Name: Matr.-Nr.:

Klausur

Algorithmen und Programmierung I WS06/07

Aufgabe	1a	1b	1c	1d	2	3	4a	4b	5a	5b	6	$\sum$
Punkte	4	4	2	4	3	3	7	4	2	4	3	40
Erz. Punkte												

Zum Bestehen sind 20 Punkte erforderlich

Aufgabe 1) Sie betreiben ein kleines Geschäft und verwalten Ihr Warensortiment in einer Liste `waren` bestehend aus Einträgen, die jeweils einen Artikelnamen und seinen Preis (in Cent) enthalten.

a) Definieren Sie geeignete Datentypen `Artikel, Preis, Eintrag` und `Sortiment`, so dass `waren::Sortiment` vereinbart werden kann.

b) Definieren Sie eine Funktion `preis`, die angewendet auf einen Artikelnamen, den zugehörigen Preis (in Bezug auf das Sortiment `waren`) liefert.

c) Schreiben Sie eine Funktion `artikelAnzahl`, die die aktuelle Anzahl von Artikeln im Sortiment `waren` ausgibt; Sie können davon ausgehen, dass alle in `waren` aufgeführten Artikel unterschiedliche Namen haben.

d) Schreiben Sie eine Funktion `teuersterArtikel`, die denjenigen Artikelnamen aus dem Sortiment `waren` liefert, der den höchsten Preis hat.

Aufgabe 2) Gegeben seien folgende Syntaxregeln:

```
expr  →  digit | expr + expr | expr * expr | (expr)
digit →   0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Zeigen Sie, dass `(4+7)*3` ein Ausdruck ist, der aus dem Symbol `expr` hergeleitet werden kann.

Aufgabe 3) Reduzieren Sie den Ausdruck

$$(\lambda x.\ xy)\lambda z.z$$

auf seine Normalform.

Aufgabe 4)

a) Definieren Sie Funktionen `sum1`, `sum2` und `sum3`, die angewendet auf eine Liste von Zahlen, jeweils die Summe dieser Zahlen ergeben. Definieren Sie

- `sum1` durch eine einfache, rekursive Funktion
- `sum2` unter Verwendung der Akkumulatortechnik und
- `sum3` unter Verwendung des Haskell-Operators `foldr`. Zur Erinnerung:
  `foldr f s [ ] = s`
  `foldr f s (x:xs) = f x (foldr f s xs)`.

b) Beweisen Sie `sum1=sum3` durch strukturelle Induktion.

Aufgabe 5) Gegeben sei folgende Typvereinbarung:

```
data Farbe = Rot | Gelb | Blau
        deriving (Eq, Ord, Show, Read)
```

a) Definieren Sie einen algebraischen Datentyp `MischFarbe`, wobei eine Mischfarbe entweder eine Farbe ist oder aus zwei Mischfarben zusammengesetzt ist.

b) Vereinbaren Sie eine geeignete Funktion `show`, mit der Sie `MischFarbe` als Instanz der Klasse `Show` vereinbaren, so dass z. B. für eine Farbe `f`, die aus der Mischung von (Rot und Gelb) und Gelb entstanden ist, `show f` den String "RGG" ergibt.

Aufgabe 6) Definieren Sie unter Verwendung der `ZF`-Notation, eine Haskell-Funktion `mapAndFilter`, die angewendet auf eine Funktion `f`, einen Test `t` und eine Liste `xs`, all diejenigen Elemente `x` aus `xs` liefert, für die gilt: `t (f x) = True`.

Geben Sie zunächst den Typ von `mapAndFilter` an.

Viel Erfolg!