

## 6. Übung „Funktionale Programmierung“ Abgabe am 25. Mai 2005 vor der Vorlesung

---

**Wichtig:** Denken Sie bei allen definierten Funktionen/Typen an die auf dem 1. Übungsblatt formulierten Regeln (z.B. Typsignaturen, Beispielanwendungen und Beispielwerte).

**Aufgabe 1** 4 Punkte  
Formalisieren Sie die Übersetzung der `do`-Notation für Monaden. Gehen Sie von einer Folge `do a1; ...; an` aus und geben Sie Ihre Definition induktiv über die Aktionen `a1` bis `an` an. Unterscheiden Sie alle Möglichkeiten für die auftretenden `ai`.

Übersetzen Sie folgende Listcomprehension zunächst in die `do`-Notation und anschließend in einen Ausdruck, der explizite Binds (`(>>=)`) enthält.

```
[ (x,y,z) | x <- [1..], let y = [1..x], z <- map (+1) y, 2*x>=z ]
```

**Aufgabe 2** 6 Punkte  
Wir betrachten den vordefinierten Typ

```
data Either a b = Left a | Right b
```

Definieren Sie eine Instanz der Klasse `Monad` für die partielle Typapplikation `Either String`.

Erweitern Sie die in der Vorlesung vorgestellten arithmetischen Ausdrücke um die Wurzelfunktion (in Haskell `sqrt :: Float -> Float`). Hierdurch ergibt sich natürlich eine neue mögliche Fehlerquelle bei der Auswertung. Verbessern Sie die Auswertung der Ausdrücke mit Hilfe des `Either`-Typs, so dass die möglichen Fehlerursachen unterschieden werden können.

Integrieren Sie Ihre Auswertung in einen kleinen Taschenrechner, der einen Ausdruck einliest (`data Expr = ... deriving Read` macht dies möglich). Danach wird das Ergebnis der Auswertung ausgegeben (ggf. eine schöne Fehlermeldung) und der Benutzer wird zu einer neuen Eingabe aufgefordert. Ihr System soll nach Eingabe von `Q` enden. Achten Sie außerdem darauf, dass Ihr System robust gegenüber falschen Eingaben ist.

**Aufgabe 3**

5 Punkte

Zeigen Sie, dass die vorgestellte Definition von Listen und des Typs `Maybe` als Instanz der Klasse `Monad` die Monadengesetze (siehe Vorlesung) erfüllen.

Können Sie eine alternative Interpretation von Listen als Monaden angeben? Zeigen Sie auch für diese Implementierung die Gültigkeit der Monadengesetze (einschl. `MonadPlus`).

**Aufgabe 4**

5 Punkte

- a) Implementieren Sie Multiplikation und Exponentiation für Church Numerals.
- b) Beweisen Sie den in der Vorlesung vorgestellten Fixpunktsatz für den reinen  $\lambda$ -Kalkül.