
Arbeitspaket zu KW 18 zum Modul „Fortgeschrittene Programmierung“

Sommersemester 2020

In dieser Woche üben wir den Umgang mit dem Testsystem LeanCheck und den Beweissystem cyp, die wir ab jetzt zum Test und zur Verifikation unserer Haskell-Programmen einsetzen. Das Thema Verifikation wird im Buch erst sehr spät behandelt, das ist natürlich falsch. Man erinnere sich gelegentlich daran, was schon Dijkstra dazu schreibt:

A programmer has to be able to demonstrate that his program has the required properties. If this comes as an afterthought, it is all but certain that he won't be able to meet this obligation: only if he allows this obligation to influence his design, there is hope he can meet it. Pure a posteriori verification denies you that whole-some influence and is therefore putting the cart before the horse, but that is exactly what happens in the software houses where “programming” and “quality assurance” are done by different groups. (Needless to say, those houses deliver without warranty.)

in <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD13xx/EWD1305.html>

und ähnlich in <https://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD361.html>

Wir machen es also besser und werden gleich bei der Besprechung neuer Programmkonstrukte die entsprechenden Verifikationsprinzipien begleitend einsetzen.

Zunächst beginnen wir mit der Verifikation von Termgleichungen, auch mit Hilfe der prädikatenlogischen Grundlagen aus dem Modul Modellierung. Außerdem sind verschiedene Sätze mit Gleichungen zu beweisen, die Sie vielleicht schon aus der Mathematik im 1. Semester kennen.

Lesen und Verstehen

Kapitel 18 Programme verifizieren und testen

im Buch „Haskell-Intensivkurs - Ein kompakter Einstieg in die funktionale Programmierung“ (<https://link.springer.com/content/pdf/10.1007%2F978-3-642-04718-3.pdf>)

Es genügt zunächst, die kurze Einleitung des Kapitels 18 und die Abschnitte bis 18.1.2 zu lesen und dabei das Beweisprinzip Induktion zu wiederholen. Die folgenden Abschnitte passen besser zu den Themen der kommenden Wochen.

Weil das sehr wenig ist, lesen wir in dieser Woche zur Motivation und Erbauung noch

<https://bartoszmilewski.com/2020/02/24/math-is-your-insurance-policy>

und natürlich die oben angeführten Links von Dijkstra.

Begriffe und Testfragen

In dieser Woche gibt es nicht viel im Buch zu lesen und deshalb auch keine neuen Begriffe und Testfragen dazu.

Übungsaufgaben

Für jeden etwas dabei: viele tolle Aufgaben und 14 Übungspunkte in Serie 3

In den Aufgaben 3.4 und 3.5 gibt es für jede der Gleichungen einen Punkt.

Aufgaben 1 und 2 im Abschnitt 18.3 sind zum Selbsttest bei der Wiederholung von Beweisen durch vollständige Induktion geeignet. Das werden wir demnächst brauchen.

Autotool

tolle Aufgaben mit cyp