

Übungen zur „Praktischen Informatik III“, WS 2005/06

Nr. 9, Abgabe: 10. Januar 2006 vor der Vorlesung



Frohe Weihnachten und ein gutes Jahr 2006!

A. Hausaufgaben

39. Unendliche Listen

3 Punkte

Definieren Sie mit Hilfe von Prozessnetzen:

(a) Die Liste aller Zweierpotenzen: $(2^n \mid n \geq 0)$.

/ 1

(b) Die geschachtelte Liste aller Binomialkoeffizienten

/ 2

$$\left(\left(\binom{n}{k} \mid 0 \leq k \leq n \right) \mid n \geq 0 \right).$$

40. Polynomfunktionen

9 Punkte

Polynome in einer Variablen können in Haskell als unendliche Listen von Zahlen dargestellt werden: `type Poly = [Float]`.

Beispielsweise repräsentiert die Liste `(1:0:2:(-4):0:3:repeat 0)` das Polynom $1 + 2x^2 - 4x^3 + 3x^5$.

Definieren Sie die folgenden Operationen über Polynomen:

(a) `scale :: Float -> Poly -> Poly` zur Multiplikation mit einem Skalar:

/ 1

$$a * \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} a b_i x^i$$

(b) `addPoly :: Poly -> Poly -> Poly` zur Addition von zwei Polynomen

/ 1

$$\sum_{i=0}^{\infty} a_i x^i + \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i + b_i) x^i$$

(c) `mulPoly :: Poly -> Poly -> Poly` zur Multiplikation von Polynomen

/ 2

$$\sum_{i=0}^{\infty} a_i x^i * \sum_{i=0}^{\infty} b_i x^i = a_0 * \sum_{i=0}^{\infty} b_i x^i + x * \sum_{i=0}^{\infty} a_{i+1} x^i * \sum_{i=0}^{\infty} b_i x^i$$

(d) `divPoly :: Poly -> Poly -> Poly` zur Division von Polynomen

/ 2

$$\frac{\sum_{i=0}^{\infty} a_i x^i}{\sum_{i=0}^{\infty} b_i x^i} = \frac{a_0}{b_0} + x * \frac{\sum_{i=0}^{\infty} \left(a_{i+1} - \frac{a_0}{b_0} b_{i+1} \right) x^i}{\sum_{i=0}^{\infty} b_i x^i}$$

(e) Berechnen Sie die Liste der Fibonacci-Zahlen durch Ausnutzung der folgenden Gleichung:

/ 1

$$\frac{1}{1-x-x^2} = 1 + x + 2x^2 + 3x^3 + 5x^4 + 8x^5 + \dots$$

(f) Überladen Sie die arithmetischen Operationen (+),(-),(*) und (/) so, dass Sie auch für Polynomoperationen verwendet werden können. Hierzu ist es notwendig, die überladenen Operationen auf dem folgenden Datentyp zu erklären:

/ 2

```
newtype Polynom = P [Float]
```

Im Gegensatz zu dem Typsynonym `Poly`, das nicht in Instanzendeklarationen verwendet werden kann, wird mittels `newtype` ein neuer algebraischer Datentyp mit genau einem Konstruktor definiert.

B. Mündliche Aufgaben

41. Strikte/nicht-strikte Auswertung

Zum Sortieren kann die Funktion `iSort` verwendet werden:

```
iSort :: Ord a => [a] -> [a]
iSort = foldr insert []
```

```
insert :: Ord a => a -> [a] -> [a]
insert x [] = [x]
insert x (y:ys) | x <= y = x : y : ys
                 | otherwise = y : insert x ys
```

Geben Sie für den Ausdruck `iSort [3,4,2,1]` eine

(a) Leftmost-Innermost-Auswertung und eine (b) Leftmost-Outermost-Auswertung an. Zeigen Sie, wo sich die beiden Reduktionsstrategien voneinander unterscheiden.

42. Normalformen

Stellen Sie fest, ob die folgenden Ausdrücke in Normalform beziehungsweise in Kopfnormalform vorliegen. Falls die Normalform existiert, so geben Sie diese an.

- (a) `\x -> x + (3 * 2)` (d) `filter (>0)`
(b) `(\y -> y + 6) 5` (e) `map square (from 2)`
(c) `head (from 2)`

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