

13. Übung zu „Grundlagen der funktionalen Programmierung“, Abgabe: 2. Februar 2007 vor der Vorlesung WS06/07

Hinweise: Dies ist das letzte Übungsblatt. Erreichbar sind 156 Punkte und 8 Zusatzpunkte.
Kolloquien zum Erwerb eines benoteten Scheins können ab Fr., den 9.2. stattfinden.

Aufgaben

13.1 **Kostenmodellierung** Mit der operationellen Semantik von Launchbury kann durch Erweiterung der Relation \Downarrow um einen Zähler eine Kostenmodellierung für die Auswertung realisiert werden. Der Zähler gibt die Zahl der Reduktionsschritte bis zum Ergebnis an.

7 Punkte

Die Regel zur Applikation etwa sieht wie folgt aus:

$$[\text{App}] \frac{\Gamma : e \Downarrow_p \Delta : \lambda y. e' \quad \Delta : e'[y \mapsto x] \Downarrow_q \Theta : z}{\Gamma : e x \Downarrow_{p+q+1} \Theta : z}$$

- (a) Geben Sie eine um den Zähler erweiterte Version aller Reduktionsregeln des Systems (incl. Datenstrukturen) an. Begründen Sie Ihre Definition.
- (b) Ermitteln Sie für den folgenden Ausdruck mit den neuen Regeln die Zahl der benötigten Reduktionsschritte:

$let\ f = \lambda x. case\ x\ of\ \{ [] \rightarrow 1; (y : z) \rightarrow y + f\ z\} in\ f\ [f\ [2]]$

(statt *Nil* und *Cons* verwendet f die Haskell-Syntax `[]` und `(:)`)

13.2 **Metaprogrammierung mit Template Haskell**

5 Punkte

Schreiben Sie eine (rekursive) Funktion für Template Haskell, welche Code generiert, um eine beliebige Zahl mit einer im Voraus bekannten ganzen Zahl $n \in \mathbb{Z}$ zu potenzieren:

```
pow :: Integer -> Q Exp -- Parameter:n, Ergebnis:Funktion \x -> x**n
```

(Erzeugen Sie als Ergebnis einen λ -Ausdruck).

Nutzen Sie bei der Codegenerierung folgende Fälle:

$$x^n = \begin{cases} x^{\frac{n}{2}} \cdot x^{\frac{n}{2}} & \text{falls } n \text{ gerade} \\ x \cdot x^{n-1} & \text{falls } n \text{ ungerade} \end{cases}$$

und vermeiden Sie, trivialen Code zu erzeugen.

Übersetzen Sie Ihr Programm mit `ghc-6.6` und der Option `-fth`. Sie können mit Option `-ddump-splices` überprüfen, für welche Ausdrücke im Programm Code generiert wird.