

# Theorie der Programmiersprachen

## 13. Übung

**1. Aufgabe:** Geben Sie die *prozedurale Semantik* sowie die *modelltheoretische Semantik* des Logikprogramms

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

bei gegebener Zielklausel

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

an.

**2. Aufgabe:** Schreiben Sie ein Logikprogramm, das Formeln weitgehend vereinfacht – durch Eliminieren von Überflüssigen *Summanden*, die 0 sind, und *Faktoren*, die 1 sind. 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  $+(x, y)$ .

### 3. Aufgabe:

Gegeben sei ein zweistelliges Prädikat  $E(x, y)$ , das wahr ist, falls  $x$  ein Elternteil von  $y$  ist. Mit Hilfe dieses Prädikats soll ein weiteres Prädikat  $V(x, y)$  definiert werden, das wahr ist, falls  $x$  ein Vorfahre von  $y$  ist.

Vergleichen Sie die beiden möglichen Lösungen:

a)

$$\begin{aligned} E(x, y) \rightarrow V(x, y) \\ E(x, y) \wedge V(y, z) \rightarrow V(x, z) \end{aligned}$$

b)

$$\begin{aligned} E(x, y) \rightarrow V(x, y) \\ V(x, y) \wedge V(y, z) \rightarrow V(x, z) \end{aligned}$$

#### 4. Aufgabe:

Gesucht ist ein Prädikat  $P(x, y)$ , das wahr ist, falls  $x$  und  $y$  verheiratet sind. Es gibt jedoch zwei Zusatzbedingungen:

- Es darf für jedes Ehepaar nur eine Tatsachenklausel verwendet werden.
- Das Prädikat darf keine Endlosschleifen erzeugen. Dies schließt zum Beispiel die Klausel „ $P(x, y) \rightarrow P(y, x)$ “ aus.