Probeklausur zur Vorlesung

Informatik A

WiSe 2015/16

Wolfgang Mulzer

Bearbeitungszeit: mindestens 90 Minuten

Aufgabe 1 Aussagenlogik *3+3+2+2 Punkte*

Betrachten Sie den folgenden Booleschen Term t:

$$x \wedge y \vee \neg z \vee (\neg y \vee x \wedge z)$$

(a) Geben Sie eine vollständige Klammerung für t an und zeichnen Sie die Baumdarstellung für t. Bestimmen Sie den Rang von t.

(b) Stellen Sie die Wertetabelle für die dreistellige Boolesche Funktion $f(x, y, z)$ auf, die durch t definiert ist.

(c) Geben Sie die kanonische DNF für $f(x, y, z)$ an.

(d) Wie viele n-stellige Boolesche Funktionen gibt es? Begründen Sie Ihre Antwort.

Aufgabe 2 Haskell *6+4 Punkte*

(a) Bei welchen der folgenden Zeilen handelt es sich um gültige Haskell-Ausdrücke? Geben Sie im positiven Fall den Wert und den Typ des Ausdrucks an, begründen Sie im negativen Fall, warum der Ausdruck nicht gültig ist.

(i) `if 3 >= 4 then True else 10`

(ii) `10:20:if False then [] else [16,17]`

(iii) `('c', 2+3, even 11)`

Hinweis: Die Funktion `even` liefert genau dann `True`, wenn das Argument eine gerade Zahl ist.

(b) Definieren Sie eine rekursive Funktion `int2str`, welche eine *positive* ganze Zahl in einen String umwandelt. Zum Beispiel soll `int2str 10` den String `''10''` liefern. Geben Sie auch die Signatur an und erklären Sie Ihre Funktion kurz.

Hinweise: Um eine Ziffer `d`, $0 \leq$ `d` ≤ 9 in das entsprechende Zeichen umzuwandeln, benutze man den Ausdruck `chr (ord '0' + d)`.

Die Funktion `div::Int -> Int -> Int` führt eine ganzzahlige Division aus, `mod::Int -> Int -> Int` liefert den Rest. Zum Beispiel ergeben `div 10 3` den Wert `3` und `mod 10 3` den Wert `1`.

Aufgabe 3 Vermischtes *3+2+2+3 Punkte*

(a) Testen Sie mit dem Resolutionskalkül die folgende Formel auf Erfüllbarkeit:

$$t = (x \vee y \vee \neg z) \wedge (x \vee \neg y) \wedge (x \vee y \vee z) \wedge \neg x.$$

(b) Geben Sie die Definition einer *funktional vollständigen logischen Signatur*. Beweisen Sie, dass $\{\neg, \vee\}$ funktional vollständig ist. Dabei dürfen Sie verwenden, dass $\{\neg, \vee, \wedge\}$ funktional vollständig ist.

(c) Geben Sie die Definition einer *Typklasse* in Haskell. Nennen Sie zwei Operationen der Typklasse `Ord`.

(d) Schreiben Sie eine prädikatenlogische Formel, die aussagt, dass für natürliche Zahlen größer 1 gilt, dass zwischen der Zahl und ihrem Doppelten mindestens eine Primzahl liegt. Sie dürfen ein Prädikat $\text{isprim}(n)$ benutzen.

Negieren Sie dann die Formel, und formen Sie um, bis kein Negationszeichen vor einem Quantor steht.

Aufgabe 4 Listen in Haskell *3+2+5 Punkte*

(a) Schreiben Sie eine polymorphe und rekursive Funktion `concat`, welche eine Liste von Listen erhält und die gegebenen Listen zu einer Liste zusammenführt. Zum Beispiel soll `concat [[1,2],[2,3]]` das Ergebnis `[1,2,2,3]` liefern und `concat [[True, False], [], [True]]` soll `[True, False, True]` ergeben.

Geben Sie auch die Signatur an und erklären Sie Ihre Funktion.

Hinweis: Der Operator `++` verbindet zwei Listen zu einer neuen Liste.

(b) Implementieren Sie `concat` aus (a) mit `foldr`.

Erklären Sie, wie Ihre Funktion den Ausdruck `concat [[True, False],[], [True]]` auswertet, indem Sie die `foldr`-Operation explizit ausschreiben.

(c) Sei `t` ein Typ aus der Typklasse `Ord`. Schreiben Sie eine Funktion `minmax`, die zu einer gegebenen nichtleeren Liste `xs` vom Typ `t` das Minimum *und* das Maximum bestimmt. Ihre Funktion soll durch eine direkte Rekursion implementiert sein und die *Akkumulatortechnik* verwenden.

Geben Sie alle Signaturen an und erklären Sie Ihre Funktion. Zeigen Sie eine schrittweise Auswertung des Ausdrucks `minmax [1,5,0,2]`.