Ludwig-Maximilians-Universität München
Institut für Informatik
Thom Frühwirth, Slim Abdennadher

Constraint-Programmierung: Theorie und Anwendung, SS 02
Übungsblatt 3

Besprechung am 04.06.02

Nebenläufige CLP

Wir verwenden eine Untermenge von Constraint Handling Rules (CHR), um in NCLP zu programmieren. Eine NCL-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, der auch weggelassen werden kann. *Lesen Sie die Kurzbeschreibung zu CHR (siehe Vorlesungswebseite).*

Vergleichen Sie das Verhalten der folgenden 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.

```
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: {`p(a)`}, {`p(b)`} und {`p(X)`}, jeweils einzeln.

Hinweis: Die Übungen finden wie vereinbart ab nun wöchentlich am Donnerstag um 11 c.t. im Rechnerraum Z8 Gobi statt.

Aufgabe 3-1

Implementieren Sie die Varianten des NCLP-Beispiels `min` (minimum) aus der Vorlesung (siehe NCLP-Folien, via Vorlesungswebseite erreichbar) in CHR. Machen Sie sich das unterschiedliche Verhalten der Varianten anhand von geigneten Anfragen klar.

Aufgabe 3-2

Implementieren Sie die Aufgabe 2-2 in NCLP anhand der Musterlösung. Beachten Sie, was in den Wächter, und was in den Rumpf der NCL-Klauseln kommt.

Aufgabe 3-3

Implementieren Sie das NCLP-Beispiel `hamming` aus der Vorlesung (siehe NCLP-Folien, via Vorlesungswebseite erreichbar) in CHR. Um die Ergebnisse der unendlichen Zahlenfolge sichtbar zu machen, sobald sie vorliegen, schreiben Sie ein zusätzliches rekursives Prädikat `observe`, das das Ergebnis von `hamming` beobachtet und ausgibt (`write(...),nl`). Der Aufruf sieht dann so aus: `observe(S),hamming(S)`.