

---

**Arbeitspaket zu KW 26 zum Modul „Fortgeschrittene Programmierung“**  
Sommersemester 2020

---

Wir wissen schon, dass es in Haskell Ausdrücke gibt, die mehrere Redexe enthalten. In dieser Woche geht es um die Reihenfolge, in der sie dann ausgewertet werden. In Haskell geschieht das bedarfsgesteuert (lazy), Teilausdrücke werden also erst dann ausgewertet, wenn ihr Wert benötigt wird. Das ermöglicht es auch, Daten (potentiell) unendlicher Größe zu verarbeiten, z.B. unendliche Listen (Streams) und Bäume. Im Buch steht nicht viel dazu. Deshalb gibt es wieder am Freitag zur Vorlesungszeit eine Einführung.

### Lesen und Verstehen

Abschnitt 5.1.6 Lazy evaluation 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> )

Außerdem gibt es den am 19. 6. 2020 vorgestellten Foliensatz

(<http://informatik.htwk-leipzig.de/schwarz/lehre/ss20/fop/lazy.pdf>)

Ergänzend können Sie die entsprechenden Abschnitte aus dem Skript zum Modul 2019 konsultieren.

### Übungsaufgaben

Serie 10 (<https://www.imn.htwk-leipzig.de/~schwarz/lehre/ss20/fop/serie10.pdf>)

### Autotool

Neben einer Aufgabe zur Bedarfsauswertung gibt es Aufgaben zu Binärzahlen, die in der schriftlichen Übungsserie vorbereitet werden. Dabei üben wir auch wieder das Beweisen in cyp.

### Punkte für die Prüfungszulassung

Bitte vor Montag 22.6.2020 um 12:00 Uhr nur diejenigen im Wiki zur Aufgabenverteilung eintragen, die noch keinen Aufgaben-Punkt haben (danach egal).

Wer sich für eine Aufgabe eingetragen hat, dann aber feststellt, dass er diese Aufgabe nicht lösen kann, trägt sich dort bitte bis Montag mittag wieder aus.

- ein Punkt für die Zusammenfassung von Abschnitt 5.1.6
- 9 Punkte für Übungsaufgaben, je einen für
  - 10.1 t1 und t2
  - 10.1 t3
  - 10.1 t4
  - 10.1 t5
  - 10.2
  - 10.3.a
  - 10.3.c
  - 10.3.d
  - 10.3.e und f