Technische Universität Dresden
Institut für Theoretische Informatik
Dr. Monika Sturm

Wintersemester 08/09

Funktionale Programmierung und Typtheorie

6. Übungsblatt

Aufgabe 1

Gegeben seien die folgenden Definitionen:

```
data BBaum a = Empty | Node a (BBaum a) (BBaum a)
```

```
foldTree f e Empty        = e
foldTree f e (Node x l r) = f x (foldTree f e l) (foldTree f e r)
size = foldTree (\n l r -> 1 + l + r) 0
```

a) Geben Sie die Typen der Funktionen `foldTree` und `size` an.

b) Schreiben Sie eine HASKELL-Funktion `allTrees`, die zu einer Zahl n eine Liste aller Binärbäume `t :: BBaum ()` mit n Knoten, d.h. `size t == n`, erzeugt.
Beispielsweise sollen für `n = 2` die folgenden Bäume erzeugt werden:
`(Node () (Node () Empty Empty) Empty)`
`(Node () Empty (Node () Empty Empty))`.

Aufgabe 2

Definieren Sie eine Funktion `perms xs`, die im Ergebnis eine Liste ausgibt, die alle Permutationen der Elemente der Argumentliste enthält.

```
perms [0,1,2]
=> [[0,1,2],[0,2,1],[1,0,2],[1,2,0],[2,0,1],[2,1,0]]
```

Hinweis: Die Reihenfolge der Permutationen in der Ergebnisliste spielt keine Rolle, wohl aber die der einzelnen Elemente in den Teillisten. Es wird vorausgesetzt, dass die an `perms` übergebene Liste keine Duplikate enthält. Es werden somit nur Permutationen ohne Wiederholung berechnet.