Foundations of Constraint Programming
Tutorial 5 (on December 16th)

Lukas Schweizer

WS 2019/20

Exercise 5.1:
Consider the following CSP P:

$$\langle x < y; x \in [7..15], y \in [9..12] \rangle$$

Show in detail how to apply Corollary 1 (slide 33, lecture 4) to prove that P is consistent.

Exercise 5.2:
The following boolean constraints define a digital circuit:

$$y_1 = x_1 \oplus x_2, y_2 = x_2 \oplus x_3, y_3 = x_3 \oplus x_4, y_4 = x_4$$

The following CSPs are instances of the given circuit, where

$$\langle y_1 = x_1 \oplus x_2, y_2 = x_2 \oplus x_3, y_3 = x_3 \oplus x_4, y_4 = x_4; x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1 \rangle$$
$$\langle y_1 = x_1 \oplus x_2, y_2 = x_2 \oplus x_3, y_3 = x_3 \oplus x_4, y_4 = x_4; x_2 = 1, y_1 = 1, y_3 = 1, y_4 = 1 \rangle$$

a) Draw the digital circuit, where inputs are x_1, x_2, x_3 and x_4 and outputs are y_1, y_2, y_3 and y_4.

b) Show how to compute a successful derivation for the given instances yielding the values for all eight variables; at each step underline the selected constraint and give the used rule.

Hint: Use the XOR rules on slide 11 (lecture 5) or define alternative rules.

Extra Exercise 5.3:
Formulate the 2D-BinPacking Problem as constraint optimization problem. N rectangular items each with a (probably different) given height and width have to be packed into rectangular bins all of the same size $W \times H$. It can be assumed that the items are sorted according to non-increasing height. The goal is to minimize the number of bins needed to pack all items (the natural upper bound therefore is N – each item into one bin).

Exercise 5.4:
Abstract argumentation frameworks allow to represent and solve conflicting knowledge. They consist of a set of abstract arguments and a binary relation between them, denoting attacks. The inherent conflicts are solved on a semantical level by selecting sets of arguments which are *acceptable* together.

More formally, an *argumentation framework (AF)* is a pair $F = (A, R)$ where A is a set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ means that a attacks b. We say that an argument $a \in A$ is *defended* (in F) by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, there exists a $c \in S$ such that $(c, b) \in R$.

An argumentation framework can be represented as a directed graph. Let $F = (A, R)$ be an AF with $A = \{a, b, c, d, e\}$ and $R = \{(a, b), (b, c), (c, b), (d, c), (d, e), (e, e)\}$. The corresponding graph representation is depicted in Fig. 1.

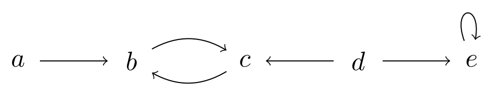

Figure 1: Example argumentation framework

Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is *conflict-free (in F)*, if there are no $a, b \in S$, such that $(a, b) \in R$. $cf(F)$ denotes the collection of conflict-free sets of F. For a conflict-free set $S \in cf(F)$, it holds that

- S is a *stable extension*, i.e. $S \in stable(F)$, if each $a \notin S$ is attacked by S;
- S is an *admissible set*, i.e. $S \in adm(F)$, if each $a \in S$ is defended by S;
- S is a *complete extension* (of F), i.e. $S \in comp(F)$, if $S \in adm(F)$ and for each $a \in A$ defended by S (in F), $a \in S$ holds.

We want to compute all extensions of a given semantics (stable, admissible or complete). Let $F = (A, R)$ be an AF, formulate for each semantics the associated CSP, such that the solutions of the CSP correspond to the extensions of the AF F.