Dr. Th. Letschert
Softwarekomponenten

THM TECHNISCHE HOCHSCHULE MITTELHESSEN
MNI Mathematik, Naturwissenschaften und Informatik

Aufgabenblatt 5

Aufgabe 1

1. Ein *Funktor-Block* (auch *For-Ausdruck*, *For-Comprehension*, *Funktor-Ausdruck*) ist ein Ausdruck, dessen Wert ein Funktor ist. Wenn in einem Funktor-Block ein *if* auftaucht, dann muss es sich um einen *filterbaren Funktor* handeln. Was macht einen Funktor zu einem filterbaren Funktor?

2. Ein Shop bietet zwei unterschiedliche Artikel an. Eine Bestellung sei durch ein Objekt mit folgendem Typ repräsentiert:

   ```
   case class Order(customer: String, item1: Int, item2: Int)
   ```

   Neben dem Namen des Kunden enthält eine Bestellung die Anzahl der gewünschten Artikel, die angeboten werden. Kann `Order` mit geeigneten Methoden zu einem filterbaren Funktor gemacht werden.

3. Wie sieht es aus mit:

   ```
   case class Order(customer: String, item1: Option[Int], item2: Option[Int])
   ```

4. Was halten Sie von:

   ```
   case class Order[Customer](customer: Customer, item1: Option[Int], item2:
       Option[Int]) {
     def withFilter(p: Customer => Boolean): Order[Customer] = ???
   }
   ```

Aufgabe 2

1. Die Scala-API enthält den Typ *scala.collection.BitSet* mit dessen Hilfe Mengen von nicht-negativen ganzen Zahlen als Bitfolgen gespeichert werden können. Diese Klasse kann als Basis für eine kompakte Darstellung von Teilmengen einer festen Basismenge M verwendet werden:

   ```
   class SetSet_BV[A]( // Bitvector based set of sub-sets of baseSet
       private val baseSet: List[A],
       private val content: List[BitSet]
     ) { ... }
   ```

   Die Potenzmenge $\mathcal{P}(M)$ einer Menge M ist die Menge aller Teilmengen von M. Kann ein Objekt der Klasse *SetSetet_BV* als Potenzmenge der Basismenge der Basismenge etwa wie folgt erzeugt werden:

   ```
   object SetSet_BV {
     def powerSet[A](as: List[A]): SetSet_BV[A] = new SetSet_BV(as, ???)
     def powerSet[A](as: A*): SetSet_BV[A] = powerSet[A](as.toList)
     def powerSet[A](as: Set[A]): SetSet_BV[A] = powerSet[A](as.toList)
   }
   ```

SetSetet_BV ein filterbarer Funktor ist und eine passende *toString*-Methode definiert wurde, dann kann man beispielsweise folgenden Code ausführen:

```
val set = Set("a", "b", "c")
val powerSet: SetSet_BV[String] = SetSet_BV.powerSet(set)

val setsSize2 =
  for(set <- powerSet;
      if set.size == 2
  ) yield set

println(setsSize2) // Set(a, b), Set(a, c), Set(b, c)
```

2. *Für begeisterte Bitfummler und Algorithmiker*

   Die Menge aller möglichen Teilmengen einer festen Basismenge ist fix. Jede einzelne kann mit einem eindeutigen Index versehen werden und eine Menge von Teilmengen kann darum durch Menge von Indizes repräsentiert werden – und die können als Bitvektor gespeichert sein.