

## Übungen zur „Praktischen Informatik III“, WS 2006/07

Nr. 3, Abgabe: 7. November 2006 vor der Vorlesung

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

### 6. Listenpermutationen

4 Punkte

Definieren Sie unter Verwendung von Listenabstraktionen

- (a) eine Funktion `removeList :: [a] -> [(a, [a])]`, die zu einer Liste alle Möglichkeiten bestimmt, ein Element zu entfernen. Als Ergebnis soll eine Liste aller möglichen Paare (entferntes Element, Restliste) zurückgegeben werden. / 2  
Beispiel: `removeList [1,2,3] =>* [(1, [2,3]), (2, [1,3]), (3, [1,2])]`
- (b) eine Funktion `perms :: [a] -> [[a]]`, die zu einer Liste alle möglichen Permutationen der Listenelemente bestimmt. / 2  
Beispiel:  
`perms [1,2,3] =>* [[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]]`

### 7. Heap Trees

5 Punkte

Ein binärer Baum heißt *Haldenbaum* (*heap tree*), falls die Knotenmarkierungen so geordnet sind, dass die Markierung eines Knotens immer kleiner oder gleich den Markierungen aller Nachfolgerknoten sind.

Implementieren Sie die folgenden Operationen auf Haldenbäumen der Form

```
data HTree a = Null | Node a (HTree a) (HTree a)
  deriving Show
```

- (a) `isEmpty :: HTree a -> Bool` testet, ob ein Haldenbaum leer ist. / 1  
`minElem :: HTree a -> a` ermittelt das kleinste Element.
- (b) `insert :: Ord a => a -> HTree a -> HTree a` fügt ein Element ein. / 2
- (c) `deleteMin :: Ord a => HTree a -> HTree a` entfernt das kleinste Element. / 2  
**Hinweis:** Definieren Sie zum Vereinigen zweier Haldenbäume eine Hilfsfunktion  
`merge :: Ord a => HTree a -> HTree a -> HTree a`.

### 8. Ausdrucksvereinfachung

3 Punkte

Gegeben sei der algebraische Datentyp `Expr` zur Definition von einfachen arithmetischen Ausdrücken:

```
data Expr = Var String | Val Int
  | Add Expr Expr | Mult Expr Expr
  deriving Show
```

Definieren Sie eine Funktion `simplify :: Expr -> Expr`, die in einem Ausdruck konstante Teilausdrücke durch ihren Wert ersetzt.

Beispiel: `simplify (Add (Mult (Val 5) (Val 3)) (Var "x"))`  
`=>* (Add (Val 15) (Var "x"))`