

## Informatik A, WS 2016/17 — 13. Übungsblatt

Abgabe bis Freitag, 27. Januar 2017, 12:00 Uhr. Aufgabe 72 ergänzt am 19. Januar

---

### 72. Strukturelle Induktion, 10 Punkte

Beweisen Sie

$$\text{elem } x \text{ (} a ++ b \text{)} = \text{elem } x \text{ } a \text{ } || \text{elem } x \text{ } b \quad (1)$$

für endliche Listen  $a$  und  $b$  unter Verwendung folgender Definitionen:

```
elem :: (Eq a) => a -> [a] -> Bool
elem x []      = False           -- elem.1
elem x (y:ys) = x==y || elem x ys -- elem.2
(++ ) :: [a] -> [a] -> [a]
[]      ++ ys = ys              -- (++ .1)
(x:xs) ++ ys = x : (xs ++ ys) -- (++ .2)
(||) :: Bool -> Bool -> Bool
True  || _ = True              -- (|| .1)
False || x = x                 -- (|| .2)
```

Begründen Sie jeden Schritt, indem Sie angeben, welche Definitionsgleichung (zum Beispiel  $(++ .1)$  oder  $\text{elem}.2$ ) oder welche anderen Eigenschaften sie verwenden.

### 73. Strikte Funktionen, 10 Punkte

- (a) Ist die Funktion `elem` aus der vorigen Aufgabe strikt im ersten Argument?
- (b) Ist sie strikt im zweiten Argument?
- (c) Gilt die Gleichung (1) auch für unendliche Listen  $a$  und/oder  $b$ ? (Die Gleichung ist auch dann erfüllt, wenn auf beiden Seiten das Ergebnis  $\perp$  herauskommt.)
- (d) Gilt sie auch für Listen mit undefinierten Elementen?

Begründen Sie Ihre Antworten.

### 74. Träge Auswertung, 10 Punkte

Zeigen Sie Schritt für Schritt, wie HASKELL den Wert `h 3 5` im Zusammenhang mit folgenden Definitionen auswertet:

```
h :: Int -> Int -> Int
h m n | nichtLeer xs = vorne xs
      | otherwise    = n
      where xs = ints m

ints m = m : ints (m+1)

vorne (x:y:zs) = x+y
vorne [x]     = x

nichtLeer []    = False
nichtLeer (_:_) = True
```

### 75. Träge Multiplikation, Programmieraufgabe, 0 Punkte

Schreiben Sie eine *polymorphe* Funktion `mult` zur Multiplikation zweier Zahlen, die das zweite Argument nicht auswertet, wenn das erste Argument 0 ist.

76. Funktionsdefinitionen, 0 Punkte

- (a) Welche der folgenden Definitionen bewirkt, dass die Funktion  $g = \text{entf } n \text{ ' '}$  alle Leerzeichen aus einer Zeichenkette entfernt?

```
entf1 z l      = [if x==z then "" else [x] | x <- l]
entf2 z l      = [x | x <- l, x/=z]
entf3 z        = concat . map (\ x -> if x==z then "" else x)
entf4 z []     = []
entf4 z (z:zs) = entf4 z zs
entf4 z (x:xs) = x: entf4 z xs
entf5 z        = foldr (\ x r -> if x==z then r else x:r) []
```

Bestimmen Sie für jede Funktion  $\text{entf } n$ , ob sie überhaupt nach den Haskell-Regeln gültig ist, und stellen Sie gegebenenfalls ihren Typ fest.

Begründen Sie Ihre Antworten.

- (b) Was macht die Funktion

```
entf ' ' . entf ', ' . entf ' ' . entf ';' . entf '.'
```

wenn  $\text{entf}$  die richtige Lösung zu Aufgabe (a) ist?

77. Sichtbarkeitsbereiche, 0 Punkte

- (a) Geben Sie für jeden im folgenden Programmstück eingeführten Namen den Sichtbarkeitsbereich an.

```
h a b
| r > 0      = a+r
| otherwise = -a+b
where
  g c a = a*b*c - 1
  r      = g a (-b)
```

- (b) Benennen Sie die Variablen so um, dass jeder Name nur an einer einzigen Stelle definiert ist.  
 (c) Bestimmen Sie  $[h \ (-1) \ (-1), h \ (-2) \ 1, h \ (-1) \ 2]$ .

78. Strukturelle Induktion, 0 Punkte

Wie kann man  $\text{foldr } g \ z \ (a \ ++ \ b)$  durch  $\text{foldr}$  mit den Listen  $a$  und  $b$  ausdrücken?

Beweisen Sie Ihre Formel, indem Sie für  $(++)$  die Definition von Aufgabe 72 und folgende Definition für  $\text{foldr}$  verwenden:

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f z []      = z                -- foldr.1
foldr f z (x:xs) = f x (foldr f z xs) -- foldr.1
```

79. Alle drin, Programmieraufgabe, 0 Punkte

192 ist die kleinste ganze 3-stellige Zahl  $n$ , so dass  $n$ ,  $2n$  und  $3n$  gemeinsam alle Ziffern von 1 bis 9 enthalten. Berechnen Sie eine Liste  $\text{hasAllDigits} :: [\text{Int}]$  mit allen 3-stelligen Zahlen  $n$ , für die  $n$ ,  $2n$  und  $3n$  alle Ziffern von 1 bis 9 enthalten. Sie können auch eine allgemeinere Funktionen schreiben, also zum Beispiel, dass in der Eingabe mit spezifiziert ist, was drin sein soll usw.