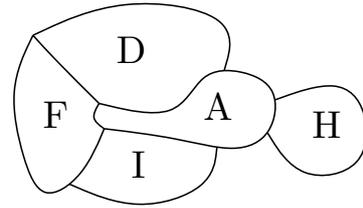


Informatik A, WS 2016/17 — 4. Übungsblatt

Abgabe bis Freitag, 11. November 2016, 12:00 Uhr. Aufgabe 17a klargestellt am 10.11.

17. Färben einer Landkarte, 10 Punkte

Die nebenstehende vereinfachte Landkarte von Europa soll mit drei Farben so gefärbt werden, dass Länder mit einer gemeinsamen Grenze verschiedene Farben haben.



- Schreiben Sie unter Verwendung geeigneter Boolescher Variablen eine Boolesche Formel, die die gültigen Färbungen beschreibt. Erklären Sie, was jede Variable bedeutet. Wieviele Variablen hat Ihre Formel? (Wenn Ihre Formel einem Schema folgt, können Sie ihn nach einigen Beispielen mit „usw.“ abkürzen, statt den ganzen Ausdruck explizit anzugeben.)
- Wie (a), aber Ihre Formel soll in konjunktiver Normalform sein. Wieviele Klauseln und Variablen hat Ihre Formel?
- Wieviele Möglichkeiten muss man betrachten, wenn man die Formel von Teil (b) durch Erstellen der Wahrheitstafel auf Erfüllbarkeit testen möchte?
- Wie lange würde das ungefähr dauern, wenn das Überprüfen einer Möglichkeit im Computer 0.02 Mikrosekunden dauert?

18. Programmieraufgabe, 10 Punkte.

Schreiben Sie eine Funktion `kleinerAlsDurchschnitt`, die berechnet, wie viele Zahlen in einer Liste von `Integer`-Werten *kleiner* als der Durchschnittswert sind. Bestimmen Sie selbst die richtige Signatur. Achten Sie auf die korrekte Behandlung der leeren Liste.

19. Preisberechnung, Programmieraufgabe, 10 Punkte

Listen der beiden folgenden Datentypen

```
type Einkaufsliste = [(String, Float)]
type Preisliste    = [(String, Float)]
```

geben an, *was* gekauft werden soll und *wieviele* (in einer passenden Einheit, z. B. kg), zum Beispiel `[("Mehl",0.5), ("Butter",0.25)]`, und andererseits den Preis in Euro pro Einheit für jeden Artikel.

- Definieren Sie eine Funktion

```
preis :: Preisliste -> Einkaufsliste -> (Float, [String])
```

zur Berechnung des Gesamtpreises aller Artikel einer Einkaufsliste. Die zweite Komponente des Ergebnisses soll die Liste der Artikelnamen enthalten, die in der Preisliste nicht gefunden wurden.

- Zusatzaufgabe, 0 Punkte. Erweitern Sie die Funktion so, dass der Preis für jeden gekauften Artikel der Einkaufsliste auf Cent gerundet wird. (Die `HASKELL`-Funktion `round` rundet eine Zahl zur nächsten ganzen Zahl.)

20. Die de-Morgan'schen Gesetze, 0 Punkte: $\overline{p \wedge q} \equiv \bar{p} \vee \bar{q}$, $\overline{p \vee q} \equiv \bar{p} \wedge \bar{q}$

Leiten Sie das zweite Gesetz aus dem ersten her, indem Sie im ersten Gesetz (i) für p und q die Ausdrücke \bar{x} und \bar{y} einsetzen, (ii) beide Seiten negieren, (iii) doppelte Negationen eliminieren und (iv) Variablen umbenennen. Funktioniert dieser Vorgang auch in die umgekehrte Richtung?

21. Äquivalenz und Antivalenz, 0 Punkte

- (a) Beweisen Sie, dass die Junktoren \oplus und \Leftrightarrow kommutativ und assoziativ sind.
- (b) Beweisen Sie: $\bar{x} \oplus y \equiv \overline{x \oplus \bar{y}}$, $x \oplus \bar{y} \equiv \overline{x \oplus y}$. (Verneinung wandert auf die höhere Ebene.)
- (c) Beweisen Sie: $x \oplus x \equiv falsch$, $x \oplus falsch \equiv x$, $\bar{x} \equiv x \oplus wahr$

22. NAND, 0 Punkte

Ist der Junktor $\bar{\wedge}$ assoziativ? Geben Sie einen Beweis oder ein Gegenbeispiel.

23. Zweierpotenzen und Zweierlogarithmen, 0 Punkte

Genauso wie die Zweierpotenzen spielt auch die Umkehrfunktion, der Logarithmus zur Basis 2, in der Informatik eine große Rolle:

$$b = \log_2 x \iff x = 2^b$$

- (a) Bestimmen Sie ohne Benutzung eines Taschenrechners die erste Dezimalziffer von $\log_2 32$, $\log_2 100$, $\log_2 1000$, und $\log_2 0.001$.
- (b) Wieviele Bits benötigt man, um ein Symbol abzuspeichern, das 26 verschiedene Werte annehmen kann?
- (c) Wieviele Bits benötigt man für n verschiedene Werte? Schreiben Sie eine Formel für die Anzahl der notwendigen Bits auf.
- (d) Schreiben Sie eine HASKELL-Funktion für Ihre Antwort aus (c). Den Logarithmus zu einer beliebigen Basis kann man mit der Funktion `logBase` bestimmen.

24. Geometrie, 0 Punkte

Ein Kreis in der Ebenen kann als ein Tupel von 3 Gleitkommazahlen dargestellt werden. Die ersten beiden werden interpretiert als die x - y -Koordinaten des Mittelpunktes, die dritte als Radius. Schreiben Sie ein Haskell-Programm, das die folgenden Funktionen implementiert.

- (a) Die Funktion `istKreis` überprüft, ob die drei Zahlen tatsächlich einen Kreis mit nichtleerer Fläche definieren.
- (b) Die Funktion `flächeBBox` berechnet die Fläche der sogenannten *bounding box* zweier Kreise, also des kleinsten achsenparallelen Rechtecks, das beide enthält.
- (c) Die Funktion `istEnthalten` überprüft von zwei Kreisen, ob einer davon im anderen enthalten ist. (Eine Möglichkeit ist, den Abstand der Mittelpunkte zu verwenden.)

25. Bedingte Ausdrücke, 0 Punkte

Schreiben Sie die folgende Funktion als einen *einzigsten* nicht geschachtelten if-then-else-Ausdruck:

```
test :: Int->Int->Int->Bool
test x y z
  | x <= y    = True
  | y <= z    = False
  | otherwise = x < z
```

Geht es auch ganz ohne if-then-else-Ausdruck und ohne Wächterbedingungen?

(Die einfachste mir bekannte Lösung passt auf eine Zeile mit 15 Zeichen.)