

2. Übungsblatt

Ausgabe: 28.10.14

Abgabe: 07.11.14

2.1 Suchen

7 Punkte

Mit den selbstgebauten Zeichenketten aus der Vorlesung kann man jetzt noch nicht so viel anfangen. Implementieren Sie eine Funktion

`occurs :: MyString → MyString → Bool`

welche prüft, ob die erste Zeichenkette in der zweiten auftritt. Dabei soll in der ersten Zeichenkette das Zeichen `*` für einen beliebige Zeichenkette stehen, also soll beispielsweise die Zeichenkette `ab*x` in den Zeichenkette `rabllyxz` oder `rlabx` vorkommen (aber nicht in `arblx`).

Hinweis: Definieren Sie zuerst eine Hilfsfunktion

`matchStart :: MyString → MyString → Bool`

die prüft, ob die gesuchte Zeichenkette *am Anfang* der zweiten Zeichenkette auftritt. Danach können sie `occurs` unter Nutzung von `matchStart` einfach implementieren.

2.2 Ischa' Freimaak!

13 Punkte

Es ist die fünfte Jahreszeit, und das geht auch an diesem Übungsblatt nicht spurlos vorüber. Um auch in etwas aufgeheiteter Stimmung noch sicher den Weg über den Freimarkt zu finden, wollen wir unsere Modellierung von ungerichteten Graphen nutzen, um unseren Freimarktbummel zu planen.

Auf der Webseite finden Sie einen Übersichtsplan des Bremer Freimarkts 2014. Abb. 1 zeigt eine Abstraktion hiervon als einen ungerichteten Graphen, wie wir ihn in der Vorlesung kennengelernt haben. Sie finden auf der Webseite auch eine Datei `Freimaak.hs`, welche diese Abstraktion in Haskell modelliert.

Ein Bummel über den Freimarkt ist in dieser Modellierung eine Traversal des Graphen, welche eine Menge von vorgegebenen Knoten in beliebiger Reihenfolge beinhaltet. Dabei wollen wir natürlich nicht schon vorher besuchte Knoten noch einmal besuchen.

1. Definieren Sie zuerst einen Datentyp `Ziele`, welcher eine Menge von Zielen (in der Form von eindeutigen Bezeichnungen des Graphen, i.e. `Typ Id`) repräsentiert. Die Ziele können entweder leer sein, oder aus einem Ziel und weiteren Zielen bestehen.

2. Damit implementieren Sie eine Funktion

`bummel :: Ziele → Lab → Trav`

die eine Traversal des Graphen zurückgibt, welche die Ziele in beliebiger Reihenfolge abläuft.

3. In der vorherigen Funktion können Sie davon ausgehen, dass Sie in dem letzten angelaufenen Ziel einfach versacken. Ändern Sie jetzt ihre Funktion so ab, dass Sie in einer definierten Endposition den Freimarktbummel beenden.



Abbildung 1: 979. Bremer Freimarkt (schematische Darstellung)

? Verständnisfragen

Auf diesem und den folgenden Übungsblättern finden sich Verständnisfragen zur Vorlesung. Diese sind nicht Bestandteil der Abgabe, können aber im Fachgespräch thematisiert werden. Wenn Sie das Gefühl haben, diese Fragen nicht sicher beantworten zu können, wenden Sie sich gerne an Ihren Tutor, an Berthold Hoffmann in seiner Fragestunde, oder an den Dozenten.

1. Welche zusätzliche Mächtigkeit wird durch Rekursion bei algebraischen Datentypen in der Modellierung erreicht? Was lässt sich mit rekursiven Datentypen modellieren, was sich nicht durch nicht-rekursive Datentypen erreichen lässt?
2. Was ist der Unterschied zwischen Bäumen und Graphen, in Haskell modelliert?
3. Was sind die wesentlichen Gemeinsamkeiten, und was sind die wesentlichen Unterschiede zwischen algebraischen Datentypen in Haskell, und Objekten in Java?