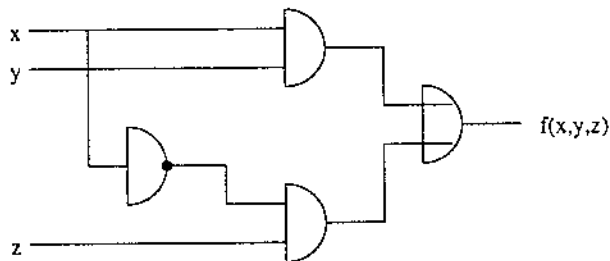


1. Aufgabe (9 Punkte): Boolesche Funktionen, Boolesche Terme, Schaltnetze

- (a) (3 Punkte) Bestimmen Sie die Wertetabelle der Booleschen Funktion  $f : B^3 \rightarrow B$ , die durch unten dargestelltes Schaltnetz realisiert wird und geben Sie den zum Schaltnetz gehörenden Booleschen Ausdruck  $t$  an!
- (b) (3 Punkte) Bestimmen Sie aus der Wertetabelle die kanonische DNF der Funktion und beweisen Sie durch Anwendung der Gesetze der Booleschen Algebra die semantische Äquivalenz dieser DNF und  $t$ .
- (c) (3 Punkte) Finden Sie ein Schaltnetz, das einen semantisch äquivalenten Booleschen Ausdruck zu  $\text{FALSE} : B \rightarrow B$  realisiert und nur aus NOR-Gattern besteht. Leiten Sie zunächst den Ihrem Schaltnetz zu Grunde liegenden Ausdruck, der nur NOR-Operatoren benutzt, her.



2. Aufgabe (9 Punkte): Listenfunktionen in Haskell

Definieren Sie folgende Funktionen in Haskell und kommentieren Sie Ihre Definitionen.

- (a) (3 Punkte) Eine Funktion `ordTrip`, die für Listen über einem allgemeinen Ordnungstyp `a` aufeinanderfolgende Tripel von Listeneinträgen jeweils in sich aufsteigend sortiert. Ist die Listenlänge kein Vielfaches von 3, so wird der Rest abgeschnitten.  
Beispiel: `ordTrip [12,3,5,4,6,1,3] = [3,5,12,1,4,6]`
- (b) (3 Punkte) Schreiben Sie eine Funktion `smaller :: Ord a => [[a]] -> a -> Bool`, die testet, ob in einer Liste von Listen über einem Ordnungstyp `a` eine Liste enthalten ist, deren sämtliche Einträge echt kleiner als ein vorgegebener Schwellwert sind.
- (c) (3 Punkte) Eine Funktion, die für Listen `xs` über einem Typ `a` und ein Prädikat `pred :: a -> Bool` eine Liste der Indizes von Einträgen in `xs` bestimmt, die das Prädikat erfüllen.

3. Aufgabe (8 Punkte): Algebraische Typen, Typklassen

- (a) (2 Punkte) Definieren Sie einen algebraischen Typ `TernTree` von ternären Bäumen, wobei in den Knoten ganze Zahlen abgespeichert sein sollen. Ein ternärer Baum ist im Gegensatz zu binären Bäumen einer, bei dem innere Knoten nicht zwei sondern 3 Söhne haben.
- (b) (3 Punkte) Machen Sie `TernTree` zur Instanz der Klasse `Eq`, wobei zwei Bäume gleich sein sollen, wenn die beiden Summen der Absolutbeträge ihrer Knoteneinträge gleich sind.
- (c) (3 Punkte) Schreiben Sie eine Funktion `isBalanced :: TernTree -> Bool`, die feststellt, ob ein ternärer Baum balanciert ist, das heißt, alle Blätter die gleiche Tiefe haben.

Hinweis: Definieren Sie geeignete Hilfsfunktionen.

4. Aufgabe (8 Punkte): Noch ein bisschen "Theorie"

(a) (5 Punkte) Beweisen Sie mittels vollständiger Induktion über die Listenlänge, dass für alle Listen  $xs$  gilt:

$$\text{filter } p \ (\text{filter } q \ xs) = \text{filter } (p \ \&\&\ q) \ xs,$$

wobei  $p \ \&\&\ q$  die Funktion ist, die die Ergebnisse der beiden Booleschen Prädikate  $p, q$  auf einem Argument durch Konjunktion verbindet. Zur Erinnerung:

$$\text{filter } p \ [] = [] \quad (\text{filter.1})$$

$$\text{filter } p \ (x:xs) \\ | \ p \ x \quad = \ x: \ \text{filter } p \ xs \quad (\text{filter.2})$$

$$| \ \text{otherwise} = \ \text{filter } p \ xs \quad (\text{filter.3})$$

(b) (3 Punkte) Richtig oder falsch? Begründen Sie kurz Ihre Antwort.

Ist in einem optimalen Präfixcode für eine Menge von Zeichen mit gegebener Wahrscheinlichkeitsverteilung ein Codewort der Länge  $k \geq 1$  enthalten, so gibt es im Code auch ein Codewort für jede Länge  $l$  mit  $1 \leq l \leq k$ .

1. Aufgabe (9 Punkte): Boolesche Funktionen, Boolesche Terme, Schaltnetze

Gegeben sei die Boolesche Funktion  $f : B^3 \rightarrow B$  durch

$$f(x_1, x_2, x_3) = (x_3 \Rightarrow x_2) \vee (\neg x_3 \Rightarrow x_1)$$

- (a) (2 Punkte) Bestimmen Sie die Wertetabelle der Funktion  $f$ !
- (b) (3 Punkte) Bestimmen Sie aus der Wertetabelle die kanonische DNF der Funktion und vereinfachen Sie diese nach Möglichkeit.
- (c) (1 Punkte) Geben Sie eine direkte Realisierung der von Ihnen gefundenen vereinfachten DNF mittels eines Schaltnetzes wieder, das NOT-, AND- und OR-Gatter enthalten darf.
- (d) (3 Punkte) Finden Sie ein Schaltnetz, das einen semantisch äquivalenten Ausdruck realisiert und nur aus NAND-Gattern besteht.

2. Aufgabe (9 Punkte): Listenfunktionen in Haskell

Definieren Sie folgende Funktionen in Haskell und kommentieren Sie Ihre Definitionen.

- (a) (3 Punkte) Eine Funktion `firstPrimeLarger`, die für eine ganzzahlige Eingabe `n` die kleinste Primzahl `p` (die ist natürlich positiv) echt größer als `n` bestimmt.
- (b) (3 Punkte) Eine möglichst allgemeine Funktion `powerOfTwo`, die von Listen testet, ob ihre Länge eine Zweierpotenz  $2^i, i \geq 0$ , ist.
- (c) (3 Punkte) Eine Funktion, die für nichtleere Listen ganzer Zahlen die beiden arithmetischen Mittelwerte der Listeneinträge mit geraden bzw. ungeraden Indizes berechnet und als Paar ausgibt. (Der erste Listeneintrag hat Index 1 usw.) Für leere Listen sei der Funktionswert gleich `(0,0)`.

3. Aufgabe (8 Punkte): Algebraische Typen, Typklassen

- (a) (2 Punkte) Definieren Sie einen algebraischen Typ `Getränk`, der die einstelligen Konstruktoren `Wodka Int`, `OSaft Int`, `Eis Int` mit jeweils ganzzahligem Parameter (entspricht Volumenangabe) sowie einen dreistelligen Konstruktor `Mix Getränk Getränk Getränk` enthält.
- (b) (1 Punkt) Welche Art von bewerteten Bäumen liegt den Objekten des Typs `Getränk` zu Grunde?
- (c) (3 Punkte) Schreiben Sie eine Funktion `anteile :: Getränk -> (Int, Int, Int)`, die den Wodka-, Eis- und Orangensaftanteil in einem Getränk bestimmt.
- (d) (2 Punkte) Machen Sie `Getränk` zur Instanz von `Eq`, wobei zwei Getränke als gleich gelten sollen, wenn die Mischungsverhältnisse (Wodka + Orangensaft) zu Eis übereinstimmen.

4. Aufgabe (6 Punkte): Noch ein bisschen "Theorie"

- (a) (3 Punkte) Beweisen Sie mittels vollständiger Induktion über die Listenlänge, dass für alle Listen  $xs$  gilt:

$$\text{length (reverse } xs) = \text{length } xs,$$

wobei `reverse` die Funktion ist, die eine Liste umdreht, definiert mittels:

$$\text{reverse } [] = [] \quad (\text{reverse.1})$$

$$\text{reverse } (z:zs) = \text{reverse } zs ++ [z] \quad (\text{reverse.2})$$

- (b) (3 Punkte) Berechnen Sie den Huffman Code für das folgende Alphabet mit den angegebenen relativen Wahrscheinlichkeiten:

$$(a, 0.19), (b, 0.18), (c, 0.07), (d, 0.21), (e, 0.25), (f, 0.1)$$