



universität
wien

DIPLOMARBEIT / DIPLOMA THESIS

Titel der Diplomarbeit / Title of the Diploma Thesis

„Boole’sche Algebra und ihre
Anwendung in der Digitaltechnik“

verfasst von / submitted by

Doris Glatz

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Magistra der Naturwissenschaften (Mag.rer.nat.)

Wien, 2017 / Vienna, 2017

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 190 406 423

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Lehramtsstudium, UF Mathematik, UF Chemie

Betreut von / Supervisor:

ao. Univ.-Prof. Mag. Dr. Günther Hörmann

Zusammenfassung

Als George Boole im Jahr 1854 versuchte, einen algebraischen Weg zu finden, um logische Probleme zu lösen, schuf er eine komplett neue algebraische Struktur - die Boole'sche Algebra. Damals konnte noch niemand ahnen, dass diese Entdeckung einmal Grundstein für ein komplett neues technisches Gebiet - die Digitaltechnik - sein würde. Und dass sich aus der Boole'schen Algebra die wohl für unseren Alltag bedeutendste Errungenschaft - die Erfindung des Computers - entwickeln könnte, war damals noch undenkbar. Möglich wurde dies, als der junge Student Claude E. Shannon 1937 die Analogie der Boole'schen Algebra zu realen Schaltkreisen erkannte und somit ein neues Anwendungsgebiet für die Boole'schen Algebra schuf. Mit dieser Grundlage stand der Realisierung von zahlreichen Maschinen und Rechenmaschinen - wie auch der Computer eine ist - nichts mehr im Wege.

Auch wenn die Digitaltechnik heute eine eigene Disziplin darstellt, ist sie schon aus ihrer Geschichte heraus direkt mit der Mathematik verbunden. Diese Arbeit beschränkt sich daher nicht nur auf eine rein mathematische Abhandlung der Boole'schen Algebra und eine rein technische Abhandlung der Digitaltechnik, sondern betrachtet auch die Digitaltechnik aus einem mathematischen Blickwinkel.

Abstract

In 1854 George Boole set out to find an algebraic way, to solve logic problems. In this process he found a completely new algebraic structure - the Boolean Algebra. Even George Boole was not aware of the fact, that this discovery would lead to a completely new field - the digital technology. And nobody would have dared to imagine, that the computer - one of the most important inventions in our everyday life - would evolve from this basic principle. This evolution was enabled by Claude E. Shannons findings in 1937. He established the analogy between Boolean Algebra and real switching circuits. Therefore, he created a completely new field of application for Boolean Algebra. Shortly thereafter the first calculating machines and subsequently computers were built.

Nowadays the digital technology has grown into a whole new field. However the observation of its history shows its direct connection to mathematics. Taking this into account, this thesis will not only be limited to a pure mathematical view of the Boolean Algebra or a pure technical view of digital technology, but approaches the digital technology from an mathematical point of view.

Danksagung

Ich möchte diese Gelegenheit nutzen, um mich bei all jenen zu bedanken, die mich bei dieser Diplomarbeit und im Laufe meines Studiums unterstützt haben.

Ein großer Dank gebührt ao. Univ.-Prof. Mag. Dr. Günther Hörmann für die wirklich ausgezeichnete Betreuung. Er war jederzeit für meine Fragen erreichbar und sein konstruktives Feedback hat sehr zum Gelingen dieser Arbeit beigetragen.

Danke auch an all meine Freundinnen und Freunden, die dazu beigetragen haben, dass ich mein Studium immer in bester Erinnerung behalten werden. Besonders möchte ich meine beste Freundin Sara erwähnen, die mich vom ersten Semester an im Mathe-Studium begleitet hat, und auch privat immer für mich da ist.

Ein herzlicher Dank gilt auch meinem Partner Nikolaus. Er hat mich stets ermutigt und war mir während meines Studiums immer eine große Stütze. Danke außerdem für das Korrekturlesen dieser Arbeit.

Mein größter Dank gilt allerdings meiner Familie - vor allem meinen Eltern und meinen Großeltern - die stets an mich geglaubt haben und mich - nicht nur in finanzieller Hinsicht - immer unterstützt haben.

Inhaltsverzeichnis

1	Einleitung	9
2	Grundlagen der Aussagenlogik	11
2.1	Aussagen	12
2.2	Ausdrücke	15
2.2.1	Tautologien und Kontradiktionen	16
3	Boole'sche Algebra	17
3.1	Axiomatik nach Huntington	18
3.2	Folgerungen aus Huntingtons Axiomatik	21
3.3	Vereinfachung von boole'schen Ausdrücken	24
3.4	Boole'sche Funktionen	26
3.5	Weitere Modelle der Boole'schen Algebra	26
4	Schaltalgebra	27
4.1	Grundlagen der Schaltalgebra	28
4.2	Kombination von Schaltungen	28
4.2.1	Negation	29
4.2.2	Parallelschaltung	29
4.2.3	Serienschaltung	30
4.3	Die Schaltalgebra - eine Boole'sche Algebra	32
4.4	Schaltfunktionen	33
4.5	Technische Realisierung	34
4.5.1	Gatterdarstellung	35
4.6	Illustrationen einiger Gesetze der Schaltalgebra	37
5	Entwicklung von Schaltnetzen	39
5.1	Problemerkfassung	40
5.2	Funktionsgleichung aufstellen	40
5.2.1	Vollkonjunktionen und Volldisjunktionen	41
5.2.2	Disjunktive Normalform (DNF)	44

5.2.3	Konjunktiven Normalform (KNF)	48
5.2.4	Normalformen von partiell definierten Funktionen	51
5.3	Minimierungsverfahren	52
5.3.1	Minimierung mit Hilfe von KV-Diagrammen	53
5.3.2	Quine-McCluskey	63
5.4	Bauteilauswahl und Realisierung	68
6	Ausblick	69
	Tabellenverzeichnis	71
	Literaturverzeichnis	73

Kapitel 1

Einleitung

It appeared to me that, although Logic might be viewed with reference to the idea of quantity, it had also another and a deeper system of relations. If it was lawful to regard it from without [...], it was lawful also to regard it from within, as based upon facts of another order which have their abode in the constitution of the Mind.

George Boole, 1847¹

Die folgenden Überlegungen inspirierten George Boole im Jahr 1847 zu seinem Essay „The Mathematical Analysis of Logic“, das von vielen als der Grundstein der Mathematisierung der Logik angesehen wird. Mit dem Ziel eine formale Methode zu finden, um logische Probleme zu lösen, erschuf er eine völlig neuartige algebraische Struktur. Er limitierte dabei die quantitative Algebra auf zwei Zustände und verknüpfte bereits bekannte Operationen mit logischen Operationen. Bald stellte sich heraus, dass diese neuartigen Betrachtungen nicht nur in der Aussagenlogik Anwendung finden konnten. Zu seinen Ehren nannte man das von ihm begründete Verknüpfungsgebilde „Boole’schen Algebra“.²

Aus diesem Grund soll auch diese Arbeit mit der Aussagenlogik beginnen. Anhand dieser Grundlage soll später im Kapitel 3 die axiomatische Konstruktion der Boole’schen Algebra motiviert werden und in Kapitel 4 mit der für diese Arbeit wichtigsten Boole’schen Algebra - der Schaltalgebra - fortgesetzt werden.

Diese postulierte Claude Elwood Shannon fast ein Jahrhundert später in seinem Paper „A symbolic analysis of relay and switching circuits“. Er erkannte die Analogie zwischen realen Schaltkreisen und der Boole’schen Algebra und schuf somit die

¹s. [22], S. 49

²vgl.[22], S. 51-64

Grundlage für die mathematische Analyse und Entwicklung von Schaltkreisen.³ Seit dieser Veröffentlichung 1937 wurde die Entwicklung der Schaltalgebra so sehr vorangetrieben, dass bald die ersten Rechenmaschinen und nicht viel später sogar die ersten Computer auf Grundlage der Schaltalgebra gebaut wurden. Wir alle wissen, wie rasant die Entwicklung auf diesem Gebiet vorangetrieben wurde und so ein komplett neues Gebiet der Informationstechnologie und der Digitaltechnik entstanden ist. Aus diesem Grund wird Claude Shannon auch oft als „Vater der Informationstechnologie“ bezeichnet.

Daher soll sich auch diese Arbeit in Kapitel 5 mit einer Weiterführung der Schaltalgebra beschäftigen - und zwar mit der Konstruktion von Schaltnetzen. Eine weiterführende Behandlung würde den Rahmen dieser Diplomarbeit sprengen. Daher soll das Kapitel 6 noch die unzähligen Möglichkeiten vor Augen führen, die mit Hilfe der Boole'schen Algebra und im speziellen der Schaltalgebra realisierbar sind.

³vgl. [22], S. 134ff

Kapitel 2

Grundlagen der Aussagenlogik

Wie das Wort „schön“ der Ästhetik und „gut“ der Ethik, so weist „wahr“ der Logik die Richtung. [...] Wahrheiten zu entdecken ist Aufgabe aller Wissenschaften: der Logik kommt es zu, die Gesetze des Wahrseins zu erkennen.

Gottlob Frege, 1918 ¹

Schon im 4. Jahrhundert vor Christus beschäftigten sich griechische Philosophen und Mathematiker - vor allem Aristoteles - mit dem Begriff „Wahrheit“ und mit Themen, die der heutigen Aussagenlogik schon sehr nahe kommen. Doch erst ab dem 19. Jahrhundert gelang es, diese auch mathematisch zu betrachten. Einer der bekanntesten Vertreter war George Boole. Sein Ziel war es, mit Hilfe von Gleichungen den Wahrheitsgehalt von Sätzen herauszufinden und ebendiese so zu verändern, dass ein einfacherer Satz mit gleicher Bedeutung erhalten wird.² Daher war er der erste, der algebraische Methoden systematisch verwendete, um logische Probleme zu lösen. So entwickelte er eine komplett neue algebraische Struktur, die nicht nur in der Aussagenlogik Anwendung findet - die Boole'sche Algebra.

Um diese geschichtliche Entwicklung nachzuvollziehen, widmet sich auch dieses Kapitel zuerst der Aussagenlogik. Diese Grundlagen benötigen wir außerdem, um in Kapitel 3 die axiomatische Konstruktion der Boole'schen Algebra zu motivieren. So wird sich dieses Kapitel nach der grundlegenden Definition einer Aussage hauptsächlich verschiedenen Verknüpfungen und Wahrheitstabellen widmen. Da sehr viele verschiedene Schreibweisen gebräuchlich sind, werden außerdem die in dieser Arbeit verwendeten Symbole eingeführt.

Als Quellen für dieses Kapitel dienten [11], S. 7-18, [4], S. 1-2, [16] sowie [10], S. 6-12. Wörtliche Zitate werden direkt im Text angeführt.

¹s. [1], S. 198

²vgl. [22], S. 58

2.1 Aussagen

Schon im Alltag begegnen uns unzählige Aussagen. Dieser Begriff ist uns daher schon sehr geläufig. Aber auch in der Logik versteht man unter einer Aussage einen Satz, von dem man sagen kann, dass er entweder wahr oder falsch ist.

Definition 2.1.1. Eine Aussage A ist ein Satz (in Umgangssprache oder als mathematische Behauptung formuliert), dem eindeutig ein Wahrheitswert W , nämlich „falsch“ oder „wahr“, zugeordnet werden kann. Für den Wahrheitswert „wahr“ setzen wir 1, für „falsch“ 0. Falls A wahr ist, gilt $W(A) := 1$, ist A falsch, $W(A) := 0$.³

Daher sind folgende Beispiele Aussagen:

- Ich wohne in Wien.
- $3 + 3 = 7$

Keine Aussagen sind hingegen:

- Wie alt bist du?
- Mach das!

Natürlich kann man auch Aussagen kombinieren und so neue Aussagen schaffen. Diese verknüpften Aussagen sind dann ebenfalls wieder für sich als ganzes wahr oder falsch. In der Alltagssprache verwenden wir besondere Verknüpfungswörter oder -phrasen um die Verknüpfung auszudrücken. Beispiele dafür sind „und“, „oder“, „entweder - oder“, „wenn - dann“ oder „genau dann - wenn“.

In der Mathematik werden solche Verknüpfungen Junktoren (logische Operatoren) genannt. Viele der Junktoren tragen einen eigenen Namen (z.B. „Konjunktion“), ihnen wird ein eigenes Symbol zugeordnet (z.B. „ $\wedge$ “) und viele entsprechen auch den vorhin genannten Verknüpfungswörtern aus der Alltagssprache (in diesem Fall „und“).

Beispiel 2.1.2. $\underbrace{3 \text{ ist eine Primzahl}}_A$ und $\underbrace{12 \text{ ist eine Primzahl}}_B$.

Diese Aussage besteht aus einer wahren Aussage A und einer falschen Aussage B . Verknüpft mit einer UND-Verknüpfung ergeben die Aussagen eine falsche Aussage. Aus dem Alltag wissen wir: Der Satz „A und B“ ist nur dann wahr, wenn sowohl A als auch B wahr sind.

Definition 2.1.3. Eine Verknüpfung von zwei Aussagen, die nur genau dann wahr ist, wenn beide verknüpften Aussagen für sich genommen wahr sind, nennt man Konjunktion. Ihr wird das Symbol $\wedge$ zugeordnet und sie entspricht dem „und“ der Alltagssprache.

$$W(A \wedge B) = 1 \Leftrightarrow (W(A) = 1 \wedge W(B) = 1)$$

³vgl. [4], S.1

Wir können uns daher eine Tabelle für die UND-Verknüpfung erstellen:

$W(A)$	$W(B)$	$W(A \wedge B)$
0	0	0
0	1	0
1	0	0
1	1	1

Auf die gleiche Art und Weise lassen sich auch Wertetabellen für andere Verknüpfungen erstellen. Je nach Anzahl der verknüpften Aussagen spricht man dabei von einer ein-, zwei- oder n-stelligen Verknüpfung.

Betrachten wir zunächst alle einstelligen Junktoren, die sich nur auf eine einzige Aussage beziehen. Von ihnen gibt es genau vier - ihre wichtigste Vertreter sind die Identität und die Negation (oder auch Komplement genannt).

Tabelle 2.1: **Alle 1-stelligen Verknüpfungen**

		Identität	Negation	
$W(A)$		$W(A)$	$\neg W(A)$	
0	0	0	1	1
1	0	1	0	1

Man überlegt sich leicht an Hand aller Möglichkeiten von Wertzuweisungen, dass außer den vier, in obiger Tabelle dargestellten, einstelligen Junktoren keine weiteren gefunden werden können.

Betrachten wir nun allgemein n-stellige Junktoren, so gibt es dafür in der zweiwertigen Logik⁴ genau 2^{2^n} Möglichkeiten.

Die $16 = 2^{2^2}$ möglichen zweistelligen Junktoren sind in der Tabelle 2.2 dargestellt.⁵

Tabelle 2.2: **Alle 16 möglichen zweistelligen Junktoren**

A	B	$A \wedge \neg A$	$A \wedge B$	$A \wedge \neg B$	A	$\neg A \wedge B$	B
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1
1	0	0	0	1	1	0	0
1	1	0	1	0	1	0	1
		J_0	J_1	J_2	J_3	J_4	J_5

⁴Nur die Werte 0 und 1 können angenommen werden

⁵Tabelle 2.2, Bemerkung 2.1.4 und Beispiel 2.1.5 vgl. [10], S. 9f

A	B	$\neg(A \leftrightarrow B)$	$A \vee B$	$\neg(A \vee B)$	$A \leftrightarrow B$	$\neg B$
0	0	0	0	1	1	1
0	1	1	1	0	0	0
1	0	1	1	0	0	1
1	1	0	1	0	1	0
		J_6	J_7	J_8	J_9	J_{10}

A	B	$B \rightarrow A$	$\neg A$	$A \rightarrow B$	$\neg(A \wedge B)$	$A \vee \neg A$
0	0	1	1	1	1	1
0	1	0	1	1	1	1
1	0	1	0	0	1	1
1	1	1	0	1	0	1
		J_{11}	J_{12}	J_{13}	J_{14}	J_{15}

Bemerkung 2.1.4 (Indizierung). Die Zahlenfolge von Nullern und Einsern von oben nach unten gelesen, kann als Dualzahl aufgefasst werden. Als Index wurde die entsprechende Zahl im Dezimalsystem gewählt.

Beispiel 2.1.5. $1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$ daher J_9

Wir kennen bereits J_1 , die Konjunktion. Es gibt aber auch noch andere zweistellige Junktoren, die einen Namen tragen und für uns von Bedeutung sind.

J_7 Die Disjunktion oder auch ODER-Verknüpfung genannt (Symbol $\vee$) entspricht dem „entweder - oder“ in der Alltagssprache. $W(A \vee B)$ ist dann 1 (also wahr), wenn entweder $W(A)$ oder $W(B)$ oder beide Aussagen 1 (also wahr) sind.⁶

J_9 Die Bijunktion mit dem Symbol $\leftrightarrow$ kennen wir aus unserem Alltag, wenn wir „A genau dann, wenn B“ sagen. $W(A \leftrightarrow B)$ ist also genau dann 1, wenn $W(A) = W(B)$.

J_{13} Die letzte Junktoren, die einen eigenen Namen und ein eigenes Symbol (und zwar $\rightarrow$) hat, ist die sogenannte Subjunktion⁷. In der Alltagssprache wird sie auch mit „wenn A, so folgt B“ verwendet. $W(A \rightarrow B)$ ist nur genau dann 1 wenn $W(B) \geq W(A)$. Sie stellt also eine Folgebeziehung von A zu B dar.

⁶Es handelt sich hierbei nicht um ein ausschließendes Oder. Dieses würde J_6 entsprechen.

⁷Wie man in Tabelle 2.2 sehen kann, können die restlichen Junktoren durch Kombination der bereits erwähnten Junktoren dargestellt werden. In manchen Quellen findet man noch Symbole für weitere Junktoren, diese sind aber wenig gebräuchlich und spielen in dieser Arbeit keine Rolle.

2.2 Ausdrücke

Um mehr als zwei verschiedene Aussagen miteinander verknüpfen zu können, benötigen wir aber auch 3-stellige-, 4-stellige und allgemein n -stellige Verknüpfungen. Da es in der zweiwertigen Logik allerdings jeweils 2^{2^n} möglichen n -stelligen Verknüpfungen gibt, wäre die Vielfalt der Junktoren schnell nicht mehr überschaubar und wir könnten uns auch keine Namen und Vorschriften mehr merken.

Dies ist aber auch gar nicht notwendig, da die 2- und 1-stelligen Verknüpfungen genügen, um alle weiteren n -stelligen Verknüpfungen auszudrücken.

Definition 2.2.1. Ein Ausdruck ist eine mehrstellige Verknüpfung mit Hilfe der 1- und 2-stelligen Junktoren.

Beispiel 2.2.2. $(\neg a \wedge b) \leftrightarrow (c \rightarrow a)$

Definition 2.2.3. Werden nun nur die Verknüpfungen $\wedge$, $\vee$ und $\neg$ verwendet, so spricht man von einem Boole'schen Ausdruck.

Beispiel 2.2.4. $(a \wedge c) \vee (a \wedge \neg b)$

Bemerkung 2.2.5. Wie auch über dem Körper der reellen Zahlen, werden auch hier Ausdrücke strikt von links nach rechts abgearbeitet (außer es wurden Klammern gesetzt). Genauer wird dies in Kapitel 3 auf Seite 25 behandelt.

Beispiel 2.2.6. Beispiel für die Berechnung des Wahrheitswerts eines Boole'schen Ausdrucks mit Hilfe einer Wahrheitstabelle.

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

Wie wir einen Boole'schen Ausdruck auch ohne Wahrheitstabellen vereinfachen können wird in Kapitel 3 behandelt. Es wird ersichtlich werden, dass dabei oft Teilausdrücke gesucht werden, die immer 0 oder 1 sind. Diese sind nämlich leicht eliminierbar.

Als Grundlage für diese späteren Anwendungen betrachten wir daher formal wahre bzw. formal falsche Ausdrücke.

2.2.1 Tautologien und Kontradiktionen

Definition 2.2.7. Man nennt einen formal wahren Ausdruck, der für jede beliebige Belegung den Wahrheitswert 1 annimmt eine **Tautologie**. Im Gegensatz dazu werden formal falsche Ausdrücke, die immer den Wahrheitswert 0 annehmen, **Kontradiktion** genannt.⁸

Beispiel 2.2.8. Tautologien sind:

- Du bist entweder in Wien geboren oder nicht in Wien geboren.
- $A \vee (\neg(A \wedge B)) = 1$, wie man einfach mit der Wahrheitstabelle zeigen kann⁹

A	B	$A \wedge B$	$\neg(A \wedge B)$	$A \vee (\neg(A \wedge B))$
1	1	1	0	1
1	0	0	1	1
0	1	0	1	1
0	0	0	1	1

Kontradiktionen wären hingegen:

- 3 ist eine Primzahl und 3 ist keine Primzahl.
- $(A \wedge B) \wedge ((\neg A) \vee (\neg B)) = 0$ ¹⁰

A	B	$\neg A$	$\neg B$	$(\neg A) \vee (\neg B)$	$A \wedge B$	$(A \wedge B) \wedge ((\neg A) \vee (\neg B))$
1	1	0	0	0	1	0
1	0	0	1	1	0	0
0	1	1	0	1	0	0
0	0	1	1	1	0	0

⁸vgl. [4], S. 9f

⁹Beispiel aus [16], S. 1.

¹⁰Beispiel ebenfalls aus [16], S. 2

Kapitel 3

Boole'sche Algebra

Um die Aussagenlogik systematisch behandeln zu können, entwickelte George Boole¹ im Jahr 1854 die Boole'sche Algebra. Mit ihrer Hilfe wollte er zum Beispiel auch ohne Wahrheitstabellen beweisen, dass zwei Ausdrücke äquivalent sind. Allerdings verwendete er in seiner Version der Boole'schen Algebra die Symbole $\cdot$ für die „UND“-Verknüpfung und $+$ für die „ODER“- Verknüpfung. Wir werden sehen, dass dies bei der Schaltalgebra in Kapitel 4 wieder aufgegriffen wird, heutzutage aber meist die Symbole $\wedge$ und $\vee$ in Verwendung sind. Auch sonst wurde die Boole'sche Algebra seit George Boole vielfach weiterentwickelt.

Wenn wir nun zuerst den axiomatischen Grundaufbau der Boole'schen Algebra betrachten, so beschränken wir uns dabei auf die Axiomatik nach Huntington. Diese wird sehr häufig verwendet und ist auch für unsere Zwecke sehr praktikabel. Für diesen Teil dienten mir [23], S. 6ff, [4], S. 8ff, [3], S. 59 und [17], S. 41ff als Quellen. In Folge benötigen wir noch einige Grundlagen, um dann die wichtigsten Folgerungen aus Huntingtons Axiomatik herzuleiten. Diese spielen bei der Vereinfachung von Boole'schen Termen eine zentraler Rolle. Bei den Folgerungen habe ich mich an [4], S. 8, [11], S. 52, [19], S. 72f und [7] gehalten, für den dritten Teil dieses Kapitels - der Vereinfachungen von Termen - verwende ich [19], S. 74, [10], S. 106 und [4]. Als Vorbereitung für die Schaltalgebra in Kapitel 4 wird außerdem noch ein geeigneter Funktionsbegriff definiert, der an [11], S. 68 und [4], S. 25 angelehnt ist. Zum Abschluss dieses Kapitels soll noch gezeigt werden, dass jedes zu einer Boole'schen Algebra isomorphe Verknüpfungsgebilde mit den drei Verknüpfungen ebenfalls eine Boole'sche Algebra darstellt. Daher gibt es zahlreiche Anwendungsbeispiele der Boole'schen Algebra - nicht nur in der Aussagenlogik sondern auch zum Beispiel in der Wahrscheinlichkeit oder der Elektrotechnik. Alle Axiome, die für die Boole'sche Algebra gelten, gelten auch in diesen Verknüpfungsgebilden. Satz und Beweis sind großteils [3], S. 59 entnommen.

¹[23], S. 6

3.1 Axiomatik nach Huntington

Definition 3.1.1. Eine nichtleere Menge B zusammen mit den Verknüpfungen $\wedge$, $\vee$ und $\neg$ heißt Boole'sche Algebra, wenn folgende Axiome erfüllt sind:

$\forall a, b, c \in B$ gilt:²

1. **Abgeschlossenheit** von B bezüglich $\wedge$ und $\vee$

2. **Neutralitätsgesetze (NG):**

Es existiert ein neutrales Element bezüglich $\wedge$: $a \wedge 1 = 1 \wedge a = a$

Es existiert ein neutrales Element bezüglich $\vee$: $a \vee 0 = 0 \vee a = a$

3. **Komplementaritätsgesetz (KoG)**

Es gibt zu jedem $a \in B$ ein $\neg a$ in B , sodass gilt: $a \wedge \neg a = 0$ und $a \vee \neg a = 1$

4. **Kommutativitätsgesetz (KG)**

Bezüglich $\wedge$: $a \wedge b = b \wedge a$

Bezüglich $\vee$: $a \vee b = b \vee a$

5. **Distributivgesetze (DG)**

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

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

6. **Es gibt mindestens 2 Elemente** $\exists a, b \in B$ mit $a \neq b$

Besteht die Menge B nur aus den zwei Elementen 0 und 1 so spricht man von der Schaltalgebra, die die Grundlage für die Digitaltechnik darstellt. Mit dieser werden wir uns im Kapitel 4 genauer befassen.

Bemerkung 3.1.2.

- Dieses Axiomensystem ist nicht unabhängig. Man kann zeigen, dass man die Definition eines der neutralen Elemente und ein Kommutativgesetz für die Eindeutigkeit gar nicht zwingend braucht, da diese aus den restlichen Axiomen ableitbar sind.³ Für die Anwendungen der booleschen Algebra in der Schaltalgebra, in der jede Verknüpfung ein zusätzliches Bauteil - und somit

²Man spricht auch bei der Boole'schen Algebra von Variablen.

³Für den Beweis, dass dieses Axiomensystem nicht unabhängig ist, sei auf [17], S. 42f verwiesen.

auch zusätzliche Kosten - bedeutet, ist es allerdings oft günstiger, mehrere Axiome einzuführen, da sonst die verwendeten Ausdrücke zu kompliziert wären.

- Oft werden anstatt der Symbole $\wedge$ und $\vee$ auch andere Symbole, wie $\sqcap$ und $\sqcup$ oder auch $+$ und $\cdot$, verwendet. Dies spielt aber keine Rolle. Grundsätzlich spricht man bei Termen, die mit $\wedge$ verknüpft sind auch oft von Produkttermen und bei Termen, die mit $\vee$ verknüpft sind von Summentermen.

Um weitere Gesetze aus diesen Axiomen herleiten zu können, müssen wir zuerst den Begriff des „Boole’schen Terms“ einführen. Ein Boole’scher Term ist analog zu einem Boole’schen Ausdruck in der Aussagenlogik definiert.

Definition 3.1.3 (Boole’scher Term⁴).

1. Jede Variable über B und jede Konstante aus B ist ein Boole’scher Term.
2. Sind a und b Boole’sche Terme, so sind $a \wedge b$, $a \vee b$ und $\neg a$ ebenfalls Boole’sche Terme.

Beispiel 3.1.4. Sind a, b, c und d Variablen, so sind zum Beispiel

- $\neg(a \wedge b) \vee ((d \wedge \neg b) \vee c)$ und
- $a \wedge b \wedge (c \vee d)$ Boole’sche Terme.

Interessant ist außerdem, dass die Axiomatik selbstdual ist.⁵ Dies wird uns später sehr helfen, wenn wir weitere Gesetze aus den Axiomen ableiten wollen.

Definition 3.1.5 (Dualität⁶). Zwei Axiome, Terme oder Gesetze sind genau dann zueinander dual, wenn gilt: vertauscht man in einem von ihnen alle vorkommenden Zeichen $\wedge$ und $\vee$, sowie die Konstanten 0 und 1 jeweils gegeneinander, so erhält man das andere.

Beispiel 3.1.6. Laut der Axiomatik von Huntington gilt: $a \wedge \neg a = 0$. Durch Austauschen von $\wedge$ durch $\vee$ und 0 durch 1 erhält man sofort das zweite Neutralitätsgesetz $a \vee \neg a = 1$

Da alle Gesetze der Boole’schen Algebra dual sind, ist diese Eigenschaft für die Herleitung weiterer Gesetze und Theoreme aus der Boole’schen Algebra insofern von Bedeutung, dass wir in Folge ebenfalls immer nur ein Gesetz beweisen müssen und das duale Gesetz sofort durch Anwendung der Dualität bewiesen haben. Diesen Sachverhalt nennt man auch Dualitätsprinzip.⁷

⁴[3], S. 52

⁵Vgl. [17], S. 42

⁶Vgl. Bong, S. 54

⁷Vgl. ebenfalls [3], S. 54

Satz 3.1.7 (Dualitätsprinzip) *Wurde ein Gesetz oder ein Theorem aus den Gesetzen der Boole'schen Algebra hergeleitet, so ist die duale Aussagen ebenso gültig.*

Bevor wir nun weitere Folgerungen aus der Axiomatik finden können, benötigen wir noch die Eindeutigkeit des Komplements und der neutralen Elemente.⁸

Satz 3.1.8 (Eindeutigkeit der neutralen Elemente)

In einer Boole'schen Algebra sind die neutralen Elemente 0 (bzgl. $\vee$) und 1 (bzgl. $\wedge$) eindeutig bestimmt.

Beweis. Wir nehmen an, wir hätten zwei neutrale Elemente bezüglich $\vee$ gefunden und zwar 0 und 0^* mit $0 \neq 0^*$. Wir folgern:

$$\begin{array}{ll} 0 = 0 \vee 0^* & \text{da } 0^* \text{ ein neutrales Element ist} \\ = 0^* \vee 0 & \text{gilt auf Grund des KG} \\ = 0^* & \text{da } 0 \text{ ein neutrales Element ist} \end{array}$$

und daher kann es nur ein neutrales Element geben, das wir mit 0 bezeichnen.

Den Beweis, dass auch das neutrale Element bezüglich $\wedge$ eindeutig bestimmt ist, können wir direkt aus dem Dualitätsprinzip folgern. $\square$

Fehlt uns nur noch der Beweis, dass auch das Komplement eindeutig bestimmt ist.

Satz 3.1.9 (Eindeutigkeit der Komplements)

In einer Boole'schen Algebra ist das Komplement eindeutig bestimmt, das heißt $\forall a \in B \exists! \neg a$ für das gilt $a \wedge \neg a = 0$ und $a \vee \neg a = 1$.

Beweis. Wir nehmen wieder an, wir hätten zwei Komplemente zu a gefunden, und zwar $\neg a$ und a^* . Wir folgern daher:

$$\begin{aligned} \neg a &\stackrel{\text{NG}}{=} \neg a \vee 0 \stackrel{\text{KoG}}{=} \neg a \vee (a \wedge a^*) \stackrel{\text{DG}}{=} (\neg a \vee a) \wedge (\neg a \vee a^*) \stackrel{\text{KG}}{=} \underbrace{(a \vee \neg a)}_1 \wedge (\neg a \vee a^*) \\ &= 1 \wedge (\neg a \vee a^*) \stackrel{\text{KoG}}{=} (a \vee a^*) \wedge (\neg a \vee a^*) \stackrel{\text{KG}}{=} (a^* \vee a) \wedge (a^* \vee \neg a) \stackrel{\text{DG}}{=} a^* \vee \underbrace{(a \wedge \neg a)}_0 \\ &\stackrel{\text{KoG}}{=} a^* \vee 0 \stackrel{\text{NG}}{=} a^* \end{aligned}$$

und daher gibt es nur genau ein Komplement, und zwar $\neg a$. $\square$

Nun haben wir alle nötigen Grundlagen behandelt, um aus den Axiomen nach Huntington weitere wichtige und sehr praktische Gesetze zu folgern. Diese werden uns beim Vereinfachen von boole'schen Ausdrücken sehr nützlich sein.

⁸Beide Sätze und Beweise vgl. [3], S. 53

3.2 Folgerungen aus Huntingtons Axiomatik

Die folgenden Sätze gelten $\forall a, b, c \in B$ mit den Verknüpfungen $\wedge$, $\vee$ und $\neg$.

Zuallererst betrachten wir eine weitere Eigenschaft der boole'schen Algebra. Verknüpft man ein Element mit sich selbst, so ist dies redundant und man erhält wieder das selbe Ergebnis.

Satz 3.2.1 (Idempotenzgesetze (IG)) a) $a \wedge a = a$ b) $a \vee a = a$

Beweis.

von a): $a \wedge a \stackrel{\text{NG}}{=} (a \wedge a) \vee 0 \stackrel{\text{KoG}}{=} (a \wedge a) \vee \underbrace{(a \wedge \neg a)}_0 \stackrel{\text{DG}}{=} a \wedge \underbrace{(a \vee \neg a)}_1 \stackrel{\text{KoG}}{=} a \wedge 1 \stackrel{\text{NG}}{=} a$

von b): folgt aus dem Dualitätsprinzip (vertausche 0 gegen 1 und $\wedge$ gegen $\vee$) □

Wie in der Aussagenlogik, sowie in der Alltagssprache, ergibt bei der Boole'schen Algebra eine doppelte Verneinung wieder eine Bejahung. Bildet man also zweimal das Komplement eines Elements, so erhält wieder das Ausgangselement.

Satz 3.2.2 (Gesetz des doppelten Komplements (DK)) $\neg(\neg a) = a$

Beweis.

$\neg(\neg a)$ ist das Komplement von $\neg a$ und daher gilt auf Grund der KoG: $\neg a \vee \neg(\neg a) = 1$ und $\neg a \wedge \neg(\neg a) = 0$. Gleichzeitig gilt $\neg a \vee a = 1$ und $\neg a \wedge a = 0$ und somit ist a auch ein Komplement zu $\neg a$. Da das Komplement aber eindeutig ist, folgt $a = \neg(\neg a)$. □

Weiters lassen sich auch noch auf sehr einfache Art und Weise die Dualitätsgesetze beweisen, die nicht mit dem Dualitätsprinzip aus Satz 3.1.7 verwechselt werden dürfen, aber natürlich mit diesem in Zusammenhang steht.

Satz 3.2.3 (Dualitätsgesetze(DuG)) a) $\neg 0 = 1$ b) $\neg 1 = 0$

Beweis.

von a) $\neg 0 \stackrel{\text{NG}}{=} 0 \vee \neg 0 \stackrel{\text{KoG}}{=} 1$

von b) Dualitätsprinzip □

Ein sehr wichtiges Gesetz stellt für uns auch das Absorptionsgesetz⁹ dar, da es beim Kürzen von Boole'schen Termen sehr oft verwendet werden kann.

⁹Benennung laut [19], in anderen Quellen auch Extremalgesetz genannt

Satz 3.2.4 (Absorptionsgesetz (AbG)) a) $a \wedge 0 = 0$ b) $a \vee 1 = 1$

Beweis.

$$\text{von a):} \quad 0 \stackrel{\text{KoG}}{=} a \wedge \neg a \stackrel{\text{NG}}{=} a \wedge \underbrace{(\neg a \vee 0)}_{\neg a} \stackrel{\text{DG}}{=} (a \wedge \neg a) \vee (a \wedge 0) \stackrel{\text{KoG}}{=} 0 \vee (a \wedge 0) \stackrel{\text{NG}}{=} a \wedge 0$$

$$\text{von b):} \quad \text{Dualitätsprinzip} \quad \square$$

Etwas weniger intuitiv zu verstehen ist hingegen das Verschmelzungsgesetz¹⁰. Es besagt, dass die Belegung der Variable b irrelevant ist, wenn diese hintereinander konjunktiv und disjunktiv mit derselben Aussage verknüpft ist. Ob zuerst konjunktiv oder disjunktiv ist egal.

Satz 3.2.5 (Verschmelzungsgesetze (VG)) a) $a \vee (a \wedge b) = a$ b) $a \wedge (a \vee b) = a$

Beweis.

$$\text{von a):} \quad a \stackrel{\text{KoG}}{=} \underbrace{a \wedge 1}_a \stackrel{\text{AbG}}{=} a \wedge \underbrace{(b \vee 1)}_1 \stackrel{\text{DG}}{=} (a \wedge b) \vee \underbrace{(a \wedge 1)}_a \stackrel{\text{NG}}{=} (a \wedge b) \vee a \stackrel{\text{KG}}{=} a \wedge (a \vee b)$$

$$\text{von b):} \quad \text{Dualitätsprinzip} \quad \square$$

Einer der wichtigsten, aber auch nicht so einfach zu beweisenden Sätze, ist das Assoziativgesetz. Es besagt, dass man beliebige Klammern in einem Term, der nur Konjunktionen oder nur Disjunktionen enthält, setzen darf. Es dürfen also beliebige Teilsummen oder Teilprodukte gebildet werden.

Satz 3.2.6 (Assoziativgesetz (AG)) a) $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
b) $a \vee (b \vee c) = (a \vee b) \vee c$

Beweis.

von a): Der Beweis des AG gestaltet sich etwas komplizierter. Wir betrachten zuerst:

$$\begin{aligned} a \vee ((a \wedge b) \wedge c) &\stackrel{\text{DG}}{=} (a \vee (a \wedge b)) \wedge (a \vee c) = \\ &\stackrel{\text{VG}}{=} a \wedge (a \vee c) = \\ &\stackrel{\text{VG}}{=} a = \\ &\stackrel{\text{VG}}{=} a \vee (a \wedge (b \wedge c)) \end{aligned}$$

¹⁰Benennung laut [19], in anderen Quellen wird dieses Gesetz stattdessen Absorptionsgesetz genannt

Wir formen um:

$$\begin{aligned}
 \neg a \vee ((a \wedge b) \wedge c) & \stackrel{\text{DG}}{=} (\neg a \vee (a \wedge b)) \wedge (\neg a \vee c) = \\
 & \stackrel{\text{DG}}{=} \underbrace{((\neg a \vee a) \wedge (\neg a \vee b))}_{1} \wedge (\neg a \vee c) = \\
 & \stackrel{\text{KoG/NG}}{=} (\neg a \vee b) \wedge (\neg a \vee c) = \\
 & \stackrel{\text{DG}}{=} \neg a \vee (b \wedge c) = \\
 & \stackrel{\text{KoG/NG}}{=} \underbrace{(\neg a \vee a) \wedge (\neg a \vee (b \wedge c))}_{1} = \\
 & \stackrel{\text{DG}}{=} \neg a \vee (a \wedge (b \wedge c))
 \end{aligned}$$

Nun verknüpfen wir jeweils die linken und die rechten Seiten der beiden Gleichungen mit $\wedge$ und erhalten:

$$\begin{aligned}
 & \stackrel{\text{DG}}{\Rightarrow} [a \vee ((a \wedge b) \wedge c)] \wedge [\neg a \vee ((a \wedge b) \wedge c)] = [a \vee (a \wedge (b \wedge c))] \wedge [\neg a \vee (a \wedge (b \wedge c))] \\
 & \stackrel{\text{KoG}}{\Rightarrow} ((a \wedge b) \wedge c) \vee \underbrace{(a \wedge \neg a)}_0 = (a \wedge (b \wedge c)) \vee \underbrace{(a \wedge \neg a)}_0 \\
 & \stackrel{\text{KoG}}{\Rightarrow} (a \wedge b) \wedge c = a \wedge (b \wedge c)
 \end{aligned}$$

von b): Dualitätsprinzip

□

Die folgenden Regeln sind für die boole'sche Algebra von großer Bedeutung. Sie wurden nach dem englischen Mathematiker DeMorgan benannt, der diese formulierte.

Satz 3.2.7 (Gesetze von DeMorgan (DM)) a) $\neg(a \vee b) = \neg a \wedge \neg b$
 b) $\neg(a \wedge b) = \neg a \vee \neg b$

Beweis.

$$\begin{aligned}
 \textcircled{1} \quad (a \vee b) \wedge (\neg a \wedge \neg b) & \stackrel{\text{KG}}{=} (\neg a \wedge \neg b) \wedge (a \vee b) = \\
 & \stackrel{\text{DG}}{=} ((\neg a \wedge \neg b) \wedge a) \vee ((\neg a \wedge \neg b) \wedge b) = \\
 & \stackrel{\text{KG+AG}}{=} \underbrace{(\neg b \wedge (\neg a \wedge a))}_0 \vee \underbrace{(\neg a \wedge (\neg b \wedge b))}_0 = \\
 & = (\neg b \wedge 0) \vee (\neg a \wedge 0) = \\
 & \stackrel{\text{KoG}}{=} 0 \vee 0 \stackrel{\text{NG}}{=} 0
 \end{aligned}$$

$$\begin{aligned}
② \quad (a \vee b) \vee (\neg a \wedge \neg b) &\stackrel{\text{DG}}{=} ((a \vee b) \vee \neg a) \wedge ((a \vee b) \vee \neg b) = \\
&\stackrel{\text{KG}}{=} (\neg a \vee (a \vee b)) \wedge ((a \vee b) \vee \neg b) = \\
&\stackrel{\text{AG}}{=} (\underbrace{(\neg a \vee a)}_1 \vee b) \wedge (a \vee \underbrace{(b \vee \neg b)}_1) = \\
&= (1 \vee b) \wedge (a \vee 1) = \\
&\stackrel{\text{KoG}}{=} 1 \wedge 1 \stackrel{\text{NG}}{=} 1
\end{aligned}$$

Wegen KoG und der Eindeutigkeit des Komplements folgt aus ① und ② die erste Regel von DeMorgan. Die zweite folgt aus dem Dualitätsprinzip. $\square$

3.3 Vereinfachung von boole'schen Ausdrücken

Mit Hilfe dieser Gesetze lassen sich nun selbst sehr komplizierte Boole'sche Ausdrücke systematisch vereinfachen. Dies soll nun an einem Beispiel ¹¹ gezeigt werden.

Beispiel 3.3.1.

$$\begin{aligned}
&(\neg a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{DG}}{=} (\neg a \wedge ((\neg b \wedge \neg c) \vee (b \wedge \neg c) \vee (b \wedge c))) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{DG}}{=} (\neg a \wedge (\underbrace{((\neg b \vee b) \wedge \neg c)}_1 \vee (b \wedge c))) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{KoG}}{=} (\neg a \wedge ((1 \wedge \neg c) \vee (b \wedge c))) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{NG}}{=} (\neg a \wedge (\neg c \vee (b \wedge c))) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{DG}}{=} (\neg a \wedge \neg c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{KG}}{=} (\neg a \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c) \vee (a \wedge b \wedge c) = \\
&\stackrel{\text{DG}}{=} (\neg a \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (\underbrace{(\neg a \vee a)}_1 \wedge (b \wedge c)) = \\
&\stackrel{\text{KoG}}{=} (\neg a \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (1 \wedge (b \wedge c)) = \\
&\stackrel{\text{DG}+\text{NG}}{=} ((\neg a \vee (a \wedge \neg b)) \wedge \neg c) \vee (b \wedge c) = \\
&\stackrel{\text{DG}}{=} (\underbrace{(\neg a \vee a)}_1 \wedge (\neg a \vee \neg b) \wedge \neg c) \vee (b \wedge c) = \\
&\stackrel{\text{KoG}}{=} (1 \wedge (\neg a \vee \neg b) \wedge \neg c) \vee (b \wedge c) = \\
&\stackrel{\text{NG}}{=} ((\neg a \vee \neg b) \wedge \neg c) \vee (b \wedge c) = \\
&\stackrel{\text{DM}}{=} (\neg(a \wedge b) \wedge \neg c) \vee (b \wedge c) = \\
&\stackrel{\text{DM}}{=} \neg((a \wedge b) \vee c) \vee (b \wedge c)
\end{aligned}$$

¹¹s. [19], S. 74

Wir erhalten einen wesentlich einfacheren Term. Außerdem haben wir gesehen, dass wir durch Umformung für einen Boole'schen Term mehrere andere äquivalente Darstellungen finden können.

Definition 3.3.2. Entsteht ein Boole'scher Term durch die Umformung eines anderen Terms, so heißen die beiden Terme „äquivalent“.

Je komplexer die Terme werden, desto komplizierter wird auch die Berechnung, da immer mehr Klammern auftauchen. Daher gibt es auch in der Boole'schen Algebra sogenannte **Vorrangregeln** wie wir sie schon aus unserer Schulzeit bei den Rechnungen über dem Körper der reellen Zahlen kennen (analog zu den „Klammer-Punkt-Strich“-Regeln).

Definition 3.3.3 (Vorrangregeln¹²). Es gelten folgende Prioritäten:

- Klammern haben Vorrang vor Negation $\neg$, diese hat Vorrang vor der Konjunktion $\wedge$, diese hat Vorrang vor allen anderen Operatoren
- Bei mehreren aufeinanderfolgenden Verknüpfungen gleicher Art wird strikt von links nach rechts gearbeitet
- Das Konjunktionssymbol $\wedge$ kann weggelassen werden, wenn keine Verwechslungsgefahr besteht.

Mit Hilfe der Gesetze von DeMorgan lässt sich außerdem noch zeigen:

Satz 3.3.4 *Enthält ein Boole'scher Term nur Konjunktionen (oder Disjunktionen), so erhält man auf einfache Art und Weise das Komplement, indem man $\wedge$ durch $\vee$ ersetzt (bzw. umgekehrt) und alle Variablen und Konstanten negiert.¹³*

$$a) \neg(x_1 x_2 x_3 \dots x_n) = \neg x_1 \vee \neg x_2 \vee \neg x_3 \vee \dots \vee \neg x_n$$

$$b) \neg(x_1 \vee x_2 \vee x_3 \vee \dots \vee x_n) = \neg x_1 \neg x_2 \neg x_3 \dots \neg x_n$$

Beweis. von a):

$$\begin{aligned} \neg(x_1 x_2 x_3 \dots x_n) &= \neg((x_1 x_2 x_3 \dots x_{n-1}) x_n) && \text{Vorrangregeln} \\ &= \neg(x_1 x_2 x_3 \dots x_{n-1}) \vee \neg x_n && \text{DeMorgan} \\ &= \neg((x_1 x_2 x_3 \dots x_{n-2}) x_{n-1}) \vee \neg x_n \\ &= \neg(x_1 x_2 x_3 \dots x_{n-2}) \vee \neg x_{n-1} \vee \neg x_n \\ &= \dots \\ &= \neg x_1 \vee \neg x_2 \vee \neg x_3 \vee \dots \vee \neg x_{n-1} \vee \neg x_n \end{aligned}$$

Durch Iteration erhalten wir also das zu Zeigende.

von b): Dualitätsprinzip

□

¹²vgl. [4], S.7

¹³vgl. [4], S. 9

3.4 Boole'sche Funktionen

Vor allem für die spätere Anwendung in der Schaltalgebra ist es für uns sehr nützlich, einen passenden Funktionsbegriff zu definieren.

Definition 3.4.1. Sei $(B, \wedge, \vee, \neg)$ eine Boole'sche Algebra. Eine n -stellige **Boole'sche Funktion** ist eine Abbildung $f: B^n \rightarrow B$, wobei $B^n = \underbrace{B \times B \times \dots \times B}_{n\text{-mal}}$

die Produktmenge von B ist (also die Menge der geordneten n -Tupel).

Ist $B = \{0, 1\}$ nennt man die Abbildung aller geordneten n -Tupel $(x_1, x_2, \dots, x_n)$ mit $x_i \in B \forall i \in \mathbb{N}$ in die Menge B eine **binäre Funktion**.

Eine n -stellige Boole'sche Funktion gibt uns nun zu n verschiedenen Eingangswerten genau einen Ausgangswert. So bildet die folgende Funktion zum Beispiel für die Belegung $(x_1, x_2, x_3) = (0, 1, 1)$ auf 0 ab. Daher kann man nun auch für Boole'sche Funktionen Wertetabellen¹⁴ erstellen.

Beispiel 3.4.2. Beispiel einer binären dreistelligen Funktion:

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

x_1	x_2	x_3	$x_2 \vee \neg x_3$	$f(x_1, x_2, x_3) = x_1 \wedge (x_2 \vee \neg x_3)$
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

3.5 Weitere Modelle der Boole'schen Algebra

Wir kennen schon die Wahrheitswertalgebra, die eine Boole'sche Algebra darstellt. Es gibt auch noch zahlreiche andere Verknüpfungsgebilde, die Boole'sche Algebren sind, wie die Mengenalgebra, die Ereignisalgebra, die Teileralgebra oder die Schaltalgebra. Dies kann man sehr einfach durch Nachprüfen aller Axiome der Boole'schen Algebra beweisen.¹⁵ In Folge gelten daher in allen Modellen der Boole'schen Algebra ebenfalls alle in diesem Kapitel hergeleiteten Gesetze und Regeln.

¹⁴analog zu den Wahrheitstabellen in der Aussagenlogik

¹⁵Für die Schaltalgebra wird dies exemplarisch in Kapitel 4 auf Seite 32 gemacht.

Kapitel 4

Schaltalgebra

Die Schaltalgebra ist die Spezialform der Boole'sche Algebra, die auch außerhalb der Mathematik und in unserem Alltag die größte Rolle spielt. Denn mit Hilfe der Gesetze der Boole'schen Algebra, die auch in der Schaltalgebra gelten, ist es möglich, zu einem Schaltkreis schnell und effizient eine Vereinfachung zu finden¹. Außerdem basieren auf ihr viele technische Errungenschaften, unter anderem der Computer.

Claude Shannon war der erste, der in seiner Abschlussarbeit die Analogie zwischen der Analyse von Schaltnetzen und den Postulaten der Boole'schen Algebra erkannte. So schuf er laut Kittler „*die folgenreichste Magisterarbeit aller Zeiten*“.² In kürzester Zeit entwickelte sich aus dieser Erkenntnis ein komplett neues Wissenschaftsgebiet - die Digitaltechnik - und bald darauf wurde der erste Computer auf Grundlage der Schaltalgebra gebaut.

Der erste Teil dieses Kapitels widmet sich den Grundlagen der Schaltalgebra und ihrer technischen Realisierung. Die Informationen dafür wurden [10], S. 125f und 131ff, [23], S. 7 und 10, [3] S. 66ff und [11], S. 59ff entnommen. Für den Abschnitt über die technische Realisierung habe ich außerdem [21], S. 34 und 55ff, [11], S. 61, [23], S. 10, [20], S. 6 und [2], S. 8 verwendet.

Zum Abschluss folgt noch ein kleiner Exkurs, in dem mit Hilfe von Schaltkreisen einige Gesetze der Boole'schen Algebra veranschaulicht werden. Zur Idee und Umsetzung siehe [10], S. 136.

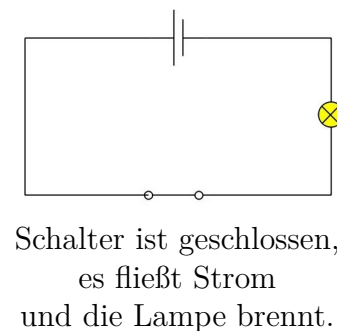
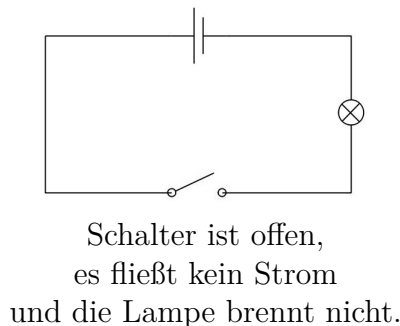
¹vgl. [22], S. 134

²s. [12], S. 331

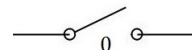
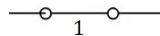
4.1 Grundlagen der Schaltalgebra

Die Schaltalgebra stellt eine spezielle Boole'sche Algebra über der Menge $B = \{0, 1\}$ mit den Verknüpfungen $+$, $\cdot$, $-$ dar. Sie wird primär dazu verwendet, Schaltungen zu realisieren und zu vereinfachen.

Aus der Physik wissen wir, dass Strom nur fließen kann, wenn der Stromkreis geschlossen ist. Fügt man in einen Stromkreis einen Schaltkontakt ein, kann man aus dem Zustand „Schalter ist geschlossen“ schon erkennen, dass „Strom fließt“. Hängt nun als Verbraucher auch noch eine Lampe im Stromkreis, so gibt es auch einen direkten Zusammenhang zwischen „Schalter ist geschlossen“ und „Lampe brennt“.³



Es kommt also alleine auf die Stellung des Schalters an. Ist dieser geschlossen, so ordnet man ihm den Leitwert 1 zu, ist er offen, den Leitwert 0.⁴



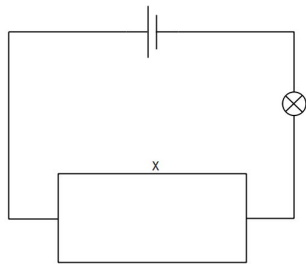
4.2 Kombination von Schaltungen

Analog zu Aussagen, können auch bei der Schaltalgebra mehrere Schalter in einem Stromkreis kombiniert werden. So erhält man ein sogenanntes Schaltnetz.

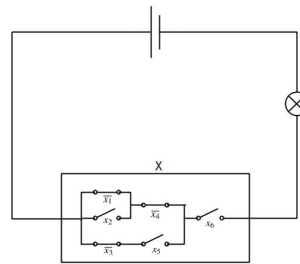
Der Leitwert des gesamten Schaltnetzes wird auch als Leitwert x bezeichnet, der Leitwert jedes einzelnen Schalters üblicherweise mit $x_1, x_2, \dots, x_n$. Diese Variablen werden auch Leitwertvariablen genannt.

³vgl. [3], S. 66

⁴Nicht zu verwechseln mit dem physikalischen Leitwert (Kehrwert des Widerstandes)



$x = 1$, wenn Strom fließt und 0, wenn kein Strom fließt.



Die Stellung der Schalter wird mit dem Leitwert x zusammengefasst.

4.2.1 Negation

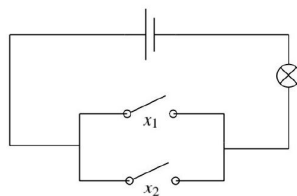
Ist ein Schalter immer dann geschlossen, wenn der Schalter mit dem Leitwert x_i offen ist (und umgekehrt), so realisiert dieser Schalter die Negation des Schalters mit dem Leitwert x_i . Für den Leitwert des Negations-Schalters verwendet man in der Schaltalgebra üblicherweise anstatt $\neg x_i$ das Zeichen $\overline{x_i}$.

In den meisten Lehrbüchern wird für die Benennung der Zustand des Schalters nach der Betätigung betrachtet. Da ein offener Schalter nach Betätigung geschlossen ist, leitet er den Strom und man bezeichnet daher einen offenen Schalter mit x_i . Für einen geschlossenen Schalter ist hingegen die Bezeichnung $\overline{x_i}$ üblich, da dieser nach Betätigung den Stromkreis unterbricht.

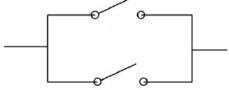
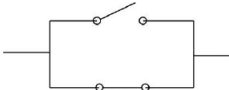
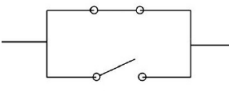
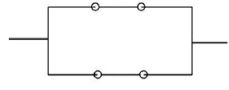
Für die Verknüpfung von einzelnen Schaltern $x_1, x_2, \dots$ gibt es nun grundsätzlich zwei verschiedene Varianten - die Serienschaltung und die Parallelschaltung.

4.2.2 Parallelschaltung

Bei der Parallelschaltung werden zwei (oder mehrere) Schalter parallel geschaltet.



Ob Strom fließt, oder nicht, hängt daher sowohl von der Stellung von x_1 als auch von der Stellung von x_2 ab. Wir betrachten die verschiedenen Möglichkeiten:

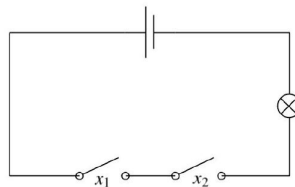
Schalterstellungen (nach der Betätigung)	Leitwert x_1	Leitwert x_2	Leitwert x
	0	0	0
	0	1	1
	1	0	1
	1	1	1

Es fließt also immer Strom, außer beide Schalter sind nach der Betätigung gleichzeitig geöffnet. Ein Beispiel wäre eine Klingel ⁵. Gibt es zwei Schalter, soll die Klingel immer läuten, sobald einer der Schalter (oder auch beide) gedrückt wird.

Wir vergleichen nun die Tabelle mit der Tabelle der Disjunktion auf S. 13. Wir sehen sofort, dass die Parallelschaltung der Disjunktion $\vee$, also der ODER-Verknüpfung entspricht. In der Schaltalgebra verwendet man aber normalerweise nicht das Zeichen $\vee$, sondern das Zeichen $+$.

4.2.3 Serienschaltung

Bei der Serienschaltung werden zwei (oder mehrere) Schalter in Serie geschaltet.



⁵Beispiel aus [3], S. 68

Ob Strom fließt, oder nicht, hängt auch bei der Serienschaltung sowohl von der Stellung von x_1 als auch von der Stellung von x_2 ab. Wir betrachten wieder die verschiedenen Möglichkeiten:

Schalterstellungen (nach der Betätigung)	Leitwert x_1	Leitwert x_2	Leitwert x
	0	0	0
	0	1	0
	1	0	0
	1	1	1

Hier fließt nur Strom, wenn beide Schalter geschlossen sind. Ein Beispiel wäre ein Fahrstuhl.⁶ Sowohl ein Bedienungsknopf im Lift, als auch die Türen in allen Stockwerken dienen als seriengeschaltene Schalter. Der Lift fährt erst los, wenn alle Türen geschlossen sind und der Bedienungsknopf gedrückt ist.

Vergleichen wir nun die Tabelle mit der Tabelle der Konjunktion auf Seite 13, so sehen wir, dass die Serienschaltung, der Konjunktion $\wedge$, also der UND-Verknüpfung entspricht. In der Schaltalgebra ist allerdings meist das Zeichen $\cdot$ in Gebrauch. Auch hier ist es üblich, ein $\cdot$ zur Verkürzung des Terms wegzulassen, falls es für die Eindeutigkeit nicht zwingend notwendig ist.

Beispiel 4.2.1. $x_1 x_2 = x_1 \cdot x_2$

Zusammengefasst gilt also bei der Schaltalgebra:

x_1	x_2	$x_1 \cdot x_2$	$x_1 + x_2$	x_1	$\overline{x_1}$
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1		
1	1	1	1		

⁶Beispiel aus [3], S. 67

4.3 Die Schaltalgebra - eine Boole'sche Algebra

Wir haben nun sowohl ein Komplement gefunden, als auch die Verknüpfungen der Konjunktion und der Disjunktion definiert.

Wir prüfen daher alle Axiome nach Huntington für das Verknüpfungsgebilde $(\{0, 1\}, +, \cdot, ^-)$ nach:

1. **Abgeschlossenheit:** leicht aus den Wertetabellen zu erkennen

2. **Neutrale Elemente:** folgt ebenfalls aus den Tabellen.

$$\text{Bzgl. } +: \quad 1 + \mathbf{0} = \mathbf{0} + 1 = 1 \quad 0 + \mathbf{0} = \mathbf{0} + 0 = 0$$

$$\text{Bzgl. } \cdot: \quad 0 \cdot \mathbf{1} = \mathbf{1} \cdot 0 = 0 \quad 1 \cdot \mathbf{1} = \mathbf{1} \cdot 1 = 1$$

3. **Komplement:** aus der Tabelle kann für alle Elemente aus $\{0, 1\}$ ein Komplement gefunden werden.

$$\text{Für } 0: \quad 0 + \overline{\mathbf{0}} = 0 + \mathbf{1} = 1 \quad 0 \cdot \overline{\mathbf{0}} = 0 \cdot \mathbf{1} = 0$$

$$\text{Für } 1: \quad 1 + \overline{\mathbf{1}} = 1 + \mathbf{0} = 1 \quad 1 \cdot \overline{\mathbf{1}} = 1 \cdot \mathbf{0} = 0$$

4. **Kommutativität:** folgt offensichtlich aus den Wertetabellen.

5. **Distributivität:** zeigen wir mit Hilfe einer Wertetabelle, für das zweite Gesetz analog

x	y	z	$x \cdot (y + z)$	$x \cdot y + x \cdot z$
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

6. **mindestens „zwei“ Elemente:** erfüllt, da B aus genau zwei Elementen besteht.

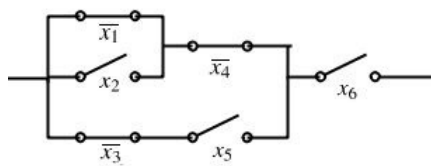
Da alle Axiome nach Huntington in der Schaltalgebra ebenfalls gelten, ist diese eine Boole'sche Algebra und alle in Kapitel 3 hergeleiteten Gesetze gelten auch in ihr.

4.4 Schaltfunktionen

Da es sich bei der Schaltalgebra um eine Boolesche Algebra handelt, können wir natürlich auch binäre Funktionen - meist Schaltfunktionen genannt - aufstellen. Wir definieren analog zu Boole'schen Funktionen:

Definition 4.4.1. Sei die Schaltalgebra $(\{0, 1\}, +, \cdot, -)$ gegeben. Eine n -stellige **binäre Funktion** ist eine Abbildung $f: \{0, 1\}^n \rightarrow \{0, 1\}$, wobei $\{0, 1\}^n$ die Produktmenge von $\{0, 1\}$ ist (also die Menge der geordneten n -Tupel).⁷

Beispiel 4.4.2. $f: \{0, 1\}^6 \rightarrow \{0, 1\}$



$$f(x_1, x_2, x_3, x_4, x_5, x_6) = ((\bar{x}_1 + x_2) \cdot \bar{x}_4 + \bar{x}_3 \cdot x_5) \cdot x_6$$

oder kurz:

$$y = ((\bar{x}_1 + x_2)\bar{x}_4 + \bar{x}_3 x_5)x_6$$

Partiell definierte Funktionen

In der Praxis treten allerdings auch Schaltfunktionen auf, denen nicht für jede Belegung $(x_1, \dots, x_n)$ eine Funktionswert zugeordnet wird. Das bedeutet, dass diese Belegung entweder nie auftritt, oder es ganz egal ist, ob die Belegung auf 0 oder 1 abbildet. Man nennt solche Funktionen „partiell definiert“ und die Belegungen, die nicht definiert sind, „don't-care“-Zustände. Man kennzeichnet sie in den Wertetabellen mit $-$.⁸

Beispiel 4.4.3.

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	0
1	0	—
1	1	1

Die folgende Funktion ist für die Belegung $(1, 0)$ nicht definiert und damit ist f eine partiell definierte Funktion.

⁷vgl. Definition 3.4.1 auf S. 26

⁸vgl. [4], S. 34

4.5 Technische Realisierung

Bis jetzt haben wir uns als Schalter immer einen ganz normalen Kontaktschalter - also einen mechanischen Schalter - vorgestellt. Man nennt diese Technik Schaltertechnik. Diese wurde wenig später zur Relaistechnik erweitert. Im Gegensatz zur Schaltertechnik werden bei der Relaistechnik die Schalter mit Hilfe eines Elektromagneten mechanisch betrieben. Im Laufe der Geschichte rückte die Schalterdarstellung allerdings immer mehr in den Hintergrund. Da es aber für die Berechnung einer Schaltfunktion keine Rolle spielt, wie diese später technisch realisiert wird, ist sie auf Grund ihrer Anschaulichkeit für alle Lernenden die erste Grundlage.

1950 wurden die Relaischalter von Halbleitern (Transistortechnik) abgelöst. Diese werden auch als rein elektronische Schaltungen bezeichnet, da durch Anlegen einer Gleichspannung von typischerweise 0,6 - 0,7 V (der sogenannten Schwellspannung) das Halbleitermaterial (meist Silizium) des Transistors leitfähig wird.⁹

Durch Kombination von mehreren Halbleitern wurde es wenig später sogar möglich, einige Logik-Funktionen gemeinsam auf einer IC-Platte zu realisieren. IC steht dabei für „integrated circuit“, oder auf deutsch integrierte Schaltungen und ist nichts anderes, als eine Halbleiterplatte, meist aus Silizium.

Mittlerweile hat sich die Technik schon so sehr weiterentwickelt, dass schon zig-Millionen Funktionen auf einer einzigen IC-Platte Platz haben. Dadurch wurde der Platzbedarf von Schaltkreisen erheblich minimiert.

Auch wenn es viele verschiedene Möglichkeiten gibt, um Schaltungen zu realisieren, ist dabei immer wichtig, dass es eine eindeutige Zuordnung zweier Zustände einer (physikalischen) Größe zu den Werten 0 und 1 gibt.¹⁰

	(physikalische Größe)	Logik-Wert 0	Logik-Wert 1
Kontaktschalter	optisch	Lampe aus	Lampe an
Relaischalter	Strom	leitet	leitet nicht
Transistor	Spannung	0 V	groß (0,6-0,7 V)

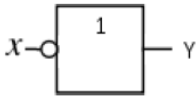
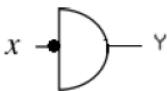
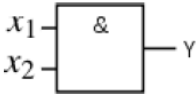
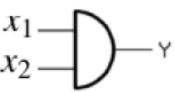
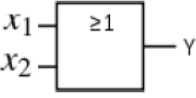
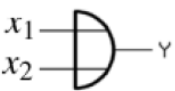
Dies ist allerdings nicht so einfach, da die Signale - anders als in der Theorie vorausgesetzt - normalerweise nicht binär sind. Betrachten wir einen Transistor. Wird keine Spannung angelegt, so ordnet man ihm den Wert 0 zu. Erst ab einer gewissen Spannung beginnt der Transistor leitfähig zu werden und ab 0,6-0,7 V wird ihm der Wert 1 zugeordnet. Dazwischen gibt es aber einen Bereich, in dem nicht entschieden werden kann, welcher Wert zugeordnet werden soll. Daher müssen die Unterschiede in der Spannung groß genug gewählt werden, sodass eine eindeutige Zuordnung möglich ist.

⁹Die genaue chemische Funktion führt in dieser Arbeit zu weit. Interessierte Leser mögen in [21], S. 55ff nachlesen.

¹⁰Tabelle vgl. [2], S. 8

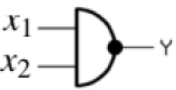
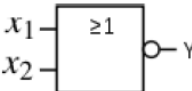
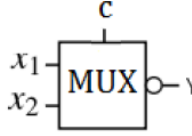
4.5.1 Gatterdarstellung

Auch die bisherige Darstellung unserer Schaltkreise entsprach den heute nicht mehr verwendeten Kontaktschaltern. Bei Funktionen, die mit einem IC realisiert werden, wird ein Schalter „gate“ (Gatter) genannt und es werden eigene Symbole verwendet. Auch hier sind verschiedene Symbole in Verwendung. Ich werde mich auf die heute theoretisch allgemein gültigen, bis jetzt aber noch nicht sehr verbreiteten Symbole nach der IEC-Norm und die alte, dafür aber sehr häufig verwendeten DIN-Norm beschränken.¹¹ Da die Definition der Schaltalgebra nur die Verknüpfungen $\cdot$, $+$ und $-$ beinhaltet, würden uns theoretisch auch das NOT-, das AND- und das OR-Gatter für alle möglichen Realisierungen reichen.

Funktion	IEC-Norm	DIN-Norm	Wahrheitstabelle																
NOT-Gatter (NICHT)			<table><tr><th>x</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	Y	0	1	1	0	$Y = \bar{x}$									
x	Y																		
0	1																		
1	0																		
AND-Gatter (UND)			<table><tr><th>x₁</th><th>x₂</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = x_1 \cdot x_2$
x ₁	x ₂	Y																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR-Gatter (ODER)			<table><tr><th>x₁</th><th>x₂</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x ₁	x ₂	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = x_1 + x_2$
x ₁	x ₂	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	

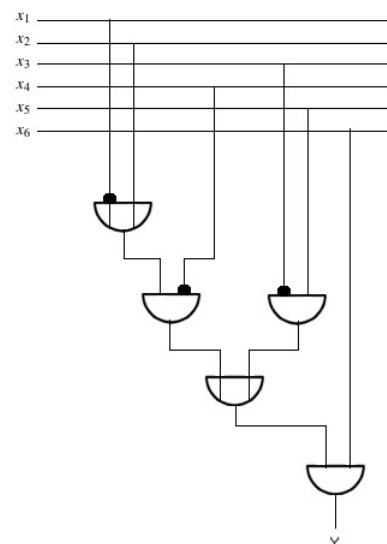
Da aber mit Hilfe von weiteren Gattern Schaltnetze mit weniger Bauteilen gebaut werden können, die daher auch billiger sind, werden noch zahlreiche weitere Schaltungen verwendet. Drei sehr wichtige Beispiele sind in der Tabelle angeführt.

¹¹Symbole leicht verändert von Tobulax, <https://de.wikipedia.org/wiki/Logikgatter>, 25.01.17

Funktion	IEC-Norm	DIN-Norm	Wahrheitstabelle																																					
NAND-Gatter (NICHT UND)			<table><tr><th>x_1</th><th>x_2</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_2	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{x_1 + x_2}$																					
x_1	x_2	Y																																						
0	0	1																																						
0	1	1																																						
1	0	1																																						
1	1	0																																						
NOR-Gatter (NICHT ODER)			<table><tr><th>x_1</th><th>x_2</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x_1	x_2	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{x_1 \cdot x_2}$																					
x_1	x_2	Y																																						
0	0	1																																						
0	1	0																																						
1	0	0																																						
1	1	0																																						
Multiplexer (WENN-DANN-SONST)			<table><tr><th>c</th><th>x_1</th><th>x_2</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	c	x_1	x_2	Y	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	1	1	1	$Y = x_1 \cdot c + x_2 \cdot \bar{c}$ das heißt $Y = \begin{cases} x_1 & \text{für } c = 1 \\ x_2 & \text{für } c = 0 \end{cases}$
c	x_1	x_2	Y																																					
0	0	0	0																																					
0	0	1	1																																					
0	1	0	0																																					
0	1	1	1																																					
1	0	0	0																																					
1	0	1	0																																					
1	1	0	1																																					
1	1	1	1																																					

Beispiel 4.5.1. Der Schaltkreis zur Funktion $f = ((\bar{x}_1 + x_2) \bar{x}_4 + \bar{x}_3 x_5) x_6$ aus dem Beispiel 4.4.2 kann daher auch mit Hilfe von Gattern dargestellt werden.

Auf den ersten Blick sieht dies nun komplizierter aus, als die Darstellung als Schaltkreis. Wird die Anzahl der Variablen und/oder Gatter allerdings erhöht, so ist die Gatterdarstellung wesentlich einfacher.



Verallgemeinerung der Gatter auf mehrere Eingangsvariablen

Bis jetzt haben wir nur Gatter betrachtet, die zu einer oder zwei Eingangsvariablen eine Ausgangsvariable liefern.

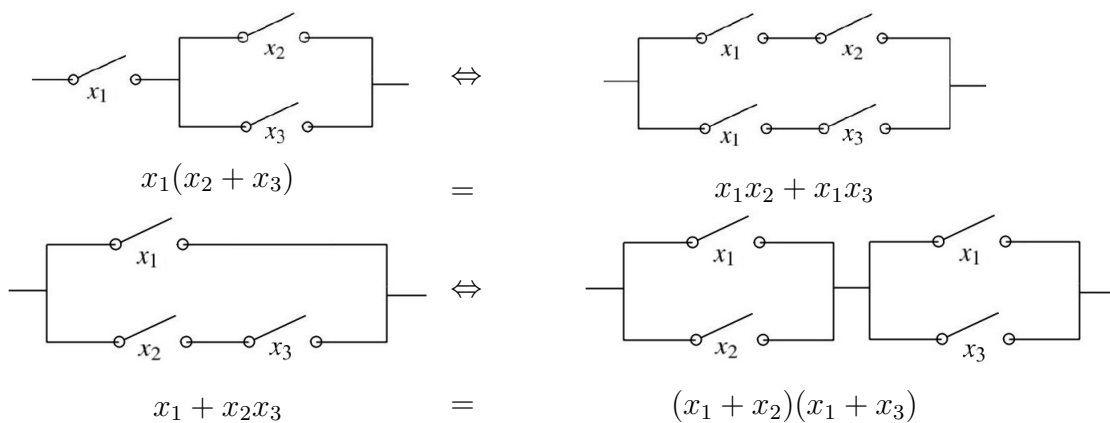
Die Gatter können allerdings noch dahingehend erweitert werden, dass zusätzliche Eingangsvariablen möglich sind. Für die ODER- und die UND-Verknüpfung sieht dies wie folgt aus:¹²



4.6 Illustrationen einiger Gesetze der Schaltalgebra

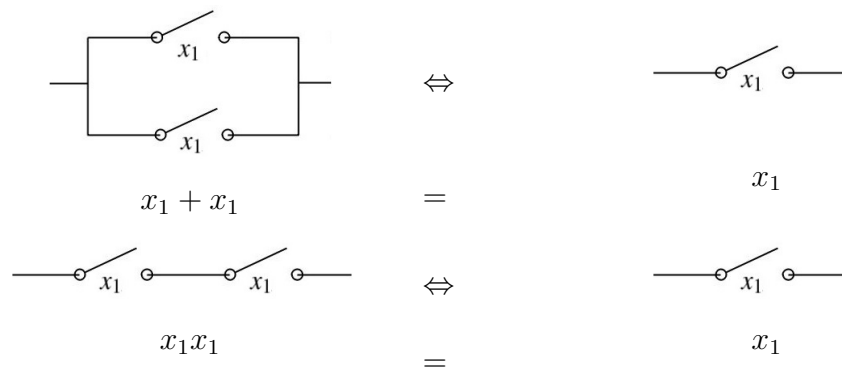
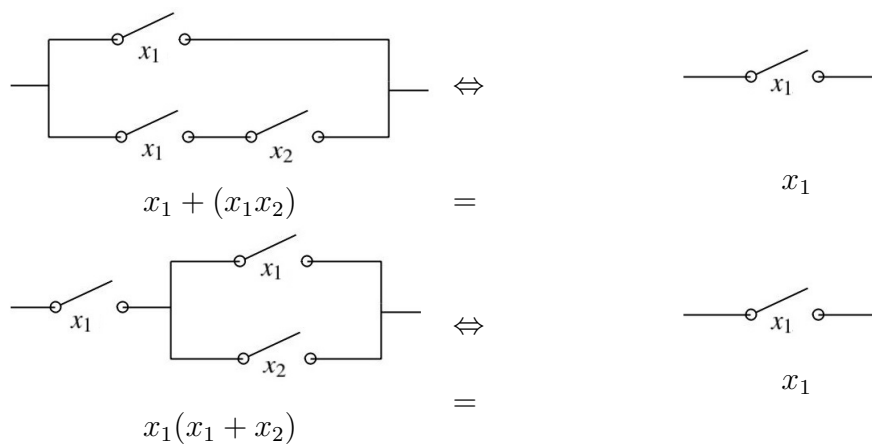
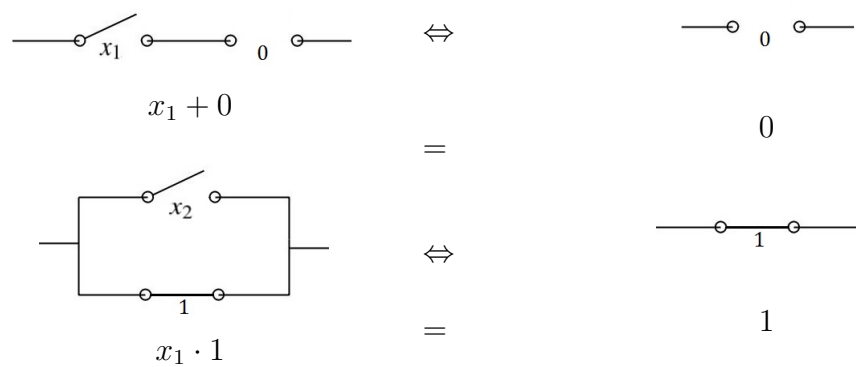
Zum Abschluss dieses Kapitels wollen wir nun noch einige Gesetze der Boole'schen Algebra mit Hilfe von Schaltkreisen (ohne Gatterdarstellung) demonstrieren, da diese dadurch sehr anschaulich werden.¹³

Distributivgesetze



¹²vgl. [6], S. 11

¹³s. [10], S. 136

Idempotenzgesetze**Verschmelzungsgesetze****Absorptionsgesetze**

Kapitel 5

Entwicklung von Schaltnetzen

Die Hauptschwierigkeit jeder Anwendung der Mathematik liegt im Übergang von dem konkreten, in Worten formulierten physikalischen oder technischen Problemen zu seiner mathematischen Fassung, [...].

Plechl, O., Duschek, A. in „Grundzüge einer Algebra der elektrischen Schaltungen“
nach [22], S. 175

Einfache Schaltnetze zu Problemen aus dem Alltag können oft durch Probieren oder simple Überlegungen realisiert werden. Soll die Funktion eines Schaltnetzes allerdings komplizierter und dabei auch noch möglichst kosteneffizient sein, so muss ein/e Techniker/in systematisch vorgehen. In der Literatur werden folgende Schritte beschrieben:

1. Problem erfassen und mit einer Wertetabelle darstellen (falls diese noch nicht gegeben ist).
2. Funktionsgleichung aufstellen.
3. Gleichung mit geeigneten Verfahren vereinfachen (um den Materialaufwand zu minimieren)
4. Bauteilauswahl und Realisierung des Schaltkreises.¹

Dieses Kapitel widmet sich den genannten Schritten in derselben Reihenfolge. Dafür dienten zahlreiche Quellen als Grundlage. 5.2 ist hauptsächlich an [10], S. 106-120 [21], S. 48, [3], S. 146, [4], S. 26 - 35 und [11], S. 70-74 angelehnt. Der Abschnitt 5.3 bezieht sich vor allem auf [3], S. 143-154 und [9], aber verwendet auch teilweise [6], S. 34 - 45 [21], S. 48ff, [23], S. 14f, [15] und [4], S. 15-17 und 36ff als Quellen.

¹vgl. [14], Folie 65

5.1 Problemerkfassung

Ein/e Techniker/in steht zu Beginn vor dem Problem, eine gewünschte Funktion für ein Bauteil oder eine Maschine zu realisieren. Dafür muss er/sie das richtige (am besten minimale) Schaltnetz finden. Ihm/ihr sind also meist die passenden Wahrheitswerte bekannt (z.B. "Wird der Schalter gedrückt, so brennt das Licht und der Wahrheitswert ist 1"), dafür kennt er/sie den zugehörigen Term oder die zugehörige Funktion nicht.

Er/sie muss sich daher zuerst die verschiedenen Bedingungen überlegen, die später realisiert werden sollen, und mit diesem Wissen eine Wertetabelle aufstellen. Falls diese bereits gegeben ist (wie bei den meisten Übungsaufgaben), fällt dieser Schritt natürlich weg.

Beispiel 5.1.1 (Alarmanlage). Es soll eine Alarmanlage so gebaut werden, dass der Alarm entweder ausgelöst wird ($f(x_1, x_2, x_3) = 1$), wenn der Panikschalter gedrückt wird (also $x_1 = 1$), oder das Haus nicht sicher ist. Sicher bedeutet dabei, dass alle Türen und alle Fenster geschlossen sind (das heißt $x_2 = 0$ und $x_3 = 0$)).

Wir folgern: ist das Haus unsicher, so ist mindestens ein Fenster oder eine Tür geöffnet und daher entspricht jedenfalls x_2 oder x_3 dem Wert 1. Mit diesem Wissen können wir nun zum Beispiel eine Wahrheitstabelle aufstellen:

Panik (x_1)	Türen (x_2)	Fenster (x_3)	Alarm (f)
0	0	0	0
0	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	1

5.2 Funktionsgleichung aufstellen

Um die Funktion eines Beispiels - wie vorhin mit der Alarmanlage - herauszufinden, kann man entweder raten (was in diesem Fall nicht sehr schwierig wäre, aber meist praktisch unmöglich ist) oder man generiert systematisch einen ersten Entwurf einer Funktion.

Wie wir bereits wissen, sind Terme nicht eindeutig bestimmt. Daher gibt es auch für eine Funktion mehrere äquivalente Darstellungen.

Beispiel 5.2.1. Folgende Terme bestimmen die selbe Funktion wie man durch Termumformungen oder Berechnung der Funktionswerte für alle Belegungen leicht zeigen kann!

- $f_1(x_1, x_2, x_3) = x_1\overline{x_3} + (\overline{x_2\overline{x_3}})x_2$
- $f_2(x_1, x_2, x_3) = x_1\overline{x_3} + x_3x_2$

Wie erhält man aus einer Reihe von Bedingungen nun solche Funktionen? Dazu sind vor allem die sogenannten Normalformen von Bedeutung. Diese ermöglichen es, direkt aus der Wertetabelle einen Funktionsterm herauszulesen (auch wenn dieser nicht immer der einfachste ist).

5.2.1 Vollkonjunktionen und Volldisjunktionen

Als Grundlage für die Normalformen betrachten wir nun zuerst die sogenannten Vollkonjunktionen (Minterme) und Volldisjunktionen (Maxterme):

Vollkonjunktion

Definition 5.2.2 (Vollkonjunktion). Ein Term $k_s^{(n)} = h_1(x_1) \cdot h_2(x_2) \cdot \dots \cdot h_n(x_n)$

mit $h_i(x_i) \equiv \begin{cases} x_j \\ \overline{x_j} \end{cases}$ oder $s \in \{0, 2^n - 1\}$ heißt Vollkonjunktion (oder Minterm)

in den Variablen $x_1, \dots, x_n$.²

Sei K_n die Menge aller Vollkonjunktionen in n Variablen. Da $h_i(x_i)$ entweder x_i oder $\overline{x_i}$ sein kann, können für K_n genau 2^n mögliche Vollkonjunktionen gefunden werden.

Beispiel 5.2.3.

$$K_1 = \left\{ \underbrace{x_1}_{k_1^{(1)}}, \underbrace{\overline{x_1}}_{k_0^{(1)}} \right\}$$

$$K_2 = \left\{ \underbrace{x_1x_2}_{k_3^{(2)}}, \underbrace{x_1\overline{x_2}}_{k_2^{(2)}}, \underbrace{\overline{x_1}x_2}_{k_1^{(2)}}, \underbrace{\overline{x_1}\overline{x_2}}_{k_0^{(2)}} \right\}$$

$$K_3 = \left\{ \underbrace{x_1x_2x_3}_{k_7^{(3)}}, \underbrace{x_1x_2\overline{x_3}}_{k_6^{(3)}}, \underbrace{x_1\overline{x_2}x_3}_{k_5^{(3)}}, \underbrace{x_1\overline{x_2}\overline{x_3}}_{k_4^{(3)}}, \underbrace{\overline{x_1}x_2x_3}_{k_3^{(3)}}, \underbrace{\overline{x_1}x_2\overline{x_3}}_{k_2^{(3)}}, \underbrace{\overline{x_1}\overline{x_2}x_3}_{k_1^{(3)}}, \underbrace{\overline{x_1}\overline{x_2}\overline{x_3}}_{k_0^{(3)}} \right\}$$

²vgl. [10], S. 106. Das Zeichen $\equiv$ bedeutet „steht für“. Es soll die Termgleichheit zeigen.

u.s.w.

Bemerkung 5.2.4. Die Nummerierung der Vollkonjunktionen erhält man, indem man jedes x_i durch eine 1 und jedes $\overline{x_i}$ durch eine 0 ersetzt. Wandelt man diese Dualzahlen nun in Dezimalzahlen um, so entstehen die Zahlen von 0 bis $2^n - 1$.

Beispiel 5.2.5. Betrachten wir eine Vollkonjunktion der Menge K_3 :

$$\overline{x_1}x_2\overline{x_3} \quad \hat{=} \quad 010 \quad \hat{=} \quad 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

Daher wählen wir für diese Vollkonjunktion in 3 Variablen die Bezeichnung $k_2^{(3)}$.

Volldisjunktion

Definition 5.2.6 (Volldisjunktion). Ein Term der Gestalt $h_1(x_1) + h_2(x_2) + \dots +$

$$h_n(x_n) \text{ mit } h_i(x_i) \equiv \begin{cases} x_j \\ \text{oder} \\ \overline{x_j} \end{cases} \quad \text{heißt Volldisjunktion (oder Maxterm) in den n Va-}$$

riablen $x_1, \dots, x_n$.

D_n sei die Menge aller Volldisjunktionen in n Variablen. Auch hier gibt es davon genau 2^n verschieden.³

Beispiel 5.2.7.

$$\begin{aligned} D_1 &= \left\{ \underbrace{x_1}_{d_1^{(1)}}, \underbrace{\overline{x_1}}_{d_0^{(1)}} \right\} \\ D_2 &= \left\{ \underbrace{x_1 + x_2}_{d_3^{(2)}}, \underbrace{x_1 + \overline{x_2}}_{d_2^{(2)}}, \underbrace{\overline{x_1} + x_2}_{d_1^{(2)}}, \underbrace{\overline{x_1} + \overline{x_2}}_{d_0^{(2)}} \right\} \\ D_3 &= \left\{ \underbrace{x_1 + x_2 + x_3}_{d_7^{(3)}}, \underbrace{x_1 + x_2 + \overline{x_3}}_{d_6^{(3)}}, \underbrace{x_1 + \overline{x_2} + x_3}_{d_5^{(3)}}, \underbrace{x_1 + \overline{x_2} + \overline{x_3}}_{d_4^{(3)}}, \right. \\ &\quad \left. \underbrace{\overline{x_1} + x_2 + x_3}_{d_3^{(3)}}, \underbrace{\overline{x_1} + x_2 + \overline{x_3}}_{d_2^{(3)}}, \underbrace{\overline{x_1} + \overline{x_2} + x_3}_{d_1^{(3)}}, \underbrace{\overline{x_1} + \overline{x_2} + \overline{x_3}}_{d_0^{(3)}} \right\} \end{aligned}$$

u.s.w.

Bemerkung 5.2.8. Die Indizierung erfolgt ebenfalls wie bei den Vollkonjunktionen. Man kann den Volldisjunktionen auch hier die Indizes 0 bis $2^n - 1$ zuordnen.

Beispiel 5.2.9. Betrachten wir eine Volldisjunktion der Menge K_3 :

$$\overline{x_1} + x_2 + \overline{x_3} \quad \hat{=} \quad 010 \quad \hat{=} \quad 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

Analog zur Vollkonjunktion wählen wir daher für diese Volldisjunktion in 3 Variablen die Bezeichnung $d_2^{(3)}$.

³vgl. [10], S. 116

Zusammenhang zwischen Vollkonjunktionen und Volldisjunktionen

Mit Hilfe der erweiterten Regeln von DeMorgan aus Satz 3.3.4 können wir folgenden Satz beweisen:

Satz 5.2.10 *Bildet man zu einer Vollkonjunktion $k_i^{(n)}$ das Komplement, so erhält man die Volldisjunktion $d_{2^n-1-i}^{(n)}$. Es gilt: $\overline{k_i^{(n)}} = d_{2^n-1-i}^{(n)}$.*

Beweisskizze Es sei $k_i \in K_s$. Auf Grund von Satz 3.3.4 ist $\overline{k_i^{(n)}} = d_j^{(n)}$ wobei $d_j^{(n)} \in D_s$. Sei z.B.:

$$\begin{aligned} i &\hat{=} 1101 \dots 101 \dots 1 \\ \text{so ist} \\ j &\hat{=} 0010 \dots 010 \dots 0 \\ \text{daher folgt mit Hilfe von Induktion:} \\ i + j &\hat{=} 1111 \dots 111 \dots 1 \hat{=} 2^{n-1} + \dots + 2^1 + 2^0 = 2^n - 1 \\ \text{es folgt:} \\ j &= 2^n - 1 - i \end{aligned}$$

Beispiel 5.2.11. Wir betrachten eine Vollkonjunktion aus K_4 (also mit 4 Variablen): $\textcolor{red}{11} = 2^4 - 1 - \textcolor{green}{4}$ und

$$\begin{aligned} \overline{k_{\textcolor{green}{4}}^{(4)}} &\hat{=} \overline{x_1 x_2 \overline{x_3} \overline{x_4}} &\hat{=} x_1 + \overline{x_2} + x_3 + x_4 &\hat{=} d_{\textcolor{red}{11}}^{(4)} \\ \textcolor{green}{4} &= 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 &1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 &= \textcolor{red}{11} \end{aligned}$$

Für die spätere Anwendung beim Quine-McCluskey-Verfahren definieren wir hier auch noch die Begriffe Implikant und Primimplikant.

Der verwandte Begriff „Implikation“ wurde schon in Kapitel 2 verwendet und bedeutet, dass aus einer Aussage schon eine zweite Aussage folgt. Bei der Implikante ist es ähnlich. Man sagt, eine Funktion f impliziert die Funktion g , wenn f für alle Belegungen, an denen g den Wert 1 annimmt, ebenfalls 1 ist. Abgekürzt kann man schreiben: $g \rightarrow f = 1$.⁴

Definition 5.2.12 (Implikant). Sei f eine binäre Funktion und i einer ihrer Konjunktionsterme. Wenn $i \rightarrow f = 1$ gilt, so nennt man i einen Implikanten der Funktion f .

Kann kein kleinerer Term gefunden werden, der die Implikante selbst, und damit auch die Ausgangsfunktion impliziert, so nennt man diese Primimplikante.

⁴vgl. [4], S. 36ff

Definition 5.2.13. Eine Konjunktion mit der minimalen Anzahl an Variablen, die eine Ausgangsfunktion impliziert, heißt Primterm oder Primimplikante.⁵

Beispiel 5.2.14.

x_1	x_2	$f(x_1, x_2)$	Minterm
0	0	1	$\overline{x_1} \overline{x_2}$
0	1	1	$\overline{x_1} x_2$
1	0	0	$x_1 \overline{x_2}$
1	1	1	$x_1 x_2$

In diesem Fall wären zum Beispiel alle Minterme, bis auf $x_1 \overline{x_2}$ Implikanten. Da ein Minterm der $\overline{x_1}$ enthält, immer auf 1 abbildet, unabhängig davon, ob er mit x_2 oder $\overline{x_2}$ verknüpft ist, ist $\overline{x_1}$ ein Primterm.

5.2.2 Disjunktive Normalform (DNF)

Definition 5.2.15. Ein Term, der aus der Disjunktion von Konjunktionstermen mit jeweils paarweise verschiedenen Variablen besteht, heißt disjunktive Normalform (DNF).⁶

Beispiel 5.2.16.

$$f(x_1, x_2, x_3) = \underbrace{x_2 x_3}_{\text{Konjunktion}} + \underbrace{x_1 \overline{x_2} x_3}_{\text{(Voll-)Konjunktion}} + \underbrace{\overline{x_1} \overline{x_2}}_{\text{Konjunktion}}$$

disjunkte Verknüpfung der Konjunktionen

Da die DNF aber im Allgemeinen nicht eindeutig bestimmt ist, kommt es zu Schwierigkeiten beim Vergleich von Funktionen. So wird gerne eine kanonische DNF verwendet.

Kanonische disjunktive Normalform (KDNF)

Definition 5.2.17. Die Erweiterung der Terme einer DNF auf vollständige n-stelligen Terme heißt kanonische disjunktive Normalform (KDNF). Jeder der vorhandenen Konjunktionsterme enthält bei der KDNF also alle n Variablen und jeder Term tritt nur einmal auf.⁷

Beispiel 5.2.18.

$$f(x_1, x_2, x_3) = \underbrace{x_1 x_2 x_3}_{\text{Vollkonjunktion}} + \underbrace{x_1 \overline{x_2} x_3}_{\text{Vollkonjunktion}} + \underbrace{\overline{x_1} \overline{x_2} x_3}_{\text{Vollkonjunktion}}$$

disjunktive Verknüpfung der Vollkonjunktionen

⁵vgl. [21] S. 48

⁶Der englische Name „sum of products“ (SOP) ist verständlicher aber im deutschen Sprachraum nicht sehr gebräuchlich.

⁷[4], S. 27

Satz 5.2.19 Jede binäre Funktion f in n Variablen besitzt eine eindeutig bestimmte Darstellung als Summe (Disjunktion) von Vollkonjunktionen.⁸

$$f(x_1, x_2, \dots, x_n) = \sum_{s=0}^{2^n-1} i_s k_s^{(n)} \quad \text{wobei} \quad i_s = \begin{cases} 0 \\ 1 \end{cases}$$

Bevor wir diesen Satz beweisen, betrachten wir noch ein Beispiel, um die Vorgehensweise des Beweises besser verstehen zu können:

Beispiel 5.2.20.

$$\begin{aligned} f(x_1, x_2, x_3) &= x_1 \bar{x}_2 + \overline{(x_1 \bar{x}_2 x_3 + \bar{x}_1)} \\ &= x_1 \bar{x}_2 + \overline{(x_1 \bar{x}_2 x_3)} x_1 && \text{DeMorgan} \\ &= x_1 \bar{x}_2 + (\bar{x}_1 + x_2 + \bar{x}_3) x_1 && \text{DeMorgan} \\ &= x_1 \bar{x}_2 + \bar{x}_1 x_1 + x_2 x_1 + \bar{x}_3 x_1 && \text{Distributivgesetz} \\ &= x_1 \bar{x}_2 + x_2 x_1 + \bar{x}_3 x_1 && \text{Komplementgesetz} \\ &= x_1 \bar{x}_2 (x_3 + \bar{x}_3) + x_2 x_1 (x_3 + \bar{x}_3) + \bar{x}_3 x_1 (x_2 + \bar{x}_2) && \text{Komplementgesetz} \\ &= x_1 \bar{x}_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 + \cancel{x_1 x_2 \bar{x}_3} + \cancel{x_1 \bar{x}_2 \bar{x}_3} && \text{Distributivgesetz} \\ &= \underbrace{x_1 x_2 x_3}_{k_7^{(0)}} + \underbrace{x_1 x_2 \bar{x}_3}_{k_6^{(0)}} + \underbrace{x_1 \bar{x}_2 x_3}_{k_5^{(3)}} + \underbrace{x_1 \bar{x}_2 \bar{x}_3}_{k_4^{(3)}} && \text{Idempotenzgesetz} \end{aligned}$$

Beweis. **Existenz**

1. Indem man die DeMorgan'schen Regeln mehrfach anwendet, kann man erreichen, dass Komplement-Striche nur noch über einzelnen Variablen stehen.
2. Wendet man nun die Distributiv-Gesetze an, so kann man alle Klammern auflösen.
3. Sollten dabei Potenzen aufgetreten sein, so kann man diese aufgrund des Idempotenzgesetzes ($a \cdot a = a$) reduzieren.
Summen, die sowohl x_i als auch $\bar{x}_i$ enthalten kann man wegen dem Komplementgesetz streichen ($a \cdot \bar{a} = 0$)
Da das Idempotenzgesetz ($a + a = a$) gilt, muss ein mehrfach vorkommender Summand nur einmal geschrieben werden.
4. Nun ergänzt man alle Terme, die weder x_i noch $\bar{x}_i$ enthalten zu Mintermen indem man sich die Komplementregel zu Hilfe nimmt. Da $x_i + \bar{x}_i = 1$ kann man einfach mit diesem Term multiplizieren und erhält nach dem Ausmultiplizieren schließlich tatsächlich eine Summe von Vollkonjunktionen.
5. Zum Schluss kann man auch hier das Idempotenzgesetz dazu verwenden, mehrfach vorkommende Terme zu streichen.

⁸Satz und Beweis, vgl. [10], S. 108f.

Eindeutigkeit Wir nehmen an, dass es für dieselbe Funktion f zwei verschiedene Darstellungen als Summe von Vollkonjunktionen gibt.

$$f(x_1, x_2, \dots, x_n) = \sum_{s=0}^{2^n-1} i'_s k_s^{(n)} = \sum_{s=0}^{2^n-1} i''_s k_s^{(n)}$$

Betrachten wir nun, was passiert, wenn wir zwei Terme der Menge K_s verknüpfen, also $k_p^{(n)} k_l^{(n)}$:

1.Fall: $p = l$

Auf Grund des Identitätsgesetz ist $k_p^{(n)} k_l^{(n)} = k_p^{(n)}$

2.Fall: $p \neq l$

Da es sich sowohl bei k_p als auch bei k_l um Vollkonjunktionen handelt und diese verschieden sind, tritt in der Verknüpfung mindestens eine Variable x_i gemeinsam mit ihrem Komplement $\bar{x}_i$ auf. Daher ist das Produkt auf Grund des Komplementgesetzes sicher 0.

Zusammengefasst:

$$k_p^{(n)} k_l^{(n)} = \begin{cases} k_p^{(n)} & \text{für } p = l \\ 0 & \text{für } p \neq l \end{cases}$$

O.b.d.A. dürfen wir aber nun annehmen, dass eine Vollkonjunktion k_m , die in der ersten KDNF vorkommt, in der zweiten KDNF nicht auftritt.

Auf Grund der vorher betrachteten Fallunterscheidung folgt nach Konjunktion vom k_m auf beiden Seiten:

$$\begin{aligned} \sum_{s=0}^{2^n-1} i'_s k_s^{(n)} &= \sum_{s=0}^{2^n-1} i''_s k_s^{(n)} \\ \sum_{s=0}^{2^n-1} i'_s k_s^{(n)} \cdot k_m &= \sum_{s=0}^{2^n-1} i''_s k_s^{(n)} \cdot k_m \\ k_m^{(n)} &= 0 \end{aligned}$$

Das ist allerdings ein Widerspruch, da $k_m^{(n)}$ nicht äquivalent zu 0 sein kann. Daher muss die DNF eindeutig sein. □

Aus Satz 5.2.19 folgt direkt:

Folgerung 5.2.21 *Zwei binäre Funktionen f und g , die dieselbe kanonische disjunktive Normalform haben, sind ident.*

Dies ermöglicht nun, zwei verschiedene Funktionen auf ihre mögliche Identität zu testen, indem man zu beiden Funktionen ihre KDNF aufstellt und vergleicht.

Aufstellen einer kanonischen disjunktiven Normalform aus der Wertetabelle

Satz 5.2.22 Eine binäre Funktion in n Variablen ist vollständig bestimmt, wenn die Stellen gegeben sind, an denen sie den Wert 1 annimmt.

Beweis. Satz 5.2.19 besagt, dass es für jede Funktion eine KDNF der Form

$$f(x_1, x_2, \dots, x_n) = \sum_{s=0}^{2^n-1} i_s k_s^{(n)} \quad \text{mit} \quad i_s = \begin{cases} 0 \\ 1 \end{cases} \quad \text{gibt.}$$

Da i_s nur 0 oder 1 sein kann, reicht es, die Stellen zu kennen, an denen die Funktion den Wert 1 annimmt. $\square$

Um eine KDNF zu erhalten, brauchen wir daher nur alle Belegungen zu betrachten, die den Wert 1 annehmen, und die zu diesen Belegungen gehörigen Vollkonjunktionen (Minterme) disjunktiv zu verknüpfen. Wir zeigen dies am Beispiel 5.1.1 von Seite 40:

Beispiel 5.2.23.

Panik (x_1)	Türen (x_2)	Fenster (x_3)	Alarm (f)	zugehöriger Minterm
0	0	0	0	$\overline{x_1} \overline{x_2} \overline{x_3}$
0	0	1	1	$\overline{x_1} \overline{x_2} x_3$
0	1	0	1	$\overline{x_1} x_2 \overline{x_3}$
0	1	1	1	$\overline{x_1} x_2 x_3$
1	0	0	1	$x_1 \overline{x_2} \overline{x_3}$
1	0	1	1	$x_1 \overline{x_2} x_3$
1	1	0	1	$x_1 x_2 \overline{x_3}$
1	1	1	1	$x_1 x_2 x_3$

Die KDNF wird durch Disjunktion aller Minterme, deren Wert 1 ist, aufgestellt. In unserem Fall:

$$f(x_1, x_2, x_3) = x_1 x_2 x_3 + x_1 x_2 \overline{x_3} + x_1 \overline{x_2} x_3 + x_1 \overline{x_2} \overline{x_3} + \overline{x_1} x_2 x_3 + \overline{x_1} x_2 \overline{x_3} + \overline{x_1} \overline{x_2} x_3$$

Da im Fall dieser Funktion aber nur eine Belegung nicht den Wert 1 annimmt stellt sich die Frage, ob man mit dieser Information nicht eine einfachere Funktion erhalten könnte.

5.2.3 Konjunktiven Normalform (KNF)

Definition 5.2.24. Ein Term, der aus der Konjunktion von Disjunktionstermen mit jeweils paarweise verschiedenen Variablen besteht, heißt konjunktive Normalform (KNF).⁹

z.B.

$$f(x_1, x_2, x_3) = \underbrace{(x_2 + x_3)}_{\text{Disjunktion}} \cdot \underbrace{(x_1 + \overline{x_2} + x_3)}_{\text{(Voll-)Disjunktion}} \cdot \underbrace{(\overline{x_1} + \overline{x_2})}_{\text{Disjunktion}}$$

konjunktive Verknüpfung der Disjunktionen

Bemerkung 5.2.25. Auch die KNF einer Funktion ist nicht eindeutig.

Kanonische Konjunkte Normalform (KKNF)

Gehen wir einen Schritt zurück, und betrachten das Komplement der KDNF unserer Alarmanlage:

Panik (x_1)	Türen (x_2)	Fenster (x_3)	Alarm (f)	kein Alarm ($\overline{f}$)
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Die KDNF von $\overline{f}$ ist also $\overline{f} = \overline{x_1} \overline{x_2} \overline{x_3} = k_0^{(3)}$.

Durch erneutes Komplementieren, folgt eine - in diesem Fall - wesentlich einfachere Darstellung von f : $f = \overline{\overline{x_1} \overline{x_2} \overline{x_3}} = x_1 + x_2 + x_3 = d_7^{(3)}$

Wir folgern allgemein:

Mit Hilfe von Satz 5.2.19 kann man das Komplement der KDNF der Boole'schen Funktion f bilden. Alle Konjunktionen, die vorher auf 1 abgebildet haben, bilden im Komplement nun auf 0 ab.

$$f(x_1, x_2, \dots, x_n) = \sum_{s=0}^{2^n-1} i_s k_s^{(n)}$$

$$\overline{f(x_1, x_2, \dots, x_n)} = \sum_{s=0}^{2^n-1} \overline{i_s} k_s^{(n)}$$

⁹Der englische Name „product of sums“ (POS) ist verständlicher aber im deutschen Sprachraum nicht sehr gebräuchlich.

Durch erneute Komplementbildung folgt nach Satz 3.3.4 auf S. 25 und Satz 5.2.10 auf S. 43:

$$f(x_1, x_2, \dots, x_n) = \overline{\sum_{s=0}^{2^n-1} \bar{i}_s k_s^{(n)}} = \prod_{s=0}^{2^n-1} \overline{\bar{i}_s k_s^{(n)}} = \prod_{s=0}^{2^n-1} (i_s + \overline{k_s^{(n)}}) = \prod_{s=0}^{2^n-1} (i_s + d_{2^n-1-s}^{(n)})$$

Wir betrachten die Fälle $i_s = 1$ und $i_s = 0$:

$$\left\{ \begin{array}{ll} i_s = 1 & 1 + d_{2^n-1-s}^{(n)} = 1 \\ & \text{Dieser Faktor kann aufgrund } 1 \cdot a = a \text{ weggelassen werden.} \\ i_s = 0 & 0 + d_{2^n-1-s}^{(n)} = d_{2^n-1-s}^{(n)} \\ & \text{Dieser Faktor entspricht der Volldisjunktion, deren Wert 0 ist.} \end{array} \right.$$

f hat also nicht nur eine Darstellung als Disjunktion (Summe) von Vollkonjunktionen (Produkten) sondern auch als Konjunktion (Produkt) von Volldisjunktionen (Summen).

Definition 5.2.26. Die Erweiterung der Terme einer KNF auf vollständige n -stelligen Terme heißt kanonische konjunktive Normalform (KKNF). Jeder der vorhandenen Disjunktionsterme enthält bei der KKNF also alle n Variablen und jeder Term tritt nur einmal auf.

Beispiel 5.2.27.

$$f(x_1, x_2, x_3) = \underbrace{(x_1 + x_2 + x_3)}_{\text{Volldisjunktion}} \cdot \underbrace{(x_1 + \bar{x}_2 + x_3)}_{\text{Volldisjunktion}} \cdot \underbrace{(\bar{x}_1 + \bar{x}_2 + x_3)}_{\text{Volldisjunktion}}$$

konjunktive Verknüpfung der Volldisjunktionen

Satz 5.2.28 Jede binäre Funktion f in n Variablen besitzt eine eindeutig bestimmte Darstellung als Produkt (Konjunktion) von Volldisjunktionen.¹⁰

$$f(x_1, x_2, \dots, x_n) = \prod_{s=0}^{2^n-1} (i_s + d_{2^n-1-s}^{(n)}) \quad \text{wobei} \quad i_s = \begin{cases} 0 \\ 1 \end{cases}$$

Beweis.

Existenz Die Existenz folgt aus der Herleitung der KKNF aus der KDNF, da die Existenz einer KDNF für jede Funktion bereits in Satz 5.2.19 bewiesen wurde.

Eindeutigkeit Mit der selben Methode, wie wir aus der kanonischen disjunktiven Normalform die kanonische konjunktive Normalform gewonnen haben, können wir nun auch die Umkehrung durchführen:

¹⁰analog zu Satz 5.2.19, zweiter Teil des Beweises vgl. [10], S. 117f

$$\begin{aligned}
f(x_1, \dots, x_n) &= \prod_{s=0}^{2^n-1} (i_s + d_{2^n-1-s}^{(n)}) \\
\overline{f(x_1, \dots, x_n)} &= \overline{\prod_{s=0}^{2^n-1} (i_s + d_{2^n-1-s}^{(n)})} = \sum_{s=0}^{2^n-1} \overline{(i_s + d_{2^n-1-s}^{(n)})} = \\
&= \sum_{s=0}^{2^n-1} \overline{i_s} \cdot \overline{d_{2^n-1-s}^{(n)}} = \sum_{s=0}^{2^n-1} \overline{i_s} k_s^{(n)}
\end{aligned}$$

Bildet man erneut das Komplement, so erhält man genau die KDNF.

$$\overline{\overline{f(x_1, \dots, x_n)}} = f(x_1, \dots, x_n) = \sum_{s=0}^{2^n-1} i_s k_s^{(n)}$$

Wir nehmen an, wir hätten zwei verschiedene KKNF der Boole'schen Funktion f gefunden. Dies hätte aber zur Folge, dass wir auch zwei verschiedene Darstellungen von f als KDNF finden können. Dies ist ein Widerspruch zu Satz 5.2.19. $\square$

Gewinnung der KKNF

Satz 5.2.29 *Eine binäre Funktion in n Variablen ist vollständig bestimmt, wenn die Stellen gegeben sind, an denen die Funktion den Wert 0 annimmt.*

Beweis. Analog zu Satz 5.2.22 auf S. 47. $\square$

Mit Hilfe dieses Satzes wollen wir nun Methoden zur Gewinnung der KKNF betrachten. Als Beispiel verwenden wir:

$$\begin{aligned}
f(x_1, x_2, x_3) &= x_1 x_2 x_3 + x_1 \overline{x_2} \overline{x_3} + \overline{x_1} x_2 x_3 + \overline{x_1} \overline{x_2} x_3 + \overline{x_1} \overline{x_2} \overline{x_3} \\
&= k_7^{(3)} + k_4^{(3)} + k_3^{(3)} + k_1^{(3)} + k_0^{(3)}
\end{aligned}$$

- a) Gewinnung über die KDNF** Man kann von einer KDNF direkt die KKNF bestimmen, indem man das Komplement mit Hilfe der Vollkonjunktionen bildet und das Ergebnis mit den DeMorgan Regeln noch einmal komplementiert.

$$\begin{aligned}
\overline{f} &= \overline{k_7^{(3)} + k_5^{(3)} + k_2^{(3)}} \\
f &= \overline{\overline{k_6^{(3)} + k_5^{(3)} + k_2^{(3)}}} = \overline{k_6^{(3)}} \cdot \overline{k_5^{(3)}} \cdot \overline{k_2^{(3)}} = d_1^{(3)} \cdot d_2^{(3)} \cdot d_5^{(3)}
\end{aligned}$$

- b) Direkte Bestimmung aus der Wertetabelle**

Wir können die KKNF direkt ablesen, indem wir die Disjunktion aller Maxterme der Zeilen, deren Belegung 0 ist, bilden.

x_1	x_2	x_3	Minterm	Maxterm	f
0	0	0	$k_0^{(3)}$	$d_0^{(3)}$	1
0	0	1	$k_1^{(3)}$	$d_1^{(3)}$	1
0	1	0	$k_2^{(3)}$	$d_2^{(3)}$	0
0	1	1	$k_3^{(3)}$	$d_3^{(3)}$	1
1	0	0	$k_4^{(3)}$	$d_4^{(3)}$	1
1	0	1	$k_5^{(3)}$	$d_5^{(3)}$	0
1	1	0	$k_6^{(3)}$	$d_6^{(3)}$	0
1	1	1	$k_7^{(3)}$	$d_7^{(3)}$	1

$$\begin{aligned}
 f &= d_2^{(3)} \cdot d_5^{(3)} \cdot d_6^{(3)} = \\
 &= (\overline{x_1} + x_2 + \overline{x_3}) \cdot (x_1 + \overline{x_2} + x_3) \cdot (x_1 + x_2 + \overline{x_3})
 \end{aligned}$$

5.2.4 Normalformen von partiell definierten Funktionen

Um die Normalformen von partiell definierten Funktionen aufstellen zu können, muss zuerst die Wertetabelle vervollständigt werden. Da es auch hier das Ziel ist, eine möglichst kurze Normalform zu erhalten, setzt man einen Don't-care-Zustand 0, wenn man eine KDNF aufstellen will, und 1, falls eine KKNF. Dies ist allerdings nur bei kanonischen Normalformen erlaubt.¹¹

Beispiel 5.2.30.

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	f_{KDNF}	f_{KKNF}
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	-	0	1
0	1	1	0	0	0
1	0	0	-	0	1
1	0	1	0	0	0
1	1	0	-	0	1
1	1	1	1	1	1

$$\begin{aligned}
 f_{\text{KDNF}} &= \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} \overline{x_2} x_3 + x_1 x_2 x_3 \\
 f_{\text{KKNF}} &= (\overline{x_1} + x_2 + x_3)(x_1 + \overline{x_2} + x_3)
 \end{aligned}$$

¹¹s. [4], S. 35

5.3 Minimierungsverfahren

Bei der Realisierung von verschiedenen Funktionen in der Praxis spielt auch die Kostenfrage eine wesentliche Rolle. Jeder Bauteil und jeder Schaltkontakt kostet extra. So ist es nicht überraschend, dass systematische Verfahren gesucht wurden, um die gewonnenen Normalformen (nicht nur rein algebraisch) so umzuformen, dass sie möglichst wenige Zeichen enthalten aber äquivalent sind.

Bevor wir Minimierungsverfahren kennenlernen, müssen wir zuerst definieren, was es bedeutet, dass eine binäre Funktion „einfacher“ ist, als eine andere.

Definition 5.3.1. Wir bezeichnen die Anzahl der Konjunktionsterme einer Funktion d mit $s(d)$ und die Anzahl der Variablen von d mit $v(d)$, wobei mehrfach vorkommende Variablen auch mehrfach zu beachten sind.

Definition 5.3.2. Seien d_1 und d_2 zwei Funktionen. d_1 ist einfacher als d_2 , wenn gilt:

$$s(d_1) < s(d_2) \text{ und } v(d_1) \leq v(d_2) \quad \text{oder}$$

$$s(d_1) \leq s(d_2) \text{ und } v(d_1) < v(d_2).$$

Beispiel 5.3.3.

- $d_1(x_1, x_2, x_3) = x_1x_3 + \overline{x_2} \overline{x_3}$ ist einfacher als $d_2(x_1, x_2, x_3) = x_1\overline{x_2}x_3 + \overline{x_2} \overline{x_3}$, da zwar beide Funktionen aus je zwei Konjunktionstermen bestehen, aber d_1 weniger Variablen enthält.
- Bei den Funktionen $d_1(x_1, x_2, x_3) = x_1x_3 + \overline{x_2} \overline{x_3}$ und $d_2(x_1, x_2, x_3) = x_1 + x_2 + \overline{x_3}$ kann man hingegen nicht feststellen, welche einfacher ist, da $s(d_1) < s(d_2)$, aber dafür $v(d_1) > v(d_2)$.

Sowohl im ersten als auch im zweiten Beispiel waren d_1 und d_2 nicht äquivalent. Da wir aber auf der Suche nach einfacheren äquivalenten Funktionen sind, werden wir uns im Folgenden nur mehr mit solchen beschäftigen.

Wir gehen aber sogar noch einen Schritt weiter - wir wollen nicht nur eine einfachere äquivalente Funktion, sondern die einfachste. Dazu definieren wir:

Definition 5.3.4. Man nennt eine Funktion minimal, wenn man keine einfachere äquivalente Darstellung finden kann.

Bemerkung 5.3.5. Bevor wir die genauen Minimierungsverfahren kennenlernen, betrachten wir noch die Funktion $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1x_2\overline{x_3}$. Mit dem Distributivgesetz kann zu $f(x_1, x_2, x_3) = x_1x_2(x_3 + \overline{x_3})$ umgeformt werden, was wiederum

auf Grund des Komplementsgesetzes und des Absorptionsgesetzes $f(x_1, x_2, x_3) = x_1 x_2$ ergibt.

Dies kann auch allgemein betrachtet werden. Wir folgern: Unterscheiden sich zwei Konjunktionsterme in genau einer Variable, so können diese Terme zusammengefasst werden. Man streicht einen der Terme und lässt beim zweiten Term die Variable weg, in der sie sich unterschieden. Die somit erhaltene Funktion ist äquivalent zur ursprünglichen. Auf diesem Prinzip basieren die folgenden Minimierungsverfahren.

5.3.1 Minimierung mit Hilfe von KV-Diagrammen

Bis zu vier (theoretisch sogar sechs) Variablen gibt es eine sehr praktische grafische Methode, um eine Funktion zu vereinfachen - das Karnaugh-Veitch-Diagramm. Auch wenn das KV-Diagramm eine graphische Methode ist, die theoretisch ohne Kenntnis der Folgerungen aus Huntingtons Axiomen durchgeführt werden kann, ist die Erklärung der Methode natürlich eine algebraische. Die Funktion beruht auf der vorhin erwähnten Regel, dass zwei Funktionsterme, die sich nur in einer Variable unterscheiden, zusammengefasst werden können.

Ein Vorteil dieses Verfahrens ist, dass sogar schon eine Wertetabelle reicht, um es anwenden zu können - eine Funktion muss nicht gegeben sein.

Das KV-Diagramm ist auch auf Grund seiner Anschaulichkeit in Schulen und auf der Universität sehr beliebt, in der Praxis werden aber andere Methoden bevorzugt verwendet. Hier überwiegt die uneingeschränkte Nutzbarkeit vor der Anschaulichkeit.

Das KV-Diagramm

Ein KV-Diagramm ist nichts anderes als eine Matrix von Mintermen. Allerdings wird (meist) nicht der Minterm selbst angegeben, sondern man schreibt 1, falls der Minterm der Funktion vorliegt (also der Funktionswert für diese Belegung 1 ist) und 0, falls nicht.¹²

KV-Diagramm mit 2 Variablen

x_1	x_2	$f(x_1, x_2)$	Minterm		x_1	$\overline{x_1}$		x_1	$\overline{x_1}$
0	0	0	$\overline{x_1} \overline{x_2}$						
0	1	1	$\overline{x_1} x_2$	$\rightarrow$	x_2	$x_1 x_2$	$\overline{x_1} x_2$	x_2	1
1	0	0	$x_1 \overline{x_2}$						
1	1	1	$x_1 x_2$		$\overline{x_2}$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_2}$	0
									0

¹²Vgl. [3], S. 149ff

KV-Diagramm und Konjunktive Normalformen

Ist eine KDNF gegeben, so kann das Karnaugh-Veitch-Diagramm direkt aus der Funktion abgelesen werden, indem man für jeden Minterm einen Einsen einträgt. Die Nullen können zur Vereinfachung weggelassen werden.

Beispiel 5.3.7.

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} x_2 x_3 \overline{x_4}$$

	$x_1 x_2$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_1} x_2$
$x_3 x_4$				
$x_3 \overline{x_4}$			1	1
$\overline{x_3} \overline{x_4}$			1	
$\overline{x_3} x_4$				1

Betrachten wir hingegen eine KKNF, so ist die Eintragung genauso einfach, wenn wir eine kurze Überlegung anstellen. Nach Satz 3.3.4 auf S. 25 können wir das Komplement einer KKNF sehr einfach bilden. Wir wissen außerdem, dass dieses Komplement wiederum eine KDNF ergibt.

Tragen wir diese KDNF ins KV-Diagramm ein, so müssen wir nur noch Einsen gegen Nullen tauschen und wir erhalten ein KV-Diagramm der ursprünglichen KKNF. Hat man dieses Prinzip verstanden, trägt man einfach für jeden Maxterm beim Minterm mit den komplementären Variablen eine 0 ein.

Beispiel 5.3.8.

$$f(x_1, x_2, x_3, x_4) = (x_1 + x_2 + x_3 + x_4) \cdot (x_1 + \overline{x_2} + x_3 + \overline{x_4}) \cdot (x_1 + x_2 + \overline{x_3} + x_4) \cdot (x_1 + \overline{x_2} + \overline{x_3} + x_4)$$

$$\overline{f(x_1, x_2, x_3, x_4)} = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} x_2 x_3 \overline{x_4}$$

	$x_1 x_2$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_1} x_2$
$x_3 x_4$				
$x_3 \overline{x_4}$			1	1
$\overline{x_3} \overline{x_4}$			1	
$\overline{x_3} x_4$				1

KV-Diagramm von $\overline{f}$

	$x_1 x_2$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_1} x_2$
$x_3 x_4$				
$x_3 \overline{x_4}$			0	0
$\overline{x_3} \overline{x_4}$			0	
$\overline{x_3} x_4$				0

KV-Diagramm von f

Benachbarte Felder

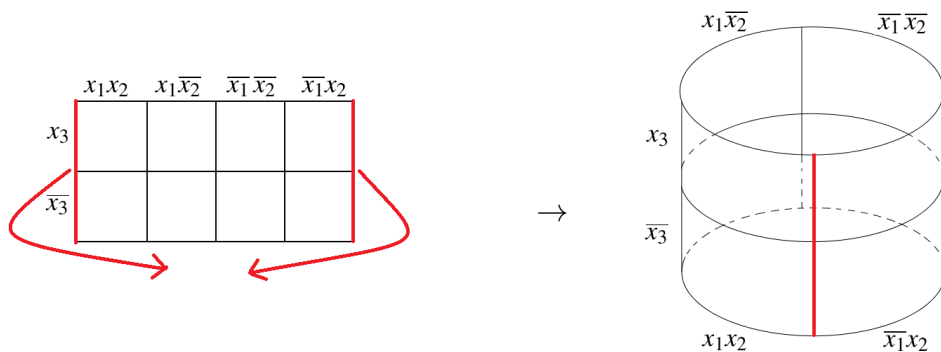
Für die spätere Anwendung bei der Vereinfachung von Normalformen ist die Eigenschaft von KV-Diagrammen, dass zwei benachbarte Felder sich in genau einer Variable unterscheiden, von zentraler Bedeutung.

Was bedeutet aber eigentlich benachbart? Intuitiv erkennen wir, dass all jene Felder benachbart sind, die eine gemeinsame Quadratseite besitzen. Betrachten wir das KV-Diagramm für drei Variablen sehen wir aber, dass auch $x_1x_2x_2$ und $\bar{x}_1x_2x_3$ sich nur in x_1 unterscheiden. Daher gelten auch Felder des linken und des rechten Randes als benachbart.

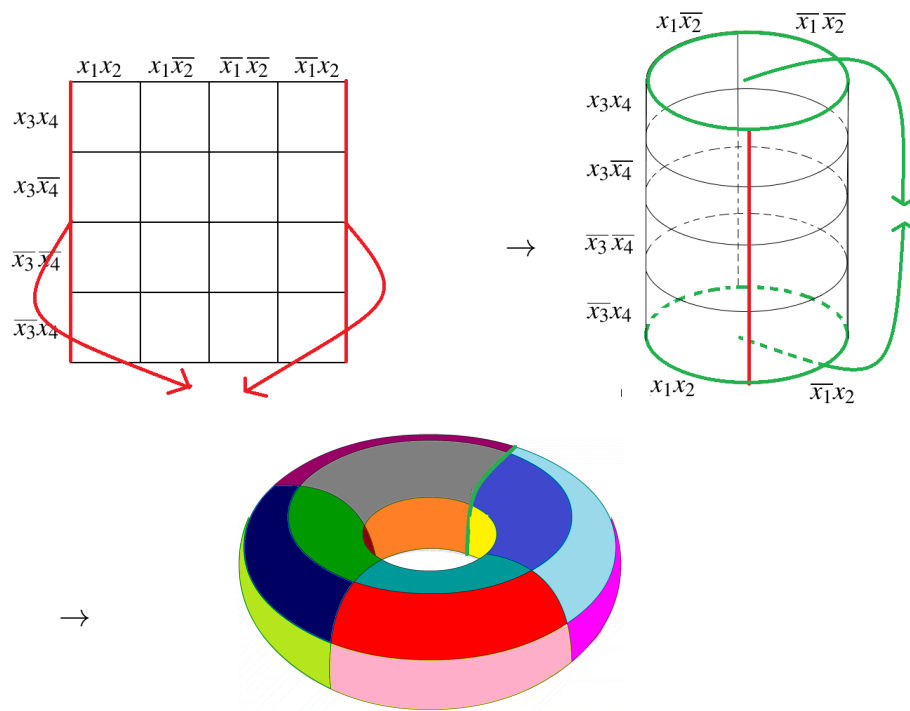
Bei vier Variablen sind zusätzlich auch noch die obersten und untersten Felder benachbart.

Anschaulich kann man sich dies so vorstellen, dass man das Diagramm mit drei Variablen zu einem Zylindermantel biegt und das Diagramm mit vier Variablen zu einem Torus (Kreisring) faltet.

Benachbarte Felder in einem KV-Diagramm mit 3 Variablen



Benachbarte Felder in einem KV-Diagramm mit 4 Variablen ¹³



Das KV-Diagramm als Minimierungsverfahren nutzen

Nun kommen wir zur Anwendung der KV-Diagramme als Minimierungsverfahren. Wie wir bereits wissen, unterscheiden sich in einem KV-Diagramm zwei benachbarte Felder nur in einer Variable.

Die Minterm-Methode

Aus der Bemerkung 5.3.5 können wir daher leicht folgern, dass sich zwei nebeneinanderliegende Einsen, die jeweils einer Vollkonjunktion und somit einem n -stelligen Term entsprechen, zu einem $(n-1)$ -stelligen Funktionsterm verkürzen lassen.

Fasst man nun aber sogar vier (acht oder sogar 16) benachbarte Einsen in Form eines Quadrats, einer ganzen Zeile oder einer ganzen Spalte zusammen, so erhält man sogar einen noch kleineren Term und zwar einen $(n-2)$ -stelligen ($(n-3)$ -stelligen oder $(n-4)$ -stelligen).¹⁴ Man geht also wie folgt vor:

¹³Letzte Abbildung verändert nach Rosario VanTulpe, https://commons.wikimedia.org/wiki/File:Karnaugh_map_KV_Torus_1.png

¹⁴Der mathematische Hintergrund dieser Behauptung wird bei den folgenden Beispielen verdeutlicht.

1. Man kreist möglichst große Gruppen von benachbarten Einsern ein. Da das Idempotenzgesetz gilt, darf ein Einser natürlich auch zu mehreren Gruppen gehören, ohne das Ergebnis zu verändern.
2. Innerhalb dieser Gruppen verbindet man die Variablen, die in allen eingekreisten Feldern gleich sind, konjunktiv und lässt die Variablen, die sich unterscheiden einfach weg.
3. Nun verknüpft man alle so erhaltenen Terme (und gegebenenfalls auch die Minterme von einzelnen Einsern, die nicht zusammengefasst werden konnten) disjunktiv.

Wir werden dies an einigen Beispielen genauer betrachten:

Beispiel 5.3.9. $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1x_2\bar{x}_3$

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3	1			
$\bar{x}_3$	1			

Man erhält vereinfacht den 3-1=2-stelligen Term:

$$y = x_1x_2$$

da gilt:

$$y = x_1x_2 \underbrace{(x_3 + \bar{x}_3)}_1 = x_1x_2$$

Wir wissen bereits, dass auch Felder ganz links beziehungsweise ganz rechts benachbart sind (da sie sich nur in einer Variable unterscheiden). Daher können wir auch diese zusammenfassen.

Beispiel 5.3.10. $f(x_1, x_2, x_3) = x_1x_2x_3 + \bar{x}_1x_2x_3$

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3	1			1
$\bar{x}_3$				

Wir erhalten daher vereinfacht:

$$y = x_2x_3$$

da gilt:

$$y = \underbrace{(x_1 + \bar{x}_1)}_1 x_2x_3 = x_2x_3$$

Nun können auch größere Gruppen gefunden und zusammengefasst werden. Einige Möglichkeiten für Gruppen von vier Einsern sind im Folgenden aufgeführt. Man erhält so Terme, die zwei Variablen weniger enthalten, als der Ausgangsterm.

Beispiel 5.3.11. $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3$

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3	1	1	1	1
$\bar{x}_3$				

Es ergibt sich ein
3-2=1-stelligen Term
 $y = x_3$

da gilt:

$$\begin{aligned}
 y &= x_3(x_1x_2 + x_1\bar{x}_2 + \bar{x}_1\bar{x}_2 + \bar{x}_1x_2) = \\
 &= x_3(x_1 \underbrace{(x_2 + \bar{x}_2)}_1 + \bar{x}_1 \underbrace{(\bar{x}_2 + x_2)}_1) = \\
 &= x_3 \underbrace{(x_1 + \bar{x}_1)}_1 = x_3
 \end{aligned}$$

Beispiel 5.3.12. $f(x_1, x_2, x_3) = x_1x_2x_3 + x_1\bar{x}_2x_3 + x_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3$

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3	1	1		
$\bar{x}_3$	1	1		

Wir erhalten:

$y = x_1$

da gilt:

$$\begin{aligned}
 y &= x_1(x_2x_3 + \bar{x}_2x_3 + x_2\bar{x}_3 + \bar{x}_2\bar{x}_3) = \\
 &= x_1(x_3 \underbrace{(x_2 + \bar{x}_2)}_1 + \bar{x}_3 \underbrace{(\bar{x}_2 + x_2)}_1) = \\
 &= x_1 \underbrace{(x_3 + \bar{x}_3)}_1 = x_1
 \end{aligned}$$

Auch bei Vierergruppen kann eine Schleife natürlich über die Ränder hinausgehen, bei vier Variablen können sogar die Eckfelder zusammengefasst werden, da diese benachbart sind, wenn wir uns den Torus vorstellen.

Beispiel 5.3.13. $f(x_1, x_2, x_3, x_4) = x_1x_2x_3x_4 + \bar{x}_1x_2x_3x_4 + x_1x_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3x_4$

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3x_4	1			1
$x_3\bar{x}_4$				
$\bar{x}_3\bar{x}_4$				
$\bar{x}_3x_4$	1			1

Man erhält daher einen
4-2=2-stelligen Term:

$y = x_2x_4$

da gilt:

$$\begin{aligned}
 y &= x_2x_4(x_1x_3 + \bar{x}_1x_3 + x_1\bar{x}_3 + \bar{x}_1\bar{x}_3) = \\
 &= x_2x_4(x_3 \underbrace{(x_1 + \bar{x}_1)}_1 + \bar{x}_3 \underbrace{(x_1 + \bar{x}_1)}_1) = \\
 &= x_2x_4 \underbrace{(x_3 + \bar{x}_3)}_1 = x_2x_4
 \end{aligned}$$

Natürlich können auch 8 Felder (oder theoretisch sogar 16 oder bei größeren KV-Diagrammen sogar noch mehr Felder) zusammengefasst werden.

Würden wir im Fall von vier Variablen allerdings alle vorhandenen 16 Felder zusammenfassen, so hätten wir eine Funktion, die alle Werte auf 1 abbildet und daher eine Tautologie.

Beispiel 5.3.14.

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 x_4 + x_1 \overline{x_2} x_3 x_4 + x_1 x_2 x_3 \overline{x_4} + x_1 \overline{x_2} x_3 \overline{x_4} + x_1 x_2 \overline{x_3} \overline{x_4} + x_1 \overline{x_2} \overline{x_3} \overline{x_4} + x_1 x_2 \overline{x_3} x_4 + x_1 \overline{x_2} \overline{x_3} x_4$$

	$x_1 x_2$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_1} x_2$
$x_3 x_4$	1	1		
$x_3 \overline{x_4}$	1	1		
$\overline{x_3} \overline{x_4}$	1	1		
$\overline{x_3} x_4$	1	1		

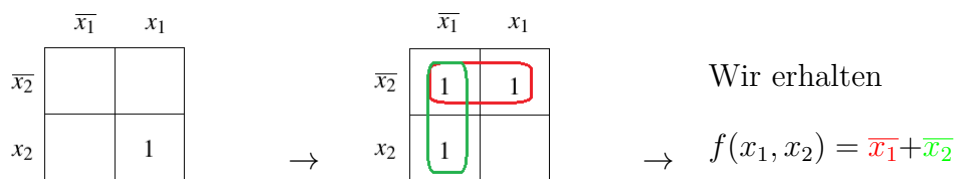
Wir erhalten vereinfacht:

$$y = x_1$$

Auf Grund der zahlreichen vorherigen Beispiele wird hier auf die genaue Herleitung verzichtet.

Nachdem wir nun die wichtigsten Möglichkeiten kennengelernt haben, wie man Gruppierungen bei KV-Diagrammen zur Minimierung nutzen kann, werden wir uns nun einigen (mehr oder weniger) komplexeren Beispielen widmen, für die mehrere Schleifen benötigt werden, um eine Minimalform zu erhalten.

Beispiel 5.3.15. Wir gehen von der Funktion $f(x_1, x_2) = \overline{x_1} \cdot \overline{x_2}$ aus. Um diese in ein KV-Diagramm eintragen zu können, stellen wir zuerst das Diagramm für das Komplement $\overline{f}(x_1, x_2)$ auf und tauschen dann 1 gegen 0. Wir erhalten:



Wir erhalten

KV-Diagramm für $\overline{f}$ KV-Diagramm für f

Man erkennt leicht, dass damit auf sehr einfache Art und Weise eine der Regeln von DeMorgan nachgewiesen wurde.

Beispiel 5.3.16.

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3x_4	1		1	1
$x_3\bar{x}_4$		1	1	1
$\bar{x}_3\bar{x}_4$	1			
$\bar{x}_3x_4$				

Durch Vereinfachung folgt:

$$y = \bar{x}_1x_3 + \bar{x}_2x_3\bar{x}_4 + x_2x_3x_4 + x_1x_2\bar{x}_3\bar{x}_4$$

Betrachten wir außerdem noch das Alarmanlagen-Beispiel 5.1.1, auch wenn dies hier relativ aufwendig ist.

Beispiel 5.3.17 (Alarmanlage).

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3	1	1	1	1
$\bar{x}_3$	1	1	0	1

Man erhält:

$$y = x_1 + x_3 + x_2$$

Die Maxterm-Methode

Da wir bereits wissen, wie wir in ein KV-Diagramm Maxterme als Nuller eintragen können, ist es umgekehrt sehr einfach, auch diese aus einem KV-Diagramm abzulesen. Für das Alarmanlagenbeispiel (das nur für eine Belegung den Wert 0 annimmt) erhält man durch Komplementierung des zugehörigen Minterms ebenfalls

$$f(x_1, x_2, x_3) = x_3 + x_2 + x_1.$$

Diese Methode wird Maxterm-Methode genannt. Sie unterscheidet sich nur sehr gering von der Minterm-Methode: ¹⁵

1. Anstatt Einser werden nun Nuller zusammengefasst.
2. Man bildet nun die Terme der Päckchen, indem man die gleich vorkommenden Variablen negiert und disjunktiv (statt konjunktiv) verknüpft. Wie bei der Minterm-Methode werden die Variablen, die sich unterscheiden, weggelassen.
3. Anschließend werden die Disjunktionsterme noch konjunktiv verknüpft (statt disjunktiv).

¹⁵vgl. [9]

Beispiel 5.3.18.

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3x_4	0	0		0
$x_3\bar{x}_4$		0		0
$\bar{x}_3\bar{x}_4$				0
$\bar{x}_3x_4$	0	0		0

Wir erhalten vereinfacht:

$$f = (x_1 + \bar{x}_2)\bar{x}_2(\bar{x}_1 + x_2 + \bar{x}_4)(\bar{x}_1 + x_2 + \bar{x}_3)$$

Vereinfachung von KV-Diagramme von Funktionen mit don't-care-Zuständen

Da es bei don't-care-Zuständen keine Rolle spielt, ob eine Funktion für die zugehörigen Belegungen die Ausgangsvariablen 1 oder 0 annimmt, kann auch bei der Minimierung ein don't-care-Zustand im KV-Diagramm einfach mit einem Strich gekennzeichnet werden und je nach Bedarf als 1 oder 0 gesehen werden.

Beispiel 5.3.19.

	x_1x_2	$x_1\bar{x}_2$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_2$
x_3x_4	1	1		—
$x_3\bar{x}_4$	1	1		
$\bar{x}_3\bar{x}_4$	1	—		
$\bar{x}_3x_4$	1	1		

Wir verwenden — als 1 und — als 0 und können daher vereinfachen:

$$f = x_1$$

5.3.2 Quine-McCluskey

Bis jetzt können wir binäre Funktionen nur vereinfachen, indem wir entweder probieren oder bei KV-Diagrammen durch gutes Hinsehen Gruppierungen finden. Ein systematischer Algorithmus ist hingegen das Quine-McCluskey-Verfahren. Hier müssen für die Vereinfachung zwei Algorithmen hintereinander durchgeführt werden.

Für das Verfahren ist zuallererst eine Wertetabelle notwendig, die dann systematisch vereinfacht wird. Im ersten Schritt werden alle Primterme konstruiert (Algorithmus von McCluskey). Da hierbei aber nicht immer die Minimalform erhalten werden kann, müssen in einem zweiten Algorithmus noch die Primterme ausgewählt werden, die benötigt werden, um die Funktion minimal zu realisieren (Algorithmus von Quine).¹⁶

Algorithmus von McCluskey Hier verwenden wir dasselbe Prinzip wie beim KV-Diagramm. Zwei Minterme, die sich in genau einer Variable unterscheiden, können zu einem Term zusammengefasst werden.

1. Zuerst erstellt man eine Liste aller Belegungen, die auf 1 abbilden, sortiert diese nach der Anzahl an vorkommenden Einser in Gruppen und nummeriert sie der Reihe nach.
2. Nun vergleicht man jeden Term einer Gruppe mit allen Termen der nächsten Gruppe. Findet man so zwei Terme, die sich nur in genau einer Variable unterscheiden, so können diese zusammengefasst werden, indem man anstatt dieser Stelle – (also ein don't-care) schreibt und die neuen Terme in eine zweite Liste schreibt.
3. Natürlich darf auch hier auf Grund der Idempotenz jeder Ausdruck mehrfach verwendet werden. Um den Überblick zu bewahren, werden bereits verwendete Terme abgehakt.
4. Man wiederholt dies so lange, bis es keine Möglichkeit mehr gibt, Terme zusammenzufassen.
5. Die minimale Funktion kann nun aufgestellt werden, indem man alle Primterme (Terme, die nicht abgehakt wurden - kommen zumindest in der letzten Tabelle vor) disjunktiv verknüpft.

Auch hier betrachten wir wieder das Alarmanlagen-Beispiel aus 5.1.1:

¹⁶vgl. [21], S.48

Beispiel 5.3.20.

Man stellt nun einer Liste von allen Belegungen, die auf 1 abbilden auf, sortiert diese nach der Anzahl der auftretenden Einsen und nummeriert anschließend:

Panik	Türen	Fenster	Alarm
0	0	0	0
0	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	1

I	0	0	1	✓
II	0	1	0	✓
III	1	0	0	✓
IV	0	1	1	✓
V	1	0	1	✓
VI	1	1	0	✓
VII	1	1	1	✓

Exemplarisch betrachten wir I und IV. (0,0,1) und (0,1,1) unterscheiden sich nur an einer Stelle. Daher dürfen wir zu (0,-,1) zusammenfassen. Ebenso können auch I und V vereinfacht werden, bei I und VI ist dies allerdings nicht möglich. Auf diese Weise vergleicht man alle Elemente. Algebraisch streckt dahinter, dass die Minterme $\overline{x_1}\overline{x_2}x_3$ und $\overline{x_1}x_2x_3$ zu $\overline{x_1}x_3$ zusammengefasst werden können.

Man erstellt eine zweite Liste mit zusammengefassten Termen und hakt die verwendeten Terme ab. Der Vorgang wird so lange wie möglich wiederholt, in diesem Fall ist nach der dritten Tabelle Schluss. Auf Grund des Idempotenzgesetz können wir alle doppelten Primterme streichen.

I, IV	0	-	1	✓
I, V	-	0	1	✓
II, IV	0	1	-	✓
II, VI	-	1	0	✓
III, V	1	0	-	✓
III, VI	1	-	0	✓
IV, VII	-	1	1	✓
V, VII	1	-	1	✓
VI, VII	1	1	-	✓

I, IV, IV, VII	-	-	1
I, IV, V, VII	1		
I, V, IV, VII	1		
II, IV, VI, VII	-	1	-
II, VI, IV, VII	1		
III, V, VI, VII	1	-	-
III, VI, V, VII	1		

Wir erhalten: $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$. In diesem Fall kann keine einfachere Form gefunden werden, aber nicht immer ist dies so.

Beispiel 5.3.21. Dazu betrachten wir die Funktion ¹⁷

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \overline{x_2} \overline{x_3} x_4 + \overline{x_1} x_2 \overline{x_3} \overline{x_4} + \overline{x_1} x_2 x_3 \overline{x_4} + \overline{x_1} x_2 x_3 x_4 + x_1 \overline{x_2} \overline{x_3} \overline{x_4} + x_1 \overline{x_2} \overline{x_3} x_4 + x_1 \overline{x_2} x_3 \overline{x_4} + x_1 x_2 x_3 x_4$$

Wir erhalten daher folgende Tabelle (schon nach Anzahl der Einsen sortiert) und vereinfachen:

I	0	0	0	1	✓	I, V	–	0	0	1						
II	0	1	0	0	✓	II, IV	0	1	–	0						
III	1	0	0	0	✓	III, V	1	0	0	–	✓					
IV	0	1	1	0	✓	III, VI	1	0	–	0	✓	III, V, VI, VIII	1	0	–	–
V	1	0	0	1	✓	IV, VII	0	1	1	–		III, VI, V, VIII	1	0	–	–
VI	1	0	1	0	✓	V, VIII	1	0	–	1	✓					
VII	0	1	1	1	✓	VI, VIII	1	0	1	–	✓					
VIII	1	0	1	1	✓	VII, IX	–	1	1	1						
IX	1	1	1	1	✓	VIII, IX	1	–	1	1						

und wir erhalten:

$$f(x_1, x_2, x_3, x_4) = \underbrace{\overline{x_2} \overline{x_3} x_4}_{\text{I, V}} + \underbrace{\overline{x_1} x_2 \overline{x_4}}_{\text{II, IV}} + \underbrace{\overline{x_1} x_2 x_3}_{\text{IV, VII}} + \underbrace{x_2 x_3 x_4}_{\text{VII, IX}} + \underbrace{x_1 x_3 x_4}_{\text{VIII, IX}} + \underbrace{x_1 \overline{x_2}}_{\text{III, V, VI, VIII}}$$

Betrachten wir aber nun die Minimierung der selben Funktion im KV-Diagramm, erhalten wir eine eindeutig einfachere Funktion. Man sieht, dass zwei der Terme, die wir durch das McCluskey-Verfahren erhalten, nicht notwendig sind, um die Funktion zu definieren.

	$x_1 x_2$	$x_1 \overline{x_2}$	$\overline{x_1} \overline{x_2}$	$\overline{x_1} x_2$
$x_3 x_4$	1	1		1
$x_3 \overline{x_4}$		1		1
$\overline{x_3} \overline{x_4}$		1		1
$\overline{x_3} x_4$		1	1	

$$f(x_1, x_2, x_3, x_4) = x_1 \overline{x_2} + x_2 x_3 x_4 + \overline{x_2} \overline{x_3} x_4 + \overline{x_1} x_2 \overline{x_4}$$

Daher reicht der McCluskey-Algorithmus alleine noch nicht aus, um eine minimale Darstellung zu erhalten. Es muss noch ein weiterer Schritt folgen.

¹⁷s. [23], S. 15f

Überdeckungstabellen nach dem Quine-Algorithmus

Um tatsächlich eine Minimalform zu erhalten müssen wir nun die sogenannte Überdeckungstabelle (auch Primimplikantentafel oder 2.Quinesche Tabelle) verwenden. So kann herausgefunden werden, welche Primterme weggelassen werden können.

1. In einem ersten Schritt werden die Nummern der erhaltenen Primterme als Zeilen gegen die Nummern der Minterme als Spalten aufgetragen.
2. Da Primterme eine Verkürzung von Mintermen darstellen, kreuzt man nun das Feld im Schnittpunkt der passenden Primtermzeile und Mintermspalte an.
3. Nun betrachten wir zuerst die Spalten. Tritt ein Kreuz nur einmal in einer Spalte auf, so markiert man dieses mit einem Kreis um das Kreuz. Die zugehörigen Primimplikanten nennt man Kernimplikanten und diese müssen in der Minimaldarstellung der Funktion auftreten.
4. Als nächstes kennzeichnet man alle Minterme, die die vorhin ermittelten Kernimplikante ebenfalls abdecken mit einem Kasten rund um das Kreuz.
5. Steht nun in jeder Spalte ein Kreis oder ein Kästchen um mindestens ein Kreuz, so wurden die wesentlichen Primterme gefunden und das Verfahren kann beendet werden. Gibt es aber noch Minterme (also Spalten), die noch von keinem Primterm abgedeckt wurden, stellt man mit diesen Mintermen und den zugehörigen Primtermen eine neue Überdeckungstabelle auf und wiederholt das Verfahren.
6. Oft endet dieses aber, ohne eine eindeutige Lösung zu liefern. In diesem Fall muss man entweder raten, oder man stellt die sogenannte Überdeckungsfunktion auf.¹⁸
 - (a) Dazu gibt man zuerst jedem noch nicht verwendeten Primimplikanten einen Namen ($A, B, \dots$).
 - (b) Die Überdeckungsfunktion verwendet nun die Variablen $w_A, w_B, \dots$, die den Kreuzen in der Tabelle entsprechen.
 - (c) Man stellt nun die Funktion auf, indem man zu jedem Minterm alle w_X disjunktiv verknüpft und anschließend die erhaltenen Terme konjunktiv.
 - (d) Durch Vereinfachung dieser Funktion nach den Axiomen kann nun herausgefunden werden, welche Primterme notwendig sind, um die Funktion vollständig zu definieren.

¹⁸vgl. [15]

Wir probieren dies gleich an dem vorherigen Beispiel 5.3.21 aus:

Beispiel 5.3.22. Mit dem McCluskey-Algorithmus haben wir bereits folgende Funktion erhalten:

$$f(x_1, x_2, x_3, x_4) = \underbrace{\overline{x_2} \overline{x_3} x_4}_{\text{I, V}} + \underbrace{\overline{x_1} x_2 \overline{x_4}}_{\text{II, IV}} + \underbrace{\overline{x_1} x_2 x_3}_{\text{IV, VII}} + \underbrace{x_2 x_3 x_4}_{\text{VII, IX}} + \underbrace{x_1 x_3 x_4}_{\text{VIII, IX}} + \underbrace{x_1 \overline{x_2}}_{\text{III, V, VI, VIII}}$$

Wir stellen nun die zugehörige Überdeckunstabelle auf:

	I	II	III	IV	V	VI	VII	VIII	IX
I, V	⊗				x				
II, IV		⊗		x					
IV, VII				x			x		
VII, IX							x		x
VIII, IX								x	x
III, V, VI, VIII			⊗		x	⊗		x	

Wir haben nun jene Primterme gefunden, die **Kernprimimplikanten** sind. Für die noch nicht überdeckten Minterme bilden wir nun eine neue Überdeckungstabelle und beginnen von vorne.

	VII	IX
A IV, VII	x	
B VII, IX	x	x
C VIII, IX		x

Wir sehen allerdings sofort, dass wir hier mit dem Verfahren nicht weiterkommen und müssen nun wieder probieren. Dies ist in diesem Fall sehr einfach, da wir entweder A ($\overline{x_1}x_2x_3$) und C ($x_1x_2x_3$) brauchen, um beide Minterme zu überdecken, oder B ($x_2x_3x_4$). Da B kostengünstiger (da einfacher) ist, als A konjugiert mit C, wählen wir B. Alternativ können wir auch die sogenannte Überdeckungsfunktion aufstellen. Diese hat im wesentlichen dieselbe Funktion wie unsere Vermutungen. Sie liefert uns die verschiedenen Optionen, aus denen man die kosteneffizienteste auswählt.

$$\ddot{u}_f = (w_A + w_B)(w_B + w_C) \stackrel{\text{DG}}{=} (w_A w_C) + w_B$$

Wir erhalten also ebenfalls die zwei Möglichkeiten „A und C “ oder „B“. Wir entscheiden uns daher für B und erhalten nun als vollständige Minimalform

$$f(x_1, x_2, x_3, x_4) = \underbrace{\overline{x_2} \overline{x_3} x_4}_{\text{I, V}} + \underbrace{\overline{x_1} x_2 \overline{x_4}}_{\text{II, IV}} + \underbrace{x_2 x_3 x_4}_{\text{VII, IX}} + \underbrace{x_1 \overline{x_2}}_{\text{III, V, VI, VIII}}$$

also genau jene Funktion, die wir auch mit dem KV-Diagramm erhalten haben.

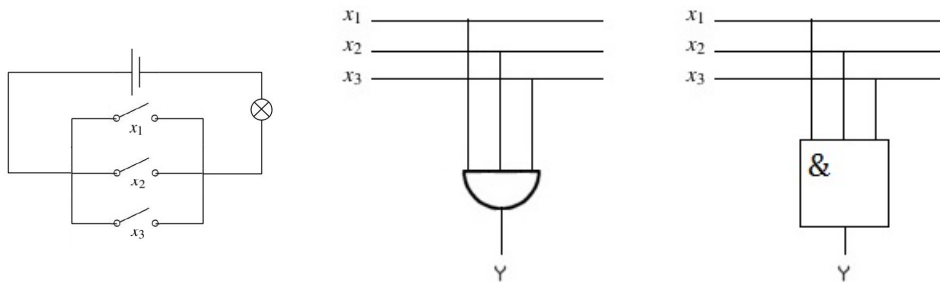
Natürlich ist das Verfahren nach McCluskey sehr aufwendig. Dafür benötigt es nur sehr geringe Rechenleistung und einen geringen Speicherbedarf. Gleichzeitig ist es auch noch sehr exakt. Wie wir gesehen haben, kann es allerdings auch manchmal sein, dass wir nur eine beinahe Minimalform erhalten. Daher gibt es noch viele Spezialregeln und Erweiterungen. Bei Interesse können diese in [5] S. 96ff nachgelesen werden.

5.4 Bauteilauswahl und Realisierung

Wurde nun die Minimalform gefunden, so ist der nächste Schritt die Bauteile zu wählen und anschließend die Schaltung zu realisieren. Schon in Unterkapitel 4.5 auf Seite 34 haben wir gesehen, mit welchen Bauteilen und auf welche Art und Weise Schaltungen realisierbar sind. Natürlich gibt es noch zahlreiche weitere Bauteile und die Technik schreitet in schneller Geschwindigkeit immer weiter voran.

Auf Grund des fehlenden Bezugs zur Mathematik beschränken wir uns daher zur Abrundung nur noch auf die (sehr einfache) Realisierung in Schalterdarstellung und in Gatterdarstellungen des Alarmanlagenbeispiels 5.1.1, das uns durch dieses Kapitel geführt hat.

Beispiel 5.4.1 (Alarmanlage). $f(x_1, x_2, x_3) = x_1 + x_2 + x_3$



Kapitel 6

Ausblick

Es ist die Ultimative Maschine – die finale Entwicklung. [...] Sie stand auf Claude Shannons Schreibtisch und machte die Leute verrückt. Nichts hätte einfacher aussehen können. Eine schlichte, kleine Holzkiste, [...] mit einem Schalter an der Vorderseite. Wer diesen Schalter umlegt, hört ein zorniges, entschlossenes Brummen. Der Deckel öffnet sich langsam, eine Hand erscheint aus dem Inneren. Sie greift aus der Kiste heraus, legt den Schalter wieder um und verschwindet rasch zurück in die Box.

Arthur C. Clarke in „Voices across the sea“, nach [8], S. 18

Claude E. Shannon gilt nicht nur als der Vater der Informationstechnologie, sondern verbrachte seine Freizeit auch damit, verschiedenste Maschinen - wie Jongliermaschinen, „Mind Reading“ Maschinen, Tischrechner, kleine Roboter oder die im Zitat beschriebene „Ultimative Maschine“- zu bauen.¹

Für die Konstruktion solcher Maschinen werden komplexere Bauteile benötigt, als die Gatter, die wir in Kapitel 4 kennengelernt haben. Durch den Einsatz dieser Bauteile können in Folge auch andere Maschinen, wie zum Beispiel Rechenmaschinen - zu denen auch der Computer zählt - gebaut werden.

Einen der komplexeren Schalter - den Multiplexer - haben wir schon kennengelernt. Weitere Beispiele wären der Codierer, der ein Signal in den Zeichen des einen Codes, in Zeichen eines anderen Codes umwandelt oder der Komparator, der zwei Binärzahlen miteinander vergleichen kann. Es gibt aber auch arithmetische Schaltnetze, die zum Rechnen verwendet werden. Konnte man zuerst nur zwei einstellige Binärzahlen mit dem Halbaddierer addieren, so bot später der Volladdierer die Möglichkeit, zwei mehrstellige Binärzahlen zu addieren. Dafür ist ein Übertrag nötig. Bei Interesse kann zum Beispiel in [4], S. 18 - 24 und 55 - 67 mehr über binäre Rechengänge und diese besonderen Schaltnetze nachgelesen werden.

¹Interessierte Leser können in [8] mehr darüber nachlesen.

Ein wichtiger Schritt in Richtung Computer stellte die sogenannte Programmierbare Logik dar.² Hierbei sind die Bauteile nicht mehr für eine spezielle Anwendung gebaut, sondern können je nach Bedarf programmiert werden. Durch sie gibt es nicht nur programmierbare Speicher (wie ROMs), sondern es können auch logische Anordnungen (sogenannte logische Arrays) nach Bedarf programmiert werden. Genaueres kann in [4], S. 88ff nachgelesen werden.

Wir alle wissen, wie schnell die Entwicklung in diesem relativ jungen Gebiet der Digitaltechnik voranschreitet, und sehen, dass es noch schier unendliche Themen gäbe, die diese Arbeit behandeln könnte. Da wir uns dabei aber auch immer weiter in die Digitaltechnik vorarbeiten und uns von der Mathematik immer weiter wegbewegen, soll mit diesem Kapitel der Abschluss gefunden werden.

Claude E. Shannon antwortete in einem Interview 1987, als er über die Zukunft von Robotern befragt wurde:

I see no limit to the capabilities of machines. As microchips get smaller and faster, I can see them getting better than we are.

Claude E. Shannon in [18], S. 64f

Auch wenn Maschinen den Menschen (noch) nicht vollständig ersetzen können, gibt es viele Bereiche, in denen der Computer den Menschen übertrifft. Das faszinierende daran ist, dass all diese Maschinen, die unseren Alltag begleiten, auf mathematischen Prinzipien basieren - und zwar unter anderem auf der Schaltalgebra und damit auch auf der Boole'schen Algebra.

Daher kann gesagt werden, dass diese - und damit auch ihre Begründer und all jene, die sie weiterentwickelt haben - einen wesentlichen Beitrag zu unserem modernen Alltag geliefert haben. Viele der Vorzüge, die wir heute genießen, wären ohne die technischen Entwicklungen aus der Boole'schen Algebra (zumindest in dieser Form) nicht möglich. Niemand kann sagen, wohin diese Entwicklung noch führen wird.

²vgl. [4], S. 88 - 103

Tabellenverzeichnis

2.1	Alle 1-stelligen Verknüpfungen	13
2.2	Alle 16 möglichen zweistelligen Junktoren	13

Literaturverzeichnis

- [1] Barth, R. *Absolute Wahrheit und endliches Wahrheitsbewusstsein*, Mohr Siebeck Thübingen, 2004
- [2] Billich, E. *Skript zur Vorlesung „Systemtheorie Digitaler Systeme“* an der TU Chemnitz, o.J. [Online in: <https://www-user.tu-chemnitz.de/~ebi/uni/lecturenotes/STDigSys.pdf>, 18.01.2016]
- [3] Bong, U. *Boolesche Algebra*, Herder GmbH & Co. KG Freiburg im Breisgau, 1977
- [4] Gremzow, C., Moser, N. *Technische Grundlagen der Informatik 1 - Digitale Systeme* an der Technischen Universität Berlin, 2010/11 [Online in: https://docs.freitagrunde.org/Lernmaterialien/TechGI_1/TechGI_1_Skript_04_2011.pdf, 08.12.2016]
- [5] Gotthardt, K. *Grundlagen der Informationstechnik*, Lit Verlag, Münster, 2005
- [6] Gumm, H.-P., Poguntke, W. *Boolesche Algebra*, Bibliographisches Institut AG, Mannheim, 1981
- [7] Hebisch, U. *Boolesche Algebra*, Freiburger Web-Vorlesung zur Klassischen Algebra, 2009 [Online in: <http://www.mathe.tu-freiberg.de/~hebisch/cafe/algebra/boolescheAlgebren.html>, 08.12.2016]
- [8] Heinz Nixdorf MuseumsForum HNF *Codes & Clowns, Claude Shannon - Jongleur der Wissenschaft*, Flyer zur Sonderausstellung, Paderborn, 2009/10, Zitat aus Clarke, A. *Voices across the sea*, 1974 [Online in: http://www.hnf.de/fileadmin/content/pictures/HNF/Sonderausstellungen/Codes_und_Clowns/Shannon_DIN-Lang_28S.pdf, 13.02.2017]
- [9] Höpping, I. *Minimierung nach Karnaugh-Veitch*, Version vom 06.02.2009, 14.32 [Online in: http://www.info-wsf.de/index.php/Minimierung_nach_Karnaugh-Veitch, 29.12.2016]

- [10] Jeger, M. *Boole'sche Algebra - Autographie der Vorlesung* an der ETH Zürich, Verlag der Fachvereine an der Eidg. Technischen Hochschule/Zürich 1974
- [11] Jehle, F. *Boolesche Algebra*, Bayrischer Schulbuch-Verlag, 1971
- [12] Kittler, F.; Berz, P.; Hauptmann, D.; Roch, A. *Claude E. Shannon - EIN AUS: Ausgewählte Schriften zur Kommunikations- und Nachrichtentheorie*, Brinkmann + Bose Berlin, 2000
- [13] Kriegel, K. *Boolesche Funktionen*, an der FU Berlin, WS 2009/10 [Online in: <http://www.inf.fu-berlin.de/lehre/WS09/infa/boole.pdf>]
- [14] Linnemann, H. *2.2 Schaltfunktionen und ihre Darstellung*, an der Beuth Hochschule für Technik Berlin, WS 2016/17 [Online in: https://prof.beuth-hochschule.de/fileadmin/user/linnemann/PDF-Dateien/TGI/TGI_SU_Teil_02.pdf, 08.12.2016]
- [15] Mandery, C. *DuE-Tutorien 16 und 17 - Tutorien zur Vorlesung "Digitaltechnik und Entwurfsverfahren"*, am KIT Karlsruher Institut für Technik, WS 2010/11 [Online in: <http://tutorium.chrismandery.de/WS1011/tutorium7.pdf>, 30.12.2016]
- [16] Metz, H.-R. *Lösungen zum Übungsblatt 1*, an der FH Gießen-Friedberg, SS 2010 [Online in: <https://homepages.thm.de/~hg8070/dm10/ueb011sg.pdf>, 08.12.2016]
- [17] Müller, D. *Diplomarbeit - Boolesche Algebra und Anwendungen*, betreut von Mitsch, H., Universität Wien, WS 2007
- [18] Redaktion des Omni Magazins *Interview Claude Shannon*, erschienen im Omni Magazin, August 1987, [Online in: <http://archive.retro.co.za/archive/Omni/Omni-Interview-Claude-Shannon.pdf>]
- [19] Schichl, H., Steinbauer, R. *Einführung in das mathematische Arbeiten* 2. Aufl., Springer-Verlag Berlin Heidelberg 2009, 2012
- [20] Schiele, I. C. *Entwicklung und Untersuchung von oberflächenmikromechanischen Relais für die Anwendung in der Hochfrequenztechnik*, Herbert Utz Verlag München, 1999
- [21] Schimmler, M. *Scriptum zur Vorlesung Systemorientierte Informatik I - Digitale Systeme* an der Technischen Fakultät der Universität Kiel WS 14/15. [Online in: <http://www.techinf.informatik.uni-kiel.de/de/lehre/vorlesungen/inf-digisys-digitale-systeme-vorlesung>, 08.12.2016]

- [22] Stanković, R.S., Astola, J. *From Boolean Logic to Switching Circuits and Automata*, Springer-Verlag Berlin Heidelberg, 2011
- [23] Teppner, U. *Skript zur Vorlesung: Grundlagen Digitaler Systeme / Introduction into Digital Systems* an der Beuth Hochschule für Technik Berlin, WS 2009/10 [Online in: https://prof.beuth-hochschule.de/fileadmin/user/svoss/GDS/Zusatzmaterial/Skript_GDS.pdf, 08.12.2016]