

Aufgabenblatt 6 zu Funktionale Programmierung

Aufgabe 6.1 (Praktikum)

Laden Sie das lückenhafte Skript mit den Funktionen zur Tautologie-Prüfung aussagenlogischer Formeln herunter und erweitern Sie das Skript um die Funktionen *show2* und *intersect*. Nutzen Sie zur Definition von *intersect* List Comprehensions oder *fold*-Funktionen.

Aufgabe 6.2 (Praktikum)

Die Funktionen, die Sie in dieser Aufgabe definieren sollen, stellen kleine Schritte auf dem Weg zu einer Lösungsfunktion für Sudokus dar.

Wir gehen von folgenden Datentypen aus:

```
type Matrix a = [Row a]
type Row a    = [a]

type Grid      = Matrix Digit
type Digit     = Char
```

Ein Sudokurätsel wird durch ein Element des Typs *Grid* repräsentiert, bei dem leere Felder durch '0' dargestellt werden.

Definieren Sie eine Funktion *nodups* :: $Eq\ a \Rightarrow [a] \rightarrow Bool$, die genau dann *True* liefert, wenn alle Elemente der Liste verschieden sind.

Verwenden Sie diese Funktion, um eine Funktion *valid* :: *Grid* → *Bool* zu definieren. Diese Funktion liefert genau dann *True*, wenn das Argument eine Matrix ist, für die die „Sudoku-Bedingung“ erfüllt ist, d. h. in jeder Zeile, jeder Spalte und jeder Box der Matrix sind die Elemente verschieden.

Hinweis: *nodups* können Sie innerhalb der Definition von *valid* sinnvoll verwenden, wenn Sie jeweils die Elemente der Zeilen, Spalten und Boxen als Liste von Werten zusammenstellen. Sehen Sie für die Boxen einen Parameter für die Boxgröße vor, damit Sie zum Testen und vor allem später zum Lösen der Sudokus mit 4×4 -Sudokus arbeiten können.