

Einführung in die funktionale Programmierung

Wintersemester 2023/2024

Aufgabenblatt Nr. 7

Abgabe: 6. Februar 2024, Besprechung 9. Februar 2024

Aufgabe 1 (20 Punkte) Typklassen

Wir wissen dass in Haskell Listen und Tupel mit $<$ verglichen werden können, denn der Listen und der Tupel Typ sind in `Ord`: D.h.

`Ord a => Ord [a]` und

`Ord a, Ord b => Ord (a,b)`.

Der Typklassentyp von $<$ ist:

`(<) :: Ord a => a -> a -> Bool`

1. Argumentieren Sie mit Hilfe der obigen Beziehungen, dass der Typ `Ord a, Ord b => [[a], [b]]` in der Typklasse `Ord` ist.
2. Warum ist der Ausdruck `[] < [[10,11], []]` getypt im polymorphen Typsystem mit Typklassen?
3. Welchen Typ (genauer Typklassentyp) hat die Funktion `\x -> x <= [[11,22], []]` ?

Aufgabe 2 (30 Punkte) Iterative Typisierung

Typisieren Sie den rekursiven Superkombinator g aus den Folien mit dem iterativen Verfahren aus Skript und Folien. Der Superkombinator ist wie folgt definiert:

`g x y = Knoten True (g x y) (g y x)`

Starten Sie mit `g :: ∀a.a`

Zur Erinnerung und als Hilfestellung:

Der Datentyp `Baum` und die Funktion `g` sind so definiert:

```
data Baum a = Leer | Knoten a (Baum a) (Baum a)
g x y      = Knoten True (g x y) (g y x)
```

Die Typen für die Konstruktoren sind

`Leer :: ∀a. Baum a`

`Knoten :: ∀a. a → Baum a → Baum a → Baum a`