

Prof. Dr. Jürgen Giesl
Peter Schneider-Kamp
René Thiemann

Übungen *Logikprogrammierung* – Blatt 10

Abgabe am Mittwoch, 21. Juni 2006, zu Beginn der Übung.

Aufgabe 1 (4 Punkte)

In Prolog gibt es die vier Prädikate `=`, `unify_with_occurs_check`, `is` und `==`, die alle eine Art der Gleichheit ausdrücken. Definieren Sie für jedes dieser Prädikate p ein entsprechendes Prädikat q mit derselben Semantik wie p , wobei Sie nur die anderen drei Gleichheits-Prädikate als eingebaute Prädikate nutzen dürfen. Falls dies nicht möglich ist, geben Sie eine kurze Begründung an. Gehen Sie davon aus, dass sich mit den eingebauten Zahlen nur endlich viele Zahlen darstellen lassen.

Um also ein Prädikat `=` mit der Semantik von `=` zu definieren, dürfen Sie nur die Prädikate `unify_with_occurs_check`, `is` und `==`, sowie andere selbst-definierte Prädikate verwenden.

Aufgabe 2 (4 + 2 + 2 Punkte)

Wir betrachten folgendes Prolog-Programm.

```
p(X, Y) :- kandidat(X), m(s(X), X, Y).
```

```
kandidat(0).  
kandidat(s(s(0))).  
kandidat(X).
```

```
m(X, 0, X).  
m(X, X, 0).  
m(s(X), s(Y), Z) :- m(X, Y, Z).
```

Hier soll das Prädikat `m` die Subtraktion zweier Zahlen berechnen. Mit Hilfe des Prädikats `p` kann man prüfen, ob ein Fehler in der Definition von `m` vorliegt: Es rät sich zuerst einen Test-Kandidaten X , und berechnet dann den Wert $Y = (X+1) - X$. Bei einer korrekten Implementierung von `m` sollte deshalb die Anfrage `p(X, Y)` offensichtlich immer die Variable Y mit `s(0)` belegen.

- a) Skizzieren Sie den SLD-Baum, den Prolog für die Anfrage `p(X, Y)` berechnet.

- b) Welche Lösungen findet Prolog und wie lauten die entsprechenden Antwortsubstitutionen? Erklären Sie, warum auch Antworten gefunden werden, die Y nicht mit $s(0)$ belegen.
- c) Welche Antworten entsprechen nicht der Semantik des entsprechenden Logikprogramms und wie muss man das Programm ändern, damit das Prolog-Programm und das Logikprogramm dieselbe Semantik haben? Geben Sie an, wie sich der SLD-Baum ändert.

Aufgabe 3 (1,5 + 1,5 Punkte)

Gegeben sei das folgende Prolog-Programm:

```
double(0,0).  
double(s(X), s(s(Y))) :- double(X, Y).  
  
doubleInt(X,Y) :- X >= 0, Y is X * 2.
```

Es definiert zwei Prädikate `double` und `doubleInt`, welche auf natürlichen Zahlen die Verdopplungsfunktion berechnen. Hierbei arbeitet `double` auf einer Zahlendarstellung mit `s` und `0`, während `doubleInt` die eingebauten Zahlen nutzt.

- a) Definieren Sie für beide Zahlendarstellungen zwei Prädikate `half(X,Y)` und `halfInt(X,Y)`, die wahr sind, falls Y die Hälfte der natürlichen Zahl X ist. Nutzen Sie dazu das Prinzip der Bidirektionalität, wenn dies möglich ist.
- b) Vergleichen Sie die Laufzeit und die Semantik von `half` und `halfInt`. Gehen Sie dabei davon aus, dass das erste Argument eine natürliche Zahl in entsprechender Zahlendarstellung ist. Geben Sie alle Unterschiede an.