Aufgabenblatt zur 1. Übung

Zeitraum: 11.04. bis 15.04.2011

Achtung: Übungsraumänderung am Dienstag, 3. DS: **jetzt PHY/C213**.

1. Aufgabe (AGS 11.1):

Programmieren Sie eine Funktion `f` in Haskell, die eine Liste von Int–Zahlen als Eingabe nimmt und eine Liste von Int–Zahlen mit folgenden Eigenschaften liefert:

- Die Ergebnisliste soll nur Zahlen größer Null enthalten.
- Die Zahlen der Ergebnisliste erscheinen in umgekehrter Reihenfolge bezüglich der Zahlen der Ausgangsliste.

2. Aufgabe (AGS 11.2):

Eine Liste $\mathtt{s} = [e_1, e_2, \ldots]$ bestehe aus Funktionswerten $f(i) = e_i$ mit $i \in I\!N^+$.
f sei wie folgt definiert:

$f(1) = 1, f(2) = 1, f(3) = 1$
$f(i+3) = f(i) + f(i+1) + f(i+2)$ für $(i \geq 1)$

(a) Geben Sie eine Haskell-Funktion `fl` an, die die oben definierte Liste erzeugt.

(b) Schreiben Sie eine Haskell-Funktion `p`, die von der Liste `s` das n-te Element ermittelt.

(c) Stellen Sie die Operationsfolgen für die Ausgabe des zweiten Elementes von `s` bei Benutzung der Auswertungsstrategien „call by value“, „call by name“und „call by need“(lazy evaluation) gegenüber.

3. Aufgabe (AGS 11.15 a, b):

(a) Schreiben Sie in Haskell eine Funktion `trans`, die eine Liste von Integer-Paaren in eine Liste von Integer-Werten so transformiert, dass für jedes Paar in der Eingabeliste, falls der erste Wert dieses Paares größer als der zweite ist, in der Ergebnisliste die Differenz der beiden Werte, anderenfalls ihre Summe erscheint. Die Reihenfolge der Zahlen in der Ergebnisliste soll der Reihenfolge der zu verarbeitenden Paare der Eingabeliste entsprechen.

(b) Eine Haskell-Funktion `liste` soll folgendes leisten:

Sie bekommt als Argumente einen Integer-Wert `n` und eine aufsteigend geordnete Liste von voneinander verschiedenen Integer-Werten.
Falls der Wert `n` in der Liste enthalten ist, soll er entfernt, ansonsten eingefügt werden, wobei die Ordnung der Liste erhalten bleiben soll. Geben Sie die Funktion `liste` einschließlich der Typ-Definitionen an.

4. Aufgabe (AGS 11.3*):

Gegeben sei der Listentyp $[l_1, l_2, ..., l_n]$ mit $l_1, l_2, ..l_n$ jeweils vom Typ `[Int]`. Es soll von jeder Liste l dieses Typs die Länge der längsten Liste, also $\max\{\text{length}(l_i) | 1 \leq i \leq n\}$, als Ergebnis berechnet werden.

(a) Geben Sie ein Beispiel für eine Liste dieses Listentyps an, und nennen Sie das zugehörige Ergebnis.

(b) Schreiben Sie in Haskell eine Funktion `max_length :: [[Int]] -> Int`, die diese Aufgabe erfüllt, und geben Sie abschließend einen Funktionsaufruf an. Wenn Sie in `max_length` Hilfsfunktionen nutzen, müssen Sie für diese die Typdeklarationen und den Programmcode aufschreiben.

Hilfestellung: Ermitteln sie zunächst aus der Liste von Listen die Liste der Längen dieser Listen.