (You are *not* required to actually convert your simplex form into BFS). [3 marks]

(d). Show the application of the simplex pivoting algorithm to solve the following problem:

$$\text{Minimize } 1 + s_1 - y \text{ subject to } \begin{array}{rcl} x & = & 1 + s_1 \\ s_2 & = & 2 - s_2 \\ s_3 & = & 4 - 2y + s_1 \\ x, y, s_1, s_2, s_3 & \geq & 0 \end{array}$$

Show the optimal value of the objective function and the corresponding values for x and y. [4 marks]

Question 4 [16 marks]

A building project has several jobs to be completed. Each job consists of a number of tasks to be performed in order, each of which requires the services of a particular tradesman (carpenter, electrician, plasterer, plumber) for a given number of hours. An example list of tasks presented in MiniZinc is as follows:

```
int: T = 13;              % The number of tasks.
int: carpenter = 1;       % Useful constants.
int: electrician = 2;
int: plasterer = 3;
int: plumber = 4;
% Each row gives a job number, tradesman, and duration in hours.
array [1..T, 1..3] of int: tasks =
    [| 1, carpenter,   3   | 1, electrician, 2
     | 1, plasterer,   4   | 2, carpenter,   5
     | 2, plumber,     2   | 2, electrician, 4
     | 2, plasterer,   6   | 3, plumber,     1
     | 3, plasterer,   2   | 3, electrician, 1
     | 4, carpenter,   7   | 4, electrician, 3
     | 4, carpenter,   4
     |];
```

(a). The building project employs one of each of the four kinds of tradesman. The work must be completed within 48 hours. Define the variables and constraints of a MiniZinc model to solve the problem, assuming the existence of a predicate $\texttt{nonoverlap}(As, Ad, Bs, Bd) = (As + Ad \leq Bs) \vee (Bs + Bd \leq As)$ which constrains the intervals A (start time As, duration Ad) and B (start time Bs, duration Bd) to not overlap. Note that if two tasks have the same job number, then the lowest numbered task must be completed before the other task can begin. [5 marks]

(b). A cumulative constraint $\texttt{cumulative}(n, s, d, r, R, start, end)$ specifies that for all tasks $i \in 1..n$ where task i starts at time $s[i]$ and runs for duration $d[i]$ consuming r resources per unit time, tasks cannot be scheduled such that more than R resources are consumed in total per unit time in the interval *start..end*. Give a MiniZinc definition of a predicate implementing the cumulative constraint. (Recall that you can introduce local variables in a predicate with a let-expression: `let {var decls} in expr`.) [4 marks]

(c). Show how the building project model can be reformulated using the cumulative constraint. [3 marks]

(d). The building project hires an extra electrician. Show the constraint you would write to allow use of the new worker in your model. [2 marks]

(e). Building regulations require that work only be conducted between 8a.m. and 8p.m. and that individual tasks cannot be broken across different days. Show the constraints you would use to model this condition. [2 marks]

Question 5 [16 marks]

(a). Give pseudocode for the DPLL algorithm used to implement Boolean SAT solvers. [3 marks]

(b). Assuming variables are labelled in the order a, b, c, d, e, f, trying to make each variable true first, show the search tree generated by the DPLL algorithm solving the following problem:

$$\{\neg a, \neg b, c\}$$
$$\{\neg a, \neg b, \neg c\}$$
$$\{\neg a, b, c\}$$
$$\{a, d\}$$
$$\{\neg a, b, \neg c\}$$
$$\{\neg b, c\}$$
$$\{\neg b, \neg c\}$$
$$\{b, c, \neg d\}$$
$$\{e, \neg a, f\}$$
$$\{\neg f, \neg b\}$$

Be sure to make it clear which labellings are obtained from pure literal elimination and unit propagation. [3 marks]

(c). Give a brief description of two highly effective optimizations that can be applied to the DPLL algorithm. [2 marks]

(d). One way to represent an integer variable $X \in 1..n$ in a SAT problem is as a collection of Boolean variables: x_k, $x_{>k}$, and $x_{<k}$ (for $k \in 1..n$) where $x_k \iff X = k$, $x_{>k} \iff X > k$, and $x_{<k} \iff X < k$. Give the logical formulae and corresponding clauses expressing the relationships between the following:

- $x_{>k}$ and $x_{>k+1}$
- $x_{<k}$ and x_k
- $x_{<k+1}$ and $x_{>k}$
- x_k, $x_{>k-1}$, and $x_{<k+1}$.

[4 marks]

(e). Explain how you would express the following constraints in a Boolean SAT problem, assuming all variables have domain $1..n$ and use the representation just described:

- $Y \leq X$
- $X = 2Y$
- $X = Y + Z$

[3 marks]

(f). Describe a key disadvantage of Boolean SAT solvers compared to FD solvers. [1 mark]