

1. Übung zur Vorlesung „Fortgeschrittene Programmierung“

Sommersemester 2020

zu lösen bis 16. April 2020

Aufgabe 1.1 (Terme)

Zur Bezeichnung von Positionen in Termen siehe die beiden Folien ab „Bezeichnungen für Teilterme“ in

<https://informatik.htwk-leipzig.de/schwarz/lehre/ss15/dp/fop15-folien.pdf>

Geben Sie Terme r, s, t über der Signatur $\Sigma_F = \{f : A \rightarrow B, g : (A \times B) \rightarrow A, h : A\}$ an, so dass das Symbol in

- r an Position $[]$ ist f ,
- s an Position $[0]$ ist g ,
- t an Position $[1]$ ist g und an Position $[1, 0]$ ist g .

Aufgabe 1.2 (Terme und Typen)

Zeichnen Sie die Termbäume für die folgenden Haskell-Ausdrücke. Geben Sie für jede Position in diesem Term jeweils

- den Typ des Symbols an dieser Position
- den Typ des Teiltermes mit der Wurzel an dieser Position

in Haskell-Syntax an.

```
xor True ( True && False )
xor ( div 7 3 == 2 ) ( not ( ( plus 2 1 ) >= ( mod 7 2 ) ) )
```

Die Operationen `xor`, `plus`, `div`, `mod` sind dabei wie im Buch definiert.

Beispiel: Term = `plus 3 1`

Position	Symbol mit Typ	Teilterm mit Typ
<code>[] (= ε)</code>	<code>plus :: Int -> Int -> Int</code>	<code>(plus 3 1) :: Int</code>
<code>[0]</code>	<code>3 :: Int</code>	<code>3 :: Int</code>
<code>[1]</code>	<code>1 :: Int</code>	<code>1 :: Int</code>

Aufgabe 1.3 (Mini-Programme)

Definieren Sie die folgenden Funktionen (Typdeklarationen nicht vergessen)

- `square`, welche jede eingegebene Zahl auf ihr Quadrat abbildet.
- `sum3`, welche drei eingegebene Zahlen auf ihre Summe abbildet.
- `mini`, welche von zwei eingegebenen Zahlen die kleinere zurückgibt.
- `eq3`, welche feststellt, ob drei eingegebene Zahlen gleich sind.

Implementieren Sie alle entworfenen Funktionen und testen Sie diese mit sinnvoll gewählten Eingaben mit `ghci`.