Teil-Musterlösung zum 2. Aufgabenblatt zur Vorlesung

MafI I: Logik & Diskrete Mathematik

(F. Hoffmann)

1. **Tautologie** Überprüfen Sie, ob folgende Formeln Tautologien sind, ohne Wertetabellen aufzustellen.

$$t_1 = [(p \vee q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow r)] \Rightarrow r$$

Lösung: Wenn die Prämisse der großen Implikation falsch ist, wird der Ausdruck zu 1 ausgewertet. Ist sie wahr, so muss p oder q 1 sein, sagen wir p. Wegen der Implikation $p \Rightarrow r$ muss dann auch r wahr sein, also auch die große Implikation. Also ist es eine Tautologie.
Man kann natürlich auch umformen:

$$[(p \vee q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow r)] \Rightarrow r$$

$$\equiv \neg(p \vee q) \vee (p \wedge \neg r] \vee [q \wedge \neg r] \vee r$$

$$\equiv \neg(p \vee q) \vee (p \vee r) \wedge 1 \vee (q \vee r) \wedge 1 \equiv \neg(p \vee q) \vee (p \vee q) \vee r \equiv 1$$

$$t_2 = [(p \Rightarrow q) \Rightarrow r] \Leftrightarrow [p \Rightarrow (q \Rightarrow r)]$$

Lösung: Mit $p = 0$ wird die rechte Seite der Äquivalenz immer wahr, die linke Seite mit zusätzlich $r = 0$ aber falsch. Also keine Tautologie.

2. **DNF und KNF**
Betrachten Sie den Booleschen Ausdruck:

$$x \Leftrightarrow y \vee \neg z \wedge y$$

Geben Sie zunächst die vollständige Klammerung und die Baumdarstellung für diesen Ausdruck an. Bilden Sie dann dazu die kanonische DNF und die kanonische KNF.

3. **Funktionale Vollständigkeit von Signaturen**

   (a) Wir wissen, dass sowohl der NAND-Operator $\{|\}$ wie auch der NOR-Operator $\{\downarrow\}$ schon funktional vollständig ist. Schreiben Sie den NOR-Operator semantisch äquivalent mittels NAND-Operators und umgekehrt.
   Lösung: Wegen $x \downarrow y = \neg(x \vee y) \equiv \neg\neg(\neg x \wedge \neg y)$ und $\neg x \equiv x|x$ ergibt sich:

$$x \downarrow y \equiv [(x|x)|(y|y)]|[(x|x)|(y|y)]$$

   Völlig analog:

$$x|y \equiv [(x \downarrow x) \downarrow (y \downarrow y)] \downarrow [(x \downarrow x) \downarrow (y \downarrow y)]$$

(b) Schreiben Sie Implikation und Antivalenz semantisch äquivalent unter Verwendung des NAND-Operators.
Lösung:

$$x \Rightarrow y \equiv \neg x \vee y \equiv \neg(x \wedge \neg y) \equiv x|(y|y)$$

$$x \oplus y \equiv \neg(x \Leftrightarrow y) \equiv \neg(x \Rightarrow y \wedge y \Rightarrow x) \equiv [x|(y|y)]|[y|(x|x)]$$

(c) Zeigen Sie, dass der Äquivalenzoperator allein nicht funktional vollständig ist.
Lösung: Wir betrachten die Variablenmenge $V = \{x_1, x_2, \ldots, x_n\}$. Jeder Term über dieser Variablenmenge, der nur den Äquivalenzoperator benutzt wird bei Belegung $(1, 1, \ldots, 1)$ zu 1 ausgewertet. Jede Boolesche Funktion, in deren Wertetabelle dem Argument $(1, 1, \ldots, 1)$ der Wert 0 zugeordnet wird ist also nicht Interpretation eines solchen Terms.
Hinweis: Nimmt man noch den Term $true$ hinzu, so gewinnt man nichts an Funktionalität, den $t \Leftrightarrow true \equiv t$. Auch $false$ reicht nicht. Man kann dann die Negation simulieren, denn $t \leftrightarrow false \equiv \neg t$, aber es ist noch nicht vollständig, ein Beweis läuft analog zu Aufgabe (e).

(d) Argumentieren Sie, dass unter der Annahme, dass $\{\oplus, \neg\}$ nicht funktional vollständig ist, auch $\{\oplus, \neg, \Leftrightarrow, true\}$ nicht funktional vollständig ist.
Lösung: Wir zeigen, dass sich alles, was man mit $\{\oplus, \neg, \Leftrightarrow, true\}$ ausdrücken kann, schon mit $\{\oplus, \neg\}$ semantisch äquivalent darstellen lässt. Dies folgt aus:

$$t \Leftrightarrow t' \equiv \neg(t \oplus t'$$

$$true \equiv t \oplus \neg t$$

(e) **Schwieriger!** Zeigen Sie, dass $\{\oplus, \neg\}$ nicht funktional vollständig ist!
Tipp: Benutzen Sie Assoziativität und Kommutativität der Antivalenz.
Lösung: Zunächst zeigt man Assoziativität und Kommutativität etwa mit Tabellenmethode. Sei $V = \{x_1, x_2, \ldots, x_n\}$ die Variablenmenge und t einTerm darüber. Wir formen t semantisch äquivalent um. Zunächst können wir wegen $\neg(t_1 \oplus t_2) \equiv \neg t_1 \oplus t_2$ annehmen, dass sich Negationszeichen nur auf einzelne Variable beziehen. Gruppieren wir gleiche Literale nebeneinander, so kommt es nur auf deren Parität an. Eine gerade Anzahl vereinfacht sich zu $false$ und das kann man streichen (wenn nicht der ganze Term $false$ ist), eine ungerade Anzahl zu einem Exemplar des Literals. Treten jetzt sowohl x als auch $\neg x$, so liefert das zusammen $true$. Eine gerade Anzahl von $true$ vereinfacht sich wieder zu $false$, bei ungerader Anzahl bleibt ein $true$ übrig.
Was haben wir erreicht? Der Ausgangsterm t ist semantisch äquivalent zu einem Term t', in dem jede Variable höchstens einmal auftritt (also als x, $\neg x$ oder gar nicht) oder der konstant $true$ bzw. $false$ ist. Wie viele solche t' kann es also syntaktisch geben? Neben den beiden konstanten Termen gibt es $3^n - 1$ viele Möglichkeiten. Für jede der n Variablen gibt es 3 Möglichkeiten, insgesamt 3^n, die 1 Möglichkeit, dass alle nicht im Term sind, können wir abziehen. Für $n > 1$ ist aber $3^n + 1 < 2^{2^n}$ und damit können nicht alle n-stelligen Booleschen Funktionen realisiert werden.

4. **Quantoren**

(a) Schreiben Sie eine quantifizierte Formel, die ausdrückt, dass wenigstens 3 verschiedene Elemente aus einem Grundbereich ein Prädikat $P(x)$ erfüllen.
Lösung:

$$\exists x \exists y \exists z : P(x) \wedge P(y) \wedge P(z) \wedge x \neq y \wedge y \neq z \wedge z \neq x$$

(b) Schreiben Sie die folgenden Formeln in Negationsnormalform, das heißt, die verwendeten Negationen beziehen sich nur auf atomare Teilformeln und es werden außer den Quantoren nur die Junktoren $\neg, \vee, \wedge$ benutzt.

i. $\neg \exists x \ \exists y \ (P(x,y) \wedge Q(x,y) \wedge R(x,y))$
Lösung:

$$\neg \exists x \ \exists y \ (P(x,y) \wedge Q(x,y) \wedge R(x,y)) \equiv \forall x \forall y \neg P(x,y) \vee \neg Q(x,y) \vee \neg R(x,y)$$

ii. $\neg \forall y \ (\exists x \ (P(x,y) \Rightarrow Q(x,y)) \wedge \exists z \ R(x,z))$
Lösung:

$$\neg \forall y \ (\exists x \ (P(x,y) \Rightarrow Q(x,y)) \wedge \exists z \ R(x,z)) \equiv \exists y \forall x (P(x,y) \wedge \neg Q(x,y)) \vee \forall z \neg R(x,z))$$