

Aufgabenblatt 2 zu Funktionale Programmierung

Aufgabe 2.1 (Praktikum)

Definieren Sie einen Datentyp `Punkt`, dessen Werte Punkte mit reellwertigen Koordinaten in der zweidimensionalen Ebene sind. Der Typ `Punkt` soll Instanz der Typklassen `Eq` und `Show` sein. Die Darstellung eines Punkts soll das Format `(x, y)` haben.

Definieren Sie einen Datentyp `Rechteck`, dessen Werte Rechtecke in der zweidimensionalen Ebene sind. Ein Rechteck soll durch einen Ursprungspunkt (linke, obere Ecke) und eine Breite und Höhe beschrieben sein. Der Typ `Rechteck` soll Instanz der Typklassen `Eq` und `Show` sein. Die Darstellung eines Rechtecks soll das Format `[Ursprung: (x, y), Breite: ..., Hoehe: ...]` haben.

Definieren Sie eine Funktion `huelle :: Rechteck → Rechteck → Rechteck`, die für zwei Rechtecke das kleinste Rechteck zurückgibt, das die beiden Rechtecke umschließt.

Aufgabe 2.2 (Praktikum)

Definieren Sie eine Funktion

$$\text{wert} :: \text{Integer} \rightarrow [\text{Char}] \rightarrow \text{Integer}$$

die den Wert einer Ziffernfolge im Stellenwertsystem einer bestimmten Basis berechnet. Als Ziffernsymbole sollen die Zeichen 0 bis 9 verwendet werden, sodass als erster Parameter nur die Zahlen zwischen 2 und 10 sinnvoll sind.

Hinweis: Die Funktion `toInteger` kann verwendet werden, um einen Wert des Typs `Int` als Wert des Typs `Integer` zu liefern. Für die Definition von Funktionen auf Listen orientieren Sie sich bitte an den bisherigen Beispielen.

Definieren Sie eine Funktion

$$\text{repr} :: \text{Integer} \rightarrow \text{Integer} \rightarrow \text{Integer}$$

die zu einer Basis (1. Argument, nur zwischen 2 und 10) und einer Zahl (2. Argument) die Repräsentation dieser Zahl im Stellenwertsystem zur angegebenen Basis liefert.

Definieren Sie schließlich unter Verwendung von `wert` und `repr` eine Funktion

$$\text{summe} :: \text{Integer} \rightarrow ([\text{Char}], [\text{Char}]) \rightarrow \text{Integer}$$

die zwei Ziffernfolgen eines Stellenwertsystems addiert und die Summe als Darstellung im gleichen Stellenwertsystem liefert.

Versuchen Sie, für *summe* auch eine Lösung zu finden, bei der die Funktion möglichst nicht-applikativ (pointless) definiert ist. Hilfreich dafür ist eine zusätzliche Funktion ähnlich zu *pair*, wobei jedoch nicht ein Paar von Funktionen auf ein Argument, sondern eine Funktion auf die Komponenten eines Paares angewendet wird.

Aufgabe 2.3 (Übung)

Zeigen Sie, dass gilt:

$$\begin{aligned} \text{cross}(f, g) \cdot \text{cross}(h, k) &= \text{cross}(f \cdot h, g \cdot k) \\ \text{case}(f, g) \cdot \text{plus}(h, k) &= \text{case}(f \cdot h, g \cdot k) \end{aligned}$$

Sie dürfen von anderen Gleichungen, die im Buch aufgeführt sind, Gebrauch machen.