

**Aufgabe 5 (Typen):****(9 + 9 + 9 = 27 Punkte)**

Bestimmen Sie zu den folgenden Haskell-Funktionen f , g und h den jeweils allgemeinsten Typ. Geben Sie den Typ an und begründen Sie Ihre Antwort. Gehen Sie hierbei davon aus, dass alle Zahlen den Typ `Int` haben und die Funktionen `+`, `-`, `length`, `==` und `>` die Typen `Int -> Int -> Int`, `[a] -> Int`, `a -> a -> Bool` und `Int -> Int -> Bool` haben.

i) $f \ 1 \ ys \ _ = ys$
 $f \ x \ (y:ys) \ z = \text{if } x > z \text{ then } f \ (x - 1) \ (x:ys) \ z \text{ else } (y:ys)$

ii) $g \ x \ (y:ys) = g \ (y \ x) \ ys$
 $g \ x \ y = x \ []$

iii) $h \ [] \ x \ y = \text{if } x == 1 \text{ then } h \ y \ (\text{length } y) \ [] \text{ else } \text{True}$
 $h \ (a:as) \ x \ y = h \ as \ (x + a) \ y$

Hinweise:

- Versuchen Sie diese Aufgabe ohne Einsatz eines Rechners zu lösen. Bedenken Sie, dass Sie in einer Prüfung ebenfalls keinen Rechner zur Verfügung haben.

i) $f :: \text{Int} \rightarrow [\text{Int}] \rightarrow \text{Int} \rightarrow [\text{Int}]$

x und y sind `Int`, da $>$ auf `Int -> Int -> Bool` definiert ist.

Wegen $(x:ys)$ muss ys auch `Int` sein, da alle Elem einer Liste den gleichen Typ haben müssen.
 Mit selb. Begründung muss auch y ein `Int` sein.

ii) $g :: b \rightarrow [a] \rightarrow c$

Zunächst setzen wir $x = b$, $y = a$, $\rightarrow (y:ys) = [a]$ und $x \ y = c$.

Da $g \ x \ (y:ys)$ sich selbst mit anderen Parametern selbst aufruft, und der Sonderfall eine nicht genau erklärte Funktion aufruft, kann man nicht sagen, welche Werte zurück gegeben werden.

Daher erhält der Rückgabewerte des Rekursionsschluss den unbekannten Typ c .

iii) $h :: [\text{Int}] \rightarrow \text{Int} \rightarrow [\text{Int}] \rightarrow \text{Bool}$

Durch "else True" steht fest, dass die Funktion einen `Bool` zurück gibt.

Durch " $x == 1$ " ergibt sich, dass x vom Typ `Int` ist und durch " $x+a$ " muss auch a vom Typ `Int` sein und dann wegen $(a:as)$ ist as vom Typ `[Int]`.

Durch " $(\text{length } y)$ " weis man direkt einmal, dass y vom Typ `[a]` ist und dadurch, dass bei " $h \ y \ (\text{length } y) \ []$ " y als erster Übergabewert genutzt wird und wir von eben wissen, dass dieser vom Typ `[Int]` ist, muss auch y den Typ `[Int]` haben.