Universität Bremen

Praktische Informatik 3 WS 18/19

Christoph Lüth
Thomas Barkowsky
Andreas Kästner
Gerrit Marquardt
Tobias Haslop
Matz Habermann
Berthold Hoffmann

5. Übungsblatt

Ausgabe: 13.11.18
Abgabe: 20.11.18 12:00

5.1 *Lagern will strukturiert sein* *10 Punkte*

Die Inhaberin der Bäckerei *Barbara's Brotlounge* hat uns gebeten, ein neues Programm für ihre Lagerverwaltung zu entwickeln. Schlaue Entwickler wie wir sind, wollen wir ein allgemeines Programm entwickeln, welches wir auch an Barbaras Konkurrenten *Pizza-Toni* verkaufen können. Damit unsere Plan funktioniert, muss unser Programm beliebige Artikel verwalten können und nicht nur Backzutaten — es gibt schließlich auch noch andere Kleingewerbetreibende mit Lagern, die verwaltet werden müssen.

Wir gehen dabei wie folgt vor:

1. In unserem Lager werden die eingelagerten Objekte als Artikel hinterlegt. Jeder Artikel besteht aus einer Beschreibung `a` und seiner Anzahl im Lager als `Natural` (wobei zwei Artikel gleich sind, wenn ihre Beschreibung gleich ist):

   ```
   data Artikel a = A a Natural
   ```

   Das Lager besteht aus einer einfachen Liste mit Artikeln:

   ```
   type Lager a = [Artikel a]
   ```

   Produkte stellen Objekte dar, die aus Artikeln aus unserem Lager hergestellt werden können. Sie enthalten jeweils einen Namen und eine Liste mit den benötigten Zuaten in Form einer Liste von Artikeln. (Auch hier sind zwei Produkte gleich, wenn ihre Beschreibung (`name`) gleich ist.)

   ```
   data Produkt a b = P {name :: b, zutaten :: [Artikel a]}
   ```

   Die wesentlichen Funktionen unseres Lagers werden das Hinzufügen und Entfernen von Artikeln sein, sowie die Überprüfung, wie viel von einem bestimmten Artikel vorhanden ist. Dazu sollen die folgenden Funktionen implementiert werden.

   ```
   enthaelt :: Eq a ⇒ Lager a → a → Natural
   hinzufuegen :: Eq a ⇒ Lager a → a → Natural → Lager a
   entfernen :: Eq a ⇒ Lager a → a → Natural → Lager a
   ```

2. Für ihr Lager wünscht sich Barbara noch eine Funktion, die beim Backen überprüft, ob sie alle Zutaten für ihr Brot im Lager und das diese Zutaten auch direkt aus dem Lager entfernt werden. Da *Barbara's Brotlounge* ist allerdings ziemlich beliebt ist backt Barbara selten nur ein einzelnes Brot zu einem gegebenen Zeitpunkt, also sollte unser Programm auch mehrere Backaufträge gleichzeitig verarbeiten können. Hierfür sollen die Funktionen

   ```
   teileVorhanden :: (Eq a, Eq b) ⇒ Produkt a b → Lager a → Bool
   entferneProdukt :: (Eq a, Eq b) ⇒ Produkt a b → Lager a → Lager a
   ```

   verwendet werden. `teileVorhanden` überprüft, ob sich im Lager alle Artikel in ausreichender Anzahl für das übergebene Produkt befinden und `entferneProdukt` entfernt alle Artikel eines Produkts aus dem Lager.

   Die Funktion

   ```
   produzieren :: (Eq a, Eq b) ⇒ [Produkt a b] → Lager a → Lager a
   ```

   soll `teileVorhanden` und `entferneProdukt` für eine Liste von Produkten umsetzten, und dabei die Artikel von allen Produkten aus dem Lager entfernen, für die sie vorhanden sind.