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

Constraint-Programmierung: Theorie und Anwendung, SS 02
Übungsblatt 2

Besprechung am 7.05.2002

Aufgabe 2-1

a) Schreiben Sie ein Prolog-Prädikat `summe(X,Y,Z)`, das genau dann zutrifft, wenn zwei ganze Zahlen `X` und `Y` zwischen `1` und `10` die Summe `Z` haben.
Beispiel: `summe(5,6,11)` trifft zu.

b) Verfolgen Sie, wie Ihr Programm das Ziel `summe(5,6,Z)` bearbeitet.

c) Welche Antwort liefert das Ziel `summe(X,Y,10)`?

Aufgabe 2-2

a) Modifizieren Sie die Definition des Prädikats `summe(X,Y,Z)` so, daß in alle Richtungen gerechnet werden kann (z.B. eine Summe `Z` kann in ihre Summanden `X` und `Y` zerlegt werden).
Hinweis: Es sollen zwei ganze Zahlen aus einem bestimmten Intervall generiert werden, deren Summe dann berechnet und auf Gleichheit mit dem Wert von `Z` überprüft wird. Wenn die Überprüfung scheitert, wird durch Rücksetzen weitergesucht.

b) Wie wird jetzt das Ziel `summe(X,Y,10)` bearbeitet.

Aufgabe 2-3

In der obigen modifizierten Definition des Prädikats `summe/3` werden solange alternative Lösungen produziert, bis ein Summandenpaar mit richtiger Summe gefunden wird. Dies zeigt, wie Prolog den gesamten Suchbaum für die Lösung eines Problems systematisch absucht.

a) Modifizieren Sie die Definition des Prädikats `summe/3` so, daß das Suchverhalten sich verbessert (Es sollen im voraus Teile des Suchbaumes ausgeschlossen werden).
Hinweis: Betrachten Sie dazu ein Ziel wie `summe(X,Y,10)`. Wenn eine ganze Zahl für `X` generiert wird, kann man `Y` direkt berechnen, weil die Summe `10` bekannt ist. Verwenden Sie die vordefinierten Prädikate `nonvar/1` und `var/1`, um zu überprüfen, welche Variablen gebunden sind.

b) Machen Sie sich aufgrund einiger Anfragen klar, wie sich das Backtracking-Verhalten verbessert hat.