WiSe 2020/2021

Funktionale Programmierung

3. Übungsblatt (für das Tutorium)

Prof. Dr. Margarita Esponda

Ziel: Auseinandersetzung mit polymorphe Funktionen, Listengeneratoren und Funktionen höherer Ordnung.

1. Aufgabe (4 Punkte)

Betrachten Sie folgende Funktionsdefinition, die die Menge aller möglichen Sublisten der Elemente einer Liste berechnet:

```
powerList [] = [[]]
powerList (x:xs) = powerList xs ++ map (x:) (powerList xs)
```

Schreiben Sie alle Reduktionsschritte für den folgenden Ausdruck:

```
powerList  [1, 2, 3, 4]
```

2. Aufgabe (2 Punkte)

Was ist die Normalform folgendes Ausdrucks? Berechnen Sie die Lösung ohne den Ausdruck in dem Haskell-Interpreter einzugeben. Begründen Sie Ihre Lösung.

```
[if x==y then 'o' else '.' | x <- [1..5], y <- [1..7], (x+y)<9]
```

3. Aufgabe (7 Punkte)

Definieren Sie eine polymorphe Funktion **poss**, die die Positionen eines Elements innerhalb einer Liste wiederum in einer Liste zurückgibt.

Anwendungsbeispiel:

```
poss  'e'  "Freie Universität Berlin"  =>  [2, 4, 10, 19]
```

a) Definieren Sie zuerst die Funktion nur unter Verwendung von expliziter Rekursion und Akkumulator-Technik.

b) Definieren Sie die Funktion unter sinnvoller Verwendung von Listengeneratoren und Funktionen höherer Ordnung.