

6. Übung zu ALGORITHMEN UND PROGRAMMIERUNG I

Abgabe bis Donnerstag, den 30. November

1. **Aufgabe** 4 Punkte
 In der Vorlesung vom 16.11.00 wurde der Vereinigungstyp *DisVer* definiert und darüber die Funktionale *c* und *plus* (s. Vorlesungsmaterial auf der ALP I-Seite). Beweisen Sie die Behauptung: Für alle *f*, *g*, *h* und *k* passenden Typen gilt $c (f, g).plus (h, k) = c (f.h, g.k)$.

2. **Aufgabe** 2 Punkte
 Formulieren Sie für Funktionen über *Int* eine Analysefunktion, die zu einem gegebenen Abschnitt die Anzahl der Werte oberhalb einer gegebenen Grenze ausgibt.
Beispiel: Tage mit einer Temperatur von 28 °C oder mehr heißen Sommertage. Ihre Funktion soll u. a. dazu verwendet werden können, die Anzahl der Sommertage einer Saison zu bestimmen.

3. **Aufgabe** 4 Punkte
 Die Fibonacci-Zahlen sind die Folge 0, 1, 1, 2, 3, 5, 8, ... Jede Zahl in dieser Folge errechnet sich aus der Summe ihrer beiden Vorgänger, wobei mit 0 und 1 gestartet wird. Schreiben Sie eine Haskell-Funktion $fib :: Nat \rightarrow Nat$, die zu jeder natürlichen Zahl *n* die *n*-te Fibonacci-Zahl berechnet.

4. **Aufgabe** 2 Punkte
 Beweisen Sie durch vollständige Induktion, dass für alle $m :: Nat$

$$NNull \text{ 'mal' } m = m$$
 gilt.

5. **Aufgabe** 2 Punkte
 Können Sie die Potenzfunktion (hoch) durch Induktion über das linke Argument definieren?

6. **Aufgabe** 4 Punkte
 Beweisen Sie das Distributivgesetz zu den in der Vorlesung definierten Operationen ($*$) und ($+$) über *Nat*, d.h. $k * (m + n) = (k * m) + (k * n)$ für alle *k*, *m* und *n* aus *Nat*.