

Algorithmen und Programmierung I

WS 2004 / 2005

Übung 6 (25 Punkte)

Ausgabe: Mo. 29.11.2004

Abgabe: Mi. 8.12.2004, 10.00

Verspätet abgegebene Lösungen werden nicht akzeptiert.

Die Abgabe erfolgt in Zweiergruppen. Die Gruppe darf nur mit ausdrücklicher Genehmigung des Tutors gewechselt werden.

Abgegeben werden müssen

a) Alle Lösungen einmal ausgedruckt. **Handschriftliche Lösungen werden nicht akzeptiert.**

b) Programme, Testdaten, Testergebnisse in elektronischer Form in Absprache mit den Tutorinnen und Tutoren.

Aufgabe 6.1 (5 Punkte)

a) Gegeben Funktionen f , g , h mit den Signaturen

$f :: \text{Int} \rightarrow \text{String}$, $g :: \text{String} \rightarrow \text{Int}$, $h :: \text{Int} \rightarrow \text{Int}$.

Streichen Sie die Ausdrücke durch, die keinen Sinn machen:

$g . h . f$ $f . h . g$
 $h . f . g$ $g . f . h$
 $f . g . h$ $h . g . f$

b) Was bedeuten:

a
 $\backslash a \backslash$
 $' a '$
 $" a "$
 $(a , 2)$
 $[a]$
 $[a , b]$
 (a , b)

?

Einige der Alternativen setzen Definitionen von a und b voraus. Machen Sie geeignete Annahmen. Die obigen Alternativen sind nicht notwendig Ausdrücke oder Teile von Ausdrücken. Beachte: Es gibt manchmal mehrere Möglichkeiten.

c) Der Ausdruck $(a \text{ 'op' } b) \text{ 'op' } c$ bedeutet dasselbe wie:

$a \text{ 'op' } (b \text{ 'op' } c)$
 $(b \text{ 'op' } a) \text{ 'op' } c$
 $op \ a \ b \text{ 'op' } c$
 $op \ c \ (op \ a \ b)$
 $op \ (op \ a \ b) \ c$

alle Alternativen bedeuten dasselbe

keine Alternative ist gleichbedeutend mit $(a \text{ 'op' } b) \text{ 'op' } c$

d) Welche der folgenden Gleichungen sind für all x s richtig , für alle x s falsch bzw. für mindestens ein x s richtig bzw. falsch? (Begründung bzw. Beispiel!)

```

[]:xs      = xs
[]:xs      = [],xs]
xs:xs      = [xs,xs]
xs:[]      = xs
xs:[] = [xs]

```

Aufgabe 6.2 (6 Punkte)

Definieren Sie die folgenden Funktionen mit Hilfe der Faltungsoperation `foldr` oder `foldl`

- a) `alleGleich :: Eq a => [a] -> Bool`

```
-- pre:  - , value: x1 == x2==x3 ==...==xn
--      fuer Argumentliste [x1..xn]
```
- b) `exists :: Eq a => (a -> Bool) -> [a] -> Bool`

```
-- exists p xs ; pre:  - , value: ∃x (x ∈xs ∧ p x)
```
- c) Die Funktion `unique`, die aus einer Liste Duplikate entfernt (siehe Aufgabe 4.4, jetzt mit `foldr` oder `foldl`)

Aufgabe 6.3 (5 Punkte)

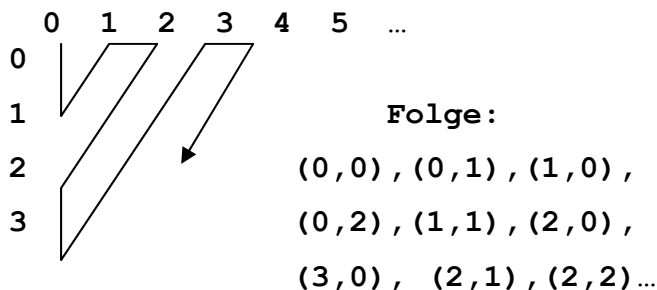
- a) Wandeln Sie eine als Zeichenkette gegebene positive Dezimalzahl in eine Zahl vom Typ `Int` um. Prüfen Sie vorher, ob die dezimale Eingabe nur aus Ziffern besteht.
- b) Machen Sie dasselbe für positive Zahlen zur Basis 2 (Eingabe: Liste von Werten vom Typ `Bit` von Übungsblatt 4 bestehend aus 0 und 1 (z.B. "OOII") sowie zur Basis 16 (Hexadezimale Ziffern sind die Dezimalziffern sowie A,B,C,D,E,F, die für die Werte 10,...,15 stehen).
- c) Schreiben Sie eine Funktion `decBin` zur Umwandlung von Zahlen in Dezimal- in solche in Binärdarstellung. Verwenden Sie den Datentyp `Bit` von Übung (z.B. `decBin 14 = IIIIO`).

Da in dieser Aufgabe Werte des Datentyps `Bit` auf Gleichheit geprüft werden müssen (z.B. `if x == I then...`), müssen die Werte des "selbst konstruierten" Typs `Bit` vergleichbar sein. Das erreicht man, in dem man die Definition folgendermaßen erweitert:

```
data Bit = I | O deriving (Show, Eq)
```

Aufgabe 6.4 (5 Punkte)

Mit Hilfe des Cantorschen Diagonalverfahrens kann man die Elemente einer Matrix mit unendlich vielen Spalten und unendlich vielen Zeilen aufzählen. Dazu muss man die Elemente der Matrix folgendermaßen diagonal durchlaufen:



Gesucht ist eine Funktion `cantor` die die ersten `n` Paare der Folge liefert.
Hinweis: Bestimmen Sie die Paare je einer Diagonalen `n=0,1,...` und fügen Sie die entstehenden Listen zusammen. Verwenden Sie die Funktion `zip` für die Paarbildung. (Keine ganz leichte Aufgabe!)

Aufgabe 6.5 (5 Punkte) Fortsetzung von 4.5

Gesucht ist eine Funktion `nBitAdd`, die zwei `n`-Bit-Register (Listen der Länge `n` von `Bit`) binär addiert und die Summe sowie den Übertrag ausgibt.

Erweitern Sie den Addierer mit Hilfe der Funktionen aus Aufgabe 6 so, dass er positive ganze Dezimalzahlen als Argumente akzeptiert, die Summe in Dezimaldarstellung ausgibt oder einen arithmetischen Überlauf signalisiert.