

Foundations of Constraint Programming

Prof. Steffen Hölldobler, Sebastian Haufe

International Master Program in Computational Logic — winter term 2010/2011

Date of exercise: 26.11.2010

Exercise 4.1

To the following CSP, apply the HYPER-ARC CONSISTENCY rule (cf. Slide 4/10) until it is closed under this rule. Indicate the chosen constraint at each step.

$$\langle x < y, x \neq z, y - x = z; x \in [1..4], y \in [1..4], z \in [2..3] \rangle$$

Exercise 4.2

Consider the following two CSPs with variables x, y, z, w :

- 1) $\langle x \neq y, z = x \cdot y; x \in \{a, b\}, y \in \{b, c\}, z \in \{bb, cc\} \rangle$, where \cdot is the string concatenation
- 2) $\langle x \neq 10, x = y + 1, \text{all_different}(x, y, z), x + y + z = w;$
 $x \in [10 \dots 13], y \in [10 \dots 12], z \in [10 \dots 12], w \in [30 \dots 32] \rangle$

Are these CSPs consistent, node consistent, arc consistent, directionally arc consistent wrt. some linear ordering \prec , hyper-arc consistent, path consistent, directionally path consistent?

Exercise 4.3

Prove the Note on Slide IV/15: A normalized CSP is path consistent iff for each subsequence x, y, z of its variables

$$C_{x,y} \subseteq C_{x,z} \cdot C_{y,z}^T, \quad C_{x,z} \subseteq C_{x,y} \cdot C_{y,z}, \quad C_{y,z} \subseteq C_{x,y}^T \cdot C_{x,z}.$$

Exercise 4.4

The goal of the following exercise is to construct an ASP encoding which solves the following *Einstein Puzzle*: There are four different persons: Marc, Joey, Sandra and Ellen. Each person likes exactly one of the sports hiking, volleyball, basketball or tennis and exactly one of the drinks tea, water, coffee or beer. The favorite sport and drink of each person differs from those of the respective other persons. Moreover you have the following clues:

- 1) Joey drinks beer.
- 2) Marc likes neither tea nor volleyball.
- 3) Either Sandra goes hiking or Joey plays basketball.
- 4) The coffee drinker plays basketball.
- 5) Ellen plays basketball if Sandra likes tea.
- 6) The water drinker plays tennis or volleyball.

Use Eclipse-Prolog together with the constraint solving library `ic` to write a program which solves the problem.

Hint: Use numbers 1,2,3,4 to encode the names of the persons and variables *Tea*, *Coffee*, *Hiking*, etc. which range over these numbers.