Informatik A, WS 2016/17 — Nachklausur

Abgabe bis Dienstag, 28. März 2017, 11:45 Uhr (90 Minuten, 40 Punkte)

---

1. Aussagenlogik, 10 Punkte

   (a) Umformung, 6 Punkte

   Beweisen Sie folgende Äquivalenzen, entweder durch Anwendung von Umformungsregeln oder durch Vergleichen der Wahrheitstafeln:

   $$(p \wedge s) \Rightarrow q \;\equiv\; (p \Rightarrow q) \vee (s \Rightarrow q) \tag{1}$$
   $$x \Leftrightarrow y \;\equiv\; (\neg x) \Leftrightarrow (\neg y) \tag{2}$$

   (b) NOR, 4 Punkte

   Der Junktor $\bar{\vee}$ (weder-noch, engl. „nor“) ist definiert als

   $$a \mathbin{\bar{\vee}} b \;\equiv\; \neg(a \vee b).$$

   Ist der Junktor $\bar{\vee}$ kommutativ?

   Ist er assoziativ?

   Geben Sie jeweils einen Beweis oder ein Gegenbeispiel.

2. Induktion, 10 Punkte

   (a) (4 Punkte) Bestimmen Sie alle Paare (k, l) von ganzen Zahlen, für die die folgende Gleichung gültig ist:

   ```
   (drop k) . (drop l) = drop (k+l)
   ```

   (Die Möglichkeit des Überlaufs bei der Berechnung von `k+l` können Sie außer Acht lassen.) *Geben Sie eine Begründung* für die Fälle, wo die Gleichung nicht gilt.

   (b) (6 Punkte) Für die Fälle, wo die Gleichung gilt, beweisen Sie sie unter Verwendung folgender Definitionen:

   ```
   drop :: Int -> [a] -> [a]
   drop n xs  | n <= 0 =  xs                  -- drop.0
   drop _ []           =  []                  -- drop.1
   drop n (x:xs)       =  drop (n-1) xs       -- drop.2

   (.) :: (b -> c) -> (a -> b) -> (a -> c)
   (f . g) x = f (g x)                        -- punkt.0
   ```

   Begründen Sie jeden Schritt, indem Sie angeben, welche Definitionsgleichung (drop.0 oder drop.1 oder drop.2 oder punkt.0) oder welche anderen Eigenschaften sie verwenden.

3. Programmieren, 10 Punkte

   (a) Listen verzahnen, 3 Punkte
   Schreiben Sie eine Funktion `jedesZweite`, die zwei Listen im Reißverschlusssystem verzahnt. Die Elemente der ersten Liste kommen an die ungeraden Positionen des Ergebnisses, und die Elemente der zweiten Liste füllen die geraden Positionen. Die Liste endet, sobald eine der Eingabelisten erschöpft ist.
   Beispiele:

   ```
   jedesZweite [4,7,9,7] [2,8,-1,2,3] = [4,2,7,8,9,-1,7,2]
   jedesZweite "Atrpazifik" "senlt" = "Asternplatz"
   ```

   Überlegen Sie selbst, welchen Datentyp die Funktion hat, und *schreiben Sie eine Typdeklaration.*

   (b) Wie viele gleiche? 7 Punkte
   Schreiben Sie eine Funktion

   `wieVieleGleiche:: Eq a => [a] -> Int`,

   die bestimmt, wie oft das häufigste Element in einer Liste vorkommt.
   Beispiele:

   `wieVieleGleiche [9,4,2,4] = 2`, weil das Element 4 zweimal vorkommt.
   `wieVieleGleiche [4,9,2,14,5] = 1`, weil jedes Element nur einmal vorkommt.

   Beachten Sie, dass die Elemente der Eingabeliste nicht zur Typklasse `Ord`, sondern nur zur Typklasse `Eq` gehören!
   Sie können beliebige Funktionen aus den Standardbibliotheken zu Hilfe nehmen. (Bei der Bewertung wird nicht darauf geachtet, ob sie die richtigen Modulnamen für den Import der Bibliotheksfunktionen schreiben.)

   (c) Zusatzfrage (2 Zusatzpunkte). Analysieren Sie die asymptotische Laufzeit Ihres Programms zu Aufgabe (b).

4. Entwurf eines Schaltnetzes, 10 Punkte

   Gesucht ist ein Schaltnetz mit drei Eingängen $a, b, c \in \{0, 1\}$ und zwei Ausgängen $u, v \in \{0, 1\}$.

   Der Ausgang u soll gleich a sein, falls $b = 0$ ist; andernfalls soll er gleich c sein.

   Wenn alle drei Eingänge den gleichen Wert haben, soll der Ausgang v gleich 0 sein; andernfalls soll v gleich b sein.

   (a) (4 Punkte) Beschreiben Sie die Ausgänge u und v durch logische Ausdrücke (nur mit Junktoren, nicht mit if-then-else-Ausdrücken).

   (b) (6 Punkte) Konstruieren ein Schaltnetz und zeichnen Sie es.
   Verwenden Sie dabei die DIN-Symbole für die Gatter. Gatter mit mehr als zwei Eingängen sind nicht erlaubt.
   Schreiben Sie zum Ausgang jedes Gatters (außer zu u und v) *einen Booleschen Ausdruck* in den Eingabevariablen, der den Wert charakterisiert.