



Universität Karlsruhe (TH)

Institut für Programmstrukturen und Datenorganisation (IPD)

Informatik I WS 2003/04

Dozent: Prof. Dr.rer.nat. G. Goos

Übungsleiter: Tom Gelhausen

<http://www.infoeins.de>

goos@ipd.info.uni-karlsruhe.de

gelhausen@fzi.de

Übungsblatt 8 - (15T / 45P)

Lambda-Kalkül und Haskell

Ausgabe: 05.11.2003

Abgabe: 12.12.2003

13:30 Uhr

Einwurf im Keller des Informatik-Hauptbaus (Geb. 50.34)

Hinweis: Benutzen Sie zum Lösen der Übungsaufgaben Ihren gesunden Menschenverstand!

1. Lambda-Kalkül (15T)

1.1 Ausdrücke (6T)

Gegeben seien folgende Ausdrücke im Lambda-Kalkül. Reduzieren Sie maximal unter Angabe der Reduktionsart.

Hinweis: +, -, *, / stehen für die üblichen arithmetischen Operationen.

a. $(\lambda x. (\lambda y. / y x)) 5 10$ (2T)

b. $(\lambda x. (\lambda x. + (- x 1)) x 3) 9$ (2T)

c. $(\lambda f. (\lambda x. f x)) (\lambda x. * x x) 2$ (2T)

1.2 Chursche Booleans (9T)

Die booleschen Wahrheitswerte repräsentieren eine Auswahl zwischen zwei Alternativen. Im Lambda-Kalkül werden sie als Funktionen dargestellt, die eines ihrer Argumente auswählen:

$$\text{true} = (\lambda x. (\lambda y. (x)))$$

$$\text{false} = (\lambda x. (\lambda y. (y)))$$

Auch logische Operationen werden als Funktionen dargestellt. Die Negation hat folgende Form:

$$\text{not} = (\lambda x. (x \text{ false true}))$$

a. Definieren Sie die Funktionen *and* und *or* im Lambda-Kalkül, welche die logischen Operationen $a \wedge b$, $a \vee b$ umsetzen. (3T)

b. Formen Sie folgenden Ausdruck um, so dass er ausschließlich hier definierten Operationen enthält. Drücken Sie ihn anschließend als Funktion im Lambda-Kalkül aus. (2T)

$$(a \vee b) \rightarrow c$$

c. Wenden Sie die Funktion auf eine Repräsentation der Belegung $a = \text{true}$, $b = \text{false}$, $c = \text{true}$ im Lambdakalkül an und reduzieren Sie maximal unter Angabe der Reduktionsart. (4T)

Hinweis: Sie brauchen die Operationen *true*, *false* und *not* nicht ausschreiben, die anderen hingegen schon.

2. Haskell (45P)

Verwenden Sie zur Lösung der Aufgaben **keine** Funktionen aus der Haskell-Library, sofern nichts anderes angegeben ist; +, -, mod... sind natürlich erlaubt! Geben Sie bitte zu jeder Funktion auch die zugehörige Signatur an.

2.1 Einfacher Einstieg (8P)

Entwerfen Sie Funktionen, die folgende Probleme lösen (je 2P):

- Addition zweier ganzer Zahlen
- Betragsberechnung einer ganzen Zahl
- Bestimmen des Minimums zweier ganzer Zahlen
- Prüfung, ob eine ganze Zahl gerade ist

Hinweis: Ausgabe soll ein boolescher Wert sein.

2.2 Beziehungskiste (4P)

Schreiben Sie eine Funktion `ehefrau`, die als Eingabe einen Männernamen erhält und den Namen der zugehörigen Ehefrau ausgibt. Bei einer falschen Eingabe soll eine Fehlermeldung ausgegeben werden. Verwenden Sie bitte folgende Beziehungen:

„Hans ist mit Barbara und Anton mit Rosa verheiratet“

Schreiben Sie die Funktion zweimal und verwenden Sie dazu unterschiedliche Ausdrücke (je 2P)!

2.3 Rekursion (6P)

- Schreiben Sie eine rekursive Funktion, die alle ganzen Zahlen von 1 bis zu einem gegebenen n miteinander multipliziert und das Produkt als Ergebnis ausgibt. Der Eingabeparameter n ist eine natürliche Zahl größer Null. (3P)
- Wandeln Sie ihre Funktion nun so um, dass sie nur das Produkt der geraden Zahlen zwischen 1 und n (inklusive) ausgibt. (3P)

2.4 Abstandsberechnung (5P)

Schreiben Sie eine Funktion, die als Eingabe zwei Punkte in x - y -Koordinaten als Kommazahlen erhält und den Abstand zwischen den Punkten berechnet. Je nachdem, ob der Abstand der Punkte größergleich 5, kleiner als 5 oder die Punkte identisch sind, soll ein anderer Text ausgegeben werden.

Hinweis: Verwenden Sie das `where`-Ausdruck zur Erleichterung.
Sie dürfen außerdem den Befehl `sqrt` verwenden.

2.5 Listen (22P)

- a. Schreiben Sie eine Funktion, die das letzte Element einer eingegebenen Liste ausgibt. (2P)
- b. Schreiben Sie eine Funktion, die zwei Listen nach dem Reißverschlussverfahren zusammenfügt.
z.B. [1, 2, 3] und [4, 5, 6] ergibt [1, 4, 2, 5, 3, 6] (3P)

Hinweis: Die beiden Listen müssen nicht gleich lang sein!

- c. Schreiben Sie eine Funktion, die aus einer Liste ganzer Zahlen das betragsmäßig größte Element heraussucht und ausgibt. (3P)
- d. Schreiben Sie eine Funktion, die eine Liste ganzer Zahlen als Eingabe erhält und prüft, ob die Listenelemente streng monoton steigend angeordnet sind. (5P)

Hinweis: Ausgabe soll ein boolescher Wert sein.

- e. Schreiben Sie eine Funktion, die eine Liste von Kommazahlen als Eingabe erhält und als Ausgabe eine geglättete Liste liefert. Dazu soll jeder Listeneintrag n durch den Mittelwert der Listeneinträge $n-1$, n und $n+1$ ersetzt werden. (9P)

Hinweise: Sie benötigen mehrere Funktionen
Passen Sie auf Randeffekte auf!