

8. Aufgabenblatt zu Funktionale Programmierung vom Mi, 06.12.2017.

Fällig: Mi, 13.12.2017 (15:00 Uhr)

Themen: *Funktionen über algebraischen Datentypen, Knochelei*

Zur Frist der Zweitabgabe: Siehe „Hinweise zu Organisation und Ablauf der Übung“ auf der Homepage der LVA.

Aufgabe

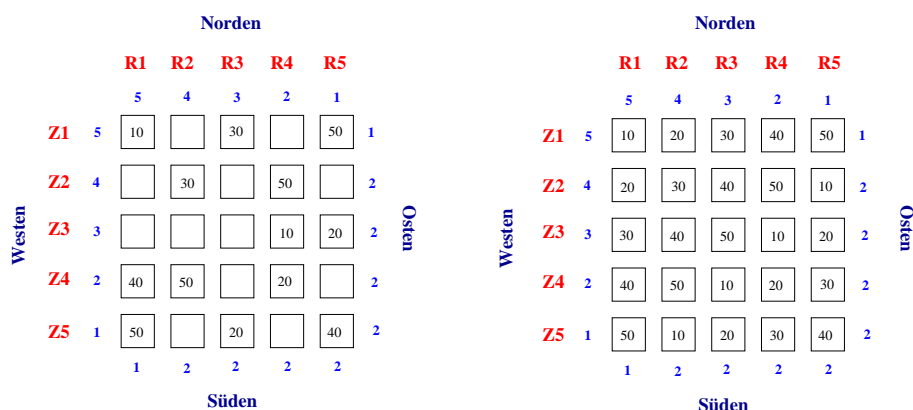
Für dieses Aufgabenblatt sollen Sie Haskell-Rechenvorschriften für die Lösung der unten angegebenen Aufgabenstellungen entwickeln und für die Abgabe in einer Datei namens **Aufgabe8.hs** ablegen. Sie sollen für die Lösung dieses Aufgabenblatts also wieder ein “gewöhnliches” Haskell-Skript schreiben. Versehen Sie wieder wie auf den bisherigen Aufgabenblättern alle Funktionen, die Sie zur Lösung brauchen, mit ihren Typdeklarationen und kommentieren Sie Ihre Programme aussagekräftig. Benutzen Sie, wo sinnvoll, Hilfsfunktionen und Konstanten.

Der Wohnungsmarkt in Wien ist angespannt und soll entlastet werden. Dazu sollen neue Stadtteile auf grüner oder fast noch grüner Wiese entwickelt werden, mit Hochhäusern unterschiedlicher Höhe. Alle Entwicklungsgebiete haben einen quadratischen Zuschnitt. In n Hochhauszeilen sollen je n Hochhäuser errichtet werden, je Zeile und Reihe genau eines mit 10, 20, ..., $n * 10$ -Stockwerken, um ein abwechslungsreiches Erscheinungsbild zu erreichen.

Um sogar aus jeder Blickrichtung ein möglichst wohlgefälliges Erscheinungsbild zu gewährleisten, erhalten die Bauträger eines Entwicklungsgebiets die zusätzliche magistratische Auflage, die Hochhäuser so zu errichten, dass Passanten, die eine Hochhauszeile oder -reihe entlangblicken genau so viele Hochhäuser sehen können wie es der Vorgabe für diese Blickrichtung entspricht.

Dabei gilt: Ein Hochhaus in einer Zeile oder Reihe verdeckt alle dahinterstehenden Hochhäuser geringerer Höhe, also mit weniger Stockwerken.

Die folgende Abbildung illustriert diese Auflagen für ein bereits teilbebautes 5×5 -Baufeld (Abb. links) zusammen mit einer vollständigen und auflagenkonformen Bebauung nach Abschluss der Stadtteilentwicklung (Abb. rechts). Beide Abbildungen sind genordet. Leere Felder in der linken Abbildung zeigen noch unbebaute Parzellen an.



Für jedes Baufeld müssen die Bauträger auf Neue mittels Probierens und Wiederprobierens mithilfe eines Modellbaukastens herausfinden, ob und wie die magistratischen Vorgaben für ein Baufeld umgesetzt und eingehalten werden können.

Abhilfe in Form eines Haskell-Programms, das diese Aufgabe automatisch löst, tut not. Tragen Sie zu diesem Programm die Haskell-Rechenvorschriften **ist_konforme_bebauung** und **berechne_bauplan** über folgenden Datentypen bei:

```

type Wahrheitswert = Bool

data Zeile    = Z1 | Z2 | Z3 | Z4 | Z5 | Z6 | Z7 | Z8 | Z9 | Z10
              -- max. 10 Zeilen von Hochhaeusern pro Baufeld.
data Reihe    = R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10
              -- max. 10 Reihen von Hochhaeusern pro Baufeld.
data Haustyp  = H00 | H10 | H20 | H30 | H40 | H50
              | H60 | H70 | H80 | H90 | H100 deriving Show
              -- Haushoehe in Stockwerken. Der Wert
              -- H00 kennzeichnet eine noch unbebaute
              -- Parzelle.

data Zahl = I | II | III | IV | V | VI | VII | VIII | IX | X -- Zahlen von 1 bis 10
type Sichtbar      = Zahl -- Anzahl sichtbarer Haeuser in einer Blickrichtung.
type Seitenlaenge  = Zahl
type Baufeldgroesse = Seitenlaenge -- Zeilen- und Reihenzahl quadratischer Baufelder.

data Zeilen_Blickrichtung = W0 -- Blickrichtung von West nach Ost.
                          | OW -- Blickrichtung von Ost nach West.
data Reihen_Blickrichtung = NS -- Blickrichtung von Nord nach Sued.
                          | SN -- Blickrichtung von Sued nach Nord.
type Zeilenauflage = Zeile -> Zeilen_Blickrichtung -> Sichtbar -- total definiert
type Reihenauflage = Reihe -> Reihen_Blickrichtung -> Sichtbar -- total definiert
type Baufeldauflage = (Zeilenauflage, Reihenauflage)

type Baufeld = Zeile -> Reihe -> Haustyp -- total definiert; beschreibt leeres bzw.
              -- (teil-) bebautes Baufeld
type Bauplan = [[Haustyp]] -- max. 10 Zeilen mit je max. 10 Reihen, d.h. max. 10 Listen
              -- der Laenge 10. Die erste Zeile (d.h. Liste) beschreibt die
              -- noerdlichste Bauzeile, die letzte die suedlichste Bauzeile.

data Bauplanung = Bauamtsauflagenskandal -- Kein vorgabenkonformer Bauplan moeglich
              | BP Bauplan deriving Show -- Vorgabenkonformer Bauplan

data Gutachten = Unvollstaendig_bebaut
              | Vollstaendig_nicht_konform_bebaut
              | Vollstaendig_konform_bebaut deriving Show

ist_konforme_bebauung :: Baufeldgroesse -> Baufeld -> Baufeldauflage -> Gutachten
berechne_bauplan :: Baufeldgroesse -> Baufeld -> Baufeldauflage -> Bauplanung

```

Angewendet auf ein leeres oder teilbebautes Baufeld und eine magistratische Bebauungsauflage

- überprüft die Wahrheitswertfunktion `ist_konforme_bebauung`, ob das (durch den Wert des Arguments vom Typ `Baufeldgroesse`) ausgewiesene Baufeld unvollständig (d.h., nicht alle Parzellen des Baufelds sind mit Werten verschieden von H00 belegt), vollständig und auflagenkonform oder vollständig, aber nicht auflagenkonform bebaut ist. Je nach Fall liefert die Funktion `ist_konforme_bebauung` den entsprechenden Wert des Typs `Gutachten`.
- berechnet die Funktion `berechne_bauplan` einen auflagenkonformen vollständigen Baufeldbebauungsplan, wenn ein solcher existiert. Sind mehrere möglich, ist es egal, welcher davon berechnet wird. In jedem Fall soll der Bauplan mit genau so vielen Zeilen und Reihen ausgegeben werden, wie der Baufeldgröße entspricht. Ist kein auflagenkonformer vollständiger Bauplan möglich, wird das Resultat `Bauamtsauflagenskandal` zurückgegeben.

Nehmen Sie weitere Typklasseninstanzbildungen vor, falls nötig oder hilfreich, um die Berechnung oder die Ausgabe von Ergebnissen im Zusammenspiel mit den vorgegebenen automatischen Instanzbildungen zu erleichtern oder ermöglichen.

Haskell Live

Der nächste *Haskell Live*-Termin findet wg. des Feiertags in dieser Woche erst am Freitag, den 15.12.2017, statt.

Haskell Private

Anmeldungen zu *Haskell Private* sind noch für kurze Zeit möglich. Nähere Hinweise und die URL zur Anmeldungsseite finden Sie auf der Homepage der Lehrveranstaltung. Nutzen Sie die Möglichkeit teilzunehmen.