PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
J.Berthold

D-35032 Marburg
Hans Meerwein Straße
Lahnberge

26.Oktober 2006

2. Übung zu „Grundlagen der funktionalen Programmierung“,

Abgabe: 2.November 2006 vor der Vorlesung **WS06/07**

Aufgaben

2.1 λ-Terme in Haskell

4 Punkte

Die (induktiv aufgebauten) λ-Terme des λ-Kalküls lassen sich in einem rekursiven Haskell-Datentyp `LExp` repräsentieren.

Haskell Code
```
data LExp = Var Int      -- interne Darstellung: Var. nummerieren.
          | Lam Int LExp
          | App LExp LExp
    deriving Show
-- Beispiel: (\ x . x y)
bsp :: LExp
bsp = (Lam 0 (App (Var 0) (Var 1)))
```

(a) Schreiben Sie die Terme S und K aus Aufgabe 2 in ihrer Haskell-Repräsentation als `LExp`. / 2

(b) Definieren Sie eine Funktion `free :: LExp -> [Int]` zur Bestimmung der freien Variablen in einem Ausdruck. (Verwenden Sie das Modul `Data.List`) / 2

2.2 Kombinatorische Logik

3 Punkte

Ein λ-Ausdruck e mit $free(e) = \emptyset$ heißt *Kombinator*. In der kombinatorischen Logik, welche eng mit dem λ-Kalkül verwandt ist, spielen die folgenden Kombinatoren eine zentrale Rolle:

$$\begin{aligned} I &= \lambda\, x.\, x \\ K &= \lambda\, x\, y.\, x \\ S &= \lambda\, x\, y\, z.\, x\, z\, (y\, z) \end{aligned}$$

Bestimmen Sie die Normalformen der folgenden Ausdrücke:

(a) $S\ I\ I$

(b) $S\ K\ K$

(c) $S\ S\ S$

Bitte wenden!

2.3 **De-Bruijn-Notation für λ-Terme** $\boxed{\text{5 Punkte}}$

Mit Hilfe einer von Nicolaas Govert de Bruijn eingeführten Syntax für λ-Terme kann das Problem der Namenskollision bei der Substitution komplett umgangen werden. λ-Terme sind hier wie folgt definiert:

- Jede Zahl $n \in \mathbb{N}$ ist λ-Term.
- Falls M und N λ-Terme sind, sind auch $(M\ N)$ und (λM) λ-Terme.

Die durch λ gebundenen Variablen werden durch Zahlen ersetzt, welche die Bindungstiefe einer Variablen, also die Anzahl der Lambda-Symbole zwischen der Verwendung und dem bindenden Lambda-Ausdruck angeben.

Beispiele (Kombinatoren aus Aufgabe 2):

$$\begin{aligned} I &: \quad \lambda 1 \\ K &: \quad \lambda\lambda 2 \\ S &: \quad \lambda\lambda\lambda 31(21) \end{aligned}$$

Zentral für diese Interpretation ist eine passende Definition der β-Reduktion und der Substitution:

$$\textbf{β-Reduktion: } (\lambda P)Q \quad \Rightarrow_\beta \quad P[1 \mapsto Q]$$

$$\begin{aligned} \textbf{Substitution:} \qquad n[m \mapsto N] \quad &:= \quad \begin{cases} n & \text{falls } n < m \\ n-1 & \text{falls } n > m \\ \text{rename}_{n,1}(N) & \text{falls } n = m \end{cases} \\ (M_1\ M_2)[m \mapsto N] \quad &:= \quad (M_1[m \mapsto N])\ (M_2[m \mapsto N]) \\ (\lambda M)[m \mapsto N] \quad &:= \quad \lambda(M[m+1 \mapsto N]) \end{aligned}$$

$$\begin{aligned} \textbf{Umnummerierung:} \qquad \text{rename}_{m,i}(n) \quad &:= \quad \begin{cases} n & \text{falls } n < i \\ n+m-1 & \text{falls } n \geq i \end{cases} \\ \text{rename}_{m,i}(M_1\ M_2) \quad &:= \quad (\text{rename}_{m,i} M_1)\ (\text{rename}_{m,i} M_2) \\ \text{rename}_{m,i}(\lambda M) \quad &:= \quad \lambda(\text{rename}_{m,i+1} M) \end{aligned}$$

(a) Geben Sie die folgenden Terme in De-Bruijn-Notation an: / 2

i. $\lambda xy.\ (\lambda x.\ y\ x)(\lambda y.\ x\ y)$

ii. $(\lambda x.x\ y)\ (\lambda z.z\ x)$

(b) Reduzieren Sie den folgenden Term nach den obigen Regeln so weit wie möglich: / 3

$$(\lambda\lambda\lambda 31(21))(\lambda\lambda 2)(\lambda\lambda 2)$$