



Übungen zu „Konzepte von Programmiersprachen“, WS 2010/11

Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 13 (letztes Blatt in der Wertung)

Abgabe: Dienstag, 1. Februar 2011 vor der Vorlesung

34. Übersetzung in Kombinatorausdrücke

5 Punkte

Gegeben sei die folgende Deklaration eines Ausdruckstyps, mit dem λ -Ausdrücke und Kombinatorikalkül-Ausdrücke definiert werden können (siehe Datei `expr.hs` mit eigener Show-Instanz auf der Vorlesungsseite).

```
data Expr = Var Int | Lambda Int Expr | App Expr Expr
          | S | K | I | B | C     deriving (Show, Eq)
```

(a) Schreiben Sie eine Haskell-Funktion `trans :: Expr -> Expr`, die die in der Vorlesung vorgestellte Übersetzung von λ -Ausdrücken in SKI-Kombinatorausdrücke implementiert. / 2

(b) Berechnen Sie mit Ihrer Übersetzungsfunktion einen zu / 0.5

$$\lambda x.\lambda y.\lambda z.x z (y z) \quad (= S)$$

äquivalenten Kombinatorausdruck.

(c) Optimieren Sie die Übersetzung in Kombinatorausdrücke, indem Sie in geeigneten Spezialfällen Ausdrücke mit den Kombinatoren B und C bilden und bei der Übersetzung berücksichtigen, dass für beliebige λ -Ausdrücke M gilt: / 2

$$S (K M) I \Leftrightarrow_C^* M$$

(d) Wiederholen Sie (b) mit der optimierten Übersetzungsfunktion. Erläutern Sie das Ergebnis. / 0.5

35. Modifikation des SECD-Interpreters

5 Punkte

(a) Modifizieren Sie die in der Vorlesung vorgestellte Haskell-Implementierung der SECD-Maschine (siehe Datei `secd_pp.hs` auf der Vorlesungsseite) so, dass anstelle der innermost- eine outermost-Auswertungsstrategie realisiert wird. Dies bedeutet, dass in Applikationen Argumentausdrücke ohne vorherige Auswertung in der Umgebung gespeichert und nur bei einem Zugriff ausgewertet werden sollen. Das Auswertungsergebnis soll in der Umgebung gespeichert werden, damit bei einem erneuten Zugriff eine Mehrfachauswertung vermieden wird. / 4

(b) Berechnen Sie mit Ihrer Maschine aus (a) oder per Hand die Normalform von $(Y(KK))$. / 1

36. Listenmonade

2 Punkte

Schreiben Sie unter Verwendung der Listenmonade eine Funktion

$$\text{goldbach} :: \text{Int} \rightarrow [(\text{Int}, \text{Int})],$$

die zu einer positiven ganzen Zahl die Liste aller Paare von Primzahlen bestimmt, deren Summe die Zahl ergibt.