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

D-35032 Marburg
Hans Meerwein Straße
Lahnberge

9.November 2006

4. Übung zu „Grundlagen der funktionalen Programmierung“,

Abgabe: 16.November 2006 vor der Vorlesung **WS06/07**

Aufgaben

4.1 Fixpunktkombinatoren

4 Punkte

(a) Zeigen Sie, dass für den Turingschen Fixpunktkombinator

$$\Theta = (\lambda\ x\ y.y\ (x\ x\ y))\ (\lambda\ x\ y.y\ (x\ x\ y))$$

gilt: $\Theta\ e \Rightarrow_\beta^* e\ (\Theta\ e)$ für beliebige λ-Ausdrücke e.

(b) Sei

$$\begin{aligned} G &\equiv \lambda a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p\,q\,s\,t\,u\,v\,w\,x\,y\,z\,r. \\ &\qquad r\,(d\,a\,s\,i\,s\,t\,e\,i\,n\,f\,i\,x\,p\,u\,n\,k\,t\,k\,o\,m\,b\,i\,n\,a\,t\,o\,r) \\ F &\equiv G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G\ G \end{aligned}$$

Zeigen Sie, dass F ein Fixpunktkombinator ist, d.h. dass $F\ e \Leftrightarrow_\beta^* e\ (F\ e)$ für beliebige λ-Ausdrücke e.

4.2 SECD-Auswertung

3 Punkte

Geben Sie die Zustandsübergänge der SECD-Maschine bei der Auswertung des Ausdrucks $(\lambda x.x\,x)(\lambda x.x\,x)$ an.

4.3 Normal-Order-Auswertung in der Haskell-SECD

5 Punkte

Die in der Vorlesung vorgestellte SECD-Maschine führt eine Auswertung in “applicative order” durch, d.h. für Applikationen wird zuerst das Argument ausgewertet, danach die applizierte λ-Abstraktion, danach die Applikation selbst.

(a) Ändern Sie die Haskell-Implementierung *secd_pp.hs* der SECD-Maschine von der Vorlesungsseite ab, so dass sie im Stil einer “normal order”-Auswertung zunächst ohne Auswertung des Arguments die Applikation durchführt. / 3

(Lassen Sie zur Vereinfachung die Striktheit der primitiven Operationen außer Acht.)

(b) Geben Sie zwei Beispielausdrücke an, welche die Effekte der geänderten Auswertungsstrategie zeigen. / 2