Constraint Technology (course 1DL023)
Autumn 2007 – Assignment 2

Pierre Flener

Department of Information Technology, Uppsala University, Sweden

— Hard deadline: 08:00 am on Monday 15 October 2007 —

Submission Instructions

1. This assignment is to be solved in teams of at most two students. Take our warning on plagiarism very seriously. *We assume that by submitting a solution you are certifying that it is solely the work of your team, and that everyone on the team contributed to elaborating this solution.*

2. Write clear answers. Justify all answers, except where explicitly not required. State any assumptions you make.

3. Document each program by mathematically describing its model and any relevant features, such as its branching heuristics, search technique, and any special data structures, as well as by giving instructions on how to compile and run it. Include sample test runs with inputs and outputs and check whether these test runs are reproducible by the programs you submit. Verify whether you are using version 1.0.1 of Gecode/J.

4. Put your answer files into a folder named *Surname.Firstname–A2* or $Surname_1.Firstname_1$–$Surname_2.Firstname_2$–*A2*. Do not use any special characters in the folder name. Include only source code files, that is remove any executable files. Textual answers must be in *.pdf* or *.txt* format. Include a *ReadMe.txt* file to explain the purpose of each other file. Comply strictly with any answer filenames imposed by the questions. Make a *.tgz* or *.zip* compressed archive of maximum 1MB from this folder. *Verify whether it decompresses properly, reproducing your folder exactly, and actually corresponds to Assignment 2.*

5. Submit your compressed folder via the Course Manager server (whose clock may differ from yours) by the deadline given above. This deadline is hard: *No* exceptions will be made. *No* other method of submission will be accepted. Late solutions will be penalised by 2 points for each 24h of delay, but solutions that are late by more than 48h will get 0 (zero) points.

6. When working in a team, *each* team member must submit a solution. Only one of the submitted solutions of a team will be graded, so make sure they are identical. The lateness penalty, if any, for a team will be determined by the moment of its *last* submitted solution.

Failure to follow the instructions above may result in 0 (zero) points, as we reserve the right to process your solutions mechanically.

Question 1: Event Sets ($3 \cdot 1 = 3$ points)

Consider the following propagators defined on stores $s : V \to 2^U$ for a set V of decision variables over a common universe U:

$$p_1(s) = \left\{ \begin{array}{l} x \mapsto \{x \mid x \in s(x) \wedge \exists y \in s(y) \text{ such that } 3 \cdot x = y\} \\ y \mapsto \{y \mid y \in s(y) \wedge \exists x \in s(x) \text{ such that } 3 \cdot x = y\} \end{array} \right\}$$

$$p_2(s) = \left\{ \begin{array}{l} x \mapsto \{x \mid x \in s(x) \wedge \exists y \in s(y) \text{ such that } 3 \cdot x = y\} \\ y \mapsto \{y \mid y \in s(y) \wedge 3 \cdot \min(x) \leq y \leq 3 \cdot \max(x)\} \end{array} \right\}$$

$$p_3(s) = \left\{ \begin{array}{ll} \left(\begin{array}{l} x \mapsto s(x) \\ y \mapsto s(y) \cap \{1,2\} \end{array} \right), & \text{if } s(x) = \{1\} \\ \left(\begin{array}{l} x \mapsto s(x) \\ y \mapsto s(y) \cap \{3,4\} \end{array} \right), & \text{if } s(x) = \{2\} \\ \left(\begin{array}{l} x \mapsto s(x) \\ y \mapsto s(y) \cap \{1,2,3,4\} \end{array} \right), & \text{otherwise} \end{array} \right\}$$

Determine the event set for each of these propagators that invokes the considered propagator the least number of times. Motivate your answer.

Question 2: Implementing $\max(x, y) = z$ ($3 + 4 = 7$ points)

Consider the $\max(x, y) = z$ constraint, which holds if and only if integer decision variable z holds the maximum value of integer decision variables x and y. Answer the following sub-questions:

a. Define a propagator for $\max(x, y) = z$ that takes subsumption into account. Is your propagator idempotent? In which situations can your propagator be rewritten into a simpler propagator?

b. Implement your propagator in Gecode/J. Use the code skeleton at `http://www.it.uu.se/research/group/astra/gecode/Max.java` and just fill in the missing parts (marked by TODO comments) after consulting the tutorial on propagator implementation at `http://www.it.uu.se/research/group/astra/gecode/implProp.pdf`.