

### 3. Übungsblatt

## Typ-basiertes Programmieren und Schließen in Funktionalen Sprachen

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

Wintersemester 2009/10

#### Aufgabe 13

Leiten Sie (per Hand) das freie Theorem für den Typ

$$\forall b.(b \rightarrow \text{Maybe}(\tau_1, b)) \rightarrow b \rightarrow \tau_2$$

her. Es seien  $\tau_1$  und  $\tau_2$  feste Typen. ◇

#### Aufgabe 14

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

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

Welche Auswirkung hat die Implementierungsvariante des *reverse* (effizient, ineffizient)? ◇

#### Aufgabe 15

Diskutieren Sie in der Typsignatur begründete Unterschiede möglicher Funktionen folgender Typen.

- $\text{nullTest} :: [a] \rightarrow \text{Bool}$  vs.  $\text{nullTest}' :: (\forall a.[a]) \rightarrow \text{Bool}$
- $\text{toEmptyList} :: \text{Int} \rightarrow [a]$  vs.  $\text{toEmptyList}' :: \text{Int} \rightarrow (\forall a.[a])$
- $\text{toList} :: a \rightarrow [a]$  vs.  $\text{toList}' :: a \rightarrow (\forall a.[a])$  vs.  $\text{toList}'' :: (\forall a.a) \rightarrow [a]$  ◇

**Aufgabe 16**

Stellen Sie *filter* mittels *unfoldr* dar.

◇

**Aufgabe 17**

Stellen Sie *foldr* mittels *destroy* dar.

◇

**Aufgabe 18**

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

◇

**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.

◇