Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik und Mathematik/ Institut für Informatik
Johann Wolfgang Goethe-Universität Frankfurt am Main

Einführung in die funktionale Programmierung

Wintersemester 2007/2008

Aufgabenblatt Nr. 5

Abgabe: Dienstag 20. November 2007 **vor!** der Vorlesung

Aufgabe 1 (10 Punkte)

Im Ausdruck

```
filter even (concat (map concat (filter ((>1) . length) xs)))
```

sei xs eine Liste vom Typ `[[[Integer]]]`.

Geben Sie eine *List Comprehension* an, die weder `map`, `filter`, noch `concat` enthält, und das gleiche Resultat wie obiger Ausdruck liefert.

Aufgabe 2 (12 Punkte)

Überführen Sie die List Comprehension

```
[w | v <- [[x,y] | x <- [1,3..20], y <- [2,4..20], y == x+1], w <- v]
```

mit dem in der Vorlesung vorgestellten Verfahren in einen Haskell-Ausdruck ohne List Comprehensions.

Aufgabe 3 (28 Punkte)

Matrizen können als Listen von (gleichlangen) Listen in Haskell dargestellt werden, z.B. kann die Matrix

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

als Liste

```
[
   [1,  2,  3,  4 ],
   [5,  6,  7,  8 ],
   [9,  10, 11, 12]
]
```

dargestellt werden.

Damit Instanzen von Typklassen für Matrizen definiert werden können, definieren wir Matrizen mittels `newtype` polymorph über den Einträgen als:

```
newtype Matrix a = Matrix [[a]]
```

a) Definieren Sie je eine geeignete Instanz der Klasse `Eq` und der Klasse `Show` für Matrizen. (5 Punkte)

b) Implementieren Sie eine Instanz der Konstruktorklasse `Functor` für Matrizen. (5 Punkte)

c) Implementieren Sie eine Instanz der Klasse `Num` für Matrizen, wobei Sie

   - für die Methoden `(+)`, `(*)` die Matrizenaddition und -multiplikation implementieren sollen;
   - für die Methode `negate` alle Einträge der Matrix negieren sollen;
   - für die Methode `abs` alle Einträge der Matrix mit ihrem Absolutbetrag ersetzen sollen;
   - für die Klassenmethoden `fromInteger` und `signum` eine geeignete Wahl treffen können. (11 Punkte)

d) Implementieren Sie eine List Comprehension, die alle $(n \times m)$ Matrizen, gefüllt mit Nullen generiert, wobei die Reihenfolge der Genierung derart sein sollte, dass zuerst alle Matrizen für $(n+m) = 2$ generiert werden, anschließend für $(n+m) = 3$ usw. D.h. die generierte Liste beginnt mit

$$\begin{pmatrix} 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}, \ldots$$

(7 Punkte)