

Klausur

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 25 Punkte erforderlich

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

Aufgabe 1) Die Fakultätsfunktion ist mathematisch durch

$\mathbf{fak} : \mathbb{N} \rightarrow \mathbb{N}$
 $\mathbf{fak} \quad n = 1 \star \dots \star n$

definiert. Geben Sie Haskell-Definitionen für **fak** an, die nicht nur auf kleinen Zahlen korrekte Ergebnisse liefert.

- a) durch eine rekursive Definition unter Verwendung eines **if-then-else**-Ausdrucks.
- b) durch eine rekursive Definition unter Verwendung von Musteranpassung.
- c) unter Verwendung der **'..'**-Notation und der Prelude-Funktion **product**.
- d) Kommentieren Sie zu Ihren Lösungen a-c jeweils was passiert, wenn **fak** auf ein negatives Argument angewendet wird.

Aufgabe 2) Definieren Sie eine Haskell-Funktion **nicht**, die angewendet auf ein Prädikat P das zu P komplementäre Prädikat P' liefert, d.h. wenn Px wahr ist gilt $P'x$ ist falsch und umgekehrt.

- a) ohne Verwendung des Kompositionsoperators,
- b) unter Verwendung des Kompositionsoperators,
- c) unter Verwendung eines **case**-Ausdrucks.

Aufgabe 3) Schreiben Sie eine Funktion **istZiffer**, die testet, ob ein Zeichen eine Ziffer ist.

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

Aufgabe 5)

- a) Schreiben Sie eine rekursive Funktion

$\mathbf{doppelt} :: \text{Num } t \Rightarrow (t \rightarrow t) \rightarrow [t] \rightarrow [t]$

so dass **doppelt** $f\ l$ aus l entsteht, indem f auf jedes Element von l zweimal angewendet wird.

- b) Rufen Sie sich die Prelude-Funktion `map` in Erinnerung, indem Sie eine rekursive Haskell-Definition für `map` angeben.
- c) Schreiben Sie eine äquivalente nicht rekursive Definition `doppelt` unter Verwendung von `map`.
- d) Schreiben Sie eine äquivalente Definition `doppelt''` unter Verwendung der ZF-Notation.
- e) Beweisen Sie mit struktureller Induktion: `doppelt = doppelt'`

Aufgabe 6) Reduzieren Sie folgenden λ -Ausdruck

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

auf seine Normalform.

Aufgabe 7) Gegeben sei die Klasse

```
class Sichtbar t where
  zeig  :: t → String
  größe :: t → Float
```

sowie der algebraische Datentyp

```
data Figur = Kreis Radius | Rechteck Länge Breite
type Radius = Float; type Länge = Float; type Breite = Float
```

Vereinbaren Sie `Figur` als Instanz der Klasse `Sichtbar`, so dass “Kreis mit Radius 4.7”, “Rechteck mit Länge 3.0 und Breite 8.1” typische Ergebnisse der Funktion `zeig` sind und `größe` als Ergebnis die jeweilige Fläche liefert.

Aufgabe 8) Eine Lehrerin modelliert die Ergebnisse ihrer Schulklasse in den schriftlichen Arbeiten eines Schulhalbjahres durch folgende Datentypen:

```
type Name      = String
type Note      = Float
type Arbeit    = (Name, Note)
type Ergebnisse = [Arbeit]
```

Schreiben Sie folgende Funktionen in Haskell:

- a) `besteSchüler :: Ergebnisse → [Name]`
Hinweis: Diese Funktion soll die Namen derjenigen Schüler liefern, die eine Arbeit mit 1,0 geschrieben haben.
- b) `eintrag :: Ergebnisse → Arbeit → Ergebnisse`
Hinweis: Diese Funktion soll aufgerufen werden, wenn eine Arbeit fertig korrigiert wurde und die Ergebnisliste hinzugefügt werden soll.
- c) `zeugnisNote :: Ergebnisse → Name → Note`
Hinweis: diese Funktion soll die Durchschnittsnote eines Schülers berechnen in Bezug auf eine vorliegende Ergebnisliste.
- d) `mittelwert :: Ergebnisse → Note`
Hinweis: Diese Funktion soll die Durchschnittsnote von allen eingetragenen Arbeiten ergeben.

Viel Erfolg!