

Übungen zur „Praktischen Informatik III“, WS 2003/04

Nr. 2, Besprechung bzw. Abgabe: 5. und 6. November in den Übungen

A. Mündliche Aufgaben

6. Listenkonstruktion mit ":" und "++"

Welche der folgenden Gleichungen auf Listen sind richtig? Begründen Sie kurz Ihre Antwort.

(a) $x:y:z = [x,y,z]$

(b) $xs:[] = [xs]$

(c) $[]:xs = xs$

(d) $(x:xs)++ys = x:(xs++ys)$

(e) $[[]]++xs = [[],xs]$

7. Zeichenkettenverarbeitung

Definieren Sie die folgenden Funktionen über Zeichenketten, die in Haskell als Listen von Zeichen realisiert sind: `type String = [Char]`.

(a) `delete :: String -> Int -> String`

entfernt beim Aufruf (`delete text i`) das *i*-te Zeichen von `text`.

(b) `insert :: String -> Int -> String -> String`

fügt beim Aufruf (`insert t1 i t2`) in `t1` ab der Position *i* `t2` ein.

(c) `slice :: String -> Int -> Int -> String`

bestimmt beim Aufruf (`slice text i n`) den Teiltext von `text`, der an Position *i* beginnt und *n* Zeichen umfasst.

8. Bildtransformationen II

In dem in Aufgabe 4 eingeführten Modul `Picture.hs` (siehe Vorlesungsseite) werden Bilder als Listen von Zeichenketten implementiert, d.h. Bilder sind Listen von Listen von Zeichen:

```
type Picture = [String] -- = [[Char]]
```

Implementieren Sie mittels Pattern Matching eigene Versionen `my_<name>` der folgenden im Modul `Picture.hs` definierten Funktionen:

(a) Die Funktion `flipV :: Picture -> Picture` spiegelt ein Bild vertikal.

(b) Die Funktion `aside :: Picture -> Picture -> Picture` ordnet zwei Bilder nebeneinander an.

(c) Die Funktion `superimpose :: Picture -> Picture -> Picture` legt zwei Bilder (gleichen Ausmasses) übereinander. Verwenden Sie die Funktion

```
superimposePixel      :: Char -> Char -> Char  
superimposePixel '.' '.' = '.'  
superimposePixel _ _   = '#'
```

zum Übereinanderlegen von einzelnen Zeichen.

B. Hausaufgaben

Hinweise: Die Lösungen sollten grundsätzlich schriftlich, Programme zusätzlich auf Diskette oder per E-Mail an Ihren Tutor abgegeben werden. Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt.

9. (a) Definieren Sie eine Funktion 5 Punkte

`power :: Int -> Int -> Int` / 1

so daß `(power k n)` den Wert k^n liefert. Gehen Sie davon aus, dass $n \geq 0$ gilt.

Der vordefinierte Operator `(^)` `:: Int->Int->Int` darf nicht verwendet werden.

- (b) Geben Sie eine ausführliche Auswertung des Ausdrucks `(power 3 3)` an. / 1

- (c) Geben Sie eine alternative Definition der Funktion `power` an, die folgende Gleichungen ausnutzt: / 3

$$\begin{aligned}k^{2n} &= (k^n)^2 \\ k^{2n+1} &= k * (k^n)^2\end{aligned}$$

Die Funktionen `div`, `mod :: Int -> Int -> Int` sind in Haskell vordefinierte Präfixfunktionen für die ganzzahlige Division und die Restbildung.

10. Listenoperationen 7 Punkte

Implementieren Sie mit Pattern Matching die folgenden Funktionen:

- (a) `member :: Eq t => [t] -> t -> Bool` / 2

soll beim Aufruf `member l e` feststellen, ob das Element `e` in der Liste `l` vorkommt.

Die in Haskell vordefinierte Funktion `elem :: Eq t => [t] -> t -> Bool` darf nicht verwendet werden.

- (b) `unique :: Eq t => [t] -> [t]` / 3

soll beim Aufruf `unique l` eine Liste derjenigen Elemente von `l` erstellen, die genau einmal in `l` vorkommen.

- (c) `flatten :: [[t]] -> [t]` / 2

soll eine Liste von Listen flachklopfen, indem alle Elementlisten aneinandergehngt werden. Zum Beispiel soll aus `[['g', 'h'], [], ['c']]` die Liste `['g', 'h', 'c']` werden. Auch hier existiert eine entsprechende vordefinierte Funktion `concat :: [[t]] -> [t]`, die nicht eingesetzt werden darf.

Hinweis: Die zu definierenden Funktionen sind *polymorph*, d.h. in den Typen der Funktionen taucht eine Typvariable `t` auf, die vom System durch einen beliebigen Typ ersetzt werden kann. Der sogenannte Kontext "`Eq t =>`" in den Typen von `member` und `unique` bedeutet, dass die Typvariable `t` nur durch solche Typen ersetzt werden darf, auf denen der Test auf Gleichheit implementiert ist. Dies ist bei den meisten vordefinierten Typen der Fall. `t` darf also zum Beispiel durch `Int` ersetzt werden. Können Sie sich einen Typ vorstellen, wo das Feststellen der Gleichheit von Elementen dieses Typs problematisch ist?