Technische Universität Dresden
Institut für Theoretische Informatik
Dr. Monika Sturm

Wintersemester 08/09

Funktionale Programmierung und Typtheorie

5. Übungsblatt

Aufgabe 1

Definieren Sie die Liste der perfekten Zahlen mithilfe einer Listenkomprehension. Eine Zahl heißt perfekt, wenn sie die Summe ihrer echten Teiler ist. Demnach ist `6` die kleinste perfekte Zahl, denn `1 + 2 + 3 = 6`. Benutzen Sie zum Testen die Funktion `take`.

Aufgabe 2

Welche der folgenden Gleichheiten über Listen sind korrekt und welche nicht?

a) `x : (xs : ys) = x : xs : ys`

b) `x : [ ] : xs = [x] : xs`

c) `x : ([x] ++ ys) = x : x : ys`

d) `(xs ++ x) : ys = xs ++ [x] ++ ys`

Aufgabe 3

In der Vorlesung wurden die Funktionen `mapTree` und `foldTree` vorgestellt.

a) Implementieren Sie mithilfe von `foldTree` die identische Funktion `treeIdentity` auf Bäumen.

b) Implementieren Sie `mapTree` mithilfe von `foldTree`.

c) Implementieren Sie mithilfe von `mapTree` eine Funktion `reciTree`, die in einem `Float`-Baum alle Zahlen durch ihren Reziprokwert ersetzt.

Aufgabe 4

Gegeben seien die Typgleichungen

$$\begin{aligned} x_1 &= g(x_3) \\ h(x_1) &= x_4 \\ x_2 &= x_3. \end{aligned}$$

Geben Sie zwei mögliche Unifikatoren σ_1, σ_2 für beide Typgleichungen an und begründen Sie mithilfe einer Abbildung $\varrho : V_T \to Typ_\Theta(V_T)$, ob entweder σ_1 oder σ_2 den allgemeinsten Unifikator darstellt.