RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHR- UND FORSCHUNGSGEBIET INFORMATIK II
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/lufgi2

Prof. Dr. Jürgen Giesl
Darius Dlugosz, Thomas v. d. Maßen, Antje Nowack

Übung *Informatik I - Programmierung* - Blatt 14

Die Übungsblätter sollen in Gruppen von je 3 Studierenden bearbeitet werden. Diese 3 Studierenden müssen aus derselben Übungsgruppe kommen. Lösungen können bis zum Dienstag, den 12. Februar, 17.45 h, in den Kasten für Ihre Übungsgruppe im Foyer (neben Aula 2) in der Ahornstr. 55 eingeworfen werden.
Bitte vergessen Sie nicht, die **Nummer Ihrer Übungsgruppe** sowie die **Namen** und **Matrikelnummer** der Mitglieder Ihrer 3er-Gruppe auf Ihre Abgabe zu schreiben.

Wichtig: Alle Programme sind **sowohl** auf **Papier** in den Kasten einzuwerfen **als auch** elektronisch mit Hilfe des **Online-Systems** abzugeben.

Aufgabe 1 (3 Punkte)

Seien `f` und `g` Funktionen mit folgenden Typen

```
f :: Int -> Int
g :: Int -> Int -> Int
```

Die Funktion `h` sei wie folgt definiert

```
h x y = f (g x y)
```

(a) Bestimmen Sie den Typ von `h`.

(b) Welche der folgenden Gleichungen sind richtig und welche nicht? Begründen Sie bitte Ihre Antwort.

(i) `h = comp f g`
(ii) `h x = comp f (g x)`
(iii) `h x y = (comp f g) x y`
(iv) `h x y = (comp f (g x)) y`
(v) `h x y = comp f (g x y)`

Bemerkung: Die Funktion `comp` ist die in der Vorlesung eingeführte Funktion und ist wie folgt definiert:

```
comp :: (b -> c) -> (a -> b) -> (a -> c)
comp f g = \x -> f (g x)
```

Aufgabe 2 (6 Punkte)

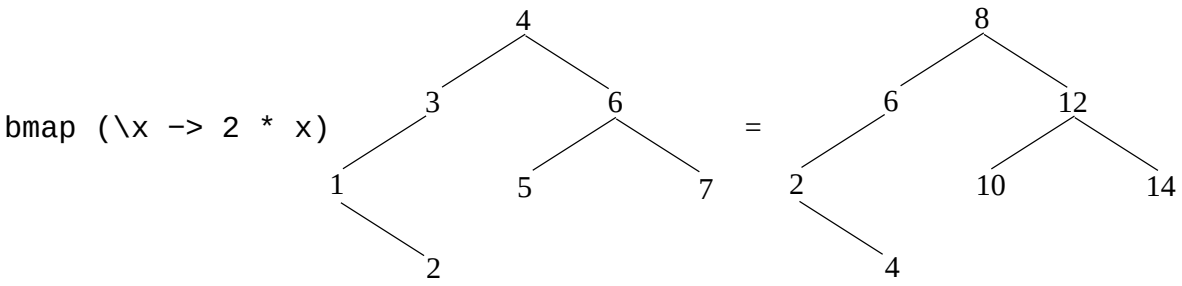
(a) Implementieren Sie in Haskell eine Funktion, welche die Quersumme einer ganzen Zahl berechnet. Die Quersumme einer Zahl ist die Summe ihrer Ziffern.

(b) Implementieren Sie in Haskell eine Funktion, welche für einen String seine Quersumme berechnet. Hierbei sei die Quersumme die Summe der Zahlen, die in dem ASCII-Code den in dem String enthaltenen Zeichen zugeordnet sind. Verwenden Sie dabei die Funktion `map` für Listen.

(c) Implementieren Sie in Haskell für die folgende Datenstruktur `Baum` eine Funktion `bmap` höherer Ordnung, welche auf das Element jedes Knotens des Baumes eine übergebene Funktion anwendet.

```
data Baum a = Knoten a [Baum a]
```

Beispiel:



Geben Sie jeweils den Typ der Funktionen an.

Zur Erinnerung:
Ein String ist eine Liste von Zeichen, d.h. der Typ `String` entspricht dem Typ `[Char]`.

Zu einem Zeichen erhält man mit Hilfe der Funktion `ord :: Char -> Int` ihre in dem ASCII-Code zugeordnete Zahl.

Aufgabe 3 (6 Punkte)

Eine Hash-Datenbank besteht aus Einträgen, die jedem Objekt eines Typs `a` eine Reihe von Objekten eines Typs `b` zuordnet.

Beispiele:

PLZ	Straßen
52074	Halifaxstr., Ahornstr., ...
52062	...
...	

oder

Stadt	PLZ
Aachen	52074, 52062, 52064, ...
Köln	50670, 50679, ...
...	

oder

Student	Freunde
Hugo	Erna, Heinz, Felix, ...
Sabine	Susi, Peter, ...
...	

(a) Definieren Sie in Haskell die Datenstruktur `Hash a b` zur Speicherung von Hash-Datenbanken.

(b) Implementieren Sie in Haskell eine Funktion `insert`, die zu einem Wert `x` des Typs `a`, dem ein Wert `y` des Typs `b` zugeordnet wird, einen Eintrag in eine Hash-Datenbank vornimmt. Dabei soll ein neuer Eintrag erzeugt werden, falls `x` noch nicht in der Hash-Datenbank vorhanden ist. Ansonsten soll die Reihe mit den Objekten, die dem `x` zugeordnet sind, um `y` erweitert werden.

(c) Implementieren Sie in Haskell eine Funktion `hmap` höherer Ordnung, welche eine Funktion `f` und eine Hash-Datenbank als Eingabe erwartet und eine Hash-Datenbank zurückliefert, bei der auf alle Elemente, die einem Wert zugeordnet sind, die Funktion `f` angewendet wird. Welches ist der Typ von `hmap`?

Beispiel: Für die Hash-Datenbank `plz` aus dem Beispiel von oben, die den Städten Postleitzahlen zuordnet, und für die Funktion `f x = x + 1` erhält man nach dem Aufruf `hmap f plz` die folgende Hash-Datenbank.

Stadt	PLZ
Aachen	52075, 52063, 52065, ...
Köln	50671, 50680, ...
...	

Aufgabe 4 (4 Punkte)

Auf einem Tisch t sind mit Nummern gekennzeichnete Blöcke gestapelt.

(a) Erstellen Sie ein Prolog-Programm, in dem die in der Abbildung 1 dargestellte Situation mit Hilfe eines Prädikats `auf` repräsentiert wird. Dabei bedeutet `auf(X,Y)`, dass `X` auf `Y` liegt.

(b) Definieren Sie ein Prädikat `ueber`, welches ausdrückt, dass ein Block über einem anderen Block bzw. auf dem Tisch steht. Hierbei dürfen höchstens zwei Klauseln definiert werden.

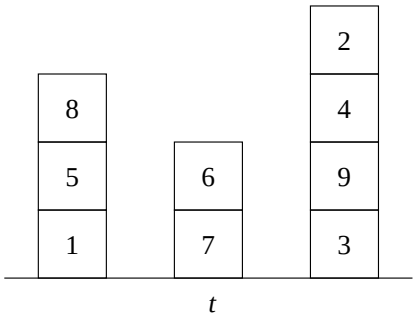


Abbildung 1:

(c) Stellen Sie die nachfolgenden Anfragen und geben Sie alle Lösungen an.

Steht Block 8 über Block 1?
Steht Block 2 über Block 7?
Was steht über Block 3?
Was steht unter Block 4?

Aufgabe 5 (6 Punkte)

(a) Erstellen Sie ein Prolog-Programm, das die nachfolgende umgangssprachliche Beschreibung formalisiert.

Peter liebt Susi.
Hans liebt Susi und Sabine.
Sabine liebt Peter und hasst Hans.
Susi liebt Peter und Felix.
Susi hasst Sabine.
Jeder liebt sich selbst.
Jemand ist ein Pechvogel, wenn seine Liebe mit Hass vergolten wird.

(b) Stellen Sie die nachfolgenden Anfragen in Prolog und geben Sie alle Lösungen an.

Liebt Peter Susi?
Liebt Susi Felix?
Wen liebt Sabine?
Wer liebt Sabine?
Wer liebt jemanden, der ihn auch liebt?
Wer liebt einen Pechvogel?

(c) Definieren Sie ein Prädikat `befreundet(X, Y)`, welches ausdrückt, dass eine Person `X` mit der Person `Y` befreundet ist. Hierbei sind zwei Personen miteinander befreundet, wenn es zwischen ihnen irgendeine Liebesbeziehung gibt oder wenn es eine Person gibt, die mit `X` und `Y` befreundet ist. Verwenden Sie so wenige Klauseln wie möglich (maximal 4).