

Funktionale Programmierung

2. Übungsblatt

Prof. Dr. Margarita Esponda

1. Aufgabe (3 Punkte)

Gegeben sei folgende Funktionsdefinition:

```
bin2dec :: [Int] -> Int
bin2dec bits = bin2dec' 0 bits
    where
        bin2dec' ac [b]    = 2*ac + b
        bin2dec' ac (b:bs) = bin2dec' (2*ac + b) bs
```

Reduzieren Sie folgenden Ausdruck, schreiben Sie die einzelnen Schritte bis zur Normalform.

`bin2dec [0,1,0,1,1,0]` => ...

2. Aufgabe (6 Punkte)

Die Zahl π kann mit der folgenden unendlichen Seriensumme berechnet werden:

$$\pi = \sum_{k=0}^{\infty} \frac{(2^{k+1})(k!)^2}{(2k+1)!}$$

Definieren Sie eine Haskell Funktion **roughlyPI**, die bei Eingabe einer natürlichen Zahl **k** die Seriensumme von **0** bis zum **k**-Wert berechnet.

Anwendungsbeispiel:

roughlyPI 1000 => 3.1415926535897922

3. Aufgabe (4 Punkte)

Schreiben Sie eine rekursive Funktion, die einen Text als Argument bekommt und alle Zeichen, die nicht Klammern sind, aus dem Text entfernt.

Anwendungsbeispiel:

onlyParenthesis "[(2+7.0)*a-(xyz), {word}]" => "[()(){}]"

4. Aufgabe (4 Punkte)

Eine Hexagonalzahl ist eine Zahl der Form $2n^2 - n$. Schreiben Sie eine rekursive Haskell-Funktion **hexagonalNums**, die bei Eingabe einer natürlichen Zahl **n** die ersten **n** Hexagonalzahlen in einer Liste zurückgibt.

Anwendungsbeispiel:

hexagonalNums 9 => [0, 1, 6, 15, 28, 45, 66, 91, 120, 153]

5. Aufgabe (5 Punkte)

Schreiben Sie eine rekursive Haskell-Funktion, die aus einer Liste von Zahlen den Durchschnitt aller Zahlen, die innerhalb des Intervalls $[a, b]$ liegen, berechnet.

Anwendungsbeispiel:

averageInInterval 2 5 [2.0, 3.0, 5.0, 1.0, 0.0, 1.0] => 3.333

6. Aufgabe (4 Punkte)

Schreiben Sie eine Funktion, die bei Eingabe einer positiven Zahl die Einsen der Binärstellung der Zahl addiert (Quersumme der Binärdarstellung der Zahl berechnet).

7. Aufgabe (4 Punkte)

Definieren Sie eine Haskell-Funktion, die bei Eingabe einer Zahl in Hexadezimal-Darstellung die Oktal-Darstellung der Zahl berechnet.

Die Zahl soll als Zeichenkette eingegeben werden.

Anwendungsbeispiel:

hex2okt "1F81F8" => "07700770"

Wichtige Hinweise:

- 1) Verwenden Sie geeignete Namen für Ihre Variablen und Funktionsnamen, die den semantischen Inhalt der Variablen oder die Semantik der Funktionen wiedergeben.
- 2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.
- 3) Kommentieren Sie Ihre Programme.
- 4) Verwenden Sie geeignete lokale Funktionen und Hilfsfunktionen in Ihren Funktionsdefinitionen.
- 5) Geben Sie für alle Funktionen die entsprechende Signatur an.
- 6) Schreiben Sie getrennte Test-Funktionen für alle Aufgaben.
- 7) Die Lösungen sollen elektronisch (nur Whiteboard-Upload) abgegeben werden. **Keine verspätete Abgabe per Email ist erlaubt.**