

Programmierung - Übung 10

Paul Sütterlin - 366676

Niklas Hempel - 349392

Aufgabe 2

das ist absolut nicht lesbar

```
r (s (d [2,3,5])) (s [3,1,4]) r (s (d [2,3,5])) (s [3,1,4]) if s [3,1,4] > 0 then s (d [2,3,5]):(r (s (d [2,3,5]))(s [3,1,4]-1)) else [] if 1 > 0 then s (d [2,3,5]) : (r (s (d [2,3,5])) (1-1)) else [] if True then s(d[2,3,5]) : (r (s (d [2,3,5])) (1-1)) else [] s (d [2,3,5]) : (r (s (d [2,3,5])) (1-1)) s(22 : d [3,5]):(r (s (22 : d [3,5])) (1-1)) s (22 : 32 : d[5]):(r (s (22 : 3 : d[5])*) (1-1)) 3*2 : __(r (*3*2*) (1-1)) 6 : __(r 6 (1-1)) 6 : (if (1-1) > 0 then 6 else [] 6 : (if 0 > 0 then 6 else []) 6 : (if False then 6 else []) 6 : [] [6]
```

Für schlechte Formatierung, fehlende * und zu viele * und _ -1,5
4,5/6

Aufgabe 4

a)

$[x : [y] : []]$: Kein typkorrekter Ausdruck. da die neue Liste Elemente vom Typ "Int" als auch Listen mit dem Typ "[Int]" enthält. Dies ist in Haskell nicht erlaubt.

$[x] ++ [y]$: Typkorrekter Ausdruck, da die beiden Listen vom Typ "[Int]". Die zusammengefügte Liste besitzt 2 Elemente und ist dann erneut vom Typ "[Int]". Der gesamte Ausdruck ist dann eine Liste mit einem Element vom Typ "[Int]" also vom Typ "[[Int]]"

1,5/1,5

b)

$[x, y] ++ xs$: Typkorrekter Ausdruck, denn $[x, y]$ ist vom Typ [Int] und besitzt 2. Elemente. Auch xs ist per Definition von diesem Typ und besitzt n Elemente. Damit hat ist die zusammengefügte Liste vom Typ "[Int]" und hat $2+n$ Elemente.

$[x] ++ [y] ++ [xs]$: ist ebenfalls typkorrekt, da $[x]++[y] = [x,y]$. Es gelten die obigen Eigenschaften analog.

Daher sind die Ausdrücke identisch

1,5/1,5

c)

$[x,y,z]$: Typkorrekter Ausdruck, da x, y und z jeweils vom Typ "Int" und damit die Liste vom Typ "[Int]" mit 3 Elementen.

$([x] ++ [y]) : [z]$: Typkorrekter Ausdruck. $([x] ++ [y])$ wird als Verkettung zweier Elemente vom Typ [Int] zu einer Liste vom Typ [Int] mit 2 Elementen $[x,y]$. Diese Liste und $[z]$ (Typ: [Int]) werden jeweils Elemente einer Liste von Listen von "Int"s ([[Int]]) Diese neue Liste besteht aus 2 Elementen: $[x,y][z]$

Die beiden Ausdrücke sind aufgrund ihrer unterschiedlichen Länge offensichtlich ungleich.

d)

$(x : []) : []$: Typkorrekter Ausdruck. x wird als Element in eine leere Liste gepackt. Damit ergibt sich $[x] : []$. Diese Liste wird dann als Element in die leere Liste eingefügt. Es ergibt sich $[[x]]$ (Länge 1)

$[x:[]] ++ [[]]$: Typkorrekter Ausdruck. $x:[x]$ wird wie oben zu $[x]$ ausgewertet. Es ergibt sich $[[x]] ++ [[]]$. Da $[x]$ vom Typ $[Int]$ und $[]$ als leere Liste ebenfalls als Liste vom Typ $[Int]$ angenommen werden kann, können beide in eine Liste zusammengefügt werden. Es ergibt sich $[[x],[]]$, also eine Liste mit dem Typ " $[[Int]]$ " und 2 Elementen.

Aufgrund der unterschiedlichen Anzahl an Elementen sind die Ausdrücke offensichtlich unterschiedlich. 1,5/1,5

e)

$x : y : z : xs$: Typkorrekter Ausdruck. x, y und z sind vom Typ "Int" und werden vorne an die Liste xs vom Typ $[Int]$ mit der Länge n angefügt. Daraus ergibt sich eine Liste vom Typ $[Int]$ mit $n+3$ Elementen.

$(x : [y]) ++ (z : xs)$: Typkorrekter Ausdruck. $x:[y]$ wird zu $[x,y]$ ausgewertet und mit der Liste $z : xs$ vereinigt. $z : xs$ wird zu einer Liste vom Typ $[Int]$ ausgewertet, da z vom Typ "Int" ist und einfach an den Anfang der Liste xs vom Typ $[Int]$ angehängt wird. Die Liste $[x,y]$ wird schlussendlich mit dieser Liste zusammengefügt. Es ergibt sich eine Liste vom Typ $[Int]$ mit der Länge $n+3$

Da die resultierenden Listen beider Ausdrücke gleich lang sind und bei beiden Listen die Elemente x,y,z (in dieser Reihenfolge) an die Int Liste xs vorne angefügt werden, ergibt sich die Gleichheit.

Aufgabe 6

1,5/1,5

Abgabe erfolgt über blatt10.hs

8/8