XI.Übungsblatt zur Vorlesung "Semantik von Programmiersprachen" SS 97

Prof. Gert Smolka und Martin Müller

Ausgabe: 3.7.1997 Abgabe: 10.7.1997

Aufgabe XI.1(Nichtstrikte Auswertung, 2 Punkte) Es sei $ones = \mathbf{rec}\ x.(1, x)$. Zeigen Sie, daß $\mathbf{fst}\ \mathbf{snd}\ ones \to_l 1$.

Aufgabe XI.2(Nicht-strikte Datenstrukturen, 4 Punkte) Der LF-Term $\mathbf{rec}\ x.(1, x)$ beschreibt die unendliche Liste von Einsen. Geben Sie analog LF-Terme zur Beschreibung

1. der Liste der natürlichen Zahlen und
2. der Liste der Fibonaccizahlen an.

Aufgabe XI.3(Typen, 4 Punkte) Ein Term $t \in \mathbf{Ter}$ heiße eindeutig typisierbar, falls für alle $\tau_1, \tau_2 \in \mathbf{GTyp}$

$$t : \tau_1 \wedge t : \tau_2 \;\Rightarrow\; \tau_1 = \tau_2$$

Zeigen Sie, daß die Terme aus TF ohne **case** eindeutig typisierbar sind, wenn wir die Typsprache um rekursive Typen und Summen abschwächen, also nur die folgende Typsprache betrachten:

$$\tau \;::=\; \mathbf{int} \mid \mathbf{bool} \mid \mathbf{unit} \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2$$

In den folgenden zwei Aufgaben bezeichen wir die strikte bzw. nicht-strikte denotationale Semantik mit $[\![\cdot]\!]^e$ *bzw.* $[\![\cdot]\!]^l$.

Aufgabe XI.4(Denotationale Semantik, 6 Punkte) Es sei $typ(x) = \mathbf{int} \to \mathbf{int}$ und $typ(f) = (\mathbf{int} \to \mathbf{int}) \to (\mathbf{int} \to \mathbf{int})$. Weiter definieren wir $\Omega_e = \mathbf{rec}\, f.(\lambda x.f\, x)$ und $\Omega_l = \mathbf{rec}\ x.x$.

1. Bestimmen Sie $[\![\lambda x.x]\!]^e$, $[\![\Omega_e]\!]^e$ und $[\![(\lambda f.f)\ \Omega_e]\!]^e$
2. Bestimmen Sie $[\![\lambda x.x]\!]^l$, $[\![\Omega_l]\!]^l$ und $[\![(\lambda x.x)\ \Omega_l]\!]^l$.
3. Bestimmen Sie $[\![\lambda x.(t\ x)]\!]^e$ für einen beliebigen Term t. Gilt $[\![t]\!]^e = [\![\lambda x.(t\ x)]\!]^e$?

Aufgabe XI.5(Denotationale Semantik, 5 Punkte) In der Vorlesung wurden die folgenden Eigenschaften der denotationalen Semantik vorgestellt. Für alle $t \in \mathbf{GTer}$, $t_1, t_2 \in \mathbf{Ter}$, $c \in \mathbf{CTer}$ und $n \in \mathbf{N}$ gilt

1. $[\![t_1]\!]^e = [\![t_2]\!]^e \quad \Rightarrow \quad t_1 \sim t_2$
2. $\exists c : t \to c \quad \Leftrightarrow \quad [\![t]\!]^e \neq \lambda\rho.\bot$
3. $t \to n \quad \Leftrightarrow \quad [\![t]\!]^e = \lambda\rho.\lfloor n \rfloor$
4. $t \to c \quad \Rightarrow \quad [\![t]\!]^e = [\![c]\!]^e$
5. $[\![t]\!]^e \neq \lambda\rho.\bot \quad \Rightarrow \quad \exists c : t \to c$

In der Vorlesung wurden (4) und (5) gezeigt. Nehmen sie diese an und zeigen Sie $(1) - (3)$. Zeigen Sie im Einzelnen

1. $(4) \wedge (5) \Rightarrow (2)$
2. $(4) \wedge (2) \Rightarrow (3)$
3. $(2) \Rightarrow (1)$

Aufgabe XI.6(VPOs, 4 Punkte) Es seien $E, D_0, \ldots, D_n, \ldots$ abzählbar viele VPOs mit $\bot$ und sei $f : D_0 \times \ldots \times D_n \to E$ eine Funktion. Die Funktion f ist *punktweise stetig*, falls für alle i, $0 \leq i \leq n$ und $d_0 \in D_0, \ldots, d_{i-1} \in D_{i-1}, d_{i+1} \in D_{i+1}, \ldots, d_k \in D_k$ die Funktion $f_i : D_i \to E$ mit

$$f_i(d) \;=\; f(d_0, \ldots, d_{i-1}, d, d_{i+1}, \ldots, d_k)$$

stetig ist. Winskel zeigt auf Seite 127:

(Lemma 8.10) $\qquad f$ ist stetig genau dann, wenn f punktweise stetig ist.

Sei nun $g : D_0 \times D_1 \times \ldots \to E$ eine Funktion, deren Wertebereich ein unendliches Produkt ist. Die Definition von punktweiser Stetigkeit verallgemeinert sich analog. Zeigen oder widerlegen Sie:

g ist stetig genau dann, wenn g punktweise stetig ist.

Aufgabe XI.7(Einschrittsemantik, 4 Punkte) Geben Sie eine Einschrittsemantik für LF an, so daß deren reflexiv-transitiver Abschluß mit der Ergebnissemantik von LF zusammenfällt (d.h. so daß $\forall t \in \mathbf{GLTer}, \forall c \in \mathbf{CLTer} : t \to_l c$ gdw. $t \to_{1,l}^* c$). NB: Es ist kein Korrektheitsbeweis verlangt!

Aufgabe XI.8(4 Zusatzpunkte) Definieren Sie in SML einen divergierenden Term in dem Sie die Datentypdefinition

```
datatype MU = REC of MU -> MU;
```

aber keine explizite Rekursion verwenden. Hinweis: Definieren Sie geeignete Funktion `rep: MU -> MU -> MU` und `abs: (MU -> MU) -> MU` und schreiben Sie damit den Term $(\lambda x.x\ x)\ (\lambda x.x\ x)$ so um, daß er typisierbar wird.