

Nachklausur

Algorithmen und Programmierung I WS07/08

Aufgabe	1	2	3	4	5	6	7	8	Σ
Punkte	12	6	2	2	12	4	4	8	50
Erz. Punkte									

Zum Bestehen sind 23 Punkte erforderlich

Ich bin damit einverstanden, dass mein Klausurergebnis unter meiner Matrikelnummer im Internet einsehbar gemacht wird.

Aufgabe 1) Gegeben sei die mathematisch definierte Funktion

$$\begin{aligned} \text{sumBis} &: \mathbb{N} \rightarrow \mathbb{N} \\ \text{sumBis } n &= \sum_{i=1}^n i \end{aligned}$$

Geben Sie Haskell-Definitionen für `sumBis` mit folgenden Merkmalen an:

- eine rekursive Definition unter Verwendung eines `if-then-else`-Ausdrucks,
- eine rekursive Definition unter Verwendung von Musteranpassung,
- eine Definition unter Verwendung der `'..'`-Notation und der Prelude-Funktion `foldr`.
- Schreiben Sie zu jeder Lösung a-c jeweils auf, was passiert, wenn `sumBis` auf ein negatives Argument angewendet wird.

Aufgabe 2) Definieren Sie eine Haskell-Funktion, `vdA` (vertausch die Argumente), die angewendet auf eine zweistellige Funktion `f` eine zweistellige Funktion `f'` liefert, sodass für alle Argumente `x` und `y` gilt: `f x y` ist gleich `f' y x`.

- ohne Verwendung der Lambda-Notation,
- unter Verwendung der Lambda-Notation.

Aufgabe 3) Schreiben Sie eine Funktion `istGerade`, die testet, ob eine ganze Zahl gerade ist.

Aufgabe 4) Ist $([x, \text{Double}] \rightarrow \text{Char} \rightarrow x, y)$ ein zulässiger Typ in Haskell? Begründen Sie Ihre Antwort.

Aufgabe 5) a) Schreiben Sie eine rekursive Funktion `tud` und `verdopple`

$$\text{tud} :: \text{Num } t \Rightarrow (t \rightarrow \text{Bool}) \rightarrow [t] \rightarrow [t]$$

so dass `tud p l` aus `l` entsteht, indem jedes Element aus `l`, das `p` erfüllt, verdoppelt wird.

- b) Rufen Sie sich die Prelude-Funktionen `map` und `filter` in Erinnerung, indem Sie deren Definitionen hinschreiben.
- c) Schreiben Sie eine zu `tud` äquivalente Funktion `tud'` unter Verwendung von `map` und `filter`.
- d) Schreiben Sie eine zu `tud` äquivalente Funktion `tud''` unter Verwendung der ZF-Notation.
- e) Beweisen Sie durch strukturelle Induktion: `tud = tud'`.

Aufgabe 6) Reduzieren Sie folgenden λ -Ausdruck auf seine Normalform:

$$\left((\lambda x . y (\lambda z . xz)) z \right) y$$

Aufgabe 7) Gegeben sei die Klasse

```
class Sichtbar t where
  zeig :: t -> String; grÖÙe :: t -> Float
```

sowie folgende Typvereinbarungen:

```
data Figur          = Quadrat Seitenlänge | Ellipse Radius Radius
type Seitenlänge    = Float; type Radius = Float
```

Vereinbaren Sie `Figur` als Instanz der Klasse `Sichtbar`, so dass “Quadrat mit Seitenlänge 4.5” oder “Ellipse mit Radien 3.0 und 2.3” typische Ergebnisse der Funktion `zeig` sind und `grÖÙe` als Ergebnis jeweils die Fläche liefert.

Hinweis: Die Fläche einer Ellipse mit Radien $r1$ und $r2$ ist $\pi \cdot r1 \cdot r2$.

Aufgabe 8) Sie haben Ihr privates Telefonverzeichnis in Haskell durch folgende Datentypen modelliert:

```
type Name          = String; type Nummer = String
type Eintrag       = (Name, Nummer); type TelBuch = [Eintrag]
```

Schreiben Sie folgende Funktionen in Haskell:

- a) `tragEin :: TelBuch -> Eintrag -> TelBuch`
Hinweis: Diese Funktion soll aufgerufen werden, wenn ein neuer Eintrag in ein Telefonbuch gemacht werden soll.
- b) `such :: TelBuch -> Name -> Nummer`
Hinweis: Diese Funktion soll die Telefonnummer einer Person liefern.
- c) `sort :: TelBuch -> TelBuch`
Hinweis: Diese Funktion soll ein Telefonbuch alphabetisch nach Namen sortieren.
- d) Schreiben Sie Typ und Definition einer Funktion `lösche`, die jeden Eintrag zu einem gegebenen Namen aus einem Telefonbuch löscht.

Viel Erfolg!