

ProInformatik II: Funktionale Programmierung

12. Übungsblatt (13. Tag)

1. Aufgabe

Vervollständigen Sie die λ -Funktionen für ganze Werte (positive und negative Zahlen), indem Sie die Subtraktion und die Multiplikation definieren.

2. Aufgabe

Ganzzahlige Werte, die mit Zahlen-Tupeln dargestellt werden, haben keine eindeutige Darstellung.

Z.B. Die positive Zahl 3 kann als (8,11) oder (0,3) dargestellt werden.

Definieren Sie einen λ -Ausdruck, der eine beliebige Zahl (a,b) in ein Tupel der Form (0,n) (positive Zahl) oder (m,0) (negative Zahl) umwandelt.

Beispiel: $\lambda z.z\ 2\ 4 \Rightarrow \lambda z.z\ 0\ 2$
 $\lambda z.z\ 8\ 3 \Rightarrow \lambda z.z\ 5\ 0$

3. Aufgabe

Definieren Sie die \geq Operation für ganzzahlige Werte (positive und negative Zahlen).

4. Aufgabe

Definieren Sie einen λ -Ausdruck, der die Länge von zwei Listen vergleicht (\leq) und die Werte 1, 0 oder -1 zurückgibt je nachdem, ob die erste Liste kleiner, gleich oder größer ist.

5. Aufgabe

- a) Vervollständigen Sie die in der Vorlesung definierten λ -Ausdrücke für Listen mit einer Funktion, die ein Element in einer Liste sucht und entsprechende Wahrheitswerte zurück gibt.
- b) Definieren Sie eine λ -Funktion, die ein Element aus einer Liste löscht.

6. Aufgabe

Schreiben Sie folgende Hilfsfunktionen für den Lambda-Kalkül-Interpreter:

`int2Lambda :: Int -> String` -- Verwandelt eine Zahl in ihren entsprechenden
 Lambda-Ausdruck

`lambda2int :: String -> Int` -- Verwandelt einen Lambda-Ausdruck in eine Zahl

Anwendungsbeispiele: `num2Lam 3 => "/z.z (/x.y.y) (/x./y.x(x(x(y))))"`

`lam2Num "/z.z (/x./y.x(x(x(y)))) (/x.y.y)" => -3`