

Funktionale Programmierung (2)

1. Ersetzen Sie in der folgenden Funktionsdefinition die Parameter `f`, `a`, `seq` so, dass die Fakultätsfunktion definiert wird:

```
1 def fakReduce(n):  
2     return reduce(f, seq, a)
```

Hinweis: Beachten Sie, dass das Argument n bei der Fakultätsfunktion eine Zahl ist, aus der Sie zunächst eine geeignete Liste `seq` erzeugen sollten. Benutzen Sie dazu `range` und `list`.

2. Sie haben bereits das Sieb des Eratosthenes kennen gelernt. Diesen Algorithmus sollen Sie nun funktional implementieren.
Schreiben Sie eine Funktion, die eine Zahl n als Parameter übergeben bekommt und alle Primzahlen bis n als Python-Liste zurück gibt!

Hinweis: Beginnen Sie mit einer Liste, die alle Zahlen bis n enthält. Nutzen Sie fortlaufend die Filter-Funktion, um Vielfache bisheriger Primzahlen aus dem Rest der Liste herauszufiltern.

3. Schreiben Sie eine Funktion, die aus einer gegebenen Linkssequenz eine Rechtssequenz generiert, wobei die Elemente in derselben Reihenfolge bleiben!
Beispiel:

$$(1, (2, (3, \text{empty}))) \mapsto (((\text{empty}, 1), 2), 3)$$

4. Definiert seien drei Funktionen:

h berechnet den Durchschnitt von drei ganzen Zahlen

g rundet eine reelle Zahl auf die nächste Ganzzahl ab

f erwartet zwei Funktionen und drei ganze Zahlen als Eingabe, so dass gilt: Werden die Funktionen h und g übergeben, dann gibt f den abgerundeten Durchschnitt der drei übergebenen Zahlen zurück.

- (a) Geben Sie Definitions-, Wertebereich und Ordnung der Funktionen an!
- (b) Schreiben Sie den Definitionsbereich von f so um, dass f eine Komposition der beiden Funktionen h und g erhalten kann, also nur noch eine Funktion und drei ganze Zahlen übergeben bekommt, aber die gleiche Abbildung realisiert, wie zuvor!
- (c) Schreiben Sie die Funktion h vollständig gecurried auf! Wie sieht ein Aufruf der Funktion für die Zahlen $(a = 2, b = 5, c = 5)$ aus?

Zusatzaufgabe

Bei dem folgenden Algorithmus handelt es sich um eine funktionale Variante des verbreiteten Sortieralgorithmus *InsertionSort* für Linkssequenzen. Dieser Sortieralgorithmus setzt sich aus den zwei Funktionen `insert` und `isort` zusammen.

Die Funktion $\text{insert}: \text{int} \times \text{LS}_{\text{int}} \rightarrow \text{LS}_{\text{int}}$ erhält eine Zahl x und eine bereits sortierte Linkssequenz xs und gibt eine Linkssequenz zurück, in der die übergebene Zahl x korrekt in xs einsortiert wurde.

Der eigentliche Sortieralgorithmus ist dann recht einfach zu definieren. Jedes Element der zu sortierenden Linkssequenz wird an die entsprechende Stelle der Ergebnis-Linkssequenz eingefügt (*Nutzen Sie hierfür die **insert** Funktion*). Die Funktion `isort` wird solange ausgeführt, bis der Basisfall erreicht wird, die Linkssequenz xs also leer ist.

*(Tipp: Reduzieren Sie erst die Linkssequenz auf **empty** und fügen Sie anschließend die Elemente an die richtigen Stelle ein. Eine leere Linkssequenz ist eine sortierte Linkssequenz.)*