Institut für Informatik
der Universität München
Dr. Andreas Abel
Dr. Hans-Wolfgang Loidl

SS 2009

Übungen zur Vorlesung
Funktionale Programmierung

Blatt 7

Aufgabe H-10:

a) Implementieren Sie in Haskell eine Funktion $unzip : [(a, b)] \to ([a], [b])$.

b) Mit Hilfe der universellen Eigenschaft von fold definieren Sie $unzip$ neu.

Aufgabe H-11:

Gegeben der Datentyp von binären Bäumen wie in der Vorlesung und die Funktion fold:

```
data BinTree a = Leaf a | Node (BinTree a)(BinTree a)

fold :: (a -> b) -> (b -> b -> b) -> BinTree a -> b
fold f comb (Leaf l)   = f l
fold f comb (Node l r) = (fold f comb l) `comb` (fold f comb r)
```

a) Schreiben Sie die universelle Eigenschaft von fold.

b) Definieren Sie die Funktion $map : (a \to b) \to (\mathit{BinTree}\, a \to \mathit{BinTree}\, b)$ mit Hilfe der universellen Eigenschaft von fold.

Aufgabe H-12: Gegeben sei der Typ von Powerlisten PList wie in der Vorlesung,

```
data PList a = Zero a | Succ (a, a)
```

Implementieren Sie in Haskell eine Funktion $twopower : \mathrm{Int} \to \mathrm{PList}\,\mathrm{Int}$ die eine Powerlist bestehend aus den Elementen $1..2^n$ erzeugt.

Aufgabe H-13:

Gegeben sei folgender Datentyp der arithmetischen Ausdrücken.

```
data Expr a = Lit a | Sum (Expr a) (Expr a) | Min (Expr a) (Expr a)
      | Mult (Expr a) (Expr a) | Div (Expr a) (Expr a)
```

a) Geben Sie den Typ eines Funktors ExprF, so dass $\mathrm{Expr}\ A = \mu X.\ \mathrm{ExprF}\ X\ A$.

b) Geben Sie den Typ des allgemeinen Datenkonstruktors in, und wie man damit die eigentlichen Datenkonstruktoren (Sum, Prod, Mult, Div und Lit) definiert.

c) Geben Sie den Typ der fold Funktion für diesen Datentyp und implementieren Sie sie in Haskell.