Constraint Programming
Prof. Dr. Thom Frühwirth, Marc Meister

assignment #8 (winter term 2005)
solutions will be presented Tuesday, 20-Dec-2005, 2 PM, o27/2203
`http://www.informatik.uni-ulm.de/pm/index.php?id=112`

Exercise 1 (Definite clause programming)**.**
Write a CHR-solver to implement the resolution calculus for ground clauses. To this end, a disjunctive clause like $x \vee \neg w \vee \neg z$ is written as a constraint `cl([+x,-w,-z])`.
Write (auxiliary) routines to translate a definite clause program (consisting of facts and rules) together with a query into disjunctive clausal form and apply the constraint solver.
Note: The query is entailed by the definite clause program whenever the solver fails.

Exercise 2 (SAT-solver)**.**
Write a CHR-solver to implement the *Davis-Putnam* procedure for SAT problems:
We *chase* a literal x (i.e. we consider x to be true) by removing all clauses with contain x and eliminating $\neg x$ in all (remaining) clauses. A definite clause triggers a chase.
As the SAT-problem is NP-complete, we (have to) implement search, chasing first x, then $\neg x$, for all atoms.

Constraintsystem B
Download `boole.pl` from the lecture web page. It contains implementations for the constraints `neg/2`, `and/3`, `or/3`, `xor/3`, and `imp/2` of the Boolean Algebra. Use this constraint-solver for the following exercises.

Exercise 3 (Equivalence)**.**

Extend `boole.pl` with rules (similar to the ones already defined) in order to cope with equivalence, i.e. implement simplifcations for a CHR-constraint `equiv(X,Y,Z)` which obey the given truth table.

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

Exercise 4 (Who lies?)**.**

Lehmann says Mueller lies.
Mueller says Schulze does not tell the truth.
Schulze says both lie.

Write a Prolog-predicate `tellTruth(Lehmann,Mueller,Schulze)` which succeeeds iff the three arguments are a valid interpretation of the given statements by Lehmann, Mueller, and Schulze. Use Boolean junctors constraints `and`, `neg`, ...
Hint: Lehman's statement can be modelled by `Lehmann=MuellerLies`, or using equivalence, with `MuellerLies` being the negation of `Mueller`.

Exercise 5 (Cross Circuit)**.**
A cross ciruit exchanges two wires/signals with the help of a logic circuit without crossing them physically. For the input pins (X, Y) and the output pins (A, B) we have $A = Y$ and $B = X$.
Write a CHR constraint `cross(X,Y,A,B)`, which implements a cross ciruit by means of Boolean constraints.
Test with queries `cross(1,0,A,B)`, `cross(1,Y,1,B)` and `cross(0,Y,A,B)`.