

## 10. Übung zur Vorlesung Programmierung und Modellierung

**A10-1 Redex** Identifizieren Sie alle Redexe in den folgenden Programmausdrücken, und geben Sie an, welche davon innerste und äußerste Redexe sind.

- a)  $(1 + 2) * (4 / 5)$
- b) `snd (1 + (2 + 3), 4 + 5)`  
wobei die Funktion `snd :: (a,b) -> b` üblicherweise definiert ist durch  $\backslash(x,y) \rightarrow y$
- c)  $(\backslash x \rightarrow (1 * 2) + x) (3+4)$
- d)  $(\backslash f \rightarrow f (1 * 2)) (\backslash x \rightarrow 3+4)$

**A10-2 Auswertestrategie** Werten Sie den Programmausdruck entweder mit der Auswertestrategie Call-By-Name oder Call-By-Value aus. Welche Auswertestrategie haben Sie jeweils gewählt und warum?

*Hinweis:* Die bisher in der Vorlesung angegebenen Definitionen für Call-By-Name und Call-By-Value beziehen sich auf den Lambda-Kalkül, dem Kern funktionaler Sprachen, also das auf Folie 10-6 vorgestellte Fragment von Haskell.

- a)  $(\backslash(x,y) \rightarrow y) (1 + (2 + 3), 4 + 5)$
- b)  $(\backslash x \rightarrow x + x) ((\backslash y \rightarrow y * y) (1+1))$

**A10-3 Verzögerte Auswertung** Betrachten Sie das folgende Haskell Programm:

```
ps = iterate (+1) 0
qs = iterate (*2) 1
rs = iterate (^2) 2

iterate f x = x : iterate f (f x)

foo (_,x:xs) ys@(y:_) zs
  | x > y      = x
  | otherwise = foo ys zs xs
```

Aufgrund der verzögerten Auswertestrategie von Haskell wird z.B. die Liste `ps` anfangs nur als ein Verweis auf den Code `iterate (+1) 0` abgespeichert. Erst sobald auf das erste Element dieser Liste zugegriffen wird, wird zeigt `ps` auf die Liste `0: iterate (+1) (0+1)`. Wird später dann auch noch das zweite Element benötigt, so zeigt der Bezeichner `ps` nun auf die Liste `0:1: iterate (+1) (1+1)` im Speicher.

Bis zu welchem Element werden die Listen `ps`, `qs`, `rs` im Speicher ausgewertet, wenn nur der Aufruf `foo ps qs rs` ausgeführt wird?

**A10-4 *Faule Fibonacci Zahlen*** Definieren Sie die Liste aller Fibonacci-Zahlen in Haskell, also `fibs :: [Integer]`. Achten Sie dabei auch auf Effizienz!

*Zur Erinnerung:* Die Liste aller Fibonacci Zahlen beginnt mit 0 und 1. Die  $i$ -te Fibonacci-Zahl ist immer die Summe ihrer beiden Vorgänger.

**H10-1 *Redex II* (2 Punkte)** (Abgabeformat: Text oder PDF)

Identifizieren Sie alle Redexe in den folgenden Programmausdrücken, und geben Sie an, welche davon innerste und äußerste Redexe sind.

a) `(\ (x,y,z) -> (0+1,y*z)) (2,3+4,(\ u->5) 6)`

b) `(\ g z -> (\ f x -> f (f x)) (\ y -> 1+2) 3)`

**H10-2 *Auswertestrategie II* (3 Punkte)** (Abgabeformat: Text oder PDF)

Werten Sie den Programmausdruck entweder mit der Auswertestrategie Call-By-Name oder Call-By-Value aus:

`(\ f x -> f (f x)) ((\ y z -> y+y) (2*2)) (3+4)`

Welche Auswertestrategie haben Sie gewählt und warum?

**H10-3 *Fixpunktkombinator* (3 Punkte)** (.hs-Datei als Lösung abgeben)

Ein Wert `x0` heißt Fixpunkt einer Funktion `f`, wenn die Gleichung `f x0 = x0` gilt. Der Fixpunktkombinator `fix` berechnet Fixpunkte von Funktionen:

`fix f = f (fix f)`

Es sei `x1 = fix foo`, dann gilt offenbar `foo x1 = x1`. Wer es nicht glaubt, kann es durch Einsetzen der definierenden Gleichungen leicht nachrechnen:

`foo x1 = foo (fix foo) = fix foo = x1`

Implementieren Sie eine Funktion zur Berechnung der  $n$ -ten Fibonaccizahl mithilfe des Fixpunktkombinators, also ohne Rekursion und ohne Funktionen der Standardbibliothek!

*Hinweis:* Der Trick besteht darin, eine Funktion zu definieren, welche zwei Argumente bekommt: das erste Argument soll die partielle Zielfunktion darstellen ("Fibonacci für Argumente bis zu einer gewissen Größe"), das zweite Argument ist das tatsächliche Argument für die Zielfunktion. Die Funktion soll die partielle Zielfunktion dann um einen Schritt erweitern. Den Rest erledigt der Fixpunktkombinator.

**Hinweis:** Themenvorschläge für die Wiederholungsvorlesung der kommenden Woche bitte im Forum [die-informatiker.net](http://die-informatiker.net) sammeln.

**Abgabe:** Lösungen zu den Hausaufgaben können bis Dienstag, den 8.07.2014, 11:00 Uhr mit UniworX abgegeben werden.

Aufgrund des Klausurbonus müssen die Hausaufgaben von Ihnen alleine gelöst werden. Abschreiben bei den Hausaufgaben gilt als Betrug und kann zum Ausschluss von der Klausur zur Vorlesung führen.