

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

Nr. 8, Abgabe: 12. Dezember 2006 vor der Vorlesung

---

19. Interaktive Ein-/Ausgabe mit `interact`

2 Punkte

Schreiben Sie unter Verwendung von `interact` ein Programm, das wiederholt eine Eingabezeile in Großbuchstaben wieder ausgibt, bis die Eingabe einer Leerzeile erfolgt. Ihr Programm sollte immer wieder Eingabeaufforderungen ausgeben und eine Schlussausgabe durchführen.

20. Listeninduktion

6 Punkte

- (a) Gegeben seien Funktionen  $h :: a \rightarrow b \rightarrow b$  und  $h' :: b \rightarrow a \rightarrow b$  sowie  $e :: b$ . Für alle  $x, z :: a$  und  $y :: b$  gelte:  $h\ x\ (h'\ y\ z) = h'\ (h\ x\ y)\ z$ . / 4

Beweisen Sie mittels Listeninduktion:

- i. Für alle  $x :: a$ ,  $y :: b$  und endliche  $xs :: [a]$  gilt:

$$h\ x\ (\text{foldl}\ h'\ y\ xs) = \text{foldl}\ h'\ (h\ x\ y)\ xs$$

- ii. Sei zusätzlich  $h\ x\ e = h'\ e\ x$  für beliebige  $x :: a$ .

Dann gilt für alle endlichen  $xs :: [a]$ :

$$\text{foldr}\ h\ e\ xs = \text{foldl}\ h'\ e\ xs$$

- (b) Definieren Sie die Listenspiegelung `reverse :: [a] -> [a]` als Instanz von `foldr` / 2 und von `foldl` und zeigen Sie mit (a) die Äquivalenz der beiden Faltungsinstanzen.

21. Programmtransformation

4 Punkte

Gegeben seien die folgenden Funktionen

```
my_unlines    :: [String] -> String
my_unlines ccs = concat (addnewline ccs)
```

```
addnewline    :: [String] -> [String]
addnewline [] = []
addnewline (cs:css) = (cs ++ "\n") : addnewline css
```

```
concat        :: [[a]] -> [a]
concat []     = []
concat (xs:xss) = xs ++ concat xss
```

- (a) Analysieren Sie den Zeitaufwand der Funktion `my_unlines`. / 1
- (b) Entwickeln Sie eine zu `my_unlines` äquivalente Funktion, die effizienter arbeitet, d.h. mit weniger Schritten dasselbe Ergebnis bestimmt. / 1
- (c) Zeigen Sie die Äquivalenz Ihrer Funktion zu `my_unlines`. / 2