PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen

D-35032 Marburg
Hans Meerwein Straße
Lahnberge
28. Januar 2004

Übungen zur „Praktischen Informatik III“, WS 2003/04

Nr. 12 (letztes Blatt), Besprechung bzw. Abgabe: 4. und 5. Februar in den Übungen

---

Hinweis: Die **Klausur** wird am

Mittwoch, 18. Februar 2004 von 8:45 bis 10:45 Uhr im HG5

(Biegenstraße) geschrieben. Es sind keine Hilfsmittel erlaubt. Mitzubringen ist lediglich Schreibzeug. Das Papier wird gestellt.

Die **Einsicht und Rückgabe** der Klausur findet am Mittwoch, 25. Februar 2004, von 15:00 bis 15:45 Uhr im Seminarraum V (Ebene D5, Lahnberge) statt. Der Termin einer eventuellen Nachholklausur wird dann bekanntgegeben.

A. Mündliche Aufgaben

56. 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)`

    (b) `(\y -> y + 6) 5`

    (c) `head (from 2)`

    (d) `filter (>0)`

    (e) `map square (from 2)`

57. Strombasierte Ein-/Ausgabe

    Mit der vordefinierten Funktion `interact :: (String -> String) -> IO ()` kann interaktives Ein-/Ausgabeverhalten strombasiert definiert werden. Die als Parameter übergebene Funktion vom Typ `(String -> String)` definiert, wie ein Eingabestrom von Zeichen auf einen Ausgabestrom abgebildet wird. Die bedarfsgesteuerte Auswertung dieser Funktion mittels `interact` bewirkt, dass von Eingaben abhängige Ausgaben genau dann erfolgen, wenn die entsprechenden Eingaben gemacht werden.

    Schreiben Sie unter Verwendung von `interact` ein Programm, das wiederholt eine Eingabezeile in Großbuchstaben wieder ausgibt, bis die Eingabe einer Leerzeile erfolgt (vgl. Aufgabe 26(a), Übung 6).

    Ergänzen Sie Ihr Programm um Eingabeaufforderungen und eine Schlussausgabe.

B. Hausaufgaben

58. 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 $((\binom{n}{k} \mid 0 \leq k \leq n) \mid n \geq 0)$. / 2

59. 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

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

(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.