

## Übungen zur „Praktischen Informatik III“, WS 2006/07

Nr. 9, Abgabe: 19. Dezember 2006 vor der Vorlesung

---

22. Gleichheitsnachweis für unendliche Listen

4 Punkte

Gegeben seien die beiden folgenden Definitionen zur Erzeugung der Liste der Fibonacci-Zahlen:

```
fibnbs, fibs :: Num a => a -> a -> [a]
fibnbs k m   = k:fibnbs m (m+k)
fibs k m     = fs
  where fs = k:m:zipWith (+) fs (tail fs)
```

Beweisen Sie für beliebige  $k, m \geq 0$ :  $\text{fibs } k \ m = \text{fibnbs } k \ m$ .

23. QuickCheck

8 Punkte

Gegeben sei der folgende Auszug aus einer Gleichungsspezifikation eines abstrakten Datentyps Bag. Ein Bag sei eine Multimenge, d.h. Elemente können mehrfach in einem Bag vorkommen.

```
abstype Bag a where
-- Operationen:
empty      :: Bag a
insert     :: Eq a => Bag a -> a -> Bag a
remove     :: Eq a => Bag a -> a -> Bag a
union      :: Eq a => Bag a -> Bag a -> Bag a
getElement :: Bag a -> a
isEmpty    :: Bag a -> Bool
contains   :: Eq a => a -> Bag a -> Bool
-- Gleichungen:
getElement (insert empty x)           = x
union (insert b1 x) b2                 = union b1 (insert b2 x)
remove (insert b x) x                  = b
contains b x -> insert (remove b x) x = remove (insert b x) x
remove (insert b (insert b x) x) x     = insert b x
```

(a) Implementieren Sie die obigen Operationen für Bags, die wie folgt definiert seien: / 3

```
type Bag a = [(a,Int)] -- [(Element,Häufigkeit)]
```

(b) Definieren Sie auf Bags einen geeigneten Gleichheitstest / 1

```
(=?=) :: Eq a => Bag a -> Bag a -> Bool.
```

(c) Testen Sie mit QuickCheck, ob Ihre Implementierung die angegebenen Gleichungen erfüllt. / 4

**Hinweis:** Verwenden Sie eine Hilfsfunktion

```
makeBag :: Eq a => [(a,Int)] -> Bag a
```

die beliebige Werte vom Typ  $[(a,Int)]$  in solche Bags umwandelt, die mittels der zur Verfügung stehenden Operationen gebildet werden können. Welche Invarianten erfüllen Bags?