WS4 – Blatt 12
Besprechung am 8. Feb., 14 Uhr, O27/2201
Veranstaltungsseite `http://www.informatik.uni-ulm.de/pm/index.php?id=82`

Constraint System $\Re$

Aufgabe 35 ($\Re$ as special RT solver).
Modify the solver for rational trees (RTs)

```
idempotency   @ X eq Y \ X eq Y  <=> true.
reflexivity   @ X eq X           <=> true.
orientation   @ T eq X           <=> var(X), X@<T | X eq T.

decompostion  @ T1 eq T2         <=> nonvar(T1), nonvar(T2) | ...

confrontation @ X eq T1, X eq T2 <=> var(X), X@<T1, T1@=<T2 | X eq T1, T1 eq T2.
```

by only changing the body of the `decomposition` rule to handle linear polynomial equations: The resulting equation `T1` minus `T2` is computed in normal form. An equation is in normal form if `X eq A + B*Y + ...` for variables, `X, Y, ...` and numbers `A, B, ....`.
Discuss completeness and termination. Test your implementation.

Aufgabe 36.
Implement an optimized solver for $\Re$ as discussed in the lecture: Based on the $\Re$ solver available at WebCHR `http://bach.informatik.uni-ulm.de/~webchr/` (note: actual rules are at the very bottom of the files), that use slack variables to turn inequations into equations use Gaussian-style variable elimination for equations, use Fourier variable elimination and the bridge rule combining the variable elimination approaches for equations and inequations containing only slack variables.

Constraint System *FD*

Aufgabe 37 (Erweiterung *FD*-Löser).
Die Datei `fd.pl` von der Veranstaltungsseite enthält einen Constraintlöser für endliche Wertebereiche. Erweitern Sie diesen Constraintlöser schrittweise.

a) Fügen Sie die passenden Regeln für die bekannten Constraints `leq`, `eq` und `neq` ein. Testen Sie z.B. auch mit
`X leq Y, Y leq Z, X in [0,3], Y in [-1,2], Z in [0,1,2,3]`

b) Implementieren Sie Regeln für das Constraint `maximum(X,Y,Z)` wobei `Z` das Maximum von `X` und `Y` ist. Orientieren Sie sich dabei an der Regel für die Addition aus der Vorlesung. Propagieren Sie nur in Richtung von `Z`. Test Sie z.B. auch mit `maximum(X,Y,Z), X in [0,1], Y in [2,4], Z in [3,5]`.

c) Das Constraint `allneq(List)` erzwingt, daß die Variablen in `List` (paarweise) verschieden sind. Testen Sie ihre Regeln für `allneq` für drei Variablen. Variieren Sie die Aufzählbereiche so, daß Testfälle mit Antwort `no` und andere mit Antwort `yes` erhalten werden.

d) Um Variablen zu binden, schreiben Sie ein Constraint `label(List)`, das die Variablen in `List` bindet (siehe Vorlesung). Ein einfacher Testfall ist bspw. `X in [0,1], Y in [0,1], X leq Y, label([X])`.