Constraint Programmierung
Prof. Dr. Thom Frühwirth, Marc Meister

WS4 – Blatt 5
Besprechung am 30. Nov., 14 Uhr, O27/2201
Veranstaltungsseite `http://www.informatik.uni-ulm.de/pm/index.php?id=82`

Herr Tom Schrijvers (Katholieke Universiteit Leuven, Belgium) spricht zum Thema
An Optimal Union-Find Algorithm in Constraint Handling Rules
(morgen [!]) am Mittwoch, den 24. November, 15 Uhr s.t., O27/121.

Aufgabe 14 (CLP Cryptoarithmetic Puzzle)**.**
Replace distinct letters by disctinct digits (numbers have no leading zeros), s.t. the following calculation holds (a literal translation to English is "Test thoroughly your strengths.").

$$\begin{array}{cccccc} & & T & E & S & T & E \\ + & & F & E & S & T & E \\ + & & D & E & I & N & E \\ \hline = & K & R & \ddot{A} & F & T & E \end{array}$$

a) Stick to the example presented in the course and use the library `clpfd`.
b) Write a predicate `distinct(L)` which is true if all elements from list `L` have a value from $\{0, 1, \ldots, 9\}$ and are (pairwise) distinct. Use the library `clpq`.

CCLP

Wir verwenden eine Untermenge von Constraint Handling Rules (CHRs), um in CCLP zu programmieren. Eine CCL-Klausel der Form

$$H \leftarrow C : D \mid G$$

wird in CHR geschrieben als

```
N @ H <=> C & D | G
```

wobei `N @` ein optionaler Name für die Regel ist.
`N @ H <=> C | G` ist die Kurzform für `N @ H <=> C & true | G`.
Weitere Informationen zu CHR im Handbuch und in den Übungen.

Aufgabe 15 (Vergleich CLP zu CCLP)**.**
Vergleichen Sie das Verhalten der folgenden CLP- und CCLP-Programme, die aus jeweils *einer* der gegebenen Regeln bestehen, indem Sie jeweils die unten angegebenen Ziele auswerten. Überprüfen Sie Ihre Antworten am Rechner und machen Sie sich klar, warum man unterschiedliche Ergebnisse erhält.

```
p(a) :- true.
p(X) :- X=a.
p(X) :- X = a, X = b.
```

```
p1 @ p(a) <=> true | true.
p2 @ p(X) <=> X=a | true.
p3 @ p(X) <=> true | X=a.
p4 @ p(X) <=> true & X = a | true.
p5 @ p(X) <=> X = a & X = b | true.
```

Ziele (Anfragen): `p(a)`, `p(b)` und `p(C)`, jeweils einzeln.

Aufgabe 16. Implementieren Sie die drei Varianten des CCLP-Beispiels `min` (minimum) aus der Vorlesung in CHR. Testen und erklären Sie das unterschiedliche Verhalten der Varianten anhand von folgenden (jeweils einzel zu stellenden) sechs Anfragen:

```
min(1,2,C).   min(A,2,1).   min(A,2,3).
min(A,A,B).   min(1,2,1).   min(1,2,3).
```