SoSe 2014

Prof. Dr. Margarita Esponda

ProInformatik II: Funktionale Programmierung

13. Übungsblatt (14. Tag)

1. Aufgabe

Programmieren Sie die show Funktion für den algebraischen Datentypen **Expr**.

2. Aufgabe

Reduzieren Sie den folgenden SKI-Ausdruck und begründen Sie die einzelnen Schritte.

SS(SI(K(KI)))(KK(S(KK)I))(KI)

3. Aufgabe

Zeigen Sie, dass folgende Lambda- und Kombinator-Ausdrücke äquivalent sind und begründen Sie die einzelnen Schritte.

$$\lambda s.\lambda x.\, s(s(x)) \;\equiv\; S(S(KS)(S(KK)I))(S(S(KS)(S(KK)I))(KI))$$

4. Aufgabe

Erweitern Sie den SKI-Parser aus der Vorlesung, sodass analog zum Lambda-Parser auch ***longmacros*** übersetzt werden können. Testen Sie Ihren Parser mit mindestens 3 vordefinierten Funktionen.

Anwendungsbeispiele:

```
parser  "x"          =>  Var "x"
parser  "{1}KK"      =>  App (App (App (App S (App (App S (App K S)) (App (App S
                             (App K K)) I))) (App K I)) K) K
```

5. Aufgabe

Definieren Sie analog zu der ersten eval-Funktion für Lambda-Ausdrücke aus der Vorlesung eine eigene **eval**-Funktion für SKI-Ausdrücke. Testen Sie Ihre **eval**-Funktion mit folgenden Ausdrücken:

Anwendungsbeispiele:

skii "(S(SI(K(KI)))(K(S(KK)I)))(S(KK)I)" => "KI"

skii "(S(S(KS)(S(S(KS)(S(KK)I))(KI)))(K(K(KI))))(S(KK)I)(KI)" => "KI"

mit **skii** = **eval . parser**

(siehe vorgegebene Funktionen in Ressources -> Material -> SKII_U10_start_functions.hs.zip)

6. Aufgabe

a) Geben Sie für folgende Haskell-Ausdrücke äquivalente Lambda Ausdrücke in Haskell an.

(2 **)

(f.g)

b) Programmieren Sie mit sinnvoller Verwendung einer anonymen Funktion und der foldl-Funktion eine Variante der reverse-Funkton für Listen.

c) Schreiben Sie folgende Haskell-Funktion

```
f 0 = 10
f n = n^2 + (f (n-1))
```

um, so dass sie als anonyme Funktion (λ-Ausdruck in Haskell) analog zu den Vorlesungsbeispielen zusammen mit der fix-Funktion (Y-Operator) das Gleiche berechnet.