
Arbeitspaket zu KW 20 zum Modul „Fortgeschrittene Programmierung“
Sommersemester 2020

In dieser Woche geht es um weitere einfache und zusammengesetzte Datentypen in Haskell und wie man diese selbst definiert.

Als weiterer rekursiver Datentyp kommen endliche Listen und die Induktion darüber dazu. Dabei werden wir (im Unterschied zum Buch) zunächst wieder vorwiegend mit einer selbst-definierten Version arbeiten.

Deshalb gibt es auch eine kurze Einführung und einen Foliensatz.

Lesen und Verstehen

Kapitel 5 Einfache Datenstrukturen (alles außer Abschnitt 5.1.6)

im Buch „Haskell-Intensivkurs - Ein kompakter Einstieg in die funktionale Programmierung“
Der Titel dieses Kapitels ist etwas irreführend, denn es geht dort überwiegend nicht um einfache, sondern um zusammengesetzte Datentypen.

Unendliche Listen und lazy evaluation sehen wir uns später noch genauer an, deshalb können wir den Abschnitt 5.1.6 jetzt weglassen.

Ergänzend konsultieren Sie bitte die Folien (<https://informatik.htwk-leipzig.de/schwarz/lehre/ss20/fop/adt.pdf>) auf der Homepage zum Modul. Diese werden auch in den Übungsaufgaben verwendet.

Begriffe und Testfragen

- 1) Was sind head und tail einer Liste?
- 2) Wie ist der Haskell-Standard für den Typ Liste ? Welche Bezeichnungen werden dabei statt Nil und Cons verwendet?
- 3) Was sind list comprehensions? Wozu können sie verwendet werden?
- 4) Was sind fst und snd in einem Paar (2-Tupel)?

Übungsaufgaben

Für Serie 5 gibt es insgesamt 6 Aufgaben-Punkte, je einen für Musterlösungen zu

- 5.1, 5.2, 5.3
- Teilaufgaben 5.4.a, 5.4.b und 5.4.a

Autotool

Haskell-Datentypen und rekursive Funktionen auf Listen