Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Dr. Clemens Ballarin

Vorlesung Semantik
Sommersemester 2004
Übungsblatt 2
Besprechung am 5. Mai 2004

Aufgabe 1 (H) (*Semantische Äquivalenz*)

Zwei Statements s_1 und s_2 heissen semantisch äquivalent, wenn sie gleiche Startzustände in gleiche Endzustände überführen:

$$s_1 \sim s_2 \quad :\Longleftrightarrow \quad \forall \sigma, \sigma' \in \Sigma.\, \langle s_1, \sigma \rangle \to \sigma' \iff \langle s_2, \sigma \rangle \to \sigma'$$

Zeigen Sie, dass gilt

$$(s_1; s_2); s_3 \;\sim\; s_1; (s_2; s_3)$$

Aufgabe 2 (H) (*Regelinduktion*)

Permutationen von Listen (über einem beliebigen Elementtyp) werden durch die folgenden vier Regeln definiert.

$$([], []) \in \mathbf{Perm} \qquad \text{(Nil)} \qquad\qquad (x\#y\#l, y\#x\#l) \in \mathbf{Perm} \qquad \text{(Swap)}$$

$$\frac{(xs, ys) \in \mathbf{Perm}}{(z\#xs, z\#ys) \in \mathbf{Perm}} \quad \text{(Cons)} \qquad\qquad \frac{(xs, ys) \in \mathbf{Perm} \qquad (ys, zs) \in \mathbf{Perm}}{(xs, zs) \in \mathbf{Perm}} \quad \text{(Trans)}$$

Dabei enthält die definierte Menge **Perm** Paare von Listen, die sich nur durch die Reihenfolge der Elemente unterscheiden. Mit # wird der *Cons*-Operator bezeichnet, d.h. $x\#xs$ ist die Liste bestehend aus dem ersten Element x und der Restliste xs. Formulieren Sie die dazugehörige Induktionsregel und beweisen Sie damit die folgenden Aussagen:

a) Für $(xs, ys) \in \mathbf{Perm}$ gilt: xs und ys haben die gleiche Länge.

b) Für $(xs, ys) \in \mathbf{Perm}$ gilt: xs und ys enthalten die gleichen Elemente.

Aufgabe 3 (Ü) (*Semantik der While-Schleife*)

Zeigen folgende **while**-Programme dasselbe Verhalten oder nicht? (Begründung)

a) Vergleichen Sie **while** $b_0 \vee b_1$ **do** s mit

1) **while** b_0 **do** s; **while** $b_0 \vee b_1$ **do** s 2) **while** $b_0 \vee b_1$ **do** s; **while** b_0 **do** s

b) Vergleichen Sie **while** $b_0 \wedge b_1$ **do** s mit

3) **while** b_0 **do** s; **while** $b_0 \wedge b_1$ **do** s 4) **while** $b_0 \wedge b_1$ **do** s; **while** b_0 **do** s

Aufgabe 4 (Ü) (*Semantik der While-Schleife*)

Die aus der Vorlesung bekannten Regeln der operationalen Semantik seien in Bezug auf die **while**-Schleife wie folgt modifiziert:

$$\frac{}{\langle \textbf{while } b \textbf{ do } s, \sigma\rangle \to \sigma} \quad \text{falls} \quad B[\![b]\!]\sigma = ff$$

$$\frac{\langle \textbf{while } b \textbf{ do } s, \sigma\rangle \to \sigma'' \quad \langle s, \sigma''\rangle \to \sigma'}{\langle \textbf{while } b \textbf{ do } s, \sigma\rangle \to \sigma'} \quad \text{falls} \quad B[\![b]\!]\sigma = tt$$

a) Verändert sich dadurch die Semantik von **while**-Programmen oder nicht? Begründen Sie Ihre Antwort oder geben Sie ein Beispielprogramm an, das bezüglich der beiden Regelmengen unterschiedliche Bedeutungen besitzt.

b) Sei $w \equiv \textbf{while } b \textbf{ do } s$. Zeigen Sie, dass es für alle σ unter der modifizierten Semantik kein σ' gibt, sodass $\langle w, \sigma\rangle \to \sigma'$. Betrachten Sie dazu die Form der Ableitungen.

Aufgabe 5 (P) (*Operationale Semantik in PROLOG*)

a) In imperativen Programmiersprachen gibt es neben dem *while*-Konstrukt oft auch ein *for*-Konstrukt, bei dem eine Laufvariable bis zu einem bestimmten Wert hochgezählt wird.

   Die Syntax der Sprache WHILE aus der Vorlesung sei nun um das Konstrukt

   for x **from** a_1 **to** a_2 **do** s

   erweitert. Geben Sie die entsprechenden Regeln für die operationelle Semantik an.

b) Syntax und Semantik der erweiterten WHILE-Sprache sollen in PROLOG programmiert werden. Zur Vereinfachung sollen für die abstrakte Syntax statt der Infix- bzw. Mixfix-Schreibweise '$a_0 + a_t$' oder '*if* b_0 *then* s_0 *else* s_1' ausschließlich Präfixkonstrukte wie '$add(a_0, a_t)$' oder '$cond(b_0, s_0, s_1)$' verwendet werden.

Allgemeines

Hinweise zum Starten von PROLOG auf den Maschinen in der Informatikhalle und Literaturhinweise finden Sie in der Datei

`/usr/proj/semantik/prog/prolog/hinweise.txt`

Einen Programm-Rahmen zur Aufgabe finden Sie im Verzeichnis

`/usr/proj/semantik/prog/prolog/big-step/`

Abgabe der Hausaufgaben (H) und Programmieraufgaben (P) vor Beginn der Übung am 5. Mai. Abgabe der Programmieraufgaben per E-Mail an ballarin@in.tum.de.