

Einführung in die funktionale Programmierung

Wintersemester 2007/2008

Aufgabenblatt Nr. 2

Abgabe: Dienstag 30. Oktober 2007 **vor!** der Vorlesung

Aufgabe 1 (20 Punkte)

Gegeben sei der folgende KFPT-Ausdruck t :

$$t = \left(\left(\left(\left(\text{case}_{\text{Bool}} ((\backslash x \rightarrow ((\backslash z \rightarrow z) (\backslash y \rightarrow y))) \text{False}) \text{True}) \right) \right) \right. \\ \left. \text{of } \{ \text{False} \rightarrow (\backslash a \rightarrow a); \right. \\ \left. \left. \text{True} \rightarrow ((\backslash u \rightarrow (\backslash v \rightarrow v)) \text{True}) \} \right) (\backslash b \rightarrow b) \right) \text{False}$$

- Werten Sie den Ausdruck t in Normalordnung aus und geben Sie für jede Ein-Schritt-Reduktion die verwendete Reduktionsregel an. (12 Punkte)
- Werten Sie den Ausdruck t in einer *anderen* Reihenfolge aus, die *nicht* der Normalordnung entspricht. (8 Punkte)

Aufgabe 2 (12 Punkte)

Die folgenden KFPT-Ausdrücke seien gegeben:

- $(\backslash y \rightarrow ((\backslash x \rightarrow x) \text{True}))$
- $(\backslash y \rightarrow \text{case}_{\text{List}} y \text{ of } \{(\text{Cons } a \ b) \rightarrow b; \text{Nil} \rightarrow \text{Nil}\}) \text{True}$
- $(\backslash z \rightarrow \text{case}_{\text{Bool}} z \text{ of } \{ \text{True} \rightarrow \text{True}; \text{False} \rightarrow (\text{True Nil}) \}) \text{True}$
- $(\text{case}_{\text{Paar}} (\text{Paar True } (\backslash z \rightarrow (z \ z))) \text{ of } \{(\text{Paar } a \ b) \rightarrow (b \ b)\})$
- $(\text{case}_{\text{Bool}} (\backslash x \rightarrow \text{True}) \text{ of } \{ \text{True} \rightarrow \text{False}; \text{False} \rightarrow \text{True} \})$
- $(\text{Cons } (\text{True True}) (\text{Cons True } (\text{Cons True Nil})))$

Geben Sie für jeden der obigen Ausdrücke an, welche der folgenden Aussagen auf ihn zutreffen.

1. Der Ausdruck *ist eine WHNF*.
2. Der Ausdruck *ist eine FWHNF*.
3. Der Ausdruck *ist eine CWHNF*.
4. Der Ausdruck *hat eine WHNF*.
5. Der Ausdruck *hat eine FWHNF*.
6. Der Ausdruck *hat eine CWHNF*.
7. Der Ausdruck *ist dynamisch ungetypt*.
8. Der Ausdruck *ist direkt dynamisch ungetypt*.
9. Der Ausdruck *terminiert*.
10. Der Ausdruck *divergiert*.

Aufgabe 3 (18 Punkte)

- a) Definieren Sie in KFPTS einen Superkombinator `first_and_last`, der als Argument ein Tripel erwartet und ein Paar bestehend aus der ersten und dritten Komponente des Tripels liefert. Sie können dabei annehmen, dass die Konstruktoren `Tripel` mit Stelligkeit 3 und `Paar` mit Stelligkeit 2 vorhanden sind. (7 Punkte)
- b) Werten Sie den Ausdruck

$$\text{first_and_last } (\text{Tripel } ((\backslash x \rightarrow x) 'a') ((\backslash y \rightarrow 'a') \text{True}) 'b')$$

aus, wobei Sie für jeden Reduktionsschritt die verwendete Regel angeben. Haben Sie dabei Normalordnung verwendet? (5 Punkte)

- c) Definieren Sie nun den Superkombinator `first_and_last` in Haskell. Dafür müssen Sie eine Datei mit der Endung `.hs` anlegen in der Sie zusätzlich zur Definition des Superkombinators die Datentypen für das Paar und das Tripel wie folgt definieren können:

```
data Paar = Paar Char Char
  deriving(Show)

data Tripel = Tripel Char Char Char
  deriving(Show)
```

Testen Sie Ihre Implementierung wahlweise mit dem HUGS oder GHCi und führen Sie einige Aufrufe der Funktion `first_and_last` durch, insbesondere mit dem selben Argument wie in Aufgabenteil b). (6 Punkte)