IX.Übungsblatt zur Vorlesung
“Semantik von Programmiersprachen” SS 97

Prof. Gert Smolka und Martin Müller

Ausgabe: 19.6.1997 Abgabe: 26.6.1997

Aufgabe IX.1(Kontexte, 2 Punkte) F-Kontexte sind wie F-Terme augebaut, die *genau ein* Loch an Termposition enthalten. Schreiben Sie $\bullet$ für das Loch und definieren Sie F-Kontexte durch eine Grammatik.

Aufgabe IX.2(Typerhaltung, 8 Punkte)

Zeigen Sie für alle $t_1, t_2 \in \mathbf{Ter}$, $t, t' \in \mathbf{GTer}$ und $\tau, \tau_1, \tau_2 \in \mathbf{GTyp}$:

1. (Substitutionslemma)

$$t_1 : \tau_1 \;\wedge\; t_2 : \tau_2 \;\wedge\; \mathit{type}(x) = \tau_2 \quad \Rightarrow \quad t_1[t_2/x] : \tau_1$$

2. (Typerhaltungssatz)

$$t : \tau \;\wedge\; t \to_1 t' \quad \Rightarrow \quad t' : \tau$$

3. (Typfehlerfreiheit)

$$t : \tau \;\wedge\; t \text{ nicht kanonisch} \quad \Rightarrow \quad \exists t' :\; t \to_1 t'$$

Gilt die Umkehrung des Typerhaltungssatzes?

Aufgabe IX.3(Operationale Semantik von $\mathbf{F}_0$, 4 Punkte) Zeigen Sie für alle $t, t' \in \mathbf{GTer}_0$ und $c \in \mathbf{CTer}_0$:

1. $\forall t, t' \; \forall c : \qquad t \to_1 t' \;\wedge\; t' \to c \Rightarrow t \to c$

   (Hinweis: Mit Induktion über die Länge der Herleitung von $t \to c$.)

2. $\forall t \; \forall c : \qquad t \to c \;\Leftrightarrow\; t \to_1^* c$

Aufgabe IX.4(Kapern, 2 Punkte) Wir sagen eine Variable x sei “frei für y in t” falls x bei Substitution für y in t nicht von einem Variablenbinder in t gekapert wird. Zum Beispiel sind im Term $\lambda y.\lambda z.(x\ \lambda u.x)$ alle Variablen außer y, z und u frei für x. Definieren Sie eine SML-Funktion, die zu einem F_0-Term $t \in \mathbf{Ter}_0$ und einer Variablen x die Menge an Variablen liefert, die *nicht* frei für x in t sind. Verwenden Sie dabei die folgenden Operator- und Datentypdeklarationen zur Darstellung von F_0-Termen in SML.

```
infixr 6 ^;
infix 7 @;
datatype Binder = \ of int;
datatype Lambda = X of int | ^ of Binder * Lambda | @ of Lambda * Lambda;
```

Typische Elemente dieses Datentyps sind

- `\1^(X 1 @ X 1)`

  für $\lambda x.x\ x$

- `(\1^X 1) @ X 1`

  für $(\lambda x.x)\ x$

- `\1^ \2^ X 1`

  für $\lambda x.\lambda y.x$

Aufgabe IX.5(SML, 4 Punkte)

1. Definieren Sie `Plus1: int -> int`, `Minus1: int-> int`, und `Gleich0: int -> bool` in SML mit den vordefinierten Datentypen `int` und `bool`. Definieren Sie außerdem den Rekursionsoperator R.
2. Definieren Sie nun `Plus: int -> int -> int`, `Mult: int -> int -> int`, und `Gleich: int -> int -> bool`, indem Sie nur die in 1 definierten Funktionen und das einfache SML-Konditional `if Gleich0 x then` e_1 `else` e_2 verwenden.