Universität Ulm
Fakultät für Informatik
Prof. Dr. Thom Frühwirth

Constraint-Programmierung, WS 02/03
Übungsblatt 9

Abgabe bis Mi, 15.01.03, 10:00

Constraintsystem BOOL

Aufgabe 9-1

Es soll ein Constraintsystem für Boolesche Algebra entworfen werden. In der Datei `bool.pl` der Email finden Sie bereits Implementierungen für die Constraints `neg/2`, `and/3`, `or/3`, `xor/3` und `imp/2`. Ergänzen Sie das Programm in analoger Weise um die Regeln, die die Äquivalenz definieren. D.h. es soll die Vereinfachungen für ein CHR-Constraint `equiv(X,Y,Z)` implementiert werden, das folgende Wahrheitstabelle befolgt:

X	Y	$Z = X \equiv Y$
0	0	1
0	1	0
1	0	0
1	1	1

Aufgabe 9-2

Eine Kreuzschaltung (engl. cross circuit) vertauscht zwei Leitungen mithilfe einer logischen Schaltung, ohne sie physikalisch zu überkreuzen: Dabei sind X und Y die Eingänge, A und

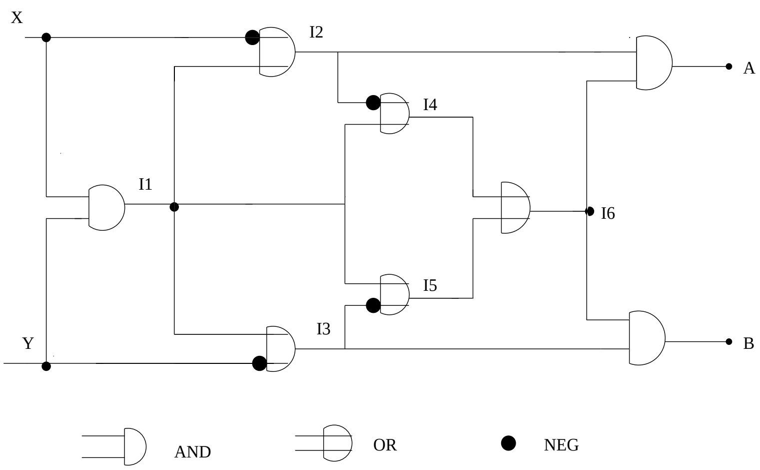

Abbildung 1: Kreuzschaltung

B die Ausgänge, und es gilt $A = Y$ und $B = X$.

Schreiben Sie ein CHR-Constraint `cross(X,Y,A,B)`, das obige Schaltung durch Boolesche Constraints implementiert.

Welche Antworten liefern die folgenden Ziele?

```
cross(1,0,A,B)
cross(1,Y,1,B)
cross(0,Y,A,B)
```

Aufgabe 9-3

Schreiben Sie ein CHR-Constraint `adder(X,Y,CI,O,CO,F1,F2,F3,F4,F5)`, das den defekten Gatter eines Volladdierers feststellt, wobei `F1,F2,F3,F4,F5` die Fehlervariablen sind.

Übersetzen Sie die in der Vorlesung vorgestellten logische Modellierung des Zusammenhangs zwischen Fehlervariable und Gatter, $\neg F_i \rightarrow G_i$, durch Umformen und Hilfsvariablen in erlaubte Boolesche Constraints. Eine Möglichkeit ist, dazu `equiv` und `imp` zu verwenden. Ein andere Möglichkeit besteht darin, ein geeignetes Constraint selbst zu implementieren.

Vergessen Sie nicht, sicherzustellen, daß maximal ein Gatter defekt sein kann. Am Ende des Programms ist es weiters sinnvoll, eine Suche/Enumeration über die Fehlervariablen zu implementieren.

Welche Antworten liefern die folgenden Ziele?

```
adder(0,0,1,0,1,F1,F2,F3,F4,F5).
adder(1,1,0,0,0,F1,F2,F3,F4,F5).
```