Constraint Technology for Solving Combinatorial Problems – Assignment 2

Justin Pearson and Magnus Ågren

Computing Science, Information Technology, Uppsala University

{justin,agren}@it.uu.se

— To be handed in before midnight June 25.[1] —

1 General Instructions

1. For all questions:
   (a) This assignment should preferably be done in pairs of two.
   (b) Discuss your results where this is relevant.
   (c) A *hard-copy* of your submission should be handed in to Magnus Ågren *before* the deadline. Please put it in his mailbox (number 55) on the fourth floor.
   (d) A *soft-copy* of your submission should also be sent by email to `agren@csd.uu.se`. Add assignment number and your names to the subject line. If your submission consists of several files, please tar or zip them together before submission.[2]
   (e) If you need a deadline extension, contact `agren@csd.uu.se` *before* the deadline.
2. For programming questions:
   (a) Submit a listing of your program(s).
   (b) Each question should be accompanied by documentation describing how your program works. This includes describing the model and any particular relevant features, e.g. heuristics used, search technique(s), special data structures (if any), etc. Don't forget to explain how to compile and run your program.
   (c) Include sample test runs of your programs with input and output. Please make sure that the test runs are reproducible by the programs you hand in.

2 Need Help?

Consult Magnus for questions 5 and 6. Consult Justin for all other questions.

[1] This means that I will empty my mailbox on the morning of June 28.

[2] You do not have to hand in parts of the assignment that you do using pen and paper electronically. (E.g. drawings of search trees.)

3 Consistency

Consider the following CSPs

- $P_1 = \langle x < y, y < z + 1, x + w > 5; w \in \{0, \ldots, 10\}, x \in \{0, \ldots, 10\}, y \in \{0, \ldots, 10\}, z \in \{0, \ldots, 10\}\rangle$
- $P_2 = \langle x + y < z, y + w < 8; w \in \{0, \ldots, 10\}, x \in \{0, \ldots, 10\}, y \in \{0, \ldots, 10\}, z \in \{0, \ldots, 10\}\rangle$
- $P_3 = \langle x + y < 3, x + z > 1; x \in \{1, 2\}, y \in \{1, 2\}, z \in \{0, 2\}\rangle$

Question 1

After making P_1 *arc consistent*, what will the domains of its variables be?

Question 2

After making P_2 *hyper-arc consistent*, what will the domains of its variables be?

Question 3

Again, after making P_3 *arc consistent*, what will the domains of its variables be?

4 Constraint Graphs and Widths

Consider the CSPs

- $P_1 = \langle\{x > y\ ,\ x + z = y\ ,\ w > y\ ,\ x \neq w\}; \mathcal{DE}\rangle$
- $P_2 = \langle\{w + y = 10, y < z, x + w = 4, u + v = z, x + y + z = 1\}; \mathcal{DE}\rangle$

Question 4

Draw the constraint graphs of P_1 and P_2.

Question 5

What is the width of P_1 with respect to the ordering $w \prec x \prec y \prec z$? **Justify your answer!**

Question 6

What is the width of P_1? **Justify your answer!**

Question 7

What is the width of P_2? **Justify your answer!**

5 *prop* Labeling Trees

Consider the CSP $P = \langle\mathcal{C}; \mathcal{DE}\rangle$ where

- $\mathcal{C} = \{w > x, x < z, y \neq z\}$, and
- $\mathcal{DE} = w \in \{1, 2, 3\}, x \in \{2, 3\}, y \in \{2, 3\}, z \in \{2, 3\}$,

Question 8

Draw the *forward checking* search tree for P.

Question 9

Draw the *partial look ahead* search tree for P with respect to the ordering $w \prec x \prec y \prec z$.

Question 10

Draw the *full look ahead (MAC)* search tree for P.

6 Implementing Forward Checking

Consider the usual model of the NQueens problem:

- $\forall i < j \in \{1, \ldots, n\} : q_i \neq q_j$
- $\forall i < j \in \{1, \ldots, n\} : q_i - i \neq q_j - j$
- $\forall i < j \in \{1, \ldots, n\} : q_i - j \neq q_j - i$

where $q_i \in \{1, \ldots, n\}, 1 \leq i \leq n$, denotes the row of the queen placed in column i. Your task is to implement a backtrack search algorithm performing forward checking that finds one solution to the NQueens problem for a given n. **You are not allowed to use any solver library for this question!** Instead, you should implement a backtrack search algorithm (e.g. similar to the one in the book) that labels each variable until they are all assigned. To do this, you will probably need to represent the variables and their current domains in some way. You will also need to represent the $x \neq y$ constraints and what values are consistent in y given an instantiation of x and vice versa.

– **Hints** –

- Start by implementing the Backtrack search algorithm (Fig. 8.20 in the book).
- When this works for small n, extend that algorithm to the Backtrack with Constraint Propagation search algorithm (Fig. 8.21 in the book).
- Keep everything simple. Don't try to implement a general solver but stick to one that only solves the NQueens problem. (Or, more generally, problems that are expressible using a set of constraints on the form $x \neq y$ and $x + c \neq y + d$.)

7 More Consistency (Optional)

Consider constraints of the form $\langle c_1 \leq x - y \leq c_2; x \in D_x, y \in D_x \rangle$ where c_1 and c_2 are given constants.

Question 11 (Optional)

First work out the propagation rules of the form:

$$\frac{\langle c_1 \leq x - y \leq c_2; x \in D_x, y \in D_y \rangle}{\langle c_1 \leq x - y \leq c_2; x \in D'_x, y \in D'_x \rangle}$$

to make the constraints arc-consistent.

Question 12 (Optional)

Now given three constraints:

$$\langle c_1 \leq x - y \leq c_2; x \in D_x, y \in D_y \rangle$$

$$\langle c_3 \leq y - z \leq c_4; y \in D_y, z \in D_z \rangle$$

and

$$\langle c_5 \leq z - x \leq c_6; z \in D_z, x \in D_x \rangle$$

work out new values of the constants $c_1, \ldots, c_6$ such that the three constraints will be path-consistent. (**Hint:** If you have trouble then first work on path consistency on concrete examples).

Question 13 (Optional)

For constraints of the above form, it can be proved that path consistency implies global consistency. Sketch an algorithm that decides if a set of these constraints has a solution or not.