Informatik I: Scheme-Programmieraufgaben zum Selbststudium

4. Zusammengesetzte Datentypen: Polynome
(Bei Fragen können Sie sich an die Tutoren des betreuten Programmierens wenden)

Aufgabe 1: Record-Datentyp für Polynome
Definition: Ein Polynom p vom Grad $n \in \mathbb{N}$ ist eine Funktion $p : \mathbb{R} \to \mathbb{R}$, wobei p die folgende Form hat: $p(x) = \sum_{i=0}^{n} a_i x^i$ mit $a_i \in \mathbb{R}, a_n \neq 0$.
Schreiben Sie eine Definition für einen zusammengesetzen Datentypen `polynomial`, welcher ein Polynom in Scheme repräsentieren soll. Dafür werden die Koeffizienten a_i des Polynoms p in einer Liste mit $n + 1$ Elementen gespeichert. Dabei steht an erster Stelle a_0, an zweiter Stelle a_1 usw.
Hinweis: Ein Aufruf `(make-polynomial empty)` erstellt im Prinzip kein richtiges Polynom. Der Fall für eine leere Koeffizientenliste ist jedoch in den nachfolgenden Aufgaben als Standardfall nützlich (s. auch nächste Aufgabe).

Aufgabe 2: Normalisierung
Für ein Polynom vom Grad n gilt die Bedingung, dass $a_n \neq 0$. Addiert man aber beispielsweise das Polynom $5+2x+3x^2$ zu $7+8x-3x^2$, erhält man $12+10x+0x^2 = 12+10x$; ein Polynom vom Grad 1. Bei den folgenden Operationen soll nun ausgeschlossen werden, dass der letzte Koeffizient 0 ist. Schreiben Sie deshalb eine Prozedur `adjust-degree`, welche für ein gegebenes Polynom (welches möglicherweise die Eigenschaft verletzt, dass die letzten Koeffizienten von 0 verschieden sind) die geforderte Eigenschaft wiederherstellt. Beachten Sie jedoch, dass wir $p(x) = 0$ als Polynom erlauben (d. h. die Bedingung $a_n \neq 0$ muss nicht gelten für $n = 0$). Dabei soll dieses Polynom dargestellt werden als Polynom mit leerer Koeffizientenliste.
Hinweis: Denken Sie in den nachfolgenden Aufgaben daran, den Grad des Polynoms immer wieder anzupassen.

Aufgabe 3: Auswertung eines Polynoms
Schreiben Sie eine Prozedur `(: evaluate-polynomial (polyomial number -> number))`, so dass ein Aufruf `(evaluate-polynomial p x0)` den Wert $p(x_0)$ berechnet.

Aufgabe 4: Addition von Polynomen
Schreiben Sie `(: polynomial-add (polynomial polynomial -> polynomial))`, so dass für zwei Polynome $p(x) = \sum_{i=0}^{n} a_i x^i$ und $q(x) = \sum_{i=0}^{m} b_i x^i$ ein Aufruf `(polynomial-add p q)` das Polynom $(p + q)(x)$ berechnet.

Aufgabe 5: Multiplikation von Polynomen
Schreiben Sie `(: polynomial-multiply (polynomial polynomial -> polynomial))` mit der Eigenschaft, dass für zwei Polynome $p(x) = \sum_{i=0}^{n} a_i x^i$ und $q(x) = \sum_{i=0}^{m} b_i x^i$ ein Aufruf `(polynomial-multiply p q)` das Polynom $(p \cdot q)(x)$ berechnet.