

# 1. Übung zu “Parallelität in funktionalen Sprachen”, SS 2006

Abgabe schriftlicher Aufgaben: Do, 27.April 2006 (vor der Vorlesung)

**Hinweise:** Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt.

Lösungen sollen grundsätzlich schriftlich abgegeben werden, Programme (mit mehr als drei Zeilen) zusätzlich per Mail an [berthold@mathematik.uni-marburg.de](mailto:berthold@mathematik.uni-marburg.de).

## Aufgaben

### 1.1 Umstellen von Listen

3 Punkte

Schreiben Sie eine Haskell-Funktion `transpose :: [[a]] -> [[a]]`, welche eine Liste von Zeilen in eine Liste von Spalten umwandelt (also transponiert):

Kann Ihre Funktion auch dann Ergebnisse liefern, wenn die Listen im Argument unterschiedliche Länge haben?

```
*Main> mytranspose [[1,2],[3,4,5],[],[6,7,8]]
[[1,3,6],[2,4,7],[5,8]]
*Main>
```

### 1.2 Programmanalyse und Parallelisierbarkeit

9 Punkte

Die `transpose`-Funktion lässt sich etwa im folgenden Programm verwenden:

```
main :: IO ()
main = putStr (concat (map find hidden))

-- find:
-----
find :: [Char] -> String
find word = word ++ " " ++ (concat dirs) ++ "\n"
  where dirs = map snd (forw ++ back)

forw = filter (any (contains word) . fst)
      [(r,"right "), (d,"down "), (dl,"downleft "), (ul,"upleft ")]
back = filter (any (contains drow) . fst)
      [(r,"left "), (d,"up "), (dl,"upright "), (ul,"downright ")]
drow = reverse word

r = grid
d = transpose grid
dl = diagonals grid
ul = diagonals (reverse grid)
```

Bitte wenden!

```

-- diagonals:
-- zipinit:
-----
diagonals [r]      = map (:[]) r
diagonals (r:rs) = zipinit r ([]:diagonals rs)

zipinit []        ys      = ys
zipinit (x:xs) (y:ys) = (x : y) : zipinit xs ys

-- contains:
-- prefix:
-- suffixes:
-----
contains xs ys = any (prefix xs) (suffixes ys)

suffixes [] = []
suffixes xs = xs : suffixes (tail xs)

prefix []      ys      = True
prefix xs     []      = False
prefix (x:xs) (y:ys) = x == y && prefix xs ys

-----
grid = [['Y', 'I', 'O', 'M', 'R', 'E', 'S', 'K', 'S', 'T'],
        ['A', 'E', 'H', 'Y', 'G', 'E', 'H', 'E', 'D', 'W'],
        ['Z', 'F', 'I', 'A', 'C', 'N', 'I', 'T', 'I', 'A'],
        ['N', 'T', 'O', 'C', 'O', 'M', 'V', 'O', 'O', 'R'],
        ['E', 'R', 'D', 'L', 'O', 'C', 'E', 'N', 'S', 'M'],
        ['Z', 'O', 'U', 'R', 'P', 'S', 'R', 'N', 'D', 'A'],
        ['O', 'Y', 'A', 'S', 'M', 'O', 'Y', 'E', 'D', 'L'],
        ['R', 'N', 'D', 'E', 'N', 'L', 'O', 'A', 'I', 'T'],
        ['F', 'I', 'W', 'I', 'N', 'T', 'E', 'R', 'R', 'C'],
        ['F', 'E', 'Z', 'E', 'E', 'R', 'F', 'T', 'F', 'I'],
        ['I', 'I', 'D', 'T', 'P', 'H', 'U', 'B', 'R', 'L'],
        ['C', 'N', 'O', 'H', 'S', 'G', 'E', 'I', 'O', 'N'],
        ['E', 'G', 'M', 'O', 'P', 'S', 'T', 'A', 'S', 'O'],
        ['T', 'G', 'F', 'F', 'C', 'I', 'S', 'H', 'T', 'H'],
        ['O', 'T', 'B', 'C', 'S', 'S', 'N', 'O', 'W', 'I']]

hidden = ["COSY", "SOFT", "WINTER", "SHIVER", "FROZEN", "SNOW",
          "WARM", "HEAT", "COLD", "FREEZE", "FROST", "ICE" ]

```

- 
- (a) Geben Sie an, wozu dieses Programm und die enthaltenen Funktionen dienen. Vervollständigen Sie die Kommentare zu den Hilfsfunktionen und die fehlenden Typdeklarationen.
- (b) Welche Teile sind voneinander unabhängig und könnten also gleichzeitig – parallel – ausgeführt werden?  
Geben Sie allgemeine Regeln zum Auffinden von Parallelität in funktionalen Programmen an.