

Inhaltsverzeichnis

Inhaltsverzeichnis	1
3.Logik	2
3.1 Zahlensysteme	2
3.2 Grundbegriffe zweiwertiger Logik.....	13
3.3 Rechengesetze für logische Ausdrücke.....	19
3.4 Logische Funktionen	24
3.5 Logische Normalformen	28
3.6 KV-Diagramm und Minimalformen	33
3.7 Technische Anwendungen und Boolesche Algebra.....	37
3.8 Binäre Codes	43
3.9 Fazit.....	48

3. Logik

Im dritten Teil der Vorlesung von Informatik I beschäftigen wir uns mit der sogenannten Booleschen Algebra. Diese ist Voraussetzung für die folgende Vorlesung "Digitaltechnik", die im 2. Semester stattfindet. Nach einer grundlegenden Einführung in die Menge der Dezimal-, Dual- und Hexadezimalzahlen (insbesondere der Arithmetik auf diesen Zahlen), folgen Basisinformationen über die zweiwertige Logik. Bei Schaltungsentwürfen ist der Begriff der logischen Funktion wesentlich. Nach der Definition dieses Begriffes werden unterschiedliche Darstellungsarten für die logischen Funktionen diskutiert. Diese können als Wertetabellen, Karnaugh-Veitch-Diagramme oder als logische (algebraische) Ausdrücke formuliert werden. Diese Formen sind logisch untereinander äquivalent. Da bei technischen Problemen der Begriff der kanonisch disjunktiven Normalform (KDN) immer zu finden ist, ist dieser Sachverhalt ausführlich dargestellt. Ein Kapitel ist dann der Minimierung von logischen Normalformen mit Hilfe von KV-Diagrammen gewidmet. Anschließend folgen mehrere technische Beispiele, die systematisch von der Problemstellung, zur Aufstellung der Wertetabelle, Formulierung und Minimierung der KDN untersucht werden. Im Abschluß dieser Ausarbeitung finden sich einfache Betrachtungen zu Codes (BCD, Aiken, Gray).

3.1 Zahlensysteme

Zehnersystem (Dezimalsystem):

Im Zehnersystem existiert als Ziffernmengende Menge $Z = \{0,1,2,\dots,8,9\}$. Die Mächtigkeit von Z (mit $|Z|$ bezeichnet) ist 10. Man nennt die Mächtigkeit von Z auch Basis B . B ist hier also 10.

Welchen Wert hat eine Ziffernfolge im Zehnersystem?

$$348 = 3 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0$$

Die Wertigkeit einer Zahl im Zehnersystem ergibt sich also als Summe von n Summanden (n ist die Anzahl der Ziffern). Jeder einzelne Summand ist das Ergebnis der Multiplikation einer Ziffer mit einer Potenz der Basis 10. Der Exponent (die Hochzahl) dieser Potenz ist $n - i$, wobei i die Ziffernposition in der Zahl angibt. In obigem Beispiel steht die Ziffer 3 an der ersten Stelle, 4 an der zweiten Stelle und 8 an der dritten Stelle.

Binärsystem (Dualsystem):

$$Z = \{0,1\}, \quad |Z| = 2 \quad .$$

Beispiel für eine Binärzahl: $(10101001)_2$

Hexadezimalsystem (Sedezimalsystem):

$$Z = \{0,1,2,3,\dots,9,A,B,C,D,E,F\}, \quad |Z| = 16 \quad .$$

Beispiel für eine Hexadezimalzahl: $(3F2_{16})$

Fragen:

1. Warum existieren unterschiedliche Zahlensysteme?
2. Welche Darstellung hat eine Zahl aus dem Zehnersystem im Dual-/Hexadezimalsystem und umgekehrt eine Zahl aus dem Dual-/Hexadezimalsystem im Zehnersystem ?
3. Wie kann man eine Arithmetik (+, -, *, /) auf den unterschiedlichen Zahlensysteme deklarieren?

Zu 1.

Im täglichen Leben gibt es unterschiedliche Zahlensysteme. Am meisten wird das 10er-System benutzt, das uns für Zählen, Messen und Rechnen wohl am Nächsten ist, da wir 10 Finger besitzen. Bei Zeitangaben verwenden wir aber aus historischen Gründen das 60er-System. Die Zeitangabe "3 Stunden 14 Minuten 52 Sekunden" entspricht $3 \cdot 60^2 + 14 \cdot 60^1 + 52 \cdot 60^0$ Sekunden. Der etwas älteren Verpackungseinheit *Gros* (1 Gros = 12 Dutzend, 1 Dutzend = 12 Stück) liegt das 12er-System zugrunde: "4 Gros 7 Dutzend 5 Stück" bedeutet $4 \cdot 12^2 + 7 \cdot 12^1 + 5 \cdot 12^0$ Stück. Das Binärsystem spielt bei Computern eine wichtige Rolle, da es die Informationsdarstellung mittels Digitalelektronik in idealer Weise widerspiegeln kann. Das Hexadezimalsystem erlaubt die Verkürzung der Darstellung von Informationen aus dem Binärsystem, indem vier Binärziffern zu einer Hexadezimalziffer zusammengefasst werden (dies findet z.B. bei Hexeditoren für Dateiinhalte oder Hauptspeicher Anwendung).

Zu 2.

Welche Darstellung hat $(10101001)_2$ im Zehnersystem?

$$\begin{aligned}(10101001)_2 &= 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 128 + 0 + 32 + 0 + 8 + 0 + 0 + 1 = (169)_{10}\end{aligned}$$

$$(343)_{16} = (?)_{10}$$

$$\begin{aligned}(343)_{16} &= 3 \cdot 16^2 + 4 \cdot 16^1 + 3 \cdot 16^0 \\ &= 768 + 64 + 3 = (835)_{10}\end{aligned}$$

$$(4AF)_{16} = (?)_{10}$$

$$\begin{aligned}(4AF)_{16} &= 4 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0 \\ &= 1024 + 160 + 15 = (1199)_{10}\end{aligned}$$

Gegeben sei eine Zahl aus dem Zehnersystem. Welche Darstellung besitzt diese im Dual-/Hexadezimalsystem?

$$(37)_{10} = (?)_2$$

Um diese Frage zu beantworten, kann man einen Divisions-Algorithmus benutzen:

$$37:2 = 18 \text{ Rest } 1$$

$$18:2 = 9 \text{ Rest } 0$$

$$9:2 = 4 \text{ Rest } 1$$

$$4:2 = 2 \text{ Rest } 0$$

$$2:2 = 1 \text{ Rest } 0$$

$$1:2 = 0 \text{ Rest } 1$$

Liest man die Reste von unten nach oben $\rightarrow (100101)_2$, so ergibt sich die Binärdarstellung von $(37)_{10}$.

$$(348)_{10} = (?)_{16}$$

Anwendung des Divisions-Algorithmus:

$$348:16 = 21 \text{ Rest } 12 = C$$

$$21:16 = 1 \text{ Rest } 5 = 5$$

$$1:16 = 0 \text{ Rest } 1 = 1$$

Liest man die Reste von unten nach oben $\rightarrow (15C)_{16}$, so ergibt sich die Hexadezimaldarstellung von $(348)_{10}$.

$$(1049)_{10} = (?)_{16}$$

$$1049:16 = 65 \text{ Rest } 9 = 9$$

$$65:16 = 4 \text{ Rest } 1 = 1$$

$$4:16 = 0 \text{ Rest } 4 = 4$$

Liest man die Reste von unten nach oben $\rightarrow (419)_{16}$, so ergibt sich die Hexadezimaldarstellung von $(1049)_{10}$.

$$(15C)_{16} = (?)_2$$

Zur Beantwortung dieser Frage bieten sich zwei Wege an.

1. Weg

Wandle die Hexadezimaldarstellung in die Dezimaldarstellung um. Wandle dann die Dezimaldarstellung mit obigem Divisions-Algorithmus in die Dualdarstellung um.

$$(15C)_{16} = 1 \cdot 16^2 + 5 \cdot 16^1 + 12 \cdot 16^0 = 256 + 80 + 12 = (348)_{10}$$

$$(348)_{10} = (?)_2$$

$$348:2 = 174 \text{ Rest } 0$$

$$174:2 = 87 \text{ Rest } 0$$

$$87:2 = 43 \text{ Rest } 1$$

$$43:2 = 21 \text{ Rest } 1$$

$$21:2 = 10 \text{ Rest } 1$$

$$10:2 = 5 \text{ Rest } 0$$

$$5:2 = 2 \text{ Rest } 1$$

$$2:2 = 1 \text{ Rest } 0$$

$$1:2 = 0 \text{ Rest } 1$$

Lies die Reste von unten nach oben $\rightarrow (101011100)_2$.

2. Weg

Stelle jede Hexadezimalziffer mit vier Dualziffern dar.

$$(1)_{16} = (0001)_2$$

$$(5)_{16} = (0101)_2$$

$$(C)_{16} = (1100)_2$$

Zusammenfassen (diesmal von oben nach unten) : $(0001) (0101) (1100) = (000101011100)_2$
 $= (101011100)_2$.

Zu 3: Arithmetik (+, -, *, /) in Zahlensystemen

Addition:

Vereinbarung

$$(0)_2 + (0)_2 = (0)_2$$

$$(0)_2 + (1)_2 = (1)_2$$

$$(1)_2 + (0)_2 = (1)_2$$

$$(1)_2 + (1)_2 = (0)_2 \text{ Übertrag } (1)_2$$

Beispiel:

Addition im Dualsystem:

$$\begin{array}{r} 1 \ 10 \ 11 \quad 27 \\ +_1 1_1 10_1 10 \quad +26 \\ \hline \underline{\underline{11 \ 01 \ 01 \quad 53}} \end{array}$$

Addition im Hexadezimalsystem:

$$\begin{array}{r} (4 \ 7 \ 3)_{16} \quad 3 + F = 2 (+1\ddot{U}) \\ + (1_1 B_1 F)_{16} \quad 7 + B_{+1\ddot{U}} = 3(+1\ddot{U}) \\ \hline \underline{\underline{(6 \ 3 \ 2)_{16}}} \quad 4 + 1_{+1\ddot{U}} = 6 \end{array}$$

Erläuterung: $3+F = 2(+1\ddot{U})$, $B+1=C$, $C+7=3(+1\ddot{U})$

Im Folgenden wird gezeigt, daß sich Subtraktion, Multiplikation und Division auf die Addition zurückführen lassen. Dazu benötigt man den Begriff des B- und des (B-1)-Komplementes.

B - Komplement und B-1 - Komplement

In einem Zahlensystem mit der Basis B bezeichnen wir als **B – Komplement** X^* zu einer gegebenen Zahl X die Ergänzung von X zur nächsten Potenz der Basis B, d.h. X^* ist Lösung der Gleichung

$$X + X^* = B^n, \text{ dabei ist } n \text{ die Stellenzahl von } X.$$

Unter dem (B-1)- Komplement \bar{X} von X verstehen wir die Lösung der Gleichung

$$X + \bar{X} = B^n - 1, \text{ dabei ist } n \text{ die Stellenzahl von } X.$$

Beachte: $X^* = \bar{X} + 1$

Beispiel 1:

$$B = 10$$

$$X = 11 \rightarrow X^* = 10^2 - 11 = 89; \text{ d.h. } 89 \text{ ist das Zehnerkomplement zu } 11$$

$$\bar{X} = 10^2 - 1 - 11 = 88; \text{ d.h. } 88 \text{ ist das Neunerkomplement zu } 11.$$

Beispiel 2:

$$B = 2$$

$$X = (11)_2 \rightarrow X^* = 2^2 - 3 = 1 = (01)_2;$$

$$\bar{X} = 0 = (00)_2.$$

Bemerkung:

Das Zweierkomplement einer Dualzahl kann so gebildet werden:

1. Schreibe alle Ziffern (von rechts gesehen) bis zur ersten dualen Eins einschließlich ab.
2. Invertiere anschließend alle Ziffern.

Besonders einfach läßt sich das Einerkomplement im Dualzahlensystem bilden. Es ist lediglich eine Zifferninversion vorzunehmen (s.u.).

Einsatz bei Subtraktion

Die folgenden Überlegungen gelten, falls bei gegebenem $a, b \in \mathbb{N}$ gilt, daß $a \geq b$. Damit ist die Differenz $a - b \geq 0$. Beide Zahlen a und b müssen mit gleicher Stellenzahl dargestellt werden, d.h. u.U sind führende Nullen zu ergänzen.

Beispiel:

Es ist im Zehnersystem die Differenz $14 - 7$ zu bilden. Diese Aufgabe wird ersetzt durch $14 - 07$. Das Ergebnis ist dann 07 .

14 ist im Dualzahlensystem 1110 . 7 ist im Dualzahlensystem 111 . Diese Folge ist um eine führende Null zu ergänzen. Also ist 7 gleich der Dualfolge 0111 .

Die Subtraktion $1110 - 0111$ wird ersetzt durch (im Zweierkomplement)

$$\begin{array}{r} 1110 \\ +1001 \end{array}$$

Das Ergebnis dieser Addition ist dann also 10111 . Wird die führende Eins gestrichen, ist das Ergebnis also 0111 . Dies entspricht der dezimalen 7 .

Einerkomplement

Bemerkung: Im Dualsystem kann wegen $x + \bar{x} = 2^n - 1$ das Einerkomplement durch Zifferninverson gebildet werden.

Beispiel:

Gegeben sei die Dualzahl 1011. Dann liefert die Invertierung 0100.

Probe 1011
 +0100
 =1111 = $[2^4 - 1]$

Zweierkomplement

Gegeben sei die Dualzahl 11110101

1. Vorgehensweise

Wie sieht das Zweierkomplement aus?

1. Betrachte Dualfolge von rechts nach links. Schreibe von rechts nach links alle Ziffern hin, bis die erste duale 1 (einschließlich) auftaucht. Danach invertiere alle Ziffern. Dies liefert im Beispiel dann 00001011

2. Vorgehensweise

1. Invertiere alle Ziffern
2. Addiere eine 1 zu der unter 1 gewonnenen Dualzahl

Beispiele für Subtraktion mit Komplementbildung

1) Es sei die Differenz $73 - 7$ (also $73-07$) zu bilden.

Das Zehnerkomplement zu 07 lautet 93.

Betrachte $73 + 93$. Dies ist 166. Streiche die führende 1. Ergebnis ist 66. Also lautet das Ergebnis für $73-07 = 66$.

Das Neunerkomplement zu 07 lautet 92.

Betrachte $73 + 92$. Das Ergebnis lautet 165. Streiche die führende 1 und addiere sie zu 65 hinzu. Das Ergebnis ist dann 66.

Betrachte $73 - 07$ im Zweiersystem. Die Aufgabe lautet also $1001001 - 0000111$.

Das Zweierkomplement zu 0000111 lautet 1111001

$$\begin{array}{r} 1001001 \\ +1111001 \\ \hline \end{array}$$

Das Ergebnis dieser Addition ist 11000010. Streiche die erste 1. Dadurch entsteht 1000010. Dies entspricht der dezimalen 66.

Das Einerkomplement zu 0000111 lautet 1111000

$$\begin{array}{r} 1001001 \\ +1111000 \\ \hline \end{array}$$

Das Ergebnis dieser Addition ist 11000010. Streiche die erste 1. Dadurch entsteht 1000010.

3.2 Grundbegriffe zweiwertiger Logik

In der deutschen Umgangssprache existiert der Begriff *Aussage*. Jede mit Worten formulierte Feststellung oder Behauptung ist eine solche Aussage ("es schneit" , "3+2 = 5"). Von solchen Sätzen lässt sich allgemein angeben, ob sie wahr oder falsch sind. Man sagt, dass der Wahrheitswert eindeutig festlegbar ist.

Definition: Ein Satz, dem eindeutig ein Wahrheitswert zugeordnet ist, heißt Aussage.

Mathematische Formulierung

A sei Aussage, $W(A)$ sei der Wahrheitswert mit

$$W(A) = \begin{cases} \text{wahr, falls A zutrifft} \\ \text{falsch, falls A nicht zutrifft} \end{cases}$$

Mehrere Aussagen lassen sich zu einer neuen Aussage zusammenfassen (Verknüpfung von Aussagen).

Beispiel: 11 ist eine Primzahl A_1
16 ist durch 4 teilbar A_2

Frage: Wie ist der Wahrheitswert der verknüpften Aussage 11 ist Primzahl und 16 ist durch 4 teilbar?

Es existieren unterschiedliche Formen der Verknüpfung zwischen Aussagen:

A_1 <u>und</u> A_2	Konjunktion
A_1 <u>oder</u> A_2	Disjunktion, einschließendes Oder
<u>entweder</u> A_1 <u>oder</u> A_2	Antivalenz, ausschliessendes (exklusives) Oder
<u>wenn</u> A_1 <u>dann</u> A_2	Implikation, Folgerung
A_1 <u>genau dann, wenn</u> A_2	Äquivalenz

Zusammengesetzte Aussagen sollten bezüglich ihres Wahrheitswertes untersucht werden. Dies geschieht dann in der Regel mit Hilfe von Wahrheitstabellen.

Beispiel: Gegeben sein die Aussagen A_1 und A_2

$$W(A_1) \begin{cases} 0, \text{ wenn } A_1 \text{ falsch ist} \\ 1, \text{ wenn } A_1 \text{ wahr ist} \end{cases}$$

$$W(A_2) \begin{cases} 0, \text{ wenn } A_2 \text{ falsch ist} \\ 1, \text{ wenn } A_2 \text{ wahr ist} \end{cases}$$

Betrachte $W(A_1)$ und $W(A_2)$

$W(A_1)$	$W(A_2)$	\wedge
0	0	0
0	1	0
1	0	0
1	1	1

} Wahrheitstabelle
Wertetabelle

Definition: Ein Zeichen (z.B. x_1, x_2), das anstelle eines Elementes aus $\{0,1\}$ steht, heißt logische Variable.

Eine Zuordnung konkreter Werte zu den Variablen heißt Belegung der Variablen.

Beispiel: x_1, x_2, x_3 seien logische Variablen. Ordnet man x_1 den Wert 1, x_2 den Wert 0, x_3 den Wert 1 zu, so heißt $(0,1,1)$ Belegung der Variablen x_1, x_2, x_3 .

Anmerkung: $x_1, x_2, x_3, x_4, \dots, x_n$ seien logische Variablen. Dann existieren 2^n Belegungen.

Folgende logische Grundverknüpfungen existieren zwischen logischen Variablen. a,b,c seien logische Variablen.

Bezeichnung	Sprechweise / Bezeichnung	Wertetabelle
-------------	---------------------------	--------------

Negation	\bar{a} (nicht a , a quer)			
		a	\bar{a}	
		0	1	
		1	0	
Konjunktion	$a \wedge b$ ($a \cdot b$)			
		a	b	$a \wedge b$
		0	0	0
		0	1	0
		1	0	0
		1	1	1
Disjunktion (ODER)	$a \vee b$ ($a + b$)			
		a	b	$a \vee b$
		0	0	0
		0	1	1
		1	0	1
		1	1	1
Antivalenz (EXOR)	$a \oplus b$ entweder a oder b			
		a	b	$a \oplus b$
		0	0	0
		0	1	1
		1	0	1
		1	1	0
Implikation (Subjunktion)	$a \rightarrow b$			
		a	b	$a \rightarrow b$
		0	0	1
		0	1	1
		1	0	0
		1	1	1
Äquivalenz	$a \leftrightarrow b$			
		a	b	$a \leftrightarrow b$
		0	0	1
		0	1	0
		1	0	0

		1	1	1
Peircefunktion (NOR)	$a \downarrow b$ (weder a noch b)			
		a	b	$a \downarrow b$
		0	0	1
		0	1	0
		1	0	0
		1	1	0
Shefferfunktion (NAND)	$a \mid b$ (nicht beide a und b)			
		a	b	$a \mid b$
		0	0	1
		0	1	1
		1	0	1
		1	1	0

Definition:

Eine Verknüpfung von endlich vielen logischen Konstanten und logischen Variablen heißt Ausdruck. Die Reihenfolge, in der die Operationen anzuwenden sind, wird durch Klammern bestimmt. Ausdrücke, die nur die Operationen ODER, UND und NICHT enthalten, heißen Boolesche Ausdrücke.

Ohne Beweis sei bereits hier bemerkt, dass jeder logische Ausdruck als boolescher Ausdruck darstellbar ist.

Logische Ausdrücke:

\bar{a} entspricht $\neg a$ } logische und
 $(a \wedge b) \vee 1$ } boolesche Ausdrücke

$((((a \vee \bar{b}) \Leftrightarrow c) \oplus a) \vee c)$ logischer, kein boolescher Ausdruck,
da \oplus und \Leftrightarrow enthalten sind

$(a \Rightarrow b) \vee (c \downarrow a)$ logischer, kein boolescher Ausdruck

$$a \Rightarrow b \oplus c$$

kein logischer Ausdruck, da die Reihenfolge
nicht eindeutig durch Klammern festgelegt ist

Um den Wahrheitswert eines logischen Ausdrucks zu bestimmen, setzt man den aktuellen Wert der Variablen in den Ausdruck ein und arbeitet ihn schrittweise ab.

Beispiel: $((a \vee b) \wedge 1) \wedge c$

a	b	c	$(a \vee b)$	$(a \vee b) \wedge 1$	$((a \vee b) \wedge 1) \wedge c$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	1	1	1

Definition:

Zwei Ausdrücke A_1 und A_2 heißen äquivalent (in Zeichen \equiv) genau dann, wenn sie bei gleicher Belegung gemeinsamer Variablen, stets den gleichen Wahrheitswert liefern. $A_1 = A_2$ heißt dann logische Gleichung.

Anmerkung: Damit zwei Ausdrücke logisch äquivalent sind, brauchen sie nicht die gleichen Variablen zu enthalten!

Beispiel 1: a, b, c seien logische Variablen.

Behauptung: $a \vee (b \wedge \bar{b}) = a \vee (c \wedge \bar{c})$

Beweis durch Wahrheitstabelle:

a	b	c	$a \vee (b \wedge \bar{b})$	$a \vee (c \wedge \bar{c})$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Anmerkung: $c \wedge \bar{c}$ genau wie $b \wedge \bar{b}$ immer falsch (d.h. 0).

Beispiel 2: a, b, c seien logische Variablen.

Behauptung: $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

Beweis durch Wahrheitstabelle:

a	b	c	$a \wedge (b \vee c)$	$(a \wedge b) \vee (a \wedge c)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

3.3 Rechengesetze für logische Ausdrücke

Satz:

Seien a, b, c logische Variable, dann gelten folgende Regeln:

Name	Vorgang	Wichtigkeit
Negation der Negation	$\overline{\overline{a}} = a$	
Kommutativ-Gesetz	$a \wedge b = b \wedge a$ $a \vee b = b \vee a$	+ +
Assoziativ-Gesetz	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$ $a \vee (b \vee c) = (a \vee b) \vee c$	+ +
Distributiv-Gesetz	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	+ +
Idempotenz-Gesetz	$a \wedge a = a$ $a \vee a = a$	
Komplement-Gesetze	$a \wedge \overline{a} = 0$ $a \vee \overline{a} = 1$	
0-1-Gesetze	$a \wedge 1 = a$ $a \vee 1 = 1$ $a \wedge 0 = 0$ $a \vee 0 = a$	
Absorptions-Gesetze	$a \wedge (a \vee b) = a$ $a \vee (a \wedge b) = a$ $(a \wedge b) \vee (a \wedge \overline{b}) = a$ $(a \vee b) \wedge (a \vee \overline{b}) = a$ $(a \wedge \overline{b}) \vee b = a \vee b$ $(a \vee \overline{b}) \wedge b = a \wedge b$	+++ +++ +++++ +++++ +++ +++
De Morgan	$\overline{(a \wedge b)} = \overline{a} \vee \overline{b}$ $\overline{(a \vee b)} = \overline{a} \wedge \overline{b}$	+++ +++

Beweise:

Es folgen Beweise von zwei Aussagen des Satzes mittels Wertetabelle:

$$1.) a \wedge \bar{a} = 0$$

a	\bar{a}	$a \wedge \bar{a}$
0	1	0
1	0	0

$$2.) (a \wedge b) \vee (a \wedge \bar{b}) = a$$

a	b	\bar{b}	$a \wedge b$	$a \wedge \bar{b}$	$(a \wedge b) \vee (a \wedge \bar{b})$	a
0	0	1	0	0	0	0
0	1	0	0	0	0	0
1	0	1	0	1	1	1
1	1	0	1	0	1	1

Anwendungen des Satzes zur Vereinfachung von logischen Ausdrücken:**Beispiel:**

$$(\bar{a} \vee b) \overline{ab} \left((\bar{a} \vee b) (\bar{a} \vee \bar{b}) \vee (ca \vee \bar{c}) \right)$$

Der Ausdruck in ausführlicher Schreibweise:

$$(\bar{a} \vee b) \wedge (\overline{a \wedge b}) \wedge \left((\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \vee ((c \wedge a) \vee \bar{c}) \right)$$

auf $(\overline{a \wedge b})$ das De Morgansche Gesetz angewandt

$$(\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \wedge ((\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \vee ((c \wedge a) \vee \bar{c}))$$

auf $(\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b})$ das Absorptions-Gesetz angewandt

$$\bar{a} \wedge ((\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \vee ((c \wedge a) \vee \bar{c}))$$

auf $(\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b})$ das Absorptions-Gesetz angewandt

$$\bar{a} \wedge (\bar{a} \vee ((c \wedge a) \vee \bar{c}))$$

Nebenrechnung:

$$((c \wedge a) \vee \bar{c}) \quad \text{Nach dem Kommutativ-Gesetz umgewandelt:}$$

$$\bar{c} \vee (c \wedge a) \quad \text{Nach dem Distributiv-Gesetz umgewandelt:}$$

$$(\bar{c} \vee c) \wedge (\bar{c} \vee a) \quad \text{Nach dem Komplement-Gesetz:}$$

$$1 \wedge (\bar{c} \vee a) \quad \text{Nach dem 0-1-Gesetz:}$$

$$(a \vee \bar{c}) \quad \text{Nach dem Kommutativ-Gesetz}$$

Also ist $((c \wedge a) \vee \bar{c}) = (a \vee \bar{c})$; das oben eingesetzt ergibt:

$$\bar{a} \wedge (\bar{a} \vee (a \vee \bar{c}))$$

auf $\bar{a} \vee (a \vee \bar{c})$ das Assoziativ-Gesetz angewandt

$$\bar{a} \wedge ((\bar{a} \vee a) \vee \bar{c})$$

nach dem Komplement-Gesetz ist $(\bar{a} \vee a) = 1$

$$\bar{a} \wedge (1 \vee \bar{c})$$

nach dem 0-1-Gesetz ist $1 \vee \bar{c} = 1$

$$\bar{a} \wedge 1$$

nach dem 0-1-Gesetz ist $\bar{a} \wedge 1 = \bar{a}$

$$\bar{a}$$

Also ist $(\bar{a} \vee b) \wedge (\overline{a \wedge b}) \wedge ((\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) \vee ((c \wedge a) \vee \bar{c})) = \bar{a}$

Shannonscher Inversionssatz (Verallgemeinerung von de Morgan)

$$\overline{a_1 \wedge a_2 \wedge \dots \wedge a_n} = \bar{a}_1 \vee \bar{a}_2 \vee \dots \vee \bar{a}_n$$

$$\overline{a_1 \vee a_2 \vee \dots \vee a_n} = \bar{a}_1 \wedge \bar{a}_2 \wedge \dots \wedge \bar{a}_n$$

Beweis erfolgt über vollständige Induktion (Vgl. Übungsaufgaben)

3.4 Logische Funktionen

Vorbemerkung:

Sei A eine Menge und n eine natürliche Zahl. Unter dem n -fachen kartesischen Produkt A^n verstehen wir die Menge $\{(a_1, a_2, \dots, a_n) \mid a_1 \in A, a_2 \in A, \dots, a_n \in A\}$, also die Menge aller n -Tupel, deren Komponenten aus Elementen der Menge A bestehen. Besitzt die Menge A die Elementanzahl m , so besitzt A^n die Elementanzahl m^n . So ist etwa $\{0,1\}^3$ die 8-elementige Menge $(8 = 2^3) \{(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)\}$.

Definition :

Es sei $D \subseteq \{0,1\}^n$.

Ferner sei $(a_1, a_2, \dots, a_n) \in D$ eine Belegung. Dann heißt die Zuordnungsvorschrift

$$f : D \rightarrow \{0,1\}, (a_1, a_2, \dots, a_n) \mapsto f(a_1, a_2, \dots, a_n)$$

n -stellige logische Funktion (mit dem Definitionsbereich D).

a_1, a_2, \dots, a_n heißen auch die unabhängigen Variablen.

Ist $D = \{0,1\}^n$, so heißt f vollständig definiert.

Ist $D \subset \{0,1\}^n$ (echte Teilmenge), so heißt f partiell definiert.

Frage: Wie können logische Funktionen dargestellt werden?

Betrachten wir dazu ein Beispiel einer reellen Funktion: $f : \mathbb{R} \setminus \{2\} \rightarrow \mathbb{R}; x \mapsto \frac{x+2}{x-2}$. Neben

der Darstellung als Funktionsgleichung kennen wir hier die Darstellung mittels Wertetabelle sowie die graphische Darstellung.

Analog ist die Darstellung einer logischen Funktion möglich über

- eine Wertetabelle
- eine graphische Darstellung oder
- einen algebraischen bzw. logischen Ausdruck.

Beispiel einer logischen Funktion

An einem Haus seien 3 Alarmsensoren befestigt. Ein Alarm ist auszulösen, wenn mindestens 2 der 3 Sensoren aktiv sind.

$$\Rightarrow f : D \subset \{0,1\}^n \rightarrow \{0,1\}$$

a_1 : Sensor 1 Zustand $\in \{0,1\}$

a_2 : Sensor 2 Zustand $\in \{0,1\}$

a_3 : Sensor 3 Zustand $\in \{0,1\}$

Da es drei Sensoren gibt, kommen wir mit einer 3-stelligen logischen Funktion (besitzt 3 unabhängige Variablen) aus.

$$\text{Also } f : D \subset \{0,1\}^3 \rightarrow \{0,1\}.$$

Wertetabelle dieser Funktion gemäß der Problemstellung:

a_1	a_2	a_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Graphische Darstellung logischer Funktionen mittels KV-Diagramm

Definition

Jedes KV-Diagramm ist ein quadratisches Schema (Matrix) von Wahrheitswerten. Jeder Eintrag der Matrix stellt einen Funktionswert für eine bestimmte Belegung der unabhängigen Variablen der Funktion dar. Die Elemente der Matrix beinhalten also die Bildelemente. An den Rändern der Matrix stehen die möglichen Belegungen der unabhängigen Variablen.

Beispiel:

		a,b			
		00	01	11	10
c	0	1			
	1			1	1

KV- Diagramm bei 4 Variablen

		a,b			
		00	01	11	10
c,d	00	$f(0,0,0,0)$	$f(0,1,0,0)$	$f(1,1,0,0)$	$f(1,0,0,0)$
	01	$f(0,0,0,1)$	$f(0,1,0,1)$	$f(1,1,0,1)$	$f(1,0,0,1)$
	11	$f(0,0,1,1)$	$f(0,1,1,1)$	$f(1,1,1,1)$	$f(1,0,1,1)$
	10	$f(0,0,1,0)$	$f(0,1,1,0)$	$f(1,1,1,0)$	$f(1,0,1,0)$

Beispiel:

		a,b			
		00	01	11	10
c,d	00	1			1
	01			1	
	11		1	1	1
	10	1			1

3.5 Logische Normalformen

Gegeben sei eine vollständig definierte logische Funktion bzw. ein logischer Ausdruck. Man sucht einen logisch äquivalenten Ausdruck dessen Darstellung standardisiert ist (logische Normalform). Zu einem logischen Ausdruck gehören immer mehrere äquivalente Normalformen. Unter diesen Normalformen gibt es allerdings eine Darstellung, die für Algorithmen interessant ist. Diese nennt man kanonisch disjunktive Normalform (KDN).

Definition: Eine Konjunktion $x_1 \wedge x_2 \wedge \dots \wedge x_n$ heißt n - stelliger Konjunktionsterm, wenn alle x_i , $i \in \{1, \dots, n\}$ entweder logische Variablen oder logisch negierte Variablen sind, und jede Variable nur einmal vorkommt.

Eine Disjunktion $x_1 \vee x_2 \vee \dots \vee x_n$ heißt n – stelliger Disjunktionsterm.

Beispiel:

$$f_1(a, b, c) = a \wedge b \wedge c$$
$$f_2(a, b, c) = a \vee b \vee c$$

Definition: Ein logischer Ausdruck der Form $K_1 \vee K_2 \vee \dots \vee K_k$, $k \in \mathbb{N}$ heißt disjunktive Normalform (DN), wenn alle K_i ($i \in \{1, \dots, k\}$) paarweise verschiedene Konjunktionsterme sind. Gilt $f(a_1, \dots, a_n) = K_1 \vee K_2 \vee \dots \vee K_k$ so heißt die DN auch disjunktive Normalform der Funktion f.

Bemerkung: Eine disjunktive Normalform für eine gegebene vollständige Funktion f ist im allgemeinen nicht eindeutig, d.h. es können in der Regel mehrer DN existieren, die logisch äquivalent sind.

Ohne Beweis gelten folgende Aussagen

Gegeben sei ein beliebiger logischer Ausdruck z.B. $(a \leftrightarrow b) \oplus c$

1. Zu diesem logischen Ausdruck kann ein äquivalenter boolescher Ausdruck konstruiert werden (vgl. Übung).
2. Zu einem booleschen Ausdruck existieren in der Regel mehrere logische äquivalente DNs.

Da für eine Funktion mehrere logische äquivalente DN existieren können, versucht man unter den DN eine zu finden, die immer existiert und die nur einmal existiert. Dies ist die sogenannte kanonisch disjunktive Normalform (KDN).

Definition: Eine DN zu einer vollständig definierten n – stelligen logischen Funktion heißt kanonisch (bzw. ausgezeichnet), wenn jeder der vorhandenen Konjunktionsterme alle n Variablen enthält und keine zwei Konjunktionsterme logisch äquivalent sind. Man bezeichnet sie auch als KDN.

Anmerkung: Zu einer beliebigen vollständig definierten logischen Funktion existiert immer nur eine KDN. Es existieren aber in der Regel mehrere logische äquivalente DN zu der Funktion f.

Frage: Gegeben sei eine DN. Wie kann man auf die zugehörige KDN schließen ?

Algorithmus:

1. Jeder Konjunktionsterm, der nicht alle n Variablen enthält, wird konjunktiv mit dem Term $(a_j \vee \bar{a}_j)$ verknüpft, wobei j die Variable ist, die in dem Konjunktionsterm nicht enthalten ist.
2. Auf den neuen Konjunktionsterm wird das Distributivgesetz angewendet.
3. Mehrfach auftretende gleiche Konjunktionsterme werden nach dem Idempotenzgesetz $(a \vee a = a)$ zusammengefasst.

Beispiel:

$$1. \quad f(a,b,c) = (a \wedge b)$$

$$a \wedge b = (a \wedge b)(c \vee \bar{c}) = (a \wedge b \wedge c) \vee (a \wedge b \wedge \bar{c})$$

$$2. \quad g(a,b,c) = (a \vee b)$$

$$\begin{aligned} (a \vee b) &= a(b \vee \bar{b}) \vee b(a \vee \bar{a}) \\ &= ab \vee a\bar{b} \vee ab \vee \bar{a}b \\ &= ab \vee a\bar{b} \vee \bar{a}b \end{aligned}$$

$$\begin{aligned}
&= ab(c \vee \bar{c}) \vee a\bar{b}(c \vee \bar{c}) \vee \bar{a}b(c \vee \bar{c}) \\
&= abc \vee ab\bar{c} \vee a\bar{b}c \vee a\bar{b}\bar{c} \vee \bar{a}bc \vee \bar{a}b\bar{c}
\end{aligned}$$

Bemerkung:

Die Konjunktionsterme einer KDN werden auch *Minterme* oder *Vollkonjunktionen* genannt.

Frage: Wie gewinnt man eine KDN aus einer Wertetabelle ?

Algorithmus:

1. Zu jeder Belegung, die den Funktionswert 1 hat, wird der zugehörige Minterm entwickelt, in dem die Variablen, die in dieser Belegung 0 sind, als negierte Variablen und die 1 sind, als einfache (nicht negierte) Variable konjunktiv verknüpft werden.
2. Die Disjunktion der Minterme ist die gesuchte KDN.

Beispiel:

<i>a</i>	<i>b</i>	<i>c</i>	<i>f(a,b,c)</i>	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	← = $\bar{a} \wedge b \wedge c$
1	0	0	0	
1	0	1	1	← = $a \wedge \bar{b} \wedge c$
1	1	0	1	← = $a \wedge b \wedge \bar{c}$
1	1	1	1	← = $a \wedge b \wedge c$

=> **KDN lautet :** $(\bar{a}bc) \vee (a\bar{b}c) \vee (ab\bar{c}) \vee (abc)$

Darstellung der KDN als KV – Diagramm

		<i>ab</i>			
		00	01	11	10
0				1	
1		1	1	1	

Beispiel:

In einem Computer seien jeweils 3 Bits zu speichern. Ein viertes Bits ist so zu setzen, daß die Anzahl der Einsen in den vier Bits immer gerade ist.

<i>a</i>	<i>b</i>	<i>c</i>	<i>f(a,b,c)</i>	
0	0	0	0	
0	0	1	1	← = $\bar{a} \wedge \bar{b} \wedge c$
0	1	0	1	← = $\bar{a} \wedge b \wedge \bar{c}$
0	1	1	0	
1	0	0	1	← = $a \wedge \bar{b} \wedge \bar{c}$
1	0	1	0	
1	1	0	0	
1	1	1	1	← = $a \wedge b \wedge c$

		<i>ab</i>			
		00	01	11	10
0			1		1
1	1			1	

KDN lautet : $(\bar{a}\bar{b}c) \vee (\bar{a}b\bar{c}) \vee (a\bar{b}\bar{c}) \vee (abc)$

Konjunktive Normalformen

Definition : Ein logischer Ausdruck der Form $D_1 \wedge D_2 \wedge \dots \wedge D_k$, $k \in \mathbb{N}$ heißt *konjunktive Normalform* wenn alle $D_i : i \in \{1, \dots, k\}$ paarweise verschiedene Disjunktionsterme sind.

Definition : Eine KN einer vollständig definierten Funktion heißt *kanonisch*, wenn jeder der auftretenden Disjunktionsterme alle n Variablen enthält und keine 2 Disjunktionsterme äquivalent sind. (Kurzbezeichnung KKN)

Anmerkung : Jede KN kann zu einer KKN erweitert werden.

Algorithmus : Gegeben sei die KN einer n-stelligen Funktion.

1. Jeder Disjunktionsterm, der nicht n-stellig ist, wird disjunktiv mit dem Term $(a_j \wedge \bar{a}_j)$ verknüpft, wobei j die Variable ist, die im Disjunktionsterm nicht enthalten ist.
2. Anwendung des Distributivgesetzes
3. Mehrfach auftretende Terme werden nach dem Idempotenzgesetz zusammengefasst.

Definition : Ein n-stelliger Disjunktionsterm über n Variablen a_1, \dots, a_n heißt n-stelliger *Maxterm oder Volldisjunktion*.

Aufstellung einer KKN aus der Wertetabelle

1. Zu jeder Belegung, die den Funktionswert „0“ hat, wird der Maxterm ermittelt, in dem die Variablen die entsprechend der Belegung „1“ sind, als negierte Variablen und die, die „0“ sind, als einfache Variablen disjunktiv verknüpft werden.
2. Die Konjunktion der Maxterme ist die KKN der Funktion.

Beispiel :

a	b	c	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Ermittlung KKN : $(a \vee \bar{b} \vee c) \wedge (\bar{a} \vee b \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$

Ermittlung KDN : $(\bar{a}\bar{b}\bar{c}) \vee (\bar{a}\bar{b}c) \vee (\bar{a}b\bar{c}) \vee (a\bar{b}c) \vee (abc)$

3.6 KV-Diagramm und Minimalformen

Bei technischen Problemstellungen, die mit der booleschen Algebra bearbeitet werden sollen, beginnt man in der Regel mit einer Wertetabelle. Daraus wird die KDN (KKN) konstruiert. Die KDN (KKN) hat „unangenehme“ Eigenschaften. Sie ist meist sehr groß. Deswegen versucht man kürzere äquivalente Darstellungen zu finden. Man versucht die zugehörige minimale DN zu konstruieren. Die folgende Vorgehensweise findet in fast allen Fällen die minimale DN. Wir nennen das Verfahren „Ermittlung einer minimalen DN mittels KV-Diagramm“. Exakt gelingt dies mit dem so genannten Quine - Mc Clusky – Verfahren.

Regeln zu Ermittlung einer minimalen DN mittels KV-Diagramm

Voraussetzung : Logische Funktion sei als 1-Muster im KV-Diagramm gegeben. Damit ist die KDN bekannt.

Schritt 1 : Man kennzeichnet nacheinander folgende 2^i - Gruppe von Einsen ($i = 0,1,2,\dots$)

- a) Jede 1-Gruppe (Einzelfeld), die zu keiner 2-Gruppe gehört
- b) Jede 2-Gruppe, die nicht in einer 4-Gruppe enthalten ist
- c) Jede 2^i -Gruppe die nicht in einer 2^{i+1} - Gruppe enthalten ist ($i = 0,1,2,\dots$)

Die Gruppen sind so zu bilden, daß möglichst viele noch nicht berücksichtigte Einsen erfaßt sind, mindestens aber eine neue 1 hinzukommt. Mann verwendet verschiedenfarbige Markierungen.

Zusatz bei partiell definierten Funktionen. Die nicht definierten Felder (don't care – Felder) werden wie „1“ behandelt, wenn sich dadurch eine 2^i - Gruppe vergrößern läßt. Die übrigen „d“- Felder werden wie „0“ behandelt.

Schritt 2 : Unter den gekennzeichneten Gruppen werden solche ausgesondert, die von den anderen *völlig überdeckt* werden.

Schritt 3 : Zu den verbleibenden Gruppen bildet man die zugehörigen Konjunktionsterme. Die disjunktive Verknüpfung liefert in der Regel die minimale DN.

Beispiele :

$$n = 2$$

	a	
	0	1
b	0	1
1	1	1

$$\text{KDN} : (\bar{a} \wedge \bar{b}) \vee (a \wedge b)$$

Hier ist keine Minimierung über das KV-Diagramm möglich

	a	
	0	1
b	0	1
1	1	1

$$\text{KDN} : (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b}) = \bar{b}$$

In obigem Beispiel ist ein Zweierblock zu erkennen

	a	
	0	1
b	0	1
1	1	1

$$\text{KDN} : (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge b) \vee (a \wedge b) = \bar{a} \vee b$$

Hier werden zwei Zweierblöcke gebildet.

Beispiele :

$n = 3$

		a,b			
		00	01	11	10
c	0	1		1	
	1		1		1

- es lässt sich keine Vereinfachung
im KV-Diagramm angeben

		a,b			
		00	01	11	10
c	0	1			1
	1				

minimale DN : $\bar{b} \bar{c}$

Hier wird ein Zweier-Block "über Eck" gebildet

		a,b			
		00	01	11	10
c	0		1	1	
	1		1	1	1

minimale DN : $b \vee ac$

Hier wird ein Viererblock und ein
Zweierblock konstruiert

		a,b			
		00	01	11	10
c	0	1			1
	1	1			1

minimale DN : \bar{b}

Die vier vorhandenen Einsen werden zu einem
Viererblock zusammengesetzt

		a,b		
	00	01	11	10
c	0	1	1	
	1		1	1

		a,b		
	00	01	11	10
c	0		1	1
	1	1	1	1

minimale DN :

$$(\bar{a} \wedge \bar{c}) \vee (\bar{a} \wedge b) \vee (a \wedge c)$$

Hier werden drei Zweierblöcke gebildet.

minimale DN : $c \vee b$

In diesem Beispiel sind zwei Viererblöcke zusammengefaßt

Beispiele :

$$n = 4$$

		a,b		
	00	01	11	10
c,d	00			
	01			
	11			
	10	1		1

		a,b		
	00	01	11	10
c,d	00	1		1
	01			
	11			
	10	1		1

minimale DN : $\bar{b}c\bar{d}$

minimale DN : $\bar{b}\bar{d}$

In den beiden obigen Beispielen sieht man, daß die Nachbarschaftseigenschaft des Absorptionsgesetzes auch "über Rand" gilt. Im ersten Beispiel wird ein 2-er Block gebildet, im zweiten Beispiel über Eck ein Viererblock konstruiert.

		a,b			
		00	01	11	10
c,d	00		1	1	
	01	1	1	1	1
	11				
	10				

minimale DN : $b\bar{c} \vee \bar{c}d$

In dem Beispiel werden zwei Vierer-Blöcke gebildet.

3.7 Technische Anwendungen und Boolesche Algebra

Häufig hat bei technischen Problemstellungen in der Hardwareentwicklung folgende Vorgehensweise, um Schaltungen zu realisieren.

1. Technische Problemstellung ist als Text gegeben
2. Ableitung der Wertetabelle aus diesem Text
3. Ermittlung der KDN aus Wertetabelle
4. Übertragung in ein KV-Diagramm
5. Minimierung der eingetragenen KDN
6. Realisierung der Schaltung.

Es ist festzuhalten, daß diese Vorgehensweise nicht unbedingt die kürzeste Darstellung einer Funktion realisiert. Es wird versucht, eine möglichst kurze DN zu realisieren. In Beispielen wurde bereits mehrfach in der Vorlesung gezeigt, daß zu einer gegebenen logischen Funktion f die zugehörige minimale DN nicht optimal ("minimale Länge") sein muß, d.h. es existieren u.U. kürzere Darstellungen zu f , die keine DN sind. Das folgende Beispiel 1 zeigt dies noch mal.

Problem 1: Es seien 4 Bits (Halbwort) in einem Computer zu speichern. Ein fünftes Bit (Parity – Bit) ist so zu setzen, dass die Anzahl der Einsen in dem 5 Bit gerade ist. Es sei eine kanonisch disjunktive Normalform zu entwickeln. Diese soll mittels KV-Diagramm minimiert werden.

a	b	c	d	$f(a,b,c,d)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

	1		1
1		1	
	1		1
1		1	

Keine Minimierung über KV – Diagramm möglich

$$f(a,b,c,d) = (\bar{a}\bar{b}\bar{c}d) \vee (\bar{a}\bar{b}c\bar{d}) \vee (\bar{a}b\bar{c}\bar{d}) \vee (\bar{a}bcd) \vee (a\bar{b}\bar{c}\bar{d}) \vee (a\bar{b}cd) \vee (ab\bar{c}d) \vee (abcd)$$

Es ist aber eine Minimierung über die Rechengesetze (Kapitel 3.3) möglich. Beispielsweise kann man a, b in den ersten drei Termen ausklammern etc. Führt man dies entsprechend durch, so erhält man einen „kurzen“ logischen Ausdruck, der allerdings keine disjunktive Normalform ist.

Problem 2: Mit einer 4-Bit – Dualzahl lassen sich die Dualzahlen 0000 bis 1111 darstellen. Will man eine der Dezimalziffern von 0 bis 9 darstellen (dies heißt auch BCD – Darstellung), sind die Dualzahlen 1010 bis 1111 “überflüssig“. Man nennt diese daher Pseudotetraden. Es ist eine Funktion anzugeben, die die Pseudotetraden erkennt. Mittels KV-Diagramm sei eine minimale DN zu entwickeln.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	$f(a,b,c,d)$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

		1	
		1	
		1	1
		1	1

$$f(a,b,c,d) = ab \vee ac$$

Weiteres algebraisches Rechnen liefert $a \wedge (b \vee c)$.

Problem 3: (Halbaddierer; Volladdierer): Es seien zwei einstellige Dualzahlen zu addieren. Wie sieht die entsprechende logische Funktion aus?

a	b	s_1	\ddot{u}_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\ddot{u}_1(a,b) = a \wedge b$$

$$s_1(a,b) = (\bar{a} \wedge b) \vee (a \wedge \bar{b})$$

Sind zwei n-stellige Dualzahlen zu addieren, muß der Übertrag aus der vorhergehenden Stellenzahl berücksichtigt werden. Das heißt bei dieser Addition sind immer drei Dualzahlen (3 Eingangsvariable) zu addieren.

Volladdierer

0	1	0	1	
1	1	1	1	
1	1	1	1	← Übertrag
1	0	1	0	0

a	b	c	s_2	\ddot{u}_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

 s_2

	1		1
1		1	

 \ddot{u}_2

		1	
	1	1	1

Hier sei keine Minimierung betrachtet, sondern lediglich der Zusammenhang zwischen Halb- und Volladdierer betrachtet.

$$s_2 = (\bar{a}bc) \vee (\bar{a}b\bar{c}) \vee (a\bar{b}\bar{c}) \vee (abc) =$$

$$\text{Schritt 1.)} = c(ab \vee \bar{a}\bar{b}) \vee \bar{c}(\bar{a}b \vee \bar{a}\bar{b})$$

$$\text{Schritt 2.)} = c(\overline{\bar{a}b}) \vee (\bar{a} \wedge b) \vee \bar{c}((\bar{a}b) \vee (\bar{a}\bar{b}))$$

$$\text{Schritt 3.)} = c \wedge \bar{s}_1 \vee \bar{c} \wedge s_1$$

Nebenrechnung von Schritt 1.) zu Schritt 2.)

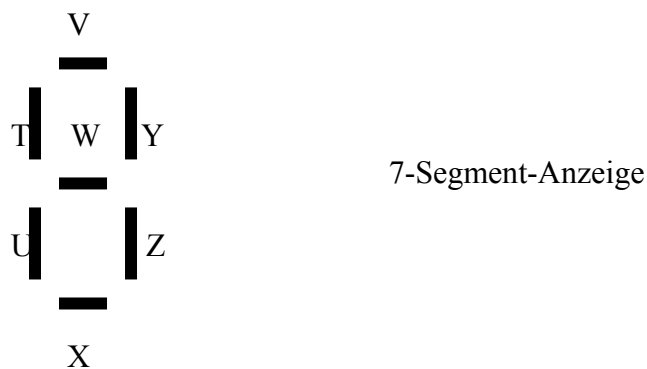
$$\overline{(\bar{a}b) \vee (\bar{a} \wedge b)} = (\bar{a} \vee b) \wedge (a \vee \bar{b})$$

$$= (\bar{a} \wedge a) \vee (b \wedge a) \vee (\bar{a} \wedge \bar{b}) \vee (b \wedge \bar{b})$$

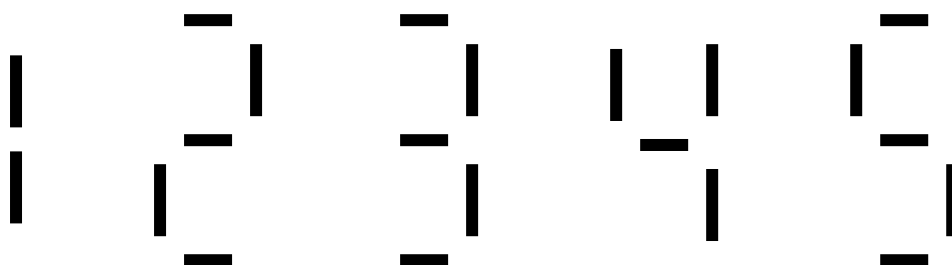
$$= 0 \vee (a \wedge b) \vee (\bar{a} \wedge \bar{b}) \vee 0$$

$$= (ab) \vee (\bar{a}\bar{b})$$

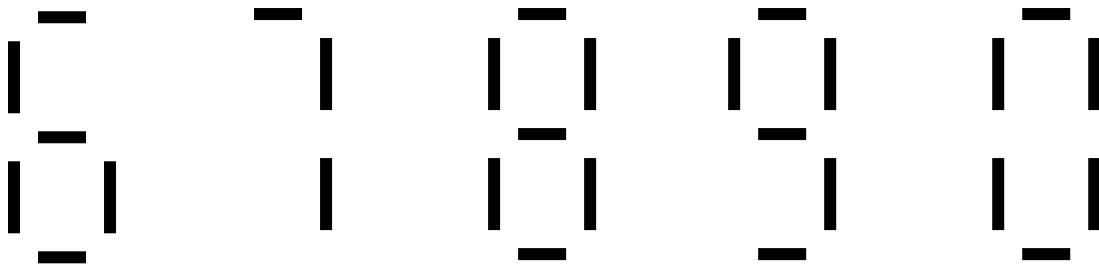
Problem 4: Zur Anzeige der Dezimalziffern 0 bis 9 soll eine Siebensegment-Anzeige verwendet werden. In der nachstehenden Graphik ist eine solche Segmentanzeige dargestellt. Die sieben Segmente seien mit T,U,V,W,X,Y,Z bezeichnet. Es sind Schaltfunktionen zur Darstellung der Dezimalziffern anzugeben.



Darstellung der Ziffern von 1 bis 5:



Darstellung der Ziffern von 6-9 und der Ziffer 0:



Die folgende Wertetabelle beschreibt die Funktionen: Beachten Sie, daß die Variablen mit d, c, b, a , nicht mit a, b, c, d bezeichnet sind. Die Dezimalziffern 0 bis 9 sind also mit den Variablen d, c, a, b dual dargestellt.

d	c	b	a	Z	Y	X	W	V	U	T
0	0	0	0	1	1	1	0	1	1	1
0	0	0	1	1	1	0	0	0	0	0
0	0	1	0	0	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	0	0
0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	1	0	1	1	1	0	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	0	0	1	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	0	1

1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

} nicht definiert

d, c

1		d	1
		d	
		d	d
1	1	d	d

Darstellung von U

$$0010 \vee 0110 \vee 1110 \vee 1010 = b\bar{a}$$

$$0000 \vee 1000 \vee 0010 \vee 1010 = \bar{a}\bar{c}$$

$$= b\bar{a} \vee \bar{c}\bar{a} = \bar{a}(b \vee \bar{c})$$



Boolescher Ausdruck

Bei einer partiell definierten Funktion sind die d's so zu schalten (d.h. auf 0 oder 1 zusetzen), dass größtmögliche Blöcke entstehen.

Darstellung von Z im KV-Diagramm:

		d,c	
1	1	d	1
1	1	d	1
1	1	d	d
	1	d	d

= c
 = \bar{b}
 = a

$$Z = a \vee \bar{b} \vee c$$

3.8 Binäre Codes

Ein Code ist die Zuordnung von zwei Mengen von Zeichen. Ein Beispiel für einen Code ist das Morse-Alphabet, bei dem die Buchstaben des Alphabetes bestimmten Punkt-Strich-Kombination zugeordnet sind. Beim Morsealphabet ist der Elementarvorrat an Zeichen die Menge $\{., -\}$.

Eine Möglichkeit, Dezimalzahlen binär darzustellen, besteht darin, dass jeder Dezimalziffer einer Zahl ein Binärwert (z.B. 4Bits) zugeordnet wird.

$$\text{z.B. } (4711)_{10} = \underbrace{0100}_4 \underbrace{0111}_7 \underbrace{0001}_1 \underbrace{0001}_1$$

Man spricht von binär codierter Darstellung bzw. von BCD-Codes (Binary Coded Decimal) oder dekadischen Codes. Eine andere Möglichkeit, Dezimalzahlen zu codieren, wäre die Umwandlung einer Dezimalzahl in die zugehörige Dualzahl (Dualcode).

$$(4711)_{10} = (1001001100111)_2$$

Die Auswahl eines Computercodes hängt von einer Vielzahl von Anforderungen ab. Die wichtigsten Anforderungen sind:

- a. Verwendung einer geringen Stellenzahl
- b. fester Stellenwert. Man spricht auch von dem Gewicht der Binärstelle. Der BCD-Code ist ein 8-4-2-1-Code.
- c. Vermeidung von Null/Eins-Worten. Die Kombinationen 0000 und 1111 können durch Störungen entstehen.
- d. Einfache Bildung des Neunerkomplements, um die Subtraktion auf eine Addition zurückführen zu können.
- e. Lexikographische Anordnung. Für einen lexikographisch aufgebauten Code lassen sich i.a. einfache Zähler aufbauen. Beispielsweise sind folgende Worte lexikographisch geordnet: 000, 001, 010, 011, 100.

BCD-Code (8-4-2-1-Code)

Vorteile	Nachteile
a. Fester Stellenwert	a. Nullwort ist vorhanden
b. Lexikographische Anordnung	b. 9er-Komplement-Bildung ist schwierig
c. Einprägsamkeit	

Exzeß-3-Code (Stibitz-Code)

Exzeß-3 bedeutet "beginnend mit 3"

Dezimalzahl	2^3	2^2	-2^1	-2^0
0	0	0	1	1
1	0	1	0	0
2	0	1	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Pseudotetraden sind die Worte

0000, 0001, 0010, 1101, 1110, 1111

Vorteile

- Null/Eins-Wort wird vermieden
- 9er Komplement-Bildung lässt sich einfach bilden
- lexikographische Anordnung
- leichte Einprägsamkeit

Die Schreibweise -2^1 und -2^0 bedeutet, daß die Gewichte 2^1 und 2^0 mit dem negierten Wert in der entsprechenden Spalte zu multiplizieren und zu subtrahieren sind. So ergibt sich die dezimale 7 aus

$$1 * 2^3 + 0 * 2^2 + 0 * (-2^1) + 1 * (-2^0) = 8 - 1 = 7.$$

Vorteile:

- fester Stellenwert
- Fehlen von Null / Einswort
- einfache Komplementbildung
- lexikographische Anordnung.

Gray-Code:

Dezimalzahl	Gray-Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101

Bemerkenswert am Gray-Code ist die Tatsache, daß sich benachbarte Codewörter nur in einer Stelle unterscheiden. Der Gray-Code findet u.a. bei KV-Diagrammen Verwendung, um Belegungen für Variablen zu symbolisieren.

Codierer / Decodierer wandeln Dualwerte von einem Code in einen anderen Code um.

Dezimalcode	BCD-Code	Gray-Code
	D C B A	W X Y Z
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
9	1 0 0 1	1 1 0 1

$W(d,c,b,a) \rightarrow W = d$
 00 01 11 10

		D	1
		D	1
		D	D
		D	D

$$X(d,c,b,a) \rightarrow X = d \vee c$$

	00	01	11	10
		1	D	1
		1	D	1
		1	D	D
		1	D	D

$$Y(d,c,b,a) \rightarrow Y = \bar{b}c \vee b\bar{c}$$

	00	01	11	10
		1	D	
		1	D	
1			D	D
1			D	D

$$Z(d,c,b,a) \rightarrow Z = \bar{b}a \vee b\bar{a}$$

	00	01	11	10
			D	
1		1	D	1
			D	D
1		1	D	D

3.9 Fazit

Im dritten Teil der Vorlesung Informatik I haben wir uns mit der Menge $\{0,1\}$ und Verknüpfungen auf dieser Menge beschäftigt. Vergleichbar ist dies mit der Einführung der reellen Zahlen in der Vorlesung Analysis. Zunächst wird in Analysis die Menge der reellen Zahlen eingeführt (in der Logik die Menge $\{0,1\}$). Danach erfolgt die Einführung der mathematischen Operationen wie $+$, $-$, $*$, $/$. In der Logik wurden mathematische Verknüpfungsoperatoren wie $\wedge, \vee, \bar{}, \oplus, \rightarrow, \leftrightarrow$ eingeführt. Beim Rechnen mit reellen Zahlen werden dann zur Vereinfachung von arithmetischen Ausdrücken Rechengesetze wie das Kommutativ-, Assoziativ- und Distributivgesetz eingeführt. Dies geschah auch in diesem Teil der Vorlesung. Im reellen Bereich wird dann der Begriff der Abbildung bzw. reellwertigen Funktion eingeführt. Jede Abbildung läßt sich als Wertetabelle(prinzipiell), Graph oder als "algebraischer" Ausdruck darstellen. Diese Darstellung kennt man auch bei logischen Funktionen. Ein Unterschied bei reellwertigen Funktionen gegenüber den logischen Funktionen soll besonders herausgestellt werden. Definitions- und Bildmenge sind bei den reellwertigen Funktionen üblicherweise unendliche Mengen, bei den logischen Funktionen ist die Definitionsmenge eine endliche Menge (das Kreuzprodukt von n Mengen), die Bildmenge ist die Menge $\{0,1\}$.

In der reellen Analysis sind dann die Begriffe Stetigkeit und Differenzierbarkeit von Interesse, in der Logik ist die "kürzeste" Darstellung einer Funktion von Interesse, um sie schaltungsmäßig zu realisieren.

Im Fazit soll auch auf eine Verbindung zum zweiten Teil der Vorlesung (Einführung in die Algorithmentheorie) aufmerksam gemacht werden. Algorithmen spielen in der Logik eine zentrale Rolle. Im einzelnen sind dies Algorithmen zur Zahlenumwandlung vom Dezimal- in das Binär- und Headezimalzahlensystem. Weitere Algorithmen finden bei der Arithmetik auf Binärzahlen Verwendung. Als Beispiel sei hier die Komplementbildung genannt. Alle genannten Problemstellungen zählen zu der Komplexitätsklasse P.

Um die minimale disjunktive Normalform einer Funktion zu finden, gehen wir von der kanonisch disjunktiven Normalform aus. Es kann gezeigt werden, daß die geschilderte Problemstellung zur Klasse der NP-vollständigen Probleme gehört.

Will man die kürzeste Darstellung einer gegebenen logischen Funktion unter allen logisch äquivalenten Funktionen finden, so stellt sich heraus, daß dieses Problem nicht berechenbar ist. Aus diesem Grund beschränkt man sich beim Schaltungsentwurf auf die kürzeste disjunktive Normalform, die nicht die kürzeste Darstellung der Funktion sein muß.