



Funktionale Programmierung, SS 2019, Blatt 1

Fabian Kunze, Kathrin Stark,

Prof. Dr. Gert Smolka

https://courses.ps.uni-saarland.de/fp_ss19/

Grundlagen funktionaler Programmierung, Auswertungsstrategien, Streams

Erinnern Sie sich an die zentralen Ideen zur funktionalen Programmierung aus der Vorlesung Programmierung 1: Polymorphismus, höherstufige Prozeduren, Muster und Konstruktoren, Listen, Bäume, Faltungen, und (abstrakte) Datenstrukturen. (Dies entspricht in etwa den Kapiteln 3, 4, 6, 7 und 14 aus *Programmierung – Eine Einführung in die Informatik mit Standard ML* von Gert Smolka.)

Lesen Sie dann Kapitel 1–3 in *A Gentle Introduction to Haskell* (www.haskell.org/tutorial/) von Hudak, Peterson und Fasel, insbesondere Abschnitte 3.3 und 3.4 zu nicht-strikter ('lazy' oder 'by need') Auswertung. Verwenden Sie einen Haskell Interpreter, um mit einigen Beispielen aus dem Text zu experimentieren.

Aufgabe 1.1 (Streams) Wir bezeichnen eine unendliche Liste als *stream*. Abschnitt 3.4 in *A Gentle Introduction...* zeigt, wie streams als zyklische Listen definiert werden können. Geben Sie Definitionen

- für $twos = [2, 2, 2, \dots]$,
- für $powers = [1, 2, 4, 8, 16, \dots]$, bestehend aus den Potenzen 2^n ,
- für $fact = [1, 1, 2, 6, 24, \dots]$, bestehend aus den n -ten Fakultäten $n!$.
- für $fib = [1, 1, 2, 3, 5, \dots]$, bestehend aus den Fibonacci-Zahlen.

Aufgabe 1.2 (Sieve of Eratosthenes) Ein Verfahren, mit dem Primzahlen bestimmt werden können, ist das folgende: Ausgehend von $[2, 3, 4, \dots]$ werden zunächst die Vielfachen $4, 6, 8, \dots$ von 2 gestrichen. Anschliessend werden die (noch vorkommenden) Vielfachen $6, 9, 12, \dots$ von 3 (der nächst größeren Zahl der verbleibenden Elemente) gestrichen, dann die Vielfachen von 5 und so weiter. Als 'Grenzwert' erbigt sich der stream $[2, 3, 5, 7, \dots]$ aller Primzahlen.

Realisieren Sie dieses Verfahren in Haskell mittels zweier Deklarationen $primes :: [Int]$ und $sieve :: [Int] \rightarrow [Int]$.

Aufgabe 1.3 (Bedarfstgesteuerte und Strike Auswertung) Erinnern Sie sich daran, dass die Auswertungsreihenfolge in Standard ML im Gegensatz zu Haskell 'strikt' ist: Bei einer Prozeduranwendung pa wird das Argument a zu einem Wert v ausgewertet, bevor der Prozedurrumpf e ausgeführt wird (mit dem formalen Parameter x der Prozedur ersetzt durch v).

Geben Sie einen Term s an, der sich unter bedarfstgesteuerter und strikter Auswertung unterschiedlich verhält.

Aufgabe 1.4 (Unendliche Listen in Standard ML) Definieren Sie einen Datentypen *'a ilist*, der (potentiell) unendliche Listen in Standard ML darstellt. Deklarieren Sie danach testweise $[2, 2, 2, \dots]$, $[1, 2, 3, \dots]$ und $[42]$ sowie eine Prozedur $take :: int \rightarrow 'a\ ilist \rightarrow 'a\ list$, welche die ersten n Elemente als 'normale' Liste zurückgibt.

Hinweis 1: In Standard ML ist $fn () \Rightarrow e$ ein Wert. Der Rumpf e wird erst evaluiert, sobald die Abstraktion auf $()$ angewendet wird.

Hinweis 2: Konstruktortypen können auch Funktionen enthalten.

Hinweis 3: Mittels der Deklaration $val\ rec\ v = \dots$ können in SML rekursive Werte definiert werden.

Aufgabe 1.5 (Effiziente unendliche Listen in Standard ML) Lesen Sie https://en.wikipedia.org/wiki/Lazy_evaluation. Nicht-strike Auswertung wird aus Effizienzgründen oft mit Memoisation verknüpft, hier implementieren wir diese Variante in Standard ML.

- a) Inwiefern ermöglicht Memoisation, dass *twos* nur konstant viel Speicher belegt und *fib*s effizient und knapp definiert werden kann?
- b) Werte in Haskell kann man sich als Referenz auf eine 'Box' vorstellen, die entweder einen Wert oder eine noch auszuwertende Berechnung ('Thunk') enthält. Auch SML kennt Referenzen.

Implementieren Sie Haskell's bedarfsgesteuerte Auswertung für Listen möglichst originalgetreu in Standard ML. Deklarieren Sie dann die konkreten Listen und Funktionen aus Aufgabe 1.4 erneut. Testen Sie, ob alle Aspekte aus dem ersten Aufgabenteil abgebildet werden.

Hinweis: Mit der Sequenzialisierung $print\ " "; e$ können Sie zum Testen sichtbar machen, wann ein Teilausdruck e ausgewertet wird.