Hochschule Flensburg
Fachbereich 3
Angewandte Informatik

Jan Christiansen

Hochschule Flensburg
University of Applied Sciences

5. Übung zur Vorlesung „Fortgeschrittene funktionale Programmierung“

Laziness und Faltungen

Labor am Freitag, 28. Oktober 2016, 12:15 Uhr

Aufgabe 1 - Ströme

Wir betrachten einen polymorphen Datentypen *Stream a*, der Ströme darstellt. Ströme sind eine Art unendliche Liste. Das heißt, mit Hilfe des Datentyp *Stream* lassen sich — im Gegensatz zum Listendatentyp — nur unendliche Listen darstellen.

- Definieren Sie den Datentyp *Stream a*.
- Definieren Sie eine Funktion $toList :: Stream\ a \to [a]$, die Sie zum Testen ihrer Funktionen nutzen können.
- Definieren Sie eine Funktion $repeatStream :: a \to Stream\ a$, die einen Strom liefert, der unenlich häufig das gegebene Element enthält.
- Definieren Sie eine Funktion $mapStream :: (a \to b) \to Stream\ a \to Stream\ b$, die eine Funktion auf alle Elemente eines Stromes anwendet.
- Definieren Sie eine Funktion $iterateStream :: (a \to a) \to a \to Stream\ a$, die einen Startwert bekommt und eine Funktion, die beschreibt, wie aus einem Wert ein neuer Wert generiert wird. Für alle Elemente des Stroms gilt, dass das Nachfolger-Element durch Anwendung der Funktion aus seinem Vorgänger erzeugt wird.

Mit Hilfe der bisher definierten Funktionen wollen nun ein paar Ströme definieren.

- Die Konstante $nats :: Stream\ Int$ generiert einen Strom von allen natürlichen Zahlen in aufsteigender Reihenfolge, startend mit 0.
- Die Konstante $ruler :: Stream\ Int$ generiert die Ruler-Sequenz. Die ersten Zahlen der Ruler-Sequenz sind zum Beispiel die folgenden.

  $$0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 4, \ldots$$

  Dabei enthält der Strom an Position i den größten Exponenten n, so dass 2^n die Zahl i teilt.
- Versuchen Sie die Definition der Konstante *ruler* so zu ändern, dass sie ohne die Verwendung des Teilbarkeitstest auszukommen. Definieren Sie dazu eine Funktion *interleaveStreams* und überlegen Sie sich, dass man den Strom der natürlichen Zahlen mit Hilfe dieser Funktion durch die Verknüpfung von zwei Strömen realisieren kann. Verändern Sie dann Ihre Definition von *ruler* Stück für Stück, bis Sie eine Definition erhalten, die ohne den Teilbarkeitstest auskommt. Sie müssen sich dazu überlegen, wie die oben definierten Funktionen miteinander interagieren.

Aufgabe 2 - Baum-Durchläufe

Definieren Sie eine Funktion $minTree :: BinTree\ Int \rightarrow Int$, die den minimalen Wert in einem Baum liefert. Definieren Sie eine Funktion $replace :: BinTree\ a \rightarrow b \rightarrow BinTree\ b$, die alle Werte in einem Baum durch einen gegebenen Wert ersetzt. Definieren Sie nun eine Funktion

$$replaceMinRec :: BinTree\ Int \rightarrow a \rightarrow (BinTree\ a, Int)$$

die beide Berechnungen parallel ausführt. Um auf die beiden Komponenten des Paares zuzugreifen, können Sie einen **where**-Block oder einen **let**-Ausdruck verwenden.

Das heißt, die erste Komponente des Paares enthält einen Baum, in dem alle Werte durch den gegebenen Wert ersetzt sind. Die zweite Komponente des Paares enthält den minimalen Wert des Baumes. Definieren Sie mit Hilfe von *replaceMinRec* eine Funktion $replaceMin :: BinTree\ Int \rightarrow BinTree\ Int$, die alle Werte in einem gegebenen Baum durch das Minimum der Werte im Baum ersetzt.

Aufgabe 3 - Faltungen

In dieser Aufgabe sollen Sie sich mit den Funktionen *foldr* und *foldl* beschäftigen. Definieren Sie die folgenden Funktionen einmal mit *foldr* und einmal mit *foldl*.

- Die Funktion $map :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$, die eine Funktion auf alle Elemente einer Liste anwendet.
- Die Funktion $filter :: (a \rightarrow Bool) \rightarrow [a] \rightarrow [a]$, die aus einer Liste alle Elemente herausfiltert, die das Prädikat nicht erfüllen.
- Die Funktion $reverse :: [a] \rightarrow [a]$, die die Reihenfolge der Elemente in einer Liste umdreht.
- Die Funktion $concat :: [a] \rightarrow [a] \rightarrow [a]$, die zwei Listen aneinanderhängt.

Vergleichen Sie für alle Funktionen die beiden Implementierungen miteinander in Hinsicht auf Laufzeit der Funktion und Verhalten bezüglich nicht-strikter Auswertung.

Hinweis: Bitte geben Sie die Lösungen der Aufgaben per Mail oder auf Papier bis zum 04.11. ab. Die Papierlösungen können im ersten Stock von Haus A in den Schrank in das entsprechende Fach geworfen werden.