

## 4. Übungsblatt

# Typ-basiertes Programmieren und Schließen in Funktionalen Sprachen

Jun.-Prof. Dr. Janis Voigtländer / Dipl.-Math. Daniel Seidel

Wintersemester 2009/10

### Aufgaben aus vorhergehenden Übungen:

#### Aufgabe 11

Beweisen Sie, dass jede Funktion  $f :: (a, a) \rightarrow a$  mit *fst* oder *snd* semantisch äquivalent ist.  $\diamond$

#### Aufgabe 17

Stellen Sie *foldr* mittels *destroy* dar.  $\diamond$

#### Aufgabe 18

Beweisen Sie die Korrektheit der *destroy* / *unfoldr*-Regel.  $\diamond$

#### Aufgabe 19

Optimieren Sie das Beispiel *zip* (*fromTo* 1 10) (*fromTo* 'a' 'j') aus der Vorlesung weiter. Versuchen Sie, ein Programm ohne Zwischendatenstrukturen und *Maybe*-Konstrukte zu erhalten.  $\diamond$

#### Aufgabe 14

Versuchen Sie, folgende Programmfragmente mittels *foldr*-fusion und/oder *shortcut deforestation* zu optimieren:

$$\text{reverse} \circ \text{map } h \quad \text{map } h \circ \text{reverse} \quad \text{reverse} \circ \text{reverse}$$

Welche Auswirkung hat die Implementierungsvariante des *reverse* (effizient, ineffizient)?  $\diamond$

### Eine neue Aufgabe:

#### Aufgabe 20

Schreiben Sie eine Funktion  $ppf :: (a \rightarrow a \rightarrow a) \rightarrow [a] \rightarrow [a]$ , die für assoziative Funktionen auf Booleschen Werten, nicht aber auf Werten des Typs *Three*, semantisch zu *scanl1* äquivalent ist.  $\diamond$