Exercise 1

Foundations of Constraint Programming

Prof. Michael Thielscher, Sebastian Voigt

International Master Program in Computational Logic — winter term 2009/2010

Date of exercise: 28.10.2009

Exercise 1.1

Consider the task of assigning to each node of a finite graph a color in such a way that no two adjacent nodes have the same color. Such an assignment is called a coloring of the graph. A coloring of the graph involving the minimal number of colors is called the chromatic number of the graph. Formulate the problem of finding the chromatic number of a graph as a constrained optimization problem.

Exercise 1.2

Consider the following Boolean constraints (see also Slide 22 in Chapter 2):

$$
\begin{aligned}
&i_1 \wedge o_2 = y_1 \\
&i_2 \wedge o_1 = y_2 \\
&\neg y_1 = o_1 \\
&\neg y_2 = o_2
\end{aligned}
$$

For the above constraints show a successful derivation using the Boolean constraint propagation rules given on Slide 23. For each derivation step you should underline the selected constraint and give the used rule. The initial CSP is:

$$\langle i_1 \wedge o_2 = y_1, i_2 \wedge o_1 = y_2, \neg y_1 = o_1, \neg y_2 = o_2 \ ; \ o_2 = 1, i_1 = 1 \rangle$$

Exercise 1.3

Consider the CSP from Slide 33 in Chapter 2:

$$\langle x \cdot y = z; x \in [1..20], y \in [9..11], z \in [155..161] \rangle$$

Transform this CSP using the three Multiplication Rules from Slide 32 until no further rule application is possible. Give the selected constraint and the used rule for each derivation step.

Exercise 1.4

Download and install the open source Prolog version Eclipse-Prolog. It can be found together with detailed documentation at the url `http://www.eclipse-clp.org/`. After installation you can start Eclipse-Prolog using the command `eclipse` and load a file *filename.pl* via `compile(filename).`

a) Write a program in Eclipse-Prolog that solves the N Queens problem (see Slide 12 in Chapter 1). Use the hybrid integer/real interval arithmetic constraint solving library `ic` (load it with `:- lib(ic).`). Use the built-in constraint propagation so that your program is able to find all solutions of an 8-queens problem almost instantaneously.

b) Compare the run time of your program to the time needed to find one solution of the 8-queens problem using a brute force strategy without constraint propagation.

Hints:

- Constraints are given using the # operator, so for example, `N1 #\= N2` implies that number `N1` is unequal to number `N2`.
- Have a look at the predicate `#::/2` for assigning integer intervals to a list of variables.
- Additionally the predicates `labeling/1` and `findall/3` might be useful.