

9. Übung zu ALGORITHMEN UND PROGRAMMIERUNG I

Abgabe bis Donnerstag, den 21. Dezember

1. Aufgabe 4 Punkte

Angenommen, die Funktionen *konst*, *subst*, *fix* werden durch die Gleichungen:

$$\begin{aligned} \textit{konst } x \ y &= x \\ \textit{subst } f \ g \ x &= f \ x \ (g \ x) \\ \textit{fix } f \ x &= f \ (\textit{fix } f) \ x \end{aligned}$$

definiert. Leiten Sie die Typen der Funktionen her.

2. Aufgabe 6 Punkte

Erweitern Sie das Beispiel *Kassenbon* aus der Vorlesung, indem Sie einen Kopf (mit Firmennamen), die Endsumme und das Datum ausdrucken. Ferner sei die Eingabe eine Liste der Form $[(anzahl, Barcode)]$.

3. Aufgabe 2 Punkte

Definieren Sie eine Funktion *trips* so, dass *trips l* eine Liste mit allen Tripeln zurückliefert, die aus benachbarten Elementen von *l* gebildet werden können.

4. Aufgabe 6 Punkte

Schreiben Sie ein Haskell-Programm für die Multiplikation von Matrizen.

5. Aufgabe 4 Punkte

Definieren Sie eine Funktion *all* in Haskell, die zu einem Prädikat *p* und eine Liste *l* genau dann den Wert *True* liefert, wenn alle Elemente *a* aus *l* das Prädikat *p* erfüllen. Verwenden Sie die Funktion *foldr* in Ihrer Definition.

6. Aufgabe 2 Punkte

Sind folgende Gleichungen gültig?

$$\begin{aligned} \textit{foldl } (-) \ a \ la &= a - \textit{sum } la \\ \textit{foldr } (-) \ a \ la &= a - \textit{sum } la \end{aligned}$$

7. Aufgabe 4 Punkte

Gegeben sei folgende Funktion:

$$\textit{insert } a \ la = \textit{takewhile } (\leq a) \ la \ ++ \ [a] \ ++ \ \textit{dropwhile } (\leq a) \ la$$

Zeigen Sie, dass für eine nicht-fallende Liste *l* auch $(\textit{insert } a \ l)$ für alle Zahlen *a* nicht-fallend ist.