

# Algorithmen und Programmierung I

WS 2005 / 2006

Klausurvorbereitung (ohne Punkte, ca. 90 Minuten) , 7.12.05, 16:15 HS 003

Für Multiple Choice Aufgaben gilt: es kann, falls nichts anderes gesagt wird, keine, eine oder mehrere richtige Antworten geben. Eine falsche Antwort wird mit einem Punktabzug bewertet (Diesmal weder Punkte noch Abzüge ;).

Die Aufgaben sind grob nach Schwierigkeit geordnet. Wenn Sie aber mit einer Aufgabe nach ca 5 Minuten noch nichts anzufangen wissen, also auch keine Lösungsidee haben, gehen Sie zur nächsten über. Versuchen Sie sich an solchen Aufgaben dann später noch mal.

## Aufgabe 1

a) `( 95`div` )`

- (i) ist keine injektive Funktion (Wertebereich `Int`) **X**, z.B. `(95`div`) 96 = (95`div`) 97`
- (ii) ist total auf dem Argumentbereich `Int`
- (iii) ist keine Funktion
- (iv) syntaktisch fehlerhaft
- (v) hat angewandt auf ein positives Argument vom Typ `Int` immer Wert größer Null

## Aufgabe 2

`(x:xs)` ist äquivalent zu

- (i) `x ++ xs`
- (ii) `[x] ++ [xs]`
- (iii) `[x] ++ xs` **X**
- (iv) jedem der Ausdrücke (i)-(iii)

## Aufgabe 3

Welche der folgenden Funktionen definiert eine Funktion des Typs  
`([Char],Char) -> [Char]` ?

- (i) `f ((x:xs), x') = [x] ++ reverse xs ++ ['x']` **X**
- (ii) `f (x,y:ys) = [] ++ reverse ys ++ [x]`
- (iii) `f ((xs:'x'),x) = [x] ++ reverse xs ++ ['x']`

### Aufgabe 4

Welchen Wert liefern die Ausdrücke

a) `tel1 liste const' = [y | (const',y) <- liste ]`

b) `tel2 liste const' = [y | (c,y) <-liste, c==const']`

wenn `liste = ("Q", 4), ("R", 007), ("S", 2 )]`

und `const' = "R" .`

a) `[4,7,2]`

b) `[7]`

### Aufgabe 5

Geben Sie einen Ausdruck an, der die Wahrheitstafel für den Operator `(||)` als Wert liefert.

Eine Wahrheitstafel für eine Boolesche Funktion sei eine Liste von Tupeln des Typs `((Bool,Bool),Bool)`. Die erste Komponente ist ein Argument (Tupel von 2 Booleschen Werten), die zweite der Wert der Anwendung des Operators auf die Argumente.

`[((x,y),(x||y)) | x <- [True, False], y <-[True, False]]`

### Aufgabe 6

Was liefert die Funktion:

```
bizzar :: [a] -> [a]
bizzar xs = foldr (++) [] (map biz xs)
  where biz x = [x]
```

Die Identität für Listen: `bizzar xs = xs`

## Aufgabe 7

Gesucht ist ein Ausdruck für die Liste aller pythagoräischer Tripel  $(x,y,z)$  für die gilt:  $x^2=y^2+z^2$ . Dabei sind  $x,y,z$  natürliche Zahlen. Hinweis: Ein Generator muss offenbar eine unendliche Liste sein, bei mehr als einem ließe das Ergebnis auf sich warten... .

```
tripel = [(x,y,z) | x<-[1..],y<-[1..x],z<-[y..x], x^2== y^2 + z^2]
```

## Aufgabe 8

Zeigen Sie durch Induktion, dass für alle endlichen Listen gilt:

```
elem x (ys ++ zs) = elem x ys || elem x zs
```

Definition der verwendeten Funktionen:

```
elem x [] = False
elem x (y:ys)
  | x==y      = True
  | otherwise = elem x ys
```

```
[] ++ ys      = ys
(x:xs) ++ ys = x:(xs ++ ys)
Der Vollständigkeit halber auch || :
False || False = False
False || True  = True
...
```

Induktion über ys:

Induktionsanfang:

```
elem x ([]++zs) = elem x zs = False || elem x zs = elem x [] || elem x zs
```

Gelte die Induktionsvoraussetzung

Induktionsschluss:

```
elem x ((y:ys)++zs)
= elem x (y:(ys++zs))                -- ++
= if (x==y) then True else elem x (ys++zs) -- Def elem
= if (x==y) then True else (elem x ys || elem x zs) -- Induktionsvoraus.
= (if (x==y) then True else elem x ys) || elem x zs -- Assoz. von (||)
= elem x (y:ys) || elem x zs          -- Def elem
```

## Aufgabe 9

Definieren Sie eine Funktion `join`, die das Folgende leistet.

Argumente sind Listen, deren Basistyp Paare vom Typ  $(a,b)$  bzw.  $(b,c)$  sind. Der Wert ist eine Liste mit Basistyp  $(a,c)$ .  $(x,y)$  ist Element des Wertes von `join xs ys`, wenn es Elemente  $(x,a)$  in `xs` und  $(a',y)$  in `ys` gibt und  $a==a'$ .

```
join xs ys = [(x,y) | (x,z) <- xs, (z',y) <- ys, z==z']
```

## Aufgabe 10

Gesucht sind Funktionen, die einen Text mit einem sehr einfachen Verfahren verschlüsseln bzw. entschlüsseln:

```
caesar, uncaesar :: Int -> String -> String
```

Der Text enthält Buchstaben und Sonderzeichen. Verschlüsselt werden nur Buchstaben. Dabei sollen alle Kleinbuchstaben zunächst in Großbuchstaben umgewandelt werden.

Die Verschlüsselung erfolgt zeichenweise. `caesar n` verschlüsselt eine Eingabezeichenkette durch Ersetzen jedes Buchstaben `b` durch den Buchstaben `ersetzt`, der dadurch definiert ist, dass er in der ASCII-Folge `n` Zeichen hinter `b` steht (natürlich zyklisch). Beispiel für `n=2` wird 'A' durch 'C', 'C' durch 'E', 'Z' durch 'B' ersetzt.

Sie können die Funktionen

```
shiftAZ n c = toChar ((fromChar c - constA + n) `mod` 26 + constA)
  where constA = fromChar 'A'
```

sowie `toUpper` und `toLower`, die Buchstaben in Groß- bzw. in Kleinbuchstaben umwandeln, verwenden.

Was leistet die Funktion `shiftAZ` genau?

```
caesar n = map (shiftAZ n . toUpper) . (filter isChar)
uncaesar n = caesar (26-n)
```

`shiftAZ n c` bestimmt den ASCII-Wert `m` eines Zeichens `c` (Annahme: es handelt sich um einen Großbuchstaben), transformiert `m` in den Zahlbereich `0..25`. Auf diesen Wert `m` wird `n` addiert und modulo 26 betrachtet:  $n' = (m+n) \bmod 26$ . Das `n'`-te Zeichen des Alphabets ersetzt das Zeichen `c`. `shiftAZ n c` transformiert schließlich die Zahl `n'` wieder in das ASCII-Zeichen `c'`, das `c` ersetzt.

### Aufgabe 11

Definieren Sie eine Haskell-Funktion `lsort`, die eine Liste von Listen nach der Länge der Listen aufsteigend sortiert. Auf die Reihenfolge gleich langer Listen kommt es nicht an. Verwenden Sie zum Sortieren einen der Algorithmen `qsort` (Quicksort) oder `isort` (Sortieren durch Einfügen, insertion sort).

Beispiel: `lsort [[1,2,3],[3,4],[2,4,5,3],[5,4],[99]]`

Wert: `[[99],[3,4],[5,4],[1,2,3],[2,4,5,3]]`

```
lsort :: Ord a => [[a]] -> [[a]]
-- Quicksort-Algorithmus
lsort [] = []
lsort (x:xs) = lsort l ++ [x] ++ r
    where l = [y | y <- xs, length y <= length x]
          r = [y | y <- xs, length y > length x]
```