

Theorie der Programmiersprachen

13. Übung

1. Aufgabe:

Geben Sie an, was die prozedurale Semantik des Logik-Programms

$$\begin{aligned} &P(a, a). \\ &P(a, b). \\ &P(x, y) :- P(y, x). \end{aligned}$$

bei gegebener Zielklausel

$$?- P(a, z)P(z, a)$$

ist.

2. Aufgabe:

Geben Sie für das Logik-Programm

$$F = \{\{P(x, z), \neg Q(x, y), \neg P(y, z)\}, \{P(u, u)\}, \{Q(a, b)\}\}$$

bzw.

$$\begin{aligned} &P(x, z) :- Q(x, y), P(y, z). \\ &P(u, u). \\ &Q(a, b). \end{aligned}$$

eine nicht erfolgreiche sowie zwei erfolgreiche Rechnungen (mit Rechenergebnis) an bei Zielklausel $?- P(v, b)$ ($G = \{\neg P(v, b)\}$).

3. Aufgabe:

Betrachten Sie das Additionsprogramm aus der Vorlesung

$$\{\{A(x, 0, x)\}, \{A(x, s(y), s(z)), \neg A(x, y, z)\}\}.$$

Berechnen Sie $A(x, y, 2)$ und $A(x, x, s(x))$ und geben Sie alle möglichen Lösungen an.

Welches Problem tritt bei $A(x, s(x), x)$ bzw. $A(x, s(s(0)), s(0))$ auf?

4. Aufgabe:

Schreiben Sie ein Logikprogramm, das Formeln weitgehend vereinfacht – durch Eliminieren von Überflüssigen Summanden, die 0 sind, und Faktoren, die 1 sind. (Dieses Programm könnte dann mit dem Differenzierprogramm kombiniert werden). Das gesuchte Programm sollte Folgendes leisten können:

Die Eingabe einer Zielklausel

$$?- \text{Einfach}(1 * F + (G + (0 + x)) * 1, H).$$

führt zu dem Rechenergebnis

$$\text{Einfach}(1 * F + (G + (0 + x)) * 1, F + (G + x)).$$

Dabei sind $x, 0, 1$ Konstanten und F, G, H Variablen, *Einfach* ist ein zweistelliges Prädikatsymbol und $*$, $+$ sind Funktionssymbole. Zur besseren Lesbarkeit verwenden wir hierbei die Infixnotation bei $+$ und $*$, d. h. wir schreiben $x + y$ statt $+(x, y)$.