4. Übung

ALP 1: Funktionale Programmierung

WS 14/15

Klaus Kriegel

Abgabe: 18.11.2014, 10:15 Uhr

Aufgabe 1:

Polynome

(2+3+3) Punkte

Ein reelles Polynom vom Grad n ist ein Ausdruck der Form $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$

wobei alle Koeffizienten a_i reelle Zahlen sind und $a_n \neq 0$ ist. Man kann ein Polynom an einer Stelle $r \in \mathbb{R}$ auswerten, indem die Variable x durch die Zahl r ersetzt und der Wert des so entstandenen arithmetischen Ausdrucks ausgerechnet wird.

- a) Implementieren Sie eine Haskell-Funktion
- eval3 :: Float -> Float -> Float -> Float -> Float -> Float zur Auswertung von Polynomen vom Grad 3, wobei die erste Eingabestelle dem Wert r und die folgenden Stellen die Koeffizienten in der Reihenfolge a_3, a_2, a_1, a_0 angeben.
- b) Implementieren Sie eine Haskell-Funktion

nullstellenTest2 :: Float -> Float -> Float -> Bool,

die für ein durch eine Koeffizientenfolge a_2, a_1, a_0 gegebenes Polynom vom Grad ≤ 2 testet, ob es eine reelle Nullstelle hat.

Achtung: Hier kann a₂ auch den Wert 0 haben, dann ist der Grad kleiner als 2.

c) Implementieren Sie eine Haskell-Funktion

smallestVal :: Float -> Float -> Float,

die für ein durch eine Koeffizientenfolge a_2, a_1, a_0 gegebenes Polynom vom Grad ≤ 2 den kleinsten Absolutbetrag berechnet, den das Polynom bei der Auswertung annehmen kann

Hinweis: Überlegen Sie, was das mit der Teilaufgabe b) zu tun hat.

Aufgabe 2:

Char

(2+2+3) Punkte)

Einige Verschlüsselungsverfahren beruhen auf Buchstabenpermutationen. Solche Verfahren sind aber nicht sicher und können leicht "geknackt" werden! Die zyklische Verschiebung der Buchstaben im Alphabet soll bereits von Caesar benutzt worden sein.

- a) Sei Z(k,n) die zyklische Zahlenfolge $k,k+1,k+2,\ldots$, $k+n-1,k,k+1,\ldots$. Implementieren Sie eine Funktion cyclicShift :: Int -> Int -> Int -> Int -> Int, die bei einer Eingabe k n m s die Zahl m aus Z(k,n) zyklisch um s Einheiten verschiebt. Wenn m nicht in Z(k,n) liegt, soll m ausgegeben werden.
- b) Implementieren Sie eine Funktion caesarForCapitals :: Char -> Int -> Char, die bei einer Eingabe ch k das Zeichen ch um k Stellen im zyklischen Alphabet verschiebt, wenn ch ein Großbuchstabe ist, anderenfalls soll ch unverändert bleiben.
- c) Implementieren Sie eine Funktion mirrorForCapitals :: Char -> Char -> Char, die bei einer Eingabe ch1 ch2 von zwei Großbuchstaben das Zeichen ch1 im zyklischen Alphabet um das Zentrum ch2 spiegelt. Ist eines der Eingabezeichen kein Großbuchstabe, dann soll ch1 ausgegeben werden.

 --> Seite 2

Aufgabe 3: Erweiterter Euklidischer Algorithmus (3 + 2 Punkte)

- a) Bestimmen Sie mit dem erweiterten Euklidischen Algorithmus zwei ganze Zahlen s und t, so dass 54s+35t=1 gilt.
- b) Begründen Sie, dass 56s+35t=1 keine ganzzahlige Lösung besitzt. Achtung: Das Argument, dass der erweiterte Euklidische Algorithmus keine entsprechende Lösung liefert, ist keine ausreichende Begründung.