



Programmieren in MOSTflexiPL

Vorlesung im Sommersemester 2026
Prof. Dr. habil. Christian Heinlein

11. Aufgabenblatt (23. Juni 2026)

Aufgabe 19: Funktionale Programmierung

Definieren Sie folgende Operatoren:

- `map•and•using•` (vgl. §5.1.2) erhält als Parameter zwei Listen `xs` und `ys` mit beliebigen Elementtypen `X` und `Y` und eine zweistellige Funktion `f` mit Parametertypen `X` und `Y` und irgendeinem Resultattyp `Z`.

Als Resultat erhält man eine neue Liste mit Elementtyp `Z`, deren Elemente jeweils durch einen Aufruf der Funktion `f` aus den entsprechenden Elementen der Listen `xs` und `ys` entstehen.

Wenn `xs` und `ys` unterschiedlich lang sind, ist die Resultatliste so lang wie die kürzere von ihnen.

- `gen•using•and•` erhält als Parameter einen `int`-Wert `n`, einen Wert `x` mit irgendeinem Typ `X` und eine einstellige Funktion `f` mit Parameter- und Resultattyp `X`.

Als Resultat erhält man eine Liste der Länge `n` mit den Elementen `x`, `f x`, `f (f x)` usw. mit Typ `X`, d. h. jedes Element außer dem ersten entsteht durch Anwendung der Funktion `f` auf das vorige Element.

Wenn `n` nicht positiv ist, erhält man `nil`, d. h. eine leere Liste.

Aufgabe 20: Funktionale Programmierung

Verwenden Sie `gen`, `map` und `fold`, um möglichst geschickt folgende Summen als rationale Zahlen zu berechnen:

- $\sum_{k=1}^{10} \frac{1}{k}$ (Anfang der harmonischen Reihe),
- $\sum_{k=1}^{10} \frac{(-1)^{k+1}}{k}$ (Anfang der alternierenden harmonischen Reihe)
- $\sum_{k=0}^{10} \frac{1}{2^k}$ (Anfang der geometrischen Reihe mit $q = \frac{1}{2}$)

Die Funktionen für `gen` können jeweils bequem mit `add` oder `mul` aus §5.2 erzeugt werden.