



Kategorientheorie für Programmierer

Hausaufgabenblatt 2 – SS19

Tübingen, 11. November 2019

Aufgabe 1: Lektüre

Für kommende Woche lesen Sie bitte Kapitel 6 und 7.

Aufgabe 2: Isomorphismen

Im Kapitel 5.4 haben Sie Isomorphismen kennengelernt. Zur Wiederholung: Ein Morphismus $f : A \rightarrow B$ in einer Kategorie \mathcal{C} ist ein Isomorphismus gdw. ein inverser Morphismus $g : B \rightarrow A$ in der Kategorie existiert für den gilt:

$$f \circ g = \text{id}_B$$

$$g \circ f = \text{id}_A$$

Zeigen Sie, dass der inverse Morphismus eindeutig ist. Zeigen Sie also, dass falls Morphismen g und g' existieren für welche die obigen Gleichungen gelten, dass dann $g = g'$ folgt. (Hinweis: Lösen Sie die Aufgabe durch Angabe einer Gleichungskette der Form $g = \dots = \dots = g'$.)

Die Eindeutigkeit des inversen Morphismus erlaubt es uns, f^{-1} für g zu schreiben.

Aufgabe 3: Isomorphismen von Datentypen in Haskell

Laden Sie das Haskell-Projekt „Isomorphisms“ herunter und entpacken Sie es in ein beliebiges Verzeichnis. Wir haben Ihnen schon den Großteil des Codes vorgegeben. Implementieren Sie Ihre Lösungen an den Stellen an denen derzeit noch „undefined“ steht.

Als Einstieg in Haskell implementieren wir einige Isomorphismen zwischen verschiedenen Datentypen. In dieser Aufgabe werden die folgenden Konzepte aus Haskell verwendet: Integer, Tuple, Either, Unit, Maybe und Listen. Es ist hilfreich zur Vorbereitung das Kapitel 6 aus dem Buch zu lesen.

$$\text{Integer} \simeq \text{Maybe Integer}$$

$$\text{Integer} \simeq \text{Either Integer Integer}$$

$$\text{Integer} \simeq [()]$$

$$\text{Integer} \simeq (\text{Integer}, \text{Integer})$$

Hinweis: Integer ist der Haskell Datentyp für beliebig große oder kleine ganze Zahlen. Um die Isomorphismen anzugeben müssen Sie also ein wenig tricksen. Überlegen Sie sich ob die Isomorphismen auch für Int, den Haskell Datentyp für Maschinenzahlen mit fester Größe (32 oder 64 Bit) gilt.