8. Aufgabenblatt zur Vorlesung

Informatik A

WiSe 2015/16

Wolfgang Mulzer

Abgabe am 15. Dezember 2015 bis 12 Uhr in die Tutorenfächer

Aufgabe 1 Häufigkeiten mit Funktionen höherer Ordnung *10 Punkte*

Auf der Veranstaltungswebsite finden Sie das Haskell-Skript zur 14. und zur 15. Vorlesung. In der 15. Vorlesung wurde die Häufigkeitsanalyse unter Verwendung von Funktionen höherer Ordnung neu implementiert. Vergleichen Sie die beiden Implementierungen. Beantworten Sie dann folgende Fragen:

(a) Erklären Sie die neue Definition von `haeufigkeit`. Warum ist der Parameter `cs` nicht aufgeführt?

(b) Was macht die Funktion `scanr`? Geben Sie ein einfaches Beispiel für eine Anwendung von `scanr`, das nicht im Vorlesungsskript vorkommt.

(c) Wie unterscheidet sich die Funktion `erhoehe` aus der 15. Vorlesung von der Funktion `erhoehe` aus der 14. Vorlesung?

(d) Liefert die Funktion `haeufigkeit` aus der 15. Vorlesung das gleiche Ergebnis wie in der 14. Vorlesung? Begründen Sie Ihre Antwort im Detail.

(e) Was macht die Funktion `foldl1` in der neuen Implementierung von `sortiere`?

(f) Was bedeutet der Ausdruck `(/= erster)` in der neuen Implementierung von `sortiere`?

Aufgabe 2 Sortieren, Auswählen, Suchen *10 Punkte*

(a) Schreiben Sie die Funktion `sortiere` aus der Vorlesung so um, dass man sie auf beliebige Typen aus der Typklasse `Ord` anwenden kann.

(b) Sei `xs::[t]` eine Liste, wobei `t` ein Typ aus der Typklasse `Ord` ist. Sei n die Länge von `xs`. Der *Median* von `xs` ist der Eintrag, der an Position $\lfloor n/2 \rfloor$ steht, nachdem `xs` sortiert wurde.

Schreiben Sie eine Funktion `median`, die diesen bestimmt.

(c) Implementieren Sie eine Funktion `isSubstr:: (String, String) -> Bool`. Diese sollen testen, ob in dem Eingabestringpaar der erste String *Teilstring* des zweiten ist.

Dabei ist ein String Teilstring eines anderen, wenn er als Teilwort von aufeinander folgenden Zeichen auftritt. Zum Beispiel ist "orma" ein Teilstring von "informatika", "ioa" aber nicht. Groß– und Kleinschreibung soll ignoriert werden.

Geben Sie alle Signaturen an und machen Sie geeignete Testläufe. Versehen Sie Ihr Skript mit sinnvollen Kommentaren.

Aufgabe 3 Listen *10 Punkte*

(a) Sei `xs::[t]` eine Liste, wobei `t` ein Typ aus der Typklasse `Show` ist. Schreiben Sie eine Funktion `druckeListe`, welche die Einträge aus `xs` einzeln Zeile für Zeile ausgibt.

Hinweis: Benutzen Sie `putStr`. Was ist der Rückgabetyp?

(b) Eine *Listenzahl* ist eine Folge von Ziffern, welche als Liste dargestellt wird und eine natürliche Zahl repräsentiert. Beispielsweise repräsentiert die Listenzahl `[1,4,3,5]` die Zahl 1435.

Benutzen Sie `foldl`, um eine Funktion `dec2int` zu implementieren, welche eine Listenzahl in den zugehörigen `Int` umwandelt. Zum Beispiel soll `dec2int [1,4,3,5]` das Ergebnis `1435` haben.

(c) Schreiben Sie eine Funktion `addiere::[Int]->[Int]->[Int]`, welche zwei gegebene Listenzahlen addiert. Ihre Funktion soll direkt auf Listenzahlen arbeiten. Ein Umweg über `Int` oder `Integer` ist nicht erlaubt.

Achtung: Bei einer gültigen Listenzahl sind nur die Ziffern 0 bis 9 erlaubt. Führende Nullen sind auch nicht zulässig.

Geben Sie alle Signaturen an und machen Sie geeignete Testläufe. Versehen Sie Ihr Skript mit sinnvollen Kommentaren.