

Rolf Socher



Mathematik für Informatiker

Diskrete Mathematik, Lineare Algebra
und Wahrscheinlichkeitsrechnung



2., aktualisierte und erweiterte Auflage

HANSER



Ihr Plus – digitale Zusatzinhalte!

Auf HanserPLUS finden Sie zu diesem Titel kostenloses digitales Zusatzmaterial. Geben Sie dazu einfach diesen Code ein:

plus-v9jh8-dawdg

plus.hanser-fachbuch.de



Bleiben Sie auf dem Laufenden!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Rolf Socher

Mathematik für Informatiker

Diskrete Mathematik, Lineare Algebra und Wahrscheinlichkeitsrechnung

2., aktualisierte und erweiterte Auflage

HANSER

Der Autor:

Prof. Dr. Rolf Socher, ehemals Technische Hochschule Brandenburg



Alle in diesem Buch enthaltenen Informationen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt geprüft und getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en, Herausgeber) und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso wenig übernehmen Autor(en, Herausgeber) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2022 Carl Hanser Verlag München

Internet: www.hanser-fachbuch.de

Lektorat: Frank Katzenmayer

Covergestaltung: Max Kostopoulos

Titelmotiv: © stock.adobe.com/Gorodenkoff

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Satz: Rolf Socher

Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Göttingen

Printed in Germany

Print-ISBN 978-3-446-46747-7

E-Book-ISBN 978-3-446-47439-0

Vorwort

Mathematik hat mir in der Schule besonders gefallen, weil ich dafür nichts auswendig zu lernen brauchte. Ich besitze bis heute noch nicht einmal eine Formelsammlung, denn die Formeln, die ich nicht sowieso durch häufigen Gebrauch inzwischen weiß, kann ich mir meist selbst herleiten. Mathematik ist eben kein Lernfach, sondern ein Fach, in dem man durch Arbeiten mit den Strukturen Verständnis erwirbt.

In diesem Sinne ist auch das vorliegende Buch weniger ein Buch zum Lernen, sondern in erster Linie ein Buch zum Arbeiten. Neben den üblichen Übungsaufgaben am Schluss jedes Abschnitts, die der Anwendung der dort erläuterten Methoden dienen, finden Sie auch Aufgaben im laufenden Text, die der Vorbereitung und selbstständigen Erarbeitung neuer Begriffe und Methoden dienen und deren Bearbeitung ich Ihnen sehr ans Herz legen möchte!

Mit einigen dieser Aufgaben verfolge ich eine problemorientierte Herangehensweise an die Mathematik. Ausgehend von einem konkreten Problem aus der Informatik, etwa der Frage, ob in einer grafischen Oberfläche der Mausclickpunkt nahe genug an einer gegebenen Linie ist, um diese zu markieren (► Abschnitt 9.1), werden die dazu benötigten mathematischen Begriffe und Methoden entwickelt, bis schließlich alle mathematischen „Werkzeuge“ da sind, um das Problem zu lösen.

Am Schluss einiger Abschnitte finden Sie Programmieraufgaben, die der weiteren Vertiefung des Stoffes, insbesondere der algorithmischen Anteile, dienen. Deren Bearbeitung stellt meines Erachtens eine gute Brücke von der Mathematik zum eigentlichen „Kerngeschäft“ der Informatiker, dem Programmieren, dar. Die Programmbeispiele im Text habe ich in Java formuliert, da dies sicherlich die häufigste Programmiersprache in den Informatikstudiengängen an Hochschulen ist.

Dieses Buch deckt mit Ausnahme der Analysis und der Stochastik die wichtigsten mathematischen Inhalte ab, die an Bachelorstudiengängen an Fachhochschulen üblicherweise angeboten werden. Die Stoffauswahl ist seit der Umstellung von Diplom- auf Bachelor- und Masterstudiengänge schwieriger geworden, weil dabei der Umfang der Mathematikmodule deutlich gekürzt wurde. Den Stoff für dieses Buch habe ich hauptsächlich im Hinblick auf die Anwendungen in der Informatik ausgewählt. Die analytische Geometrie ist eine ganz wesentliche Grundlage der Computergrafik, die lineare Algebra wird unter anderem in der Theorie der fehlerkorrigierenden Codes angewandt, und die modulare Arithmetik spielt eine wichtige Rolle in vielen Teilen der Informatik, insbesondere in der Kryptografie.

Ich danke Marion Clausen und Susanne Hohmann für ihr sorgfältiges Korrekturlesen und -rechnen sowie Mirjam Ambrosius und Katja Orłowski für viele nützliche Hinweise. Ferner danke ich dem Carl Hanser Verlag, allen voran Frau Fritsch und Frau Wulst für die gewohnt gute Zusammenarbeit.

Berlin, im November 2010

Rolf Socher

Vorwort zur 2. Auflage

In der zweiten Auflage wurde das Buch erweitert um Kapitel 8 zur Wahrscheinlichkeitsrechnung.

Die Lösungen zu den Aufgaben finden Sie auf der Seite <https://plus.hanserfachbuch.de/>. Den Zugangscode finden Sie auf der ersten Seite des Buchs.

Ich danke dem Carl Hanser Verlag, insbesondere Christina Kubiak und Frank Katzenmayer, für die gute Zusammenarbeit bei der zweiten Auflage.

Berlin, im März 2022

Rolf Socher

Inhalt

1	Aussagenlogik	9
1.1	Aussagen und logische Junktoren	9
1.2	Rechnen mit logischen Formeln	15
1.3	Normalformen und Vereinfachung von Formeln	22
1.4	Beweisverfahren	32
2	Mengen und Relationen	40
2.1	Mengen	40
2.2	Mengenoperationen	46
2.3	Relationen	52
3	Funktionen und Abzählbarkeit	61
3.1	Funktionen	61
3.2	Injektive, surjektive und bijektive Funktionen und die Umkehrfunktion	67
3.3	Endliche und unendliche Mengen	71
4	Kombinatorik	75
4.1	Die Summen- und die Produktregel	75
4.2	Permutationen und geordnete Auswahl ohne Wiederholung	78
4.3	Die Binomialzahlen	82
4.4	Ungeordnete Auswahl mit Wiederholung	86
5	Teilbarkeit und modulare Arithmetik	88
5.1	Teilbarkeit und euklidischer Algorithmus	89
5.2	Primzahlen und Primfaktorzerlegung	96
5.3	Modulare Arithmetik	98
5.4	Die modulare Inverse	103
5.5	Rechnen in \mathbb{Z}_m	105
5.6	Der RSA-Algorithmus	111
6	Algebraische Strukturen: Gruppen, Ringe und Körper	115
6.1	Gruppen	115
6.2	Ringe und Körper	122
6.3	Polynome	124
7	Graphen	130
7.1	Grundlegende Definitionen	130
7.2	Wege, Kreise und Komponenten eines Graphen	133
7.3	Färbungen von Graphen	138
7.4	Bäume und Graphenalgorithmen	141
7.5	Boy meets girl: bipartite Graphen	148
8	Wahrscheinlichkeitsrechnung	156
8.1	Deskriptive Statistik	156
8.2	Grundbegriffe der Wahrscheinlichkeitsrechnung	160

8.3	Zufallsvariablen und Verteilungen	172
8.4	Bedingte Wahrscheinlichkeit	185
9	Analytische Geometrie in der Ebene	193
9.1	Einführung	193
9.2	Vektoren	194
9.3	Winkel, Skalarprodukt und Determinante	202
9.4	Lösung des Problems „Wohin klickt die Maus?“	206
9.5	Geraden	209
10	Analytische Geometrie im Raum	218
10.1	Vektoren im Raum	218
10.2	Ebenen	221
10.3	Spatprodukt, lineare Unabhängigkeit von 3 Vektoren, Basen	230
11	Lineare und affine Abbildungen	233
11.1	2-D-Transformationen in der Computergrafik	233
11.2	Lineare Abbildungen und Matrizen	235
11.3	3-D-Transformationen	245
11.4	Affine Abbildungen und homogene Koordinaten	250
11.5	Inverse Abbildungen	255
12	Vektorräume	258
12.1	Einführung	258
12.2	Vektorräume und Unterräume	261
12.3	Basis, Dimension und lineare Unabhängigkeit	265
13	Lineare Abbildungen und Matrizen	275
13.1	Lineare Abbildungen	275
13.2	Matrizen zur Darstellung linearer Abbildungen	282
14	Der Gauß-Algorithmus	291
14.1	Berechnung des Rangs einer Matrix	291
14.2	Berechnung der Inversen einer Matrix	296
14.3	Lösen linearer Gleichungssysteme	298
15	Fehlerkorrigierende Codes	306
15.1	Grundbegriffe	306
15.2	Lineare Codes	311
15.3	Konstruktion linearer Codes	313
	Symbolverzeichnis	318
	Sachwortverzeichnis	320

1 Aussagenlogik

1.1 Aussagen und logische Junktoren

Stellen Sie sich vor, Sie möchten ein Programm schreiben, das bei Eingabe eines Datums prüft, ob es sich um ein gültiges Datum handelt, und nicht etwa um den 35. März oder den 31. April. Unter anderem müssen Sie dabei prüfen, ob in einem bestimmten Jahr x der 29. Februar ein gültiges Datum ist, das heißt, Sie müssen herausfinden, ob das Jahr x ein Schaltjahr ist. Die Schaltjahrregeln sind recht kompliziert mit Ausnahmen und Ausnahmen von den Ausnahmen und daher ein gutes Beispiel für die Verwendung logischer Ausdrücke.

Die heutige Schaltjahrregelung wurde 1582 mit dem gregorianischen Kalender eingeführt. Sie war notwendig geworden, weil das astronomische Jahr (ein vollständiger Umlauf der Erde um die Sonne) nicht exakt 365 Tage, sondern 365,24219... Tage hat. Sie können selbst ausrechnen, nach wie viel Jahren Weihnachten auf der Nordhalbkugel mitten in den Sommer fallen würde, wenn man diesen Unterschied nicht ausglich. Damit dies nicht passiert, führt man zunächst alle 4 Jahre einen zusätzlichen Schalttag (den 29. Februar) ein. Damit schießt man jedoch ein wenig über das Ziel hinaus, denn mit dieser Regelung käme man im Schnitt auf 365,25 Tage im Jahr. Aus diesem Grund lässt man alle 100 Jahre (also in den Jahren 1800, 1900 usw.) den Schalttag wieder weg. Doch dann ist man wieder leicht unter der Zahl von 365,24219... Tagen pro Jahr. Deshalb fügt man alle 400 Jahre (also in den Jahren 1600, 2000, 2400 usw.) wieder einen Schalttag ein. Rechnen Sie nun selbst aus, wie viele Jahre es dauert, bis der gregorianische Kalender um einen ganzen Tag vom tatsächlichen Wert abweicht (► Aufgabe 1)!

Zur Entscheidung, ob ein gegebenes Jahr x ein Schaltjahr ist, reicht offenbar folgende Information aus: Ist x durch 4 (bzw. 100 bzw. 400) ohne Rest teilbar? Den Rest bei der ganzzahligen Division schreiben wir in der Form $x \% m$. Beispielsweise ist $9 \% 4 = 1$ und $12 \% 4 = 0$. Ist $x \% m = 0$, so ist x (ohne Rest) durch m teilbar. Eine andere Schreibweise für x ist *teilbar durch m* lautet $m|x$ (lies: m ist ein Teiler von x).

Schauen Sie sich folgende umständliche, dennoch korrekte Realisierung der Schaltjahrprüfung in Java an:

```
public boolean schaltjahr(int jahr){
    if (jahr%4 == 0)
        if (jahr%100 == 0)
            if (jahr%400 == 0) return true;
            else return false;
        else return true;
    else return false;
}
```

Geht's vielleicht noch komplizierter? Mal ehrlich: Verstehen Sie die Struktur dieses Programms? Die formale Logik wird uns helfen, solcherart Wildwuchs zu beschneiden. Ein Ziel der nun folgenden Ausführungen soll es sein, eine gut lesbare

und verständliche Schaltjahrformel zu entwickeln und dabei etwas über formale Logik zu lernen.

Aussagen und Aussageformen

Die Grundbausteine der formalen Logik sind die Elemente, die in Java durch die Klasse `boolean` repräsentiert werden. In der Logik heißen sie *Aussagen*. Aussagen können *wahr* oder *falsch* sein. Beispiele für Aussagen in der Programmierung (also Objekte der Klasse `boolean`) sind etwa:

```
n < array.length, jahr%4 == 0, stack.isEmpty()
```

Dagegen sind arithmetische Ausdrücke wie `array.length-1` oder `jahr%4` keine Aussagen. In der Mathematik haben wir es mit Aussagen der Art „7 ist eine Primzahl“ oder „Ist n eine natürliche Zahl, so ist $n^2 + n$ gerade“ zu tun. Das Ergebnis der Auswertung einer Aussage (wahr oder falsch) nennt man auch den *Wahrheitswert* der Aussage.

Der Satz „Heute ist Sonntag“ kann wahr oder falsch sein, jedoch abhängig davon, wann Sie den Satz lesen (oder sagen). Er enthält gewissermaßen eine Variable „Heute“, genauso wie `jahr%4 == 0` eine Variable `jahr` enthält, deren Wert erst bekannt sein muss, damit man den Wahrheitswert der Aussage bestimmen kann. Solche Ausdrücke, in denen Variablen vorkommen, und die ebenfalls wahr oder falsch sein können, heißen *Aussageformen*.

Aussagen können durch sogenannte *logische Junktoren* miteinander verknüpft werden. Die bekanntesten sind „und“, „oder“ und „nicht“. Die Zeichen p und q stehen im Folgenden für beliebige Aussagen.

Die Konjunktion

Das logische „und“, die *Konjunktion*, wird in der Mathematik mit dem Zeichen \wedge geschrieben, in Java wird das Zeichen `&&` benutzt. Die offensichtlich wahre Aussage „12 ist durch 3 und durch 4 teilbar“ besteht aus den beiden Teilaussagen „12 ist durch 3 teilbar“ und „12 ist durch 4 teilbar“, die durch ein „und“ verknüpft sind:

$$(3|12) \wedge (4|12),$$

bzw. in Javanesisch:

```
(12 % 3 == 0) && (12 % 4 == 0).
```

Der Ausdruck $p \wedge q$ ist genau dann wahr, wenn sowohl p als auch q wahr ist. Wir stellen dies mithilfe einer Verknüpfungstafel, der sogenannten *Wahrheitstafel*, dar. Dabei wird der Wahrheitswert „wahr“ durch 1, der Wahrheitswert „falsch“ durch 0 dargestellt.

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Die Disjunktion

Das logische „oder“, die *Disjunktion*, wird in der Mathematik mit dem Zeichen \vee geschrieben, in Java wird das Zeichen `||` benutzt. Die offensichtlich wahre Aussage „6 ist durch 3 oder durch 4 teilbar“ wird dargestellt durch:

$$(3|6) \vee (4|6)$$

bzw. in Javanesisch:

$$(6 \% 3 == 0) \ || \ (6 \% 4 == 0) .$$

Der Ausdruck $p \vee q$ ist genau dann wahr, wenn mindestens eine der beiden Aussagen p und q wahr ist. Wir stellen dies mithilfe einer Verknüpfungstafel dar:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Auch hier ist wieder Vorsicht angesagt mit der Übersetzung umgangssprachlicher Formulierungen. Wenn zu Ihnen jemand sagt: „Heute Abend gehe ich ins Theater *oder* ins Kino“, dann können Sie mit ziemlicher Sicherheit davon ausgehen, dass er eigentlich meint: „Heute Abend gehe ich *entweder* ins Theater *oder* ins Kino“. Dieses „ausschließende Oder“ heißt in der mathematischen Logik auch *exklusives Oder (XOR)*. Das „Oder“, das durch das Symbol \vee dargestellt wird, heißt *inklusives Oder*.

Die Negation

Das logische „Nicht“, die *Negation*, wird in der Mathematik mit dem Zeichen \neg geschrieben, in Java wird das Zeichen `!` benutzt. Die wahre Aussage „6 ist nicht durch 4 teilbar“ wird dargestellt durch:

$$\neg(4|6)$$

bzw. in Javanesisch:

$$!(6 \% 4 == 0)$$

oder noch einfacher durch $6 \% 4 \neq 0$.

Der Ausdruck $\neg p$ ist genau dann wahr, wenn p falsch ist:

p	$\neg p$
0	1
1	0

Wie lautet die Negation von „Die Flasche ist voll“? Nein, nicht „Die Flasche ist leer“, sondern „Die Flasche ist nicht voll“! Auch mit der Negation muss man ein wenig aufpassen.

Die Implikation

Das logische „wenn, ... dann“, die *Implikation*, wird in der Mathematik mit dem Zeichen \rightarrow geschrieben. Die Programmiersprache Java kennt kein Zeichen für die Implikation. Die wahre Aussageform „wenn x durch 6 teilbar ist, dann ist x durch 3 teilbar“ wird dargestellt durch:

$$6|x \rightarrow 3|x.$$

Der Ausdruck $p \rightarrow q$ ist genau dann falsch, wenn p wahr und q falsch ist:

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Die logische Implikation macht erfahrungsgemäß die meisten Probleme bei der Übersetzung umgangssprachlicher Sätze. Das liegt oft daran, dass man zwar „wenn, ... dann“ sagt, in Wirklichkeit jedoch eine andere logische Verknüpfung meint, ähnlich wie bei dem Satz „Heute Abend gehe ich ins Kino oder ins Theater“, der eigentlich ein *exklusives oder* meint. Nehmen wir an, jemand sagt: „Wenn ich 10000 Euro gespart habe, dann mache ich eine Weltreise.“ Damit meint er mit ziemlicher Sicherheit aber *mehr* als die logische Implikation. Er will damit nicht nur sagen, dass er eine Weltreise macht, wenn er genug Geld hat, sondern es heißt auch umgekehrt: Wenn er nicht genug Geld hat, dann fällt die Weltreise eben aus. Er verwendet das „wenn, ... dann“ im Sinne einer logischen Biimplikation (► nächster Abschnitt). Im alltäglichen Sprachgebrauch sind beide Bedeutungen des „wenn, ... dann“ üblich, und genau das führt zu Missverständnissen. Der Satz: „Wenn es regnet, (dann) ist die Straße nass“ meint eindeutig die logische Implikation. Ihn kann man nicht umkehren zu „Wenn es nicht regnet, dann ist die Straße nicht nass“, denn es könnte ja auch jemand die Straße mit dem Gartenschlauch wässern.

Das umgangssprachliche „wenn, ... dann“ unterscheidet sich in einem zweiten Aspekt von der logischen Implikation. Meistens schwingt im „wenn, ... dann“ ein kausaler oder finaler Kontext mit: „Wenn ich auf den Schalter drücke, dann geht das Licht an“, dieser Satz meint auch: „Das Licht geht an, *weil* ich auf den Schalter drücke.“ Man erwartet meist einen inhaltlichen Zusammenhang zwischen den beiden Sätzen, die durch „wenn, ... dann“ verbunden sind. Was meinen Sie zu dem Satz „Wenn Paris die Hauptstadt von Italien ist, dann ist Rom die Hauptstadt von Frankreich.“ Sinnlos, nicht wahr? Doch als logische Aussage ist der Satz wahr. Die erste Zeile der Wahrheitstafel besagt nämlich: Wenn sowohl p als auch q falsch ist, dann ist $p \rightarrow q$ wahr! Und das gilt sogar noch, wenn p falsch und q wahr ist (zweite Zeile). Man kann also sagen: Ist p falsch, so ist die Implikation auf jeden Fall wahr, unabhängig davon, ob q wahr oder falsch ist. Man nennt diesen Sachverhalt oft auch (lateinisch) *ex falso quodlibet*, d. h., aus einer falschen Aussage kann man alles folgern.

Ganz fremd ist aber der Umgangssprache der logische Gebrauch der Implikation nicht, wenn Sie sich folgende Redewendung vor Augen halten: „Wenn Ouagadougou

die Hauptstadt der Schweiz ist, dann bin ich der Kaiser von China.“ Der Satz ist tatsächlich wahr – egal ob er von Ihnen oder vom chinesischen Kaiser höchstselbst ausgesprochen wird.

Die Biimplikation

Das logische „genau dann ..., wenn“, die *Biimplikation*, wird in der Mathematik mit dem Zeichen \leftrightarrow geschrieben. Auch diese logische Verknüpfung gibt es in Java nicht. Die wahre Aussageform „ x ist genau dann durch 6 teilbar, wenn x durch 3 und durch 2 teilbar ist“ wird dargestellt durch:

$$6|x \leftrightarrow (2|x \wedge 3|x).$$

Der Ausdruck $p \leftrightarrow q$ ist genau dann wahr, wenn p und q denselben Wahrheitswert haben:

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

Sheffer- und Peirce-Operator

Wichtig für die Schaltalgebra, jedoch weniger gebräuchlich in der formalen Logik sind der Sheffer-Operator $|$ und der Peirce-Operator \downarrow .

Der Ausdruck $p | q$ ist genau dann falsch, wenn p und q wahr sind. Der Ausdruck $p \downarrow q$ ist genau dann wahr, wenn p und q falsch sind:

p	q	$p q$	$p \downarrow q$
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Der Sheffer-Operator entspricht dem NAND-Gatter der Schaltungslogik, und der Peirce-Operator entspricht dem NOR-Gatter (► Abbildung 1-1 auf Seite 29).

Logische Formeln

Mit den genannten Junktoren lassen sich beliebige logische Formeln (genauer gesagt: *aussagenlogische* Formeln) zusammensetzen, etwa

$$p \rightarrow (q \vee r)$$

oder

$$(p \rightarrow q) \rightarrow (q \rightarrow p).$$

Um Klammern einzusparen, vereinbart man ähnlich wie die Regel „Punkt vor Strich“ folgende Vorrangregeln für die Junktoren:

- Der Operator \neg bindet am stärksten.
- Die Operatoren \vee , \wedge binden stärker als \rightarrow und \leftrightarrow .

Zwischen \vee und \wedge ebenso wie zwischen \rightarrow und \leftrightarrow sind jedoch keine Vorrangregeln gesetzt. Da müssen Sie also auf jeden Fall Klammern setzen. Beispielsweise bedeutet $((p) \vee q) \rightarrow r$ dasselbe wie $p \vee q \rightarrow r$, während dagegen der Ausdruck $p \vee q \wedge r$ nicht eindeutig definiert ist. Es gibt Autoren, die der Konjunktion eine höhere Bindungskraft einräumen als der Disjunktion und der Implikation eine höhere als der Biimplikation. Dadurch könnte man auf die Klammern im Ausdruck $p \vee (q \wedge r)$ verzichten. Ich halte das jedoch für keine gute Idee, denn diese Regelung hat keine klare und einfach zu merkende Regel wie „Punkt vor Strich“. Aus leidvoller Erfahrung bei der Korrektur von Klausuren kann ich Ihnen nur abraten, hier an der falschen Stelle zu sparen (an den Klammern nämlich).

Falls Sie sich nicht sicher sind, so halten Sie sich am besten an die Regel: Ein Klammerpaar zu viel schadet nicht, ein Klammerpaar zu wenig kann jedoch alles falsch machen.

Der Wahrheitswert einer zusammengesetzten Formel lässt sich bestimmen, indem sukzessive deren Teilformeln ausgewertet werden.

Beispiel 1

Wir erstellen die Wahrheitstafel der Formel $(p \wedge q) \vee (p \wedge \neg q)$:

p	q	p	q	$p \wedge q$	$p \wedge \neg q$	$(p \wedge q) \vee (p \wedge \neg q)$
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

Aufgaben zu 1.1

- 1 Wie viele Jahre dauert es, bis der gregorianische Kalender um einen ganzen Tag vom tatsächlichen Wert abweicht?
- 2 Welche der folgenden Ausdrücke sind Aussagen, welche sind Aussageformen?
 - a) $x^2 + 1 > 0$
 - b) Tobias ist älter als Marlene.
 - c) $x^2 + 3x - 5$
 - d) Wie spät ist es?
- 3 Formulieren Sie die folgenden umgangssprachlichen Sätze zunächst in der „wenn, ... dann“-Form. Anschließend bilden Sie jeweils eine logische Formel unter Verwendung der Aussagen $p =$ „Es ist Freitag“ und $q =$ „Ich gehe ins Kino“.

- a) Ich gehe jeden Freitag ins Kino.
 b) Ich gehe nur freitags ins Kino.
 c) Freitags gehe ich nie ins Kino.
- 4 Erstellen Sie eine Wahrheitstafel für das *exklusive oder* („Ich gehe entweder ins Kino oder ins Theater“).
- 5 Erstellen Sie eine Wahrheitstafel für *weder ... noch* („Ich gehe weder ins Kino noch ins Theater“).
- 6 Wie viele verschiedene logische Junktoren (d.h. Verknüpfungen zwischen zwei Aussagenvariablen) kann es geben? Stellen Sie alle möglichen Wahrheitstafeln auf!
- 7 Erstellen Sie Wahrheitstafeln für folgende Formeln.
- a) $p \vee (p \rightarrow q)$
 b) $p \vee q \rightarrow p \wedge q$
 c) $p \rightarrow p$
 d) $(p \rightarrow q) \rightarrow r$
 e) $p \rightarrow (q \rightarrow r)$
- 8 Sei n eine natürliche Zahl. Wie viele Zeilen hat die Wahrheitstafel einer Formel, in der n Aussagenvariablen vorkommen?

1.2 Rechnen mit logischen Formeln

Wir erstellen die Wahrheitstafel der Formel $p \vee \neg p$:

p	$\neg p$	$p \vee \neg p$
0	1	1
1	0	1

Diese Formel ist offenbar stets wahr, ganz egal, ob p wahr oder falsch ist. Erstaunt Sie das? Setzen Sie doch einfach irgendeine Aussage für p ein, etwa „Es regnet“: Dann wird daraus „Es regnet oder es regnet nicht“. Diese Wettervorhersage ist keine große Kunst! Eine Formel, die stets wahr ist, heißt *Tautologie*.

Die Formel F heißt *Tautologie*, wenn in jeder Zeile ihrer Wahrheitstafel der Wert 1 (wahr) steht. Die Formel F heißt *Kontradiktion*, wenn in jeder Zeile ihrer Wahrheitstafel der Wert 0 (falsch) steht.

Definition
 Tautologie,
 Kontradiktion

Eine Tautologie ist stets wahr, und eine Kontradiktion ist stets falsch, unabhängig vom Wahrheitswert der Aussagen, aus denen sie bestehen.

Beispiel 2

a) Wir erstellen die Wahrheitstafel der Formel $p \rightarrow (p \rightarrow q)$:

p	q	p	$p \rightarrow q$	$p \rightarrow (p \rightarrow q)$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	1
1	1	0	1	1

Diese Formel ist ebenfalls eine Tautologie. Können Sie erkennen, wieso das so ist? Bei dieser Formel handelt es sich um eine „Übersetzung“ des *ex falso quodlibet*, das heißt der Regel: Wenn p falsch ist, dann ist die Implikation auf jeden Fall wahr, unabhängig davon, ob q wahr oder falsch ist.

b) Wir erstellen die Wahrheitstafel der Formel $(p \rightarrow q) \rightarrow q$:

p	q	$p \rightarrow q$	$(p \rightarrow q) \rightarrow q$
0	0	1	0
0	1	1	1
1	0	0	1
1	1	1	1

Kommt Ihnen diese Tafel bekannt vor? Richtig, die Ergebnisspalte für $(p \rightarrow q) \rightarrow q$ ist dieselbe wie der Disjunktion $p \vee q$. Wir sagen, die beiden Formeln $(p \rightarrow q) \rightarrow q$ und $p \vee q$ sind *logisch äquivalent*. ■

Metalogische Symbole

Wir bezeichnen Formeln im Folgenden mit großen Buchstaben, vorzugsweise F und G .

Definition
Logische
Äquivalenz

Die beiden Formeln F und G heißen (*logisch*) *äquivalent*, wenn sie in jeder Zeile ihrer Wahrheitstabellen übereinstimmen. Wir schreiben $F \Leftrightarrow G$.

Wir können daher schreiben: $(p \rightarrow q) \rightarrow q \Leftrightarrow p \vee q$ (► Beispiel 2b). Das Symbol \Leftrightarrow ist im Gegensatz zu \leftrightarrow kein logischer Junktoren. Es ist vielmehr ein *metalogisches* Zeichen, das heißt ein Zeichen der Sprache, die über logische Formeln spricht.

Das Äquivalenzzeichen \Leftrightarrow wird in der Mathematik häufig verwendet, wenn äquivalente Umformungen durchgeführt werden, etwa beim Rechnen mit Gleichungen:

$$x + 3 = 7 \Leftrightarrow x = 4.$$

In diesem Buch verwende ich statt des Zeichens \Leftrightarrow häufig die Formulierung *genau dann ..., wenn*.

Die beiden Zeichen \leftrightarrow und \Leftrightarrow sind eng miteinander verknüpft:

Die beiden Formeln F und G sind genau dann logisch äquivalent, wenn die Formel $F \leftrightarrow G$ eine Tautologie ist.

Satz

Die besondere Bedeutung der logischen Äquivalenz liegt darin, dass man in einer Formel Teilformeln durch logisch äquivalente Formeln ersetzen kann, ohne den Wahrheitswert der Formel zu ändern. Man kann dann mit Äquivalenzen rechnen wie mit Gleichungen, beispielsweise kann man Äquivalenzen benutzen, um Formeln zu vereinfachen.

In Analogie zu dem Zeichenpaar \leftrightarrow und \Leftrightarrow gibt es auch das Zeichenpaar \rightarrow und \Rightarrow . Das Zeichen \Rightarrow ist ebenfalls ein metalogisches Symbol. Wir vereinbaren, dass die metalogischen Symbole noch schwächer binden als die entsprechenden logischen Symbole.

Die Formel G heißt (*logische*) *Konsequenz* der Formel F , wenn in jeder Zeile der Wahrheitstafel, in der F wahr ist, auch G wahr ist. Wir schreiben $F \Rightarrow G$.

Definition
Konsequenz

Es gilt: Die Formel G ist eine Konsequenz der Formel F , wenn die Formel $F \rightarrow G$ eine Tautologie ist, und das ist genau dann der Fall, wenn die Formel $F \wedge \neg G$ eine Kontradiktion ist. Insbesondere gilt: Ist F eine Kontradiktion (das heißt, immer falsch), so ist jede beliebige Formel G eine Konsequenz von F , denn $F \wedge \neg G$ ist immer eine Kontradiktion unabhängig von G . Diese Tatsache ist wiederum nichts anderes als das *ex falso quodlibet*. Spielen Sie Sudoku? Dann kennen Sie das Phänomen sicherlich: Wenn Sie irgendwann eine falsche Schlussfolgerung gezogen und als Folge eine falsche Zahl eingetragen haben, dann können Sie alles, was Sie danach eingetragen haben, vergessen.

Im Hinblick auf Beispiel 2 a) können wir sagen: Die Formel $p \rightarrow q$ ist eine logische Konsequenz der Formel $\neg p$: Wenn die Aussage p falsch ist, dann ist die Formel $p \rightarrow q$ wahr, bzw. $\neg p \Rightarrow p \rightarrow q$.

Das Konsequenzzeichen wird in der Mathematik häufig für Umformungen verwendet, die keine Äquivalenzumformungen sind, etwa beim Rechnen mit Gleichungen:

$$x = -2 \Rightarrow x^2 = 4.$$

Dabei ist wichtig, dass der Implikationspfeil nicht umgedreht werden kann. Im Beispiel folgt eben aus $x^2 = 4$ nicht $x = -2$, denn x könnte auch 2 sein.

Mithilfe der logischen Äquivalenz können wir ausdrücken, dass zwei Formeln logisch gesehen gleich sind. Beispielsweise ist die Biimplikation $p \leftrightarrow q$ (wie der Name ebenso wie das Symbol schon andeuten) „nichts anderes“ als eine Implikation in beiden Richtungen:

$$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p).$$

Dies lässt sich einfach durch Vergleich der beiden Wahrheitstabellen für $p \leftrightarrow q$ und $(p \rightarrow q) \wedge (q \rightarrow p)$ feststellen. Betrachten Sie als Beispiel die Aussageform:

$$6|x \leftrightarrow 2|x \wedge 3|x$$

Tabelle 1-1
Rechenregeln der
Aussagenlogik

$F \vee G \Leftrightarrow G \vee F$	①	$F \wedge G \Leftrightarrow G \wedge F$
$(F \vee G) \vee H \Leftrightarrow F \vee (G \vee H)$	②	$(F \wedge G) \wedge H \Leftrightarrow F \wedge (G \wedge H)$
$F \wedge (G \vee H) \Leftrightarrow (F \wedge G) \vee (F \wedge H)$	③	$F \vee (G \wedge H) \Leftrightarrow (F \vee G) \wedge (F \vee H)$
$(F \vee G) \Leftrightarrow F \wedge G$	④	$(F \wedge G) \Leftrightarrow F \vee G$
$F \wedge (F \vee G) \Leftrightarrow F$	⑤	$F \vee (F \wedge G) \Leftrightarrow F$
$F \vee F \Leftrightarrow F$	⑥	$F \wedge F \Leftrightarrow F$
$F \vee 1 \Leftrightarrow 1$	⑦	$F \wedge 0 \Leftrightarrow 0$
$F \vee 0 \Leftrightarrow F$	⑧	$F \wedge 1 \Leftrightarrow F$
$F \vee F \Leftrightarrow 1$	⑨	$F \wedge F \Leftrightarrow 0$
$F \Leftrightarrow F$	⑩	
$F \rightarrow G \Leftrightarrow F \vee G$	⑪	
$F \leftrightarrow G \Leftrightarrow (F \vee G) \wedge (G \vee F)$	⑫	
$F G \Leftrightarrow (F \wedge G)$	⑬	
$F \downarrow G \Leftrightarrow (F \vee G)$	⑭	

„Eine Zahl ist genau dann durch 6 teilbar, wenn sie durch 2 und durch 3 teilbar ist.“
Dies ist logisch dasselbe wie: „Jede Zahl, die durch 6 teilbar ist, ist durch 2 und durch 3 teilbar und umgekehrt.“

$$(6|x \rightarrow 2|x \wedge 3|x) \wedge (2|x \wedge 3|x \rightarrow 6|x).$$

Logische Äquivalenzen können wie Rechenregeln benutzt werden, um Formeln zu vereinfachen. Tabelle 1-1 listet einige nützliche Rechenregeln auf. Wir führen dazu zwei logische Konstanten 1 und 0 ein, deren Wahrheitswert 1 bzw. 0 ist. Jede Tautologie ist äquivalent zu 1 und jede Kontradiktion ist äquivalent zu 0.

Wenn Sie die Regeln 1 bis 10 genau betrachten, wird Ihnen sicher auffallen, dass in jeder Zeile die Formel auf der linken Seite und die Formel auf der rechten Seite durch Vertauschen der Junktoren \vee und \wedge sowie durch Vertauschen der Konstanten 0 und 1 ineinander übergehen. Man nennt dies *Dualisieren*: Ist F eine Formel, die außer \vee , \wedge und keine weiteren Junktoren enthält, so entsteht die zu F duale Formel F' , indem man in $F \vee$ und \wedge sowie 0 und 1 miteinander vertauscht. Es gilt: Ist F eine Tautologie, so ist auch die duale Formel F' eine Tautologie.

Alle diese Äquivalenzen lassen sich durch Konstruktion der Wahrheitstabellen beweisen.

Die Regeln 11 bis 14 können benutzt werden, um das Implikationszeichen, Biimplikationszeichen, sowie Sheffer- und Peirce-Operator vollständig aus einer Formel zu eliminieren. Man kann daher stets mit Formeln arbeiten, die nur aus Disjunktion, Konjunktion und Negation aufgebaut sind.

Es genügt sogar ein einziger Junktor, nämlich der Sheffer-Operator (oder wahlweise der Peirce-Operator), um sämtliche Formeln darzustellen. Wie Sie in der folgenden Tabelle sehen, können Negation, Konjunktion und Disjunktion komplett ersetzt werden durch Formeln, die nur den Sheffer-Operator (bzw. den Peirce-Operator) enthalten.

$$\begin{aligned} F \wedge G &\Leftrightarrow (F|G)|(F|G) &\Leftrightarrow (F \downarrow F) \downarrow (G \downarrow G) \\ F \vee G &\Leftrightarrow (F|F)|(G|G) &\Leftrightarrow (F \downarrow G) \downarrow (F \downarrow G) \\ F &\Leftrightarrow F|F &\Leftrightarrow F \downarrow F \end{aligned}$$

Diese Tatsache ist besonders wichtig für den Entwurf logischer Schaltungen. Sie besagt, dass eine einzige Sorte von Bauteilen, nämlich das NAND-Gatter oder das NOR-Gatter, genügt, um sämtliche logische Schaltungen zu realisieren.

Es folgen einige Beispiele für das „Rechnen“ mit logischen Formeln.

Beispiel 3

a) Die Formel $(p \rightarrow q)$ soll vereinfacht werden:

$$(p \rightarrow q) \stackrel{\textcircled{11}}{\Leftrightarrow} (\neg p \vee q) \stackrel{\textcircled{4}}{\Leftrightarrow} p \wedge \neg q \stackrel{\textcircled{10}}{\Leftrightarrow} p \wedge \neg q.$$

b) Die Formel $p \wedge (p \rightarrow q)$ soll vereinfacht werden:

$$p \wedge (p \rightarrow q) \stackrel{\textcircled{11}}{\Leftrightarrow} p \wedge (\neg p \vee q) \stackrel{\textcircled{3}}{\Leftrightarrow} (p \wedge \neg p) \vee (p \wedge q) \stackrel{\textcircled{9}}{\Leftrightarrow} 0 \vee (p \wedge q) \stackrel{\textcircled{8}}{\Leftrightarrow} p \wedge q.$$

Beispiel 4 Die Schaltjahrformel

Wir versuchen nun, die Schaltjahrformel zu vereinfachen. Dazu müssen wir die Konstruktion `if (p) q else r` in Aussagenlogik übersetzen. Die Übersetzung lautet:

$$(p \rightarrow q) \wedge (\neg p \rightarrow r).$$

Dies ist selbstverständlich nur möglich, wenn q und r boolesche Ausdrücke sind. Als Spezialfälle erhalten wir für `if (p) q else false`:

$$(p \rightarrow q) \wedge (\neg p \rightarrow 0) \stackrel{\textcircled{11}}{\Leftrightarrow} (p \rightarrow q) \wedge (\neg p \vee 0) \stackrel{\textcircled{10}}{\Leftrightarrow} (p \rightarrow q) \wedge (\neg p \vee 0) \stackrel{\textcircled{8}}{\Leftrightarrow} (p \rightarrow q) \wedge \neg p$$

sowie für `if (p) q else true`:

$$(p \rightarrow q) \wedge (\neg p \rightarrow 1) \stackrel{\textcircled{11}}{\Leftrightarrow} (p \rightarrow q) \wedge (\neg p \vee 1) \stackrel{\textcircled{7}}{\Leftrightarrow} (p \rightarrow q) \wedge 1 \stackrel{\textcircled{8}}{\Leftrightarrow} (p \rightarrow q)$$

und schließlich für `if (p) true else false`:

$$(p \rightarrow 1) \wedge (\neg p \rightarrow 0) \stackrel{\textcircled{11}}{\Leftrightarrow} (\neg p \vee 1) \wedge (\neg p \vee 0) \stackrel{\textcircled{7}}{\Leftrightarrow} 1 \wedge (\neg p \vee 0) \stackrel{\textcircled{8}}{\Leftrightarrow} \neg p \vee 0 \stackrel{\textcircled{10}}{\Leftrightarrow} \neg p \vee 0 \stackrel{\textcircled{8}}{\Leftrightarrow} \neg p.$$

Wir setzen nun als Abkürzung $p_4 := \text{jahr}\%4 == 0$, $p_{100} := \text{jahr}\%100 == 0$, $p_{400} := \text{jahr}\%400 == 0$ und erhalten für den gesamten Ausdruck:

```

if (p4)
  {if (p100)
    {if (p400) {true;}
     else {false;}}
   else {true;}}
else {false;}

```

Den inneren Ausdruck `if (p400) {true;} else {false;}` ersetzen wir durch p_{400} :

```

if (p4)
  {if (p100)
    {p400}
   else {true;}
   else {false;}

```

Nun ersetzen wir den Ausdruck `if (p100) {p400} else {false;}` durch $p_{100} \rightarrow p_{400}$:

```

if (p4)
  (p100 → p400)
else {false;}

```

Im letzten Schritt erhalten wir:

$$(p_4 \rightarrow (p_{100} \rightarrow p_{400})) \wedge p_4.$$

Nun vereinfachen wir weiter nach den logischen Gesetzen:

$$\begin{aligned}
(p_4 \rightarrow (p_{100} \rightarrow p_{400})) \wedge p_4 &\stackrel{\textcircled{11}}{\Leftrightarrow} (p_4 \vee (p_{100} \vee p_{400})) \wedge p_4 \\
&\stackrel{\textcircled{3}}{\Leftrightarrow} (p_4 \wedge p_4) \vee ((p_{100} \vee p_{400}) \wedge p_4) \\
&\stackrel{\textcircled{9}}{\Leftrightarrow} 0 \vee ((p_{100} \vee p_{400}) \wedge p_4) \\
&\stackrel{\textcircled{8}}{\Leftrightarrow} (p_{100} \vee p_{400}) \wedge p_4
\end{aligned}$$

und zum guten Schluss rückübersetzt in javanesisch:

```

public boolean schaltjahr(int jahr) {
  return jahr%4 == 0 && (jahr%100 != 0 || jahr%400 == 0);
}

```

Stimmt das denn nun? Haben wir bei den Umformungen keinen Fehler gemacht? Versuchen wir die Formel wieder in normales Deutsch umzuwandeln: Ein Schaltjahr muss zwei Bedingungen erfüllen: 1. Es muss durch 4 teilbar sein und 2. es darf nicht durch 100 teilbar sein oder es muss durch 400 teilbar sein. Stimmt! ■

Aufgaben zu 1.2

1 Bilden Sie die Negation der folgenden Aussagen und formulieren Sie diese möglichst einfach und in gutem Deutsch (also *nicht* nach dem Muster: „Es stimmt nicht, dass Claudia Gitarre und Saxophon spielen kann“):

- a) Claudia kann Gitarre und Saxophon spielen.
- b) Christoph studiert Mathematik oder Philosophie.
- c) Heute abend gehe ich entweder ins Kino oder ins Theater.
- d) Wenn Sonja älter als Paul ist, dann ist sie auch älter als Christoph.
- e) Jeder Student ist arm.

2 Welche der folgenden Formeln sind Tautologien, welche sind Kontradiktionen?

- a) $p \rightarrow (q \rightarrow p)$
- b) $p \vee (p \rightarrow q)$
- c) $q \vee (p \rightarrow q)$
- d) $p \wedge q \wedge (p \rightarrow q)$

3 Beweisen Sie, dass die Formeln F und G jeweils äquivalent sind.

	F	G
a)	$p \rightarrow q$	$q \rightarrow p$
b)	$p \leftrightarrow q$	$(p \wedge q) \vee (\neg p \wedge \neg q)$
c)	$(p \leftrightarrow q)$	$p \leftrightarrow q$
d)	$p \rightarrow (q \rightarrow r)$	$p \wedge q \rightarrow r$

4 Beweisen Sie, dass die Formel G jeweils eine Konsequenz von F ist.

	F	G
a)	$(p \rightarrow q) \wedge (q \rightarrow r)$	$p \rightarrow r$
b)	$(p \rightarrow q)$	p
c)	q	$p \rightarrow q$
d)	$p \wedge (p \rightarrow q)$	q

5 Vereinfachen Sie folgende Formeln mithilfe der Regeln für logische Formeln.

- a) $p \rightarrow (p \rightarrow q)$
- b) $p \vee (q \wedge \neg p)$
- c) $(p \wedge q) \vee (\neg p \wedge q)$
- d) $p \wedge (q \vee (r \wedge \neg p))$
- e) $p \wedge (q \vee (r \wedge p))$

6 Vereinfachen Sie den folgenden Ausdruck:

```

if (p)
  {if (q) {p} else {false;}}
else {if (!q) {false;} else {true;}}

```

1.3 Normalformen und Vereinfachung von Formeln

Es gibt noch einen anderen Weg als den soeben beschriebenen, um die Schaltjahrformel aufzustellen. Man könnte für jede Kombination von Wahrheitswerten der drei Basisaussagen p_4 , p_{100} und p_{400} aufschreiben, ob die Schaltjahrformel wahr oder falsch sein müsste. Ist beispielsweise p_4 wahr, p_{100} wahr und p_{400} falsch, so ist das Jahr kein Schaltjahr. Auf diese Weise konstruiert man die komplette Wahrheitstafel der Formel (die man ja als Formel noch gar nicht kennt).

Aufgabe 1 Konstruieren Sie auf die genannte Weise die Wahrheitstafel der Schaltjahrformel F_S .

Lösung

p_4	p_{100}	p_{400}	F_S
0	0	0	0
0	0	1	–
0	1	0	–
0	1	1	–
1	0	0	1
1	0	1	–
1	1	0	0
1	1	1	1

Die Striche bedeuten, dass die zugehörige Kombination von Wahrheitswerten der Basisaussagen gar nicht möglich ist. Ist etwa p_{400} wahr, so kann p_{100} gar nicht falsch sein. ■

Bislang haben wir zu einer gegebenen Formel die Wahrheitstafel aufgestellt. Nun gehen wir den umgekehrten Weg und konstruieren aus der Wahrheitstafel die (besser gesagt, *eine*) Schaltjahrformel. Zu diesem Zweck betrachten wir die beiden Zeilen, in denen F_S wahr ist. In der fünften Zeile lässt sich folgende Formel ablesen:

$$p_4 \wedge p_{100} \wedge p_{400}.$$

Das ist der Schaltjahrfall, in dem das Jahr durch 4, aber nicht durch 100 und nicht durch 400 teilbar ist. Dieser Fall lässt sich noch einfacher beschreiben durch „Das Jahr ist durch 4, aber nicht durch 100 teilbar“, denn in diesem Fall folgt automatisch, dass es auch nicht durch 400 teilbar sein kann. Die Formel reduziert sich damit zu

$$p_4 \wedge p_{100}.$$

Aus der achten Zeile folgt:

$$p_4 \wedge p_{100} \wedge p_{400}.$$

Wenn p_{400} wahr ist, dann sind auch p_{100} und p_4 wahr. Die Formel in Zeile 8 reduziert sich dann zu p_{400} .

Insgesamt ergibt sich folgende Schaltjahrformel:

$$(p_4 \wedge \neg p_{100}) \vee p_{400}.$$

Dies bedeutet rückübersetzt in Alltagssprache, dass es zwei Schaltjahrfälle gibt. Fall 1: Ist das Jahr durch 4 teilbar, aber nicht durch 100, dann ist es ein Schaltjahr. Fall 2: Ist das Jahr durch 400 teilbar, so ist es ein Schaltjahr.

Diese Formel unterscheidet sich von der in Beispiel 3 erarbeiteten Formel. Die beiden Formeln sind jedoch logisch äquivalent, wenn man wieder die Abhängigkeiten zwischen den Formeln p_{400} , p_{100} und p_4 berücksichtigt.

Beispiel 5 Ein Kriminalfall

Inspektor Quak hat drei Männer, A , B und C , unter Verdacht auf einen Einbruch verhaftet. Fest steht, dass mindestens einer der Drei an der Tat beteiligt war, und dass außer den Dreien niemand als Täter infrage kommt. Der Inspektor kennt seine Pappenheimer und weiß:

- Sowohl A als auch B „arbeiten“ nie allein.
- B arbeitet nur dann mit A zusammen, wenn auch C dabei ist.

Wir erstellen die Wahrheitstafel. Dabei bedeutet p_A : A ist schuldig, p_B : B ist schuldig, p_C : C ist schuldig. Die Formel F beschreibt einen möglichen Tathergang.

p_A	p_B	p_C	F	
0	0	0	0	
0	0	1	1	$p_A \wedge \neg p_B \wedge p_C$
0	1	0	0	
0	1	1	1	$p_A \wedge p_B \wedge p_C$
1	0	0	0	
1	0	1	1	$p_A \wedge \neg p_B \wedge p_C$
1	1	0	0	
1	1	1	1	$p_A \wedge p_B \wedge p_C$

Daraus ergibt sich folgende etwas unübersichtliche Formel:

$$(p_A \wedge \neg p_B \wedge p_C) \vee (p_A \wedge p_B \wedge p_C) \vee (\neg p_A \wedge p_B \wedge p_C) \vee (\neg p_A \wedge \neg p_B \wedge p_C).$$

Bevor wir darangehen, diese Formel zu vereinfachen, halten wir die besondere Form dieser Formel fest. In jeder Zeile der Wahrheitstafel steht eine Konjunktion, in der jede Aussagenvariable genau einmal (negiert oder unnegiert) vorkommt. Wir nennen eine solche spezielle Konjunktion einen *vollständigen Minterm*. Die gesamte Formel ist eine Disjunktion von vollständigen Mintermen. Wir wollen im Folgenden auch unvollständige Minterme zulassen, das sind Minterme, die nicht notwendigerweise alle Aussagevariablen enthalten.

Definition
Disjunktive
Normalform

Ein *Literal* ist eine Aussagenvariable oder deren Negation. Zwei Literale der Form p und $\neg p$ heißen *komplementär*.

Ein *Minterm* ist eine Konjunktion von Literalen, die jede Aussagenvariable *höchstens einmal* enthält.

Ein Minterm heißt *vollständig*, wenn er jede Aussagenvariable der Gesamtformel *genau einmal* enthält.

Ein Formel F heißt in *disjunktiver Normalform (DNF)*, wenn sie eine Disjunktion von Mintermen ist. Sind alle Minterme vollständig, so heißt F in *vollständiger DNF*.

Dabei ist zu beachten, dass eine Konjunktion und eine Disjunktion auch aus nur einem einzigen Literal bestehen kann! Beispiele für Formeln in DNF sind:

$$p, \quad q, \quad (p \wedge q \wedge r) \vee (\neg p \wedge q \wedge r), \quad p \vee q, \quad p \vee (q \wedge r).$$

Die ersten drei Formeln sind sogar in vollständiger DNF, die beiden letzten dagegen nicht. Welche Bedeutung hat die Einschränkung, dass ein Minterm jede Aussagenvariable höchstens einmal enthält? Das heißt, dass folgende Fälle ausgeschlossen sind:

- identische Literale, etwa $p \wedge q \wedge p \wedge r$,
- komplementäre Literale, etwa $p \wedge q \wedge \neg p \wedge r$.

Im ersten Fall kann man nämlich eines der beiden Literale löschen, im zweiten Fall ist der ganze Minterm äquivalent zur Kontradiktion und kann daher in der Endformel vernachlässigt werden. Wenn Sie die disjunktive Normalform aus der Wahrheitstafel heraus erstellen, können die beiden genannten Fälle sowieso nicht auftreten.

Die Definition der *konjunktiven Normalform (KNF)* erhalten Sie, indem Sie in der obigen Definition der DNF einfach jeweils „Konjunktion“ durch „Disjunktion“ und „Minterm“ durch „Maxterm“ ersetzen und umgekehrt. Beispiele für Formeln in KNF erhalten Sie, indem Sie in den obigen Beispielen die Zeichen \vee und \wedge vertauschen.

Das Verfahren, mit dem man aus der Wahrheitstafel einer Formel die konjunktive Normalform abliest, funktioniert analog zum Verfahren zur Erstellung der DNF. Wir zeigen dies am Beispiel unseres Kriminalfalles. Dieses Mal betrachten wir ausschließlich die Zeilen, in denen die Formel falsch ist:

p_A	p_B	p_C	F	
0	0	0	0	$p_A \vee p_B \vee p_C$
0	0	1	1	
0	1	0	0	$p_A \vee p_B \vee p_C$
0	1	1	1	
1	0	0	0	$p_A \vee p_B \vee p_C$
1	0	1	1	
1	1	0	0	$p_A \vee p_B \vee p_C$
1	1	1	1	

Die erste Zeile besagt: Sind p_A, p_B, p_C alle falsch, so ist auch die Gesamtformel F falsch. Damit F überhaupt wahr sein kann, muss also p_A oder p_B oder p_C wahr sein (mindestens einer der Drei war an der Tat beteiligt).

Auf diese Weise erhalten wir die Formel:

$$(p_A \vee p_B \vee p_C) \wedge (p_A \vee \neg p_B \vee p_C) \wedge (\neg p_A \vee p_B \vee p_C) \wedge (\neg p_A \vee \neg p_B \vee p_C).$$

Es gibt zwei grundsätzliche Möglichkeiten, eine gegebene Formel in disjunktive oder konjunktive Normalform überzuführen: Zum einen mithilfe der Wahrheitstafelkonstruktion, zum anderen durch Anwendung der „Rechenregeln“. Wir zeigen ein Beispiel für den zweiten Weg.

Beispiel 6 Transformation in DNF

Die Formel $p \wedge (q \vee (p \wedge r))$ soll mithilfe der Rechenregeln aus Tabelle 1-1 (► Seite 18) in DNF transformiert werden:

$$p \wedge (q \vee (p \wedge r)) \stackrel{\textcircled{3}}{\Leftrightarrow} (p \wedge q) \vee (p \wedge p \wedge r) \stackrel{\textcircled{6}}{\Leftrightarrow} (p \wedge q) \vee (p \wedge r).$$

Alternativ wäre auch folgende Rechnung möglich gewesen:

$$p \wedge (q \vee (p \wedge r)) \stackrel{\textcircled{3}}{\Leftrightarrow} p \wedge (q \vee p) \wedge (q \vee r) \stackrel{\textcircled{5}}{\Leftrightarrow} p \wedge (q \vee r) \stackrel{\textcircled{3}}{\Leftrightarrow} (p \wedge q) \vee (p \wedge r).$$

KV-Diagramme zur Vereinfachung von Formeln

Im Kriminalfall aus Beispiel 5 war zuletzt folgende recht unübersichtliche Formel entstanden:

$$(p_A \wedge p_B \wedge p_C) \vee (\neg p_A \wedge p_B \wedge p_C) \vee (p_A \wedge \neg p_B \wedge p_C) \vee (\neg p_A \wedge \neg p_B \wedge p_C).$$

Unser nächstes Ziel ist es, diese Formel zu vereinfachen, in der Hoffnung, dann den oder die Täter entlarven zu können. Die Grundregel zur Vereinfachung von Formeln in konjunktiver oder disjunktiver Normalform lautet:

Eine Formel in DNF oder KNF kann nur dann vereinfacht werden, wenn sie komplementäre Literale enthält.

Die Formel $(p \wedge q) \vee (p \wedge r)$ beispielsweise kann nicht weiter vereinfacht werden. Handelt es sich jedoch um eine vollständige DNF – was auf die aus der Wahrheitstafel abgelesene Form stets zutrifft – so gibt es stets komplementäre Literale, vorausgesetzt, es ist mehr als ein Minterm vorhanden. Es sei angemerkt, dass die komplementären Literale aufgrund der Definition eines Minters in unterschiedlichen Minternen auftreten müssen.

Aber nicht jede DNF-Formel, die komplementäre Literale enthält, lässt sich auch vereinfachen. Beispielsweise kann die Formel $(p \wedge q) \vee (\neg p \wedge q)$ trotz komplementärer Literale nicht weiter vereinfacht werden.

In obiger Formel sind jedenfalls genügend komplementäre Literale vorhanden, die Anlass zur Vereinfachung geben. Wir wollen im Folgenden das Verfahren von Karnaugh und Veitch¹ zur Vereinfachung von DNF- bzw. KNF-Formeln vorstellen. Das Verfahren geht von einer vollständig ausgefüllten Wahrheitstafel aus. Diese Wahrheitstafel wird in ein rechteckiges Schema – das sogenannte KV-Diagramm – der Größe 4, 8, 16 ... je nach der Anzahl n der beteiligten Aussagenvariablen umgewandelt. Die folgende Abbildung zeigt jeweils Wahrheitstafel und das zugeordnete KV-Diagramm für $n = 2$. Die Ziffern ① bis ④ geben an, an welcher Stelle die Zeilen der Wahrheitstafel im KV-Diagramm untergebracht werden.

p	q	F
0	0	①
0	1	②
1	0	③
1	1	④

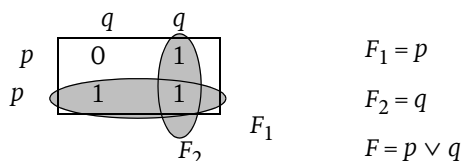
	q	q
p	①	②
p	③	④

Um die Einträge des KV-Diagramms in Minterme zu übersetzen, fasst man so viele nebeneinanderstehende Einsen wie möglich zusammen. Wir betrachten folgende Wahrheitstafel (links):

p	q	F
0	0	0
0	1	1
1	0	1
1	1	1

	q	q
p	0	1
p	1	1

Wir tragen zunächst die Wahrheitswerte der Formel F in das KV-Diagramm ein (rechts). Nun werden benachbarte Einsen zusammengefasst.



Die beiden nebeneinanderstehenden Einsen an Position 3 und 4 entsprechen der Formel:

$$F_1 = (p \wedge \bar{q}) \vee (p \wedge q) \Leftrightarrow p \wedge (\bar{q} \vee q) \Leftrightarrow p.$$

Die beiden nebeneinanderstehenden Einsen an Position 2 und 4 entsprechen der Formel:

$$F_2 = (\bar{p} \wedge q) \vee (p \wedge q) \Leftrightarrow (\bar{p} \vee p) \wedge q \Leftrightarrow q.$$

1. <http://de.wikibooks.org/wiki/Karnaugh-Veitch-Diagramm>. Eine Visualisierung des Verfahrens finden Sie unter <http://www.iain.ira.uka.de/users/asfour/TI/KVD/>.

Insgesamt ergibt sich die vereinfachte disjunktive Normalform $p \vee q$.

Ein KV-Diagramm mit 3 Variablen p, q und r ist stets folgendermaßen aufgebaut:

p	q	r	F
0	0	0	①
0	0	1	②
0	1	0	③
0	1	1	④
1	0	0	⑤
1	0	1	⑥
1	1	0	⑦
1	1	1	⑧

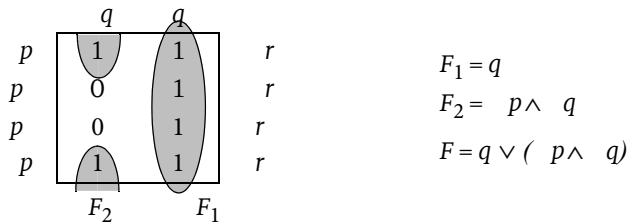
⑤⑥⑦⑧

	q	q	
p	①	③	r
p	⑤	⑦	r
p	⑥	⑧	r
p	②	④	r

Das KV-Diagramm ist so gemacht, dass sich zwei benachbarte Felder stets um genau eine Variable unterscheiden. Ziel des Verfahrens ist es, so viele nebeneinanderstehende Einsen wie möglich zusammenzufassen. Dabei ist Folgendes zu beachten:

- Es sollen so viele Einsen wie möglich zusammengefasst werden. In einem 3er-Diagramm können jeweils 2, 4 oder 8 Einsen zusammengefasst werden.
- Zwei Felder sind „benachbart“, wenn sie sich in genau einem komplementären Literal unterscheiden. In diesem Sinne sind auch die beiden Felder links oben (①) und links unten (②) benachbart, weil sie sich im komplementären Literal (r bzw. \bar{r}) unterscheiden. Dasselbe gilt für die beiden Felder ③ und ④. Stellen Sie sich einfach vor, der obere und der untere Rand des Diagramms wären zusammengeklebt.

Wir betrachten dazu folgendes Beispiel mit drei Variablen:



Die Zusammenfassung ergibt:

- In der rechten Spalte können 4 Einsen zur Formel $F_1 = q$ zusammengefasst werden.
- Die beiden Felder links oben ($p \wedge q \wedge r$) und links unten ($p \wedge q \wedge \bar{r}$) werden zur Formel $F_2 = p \wedge q$ zusammengefasst.

Nun kommen wir zur Lösung unseres Kriminalfalls (► Beispiel 5 auf Seite 23):

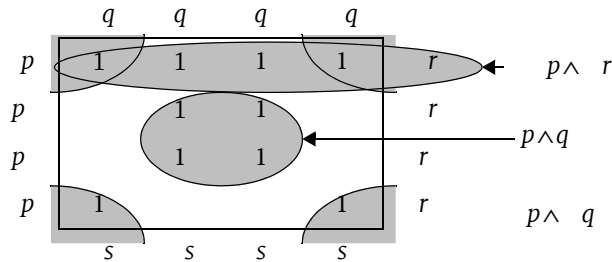
	p_B	p_B	
p_A	0	0	p_C
p_A	0	0	p_C
p_A	1	1	p_C
p_A	1	1	p_C

F

$F = p_C$

Hier können vier Einsen zur Formel $F = p_C$ zusammengefasst werden. Das bedeutet: C ist der Täter – doch halt, was heißt „der Täter“? Die Formel p_C sagt ja gar nichts aus über A und B. Beide können schuldig oder unschuldig sein. Inspektor Quak kann daher lediglich C überführen. Die beiden anderen muss er laufen lassen, weil er ihnen nichts nachweisen kann.

Schließlich noch ein Diagramm mit vier Variablen:



Der Minterm $p \wedge q$ entsteht durch Zusammenfassung der Einsen in den vier Ecken. Die Gesamtformel ergibt sich zu:

$$(p \wedge q) \vee (p \wedge q) \vee (p \wedge r).$$

Man hätte auch alternativ statt der vier Einsen in der ersten Reihe die vier Einsen in der Mitte oben zu $q \wedge r$ zusammenfassen können und hätte dann die Gesamtformel

$$(p \wedge q) \vee (p \wedge q) \vee (q \wedge r)$$

erhalten.

Wozu kann man das Verfahren gebrauchen, außer zur Lösung von Kriminalrätseln?

Logische Schaltungen

Eine wichtige Anwendung der dargestellten Verfahren zur Vereinfachung logischer Formeln ist der Entwurf logischer Schaltungen. In diesem Kontext spricht man von *booleschen Funktionen* anstelle von Formeln und benutzt auch eine andere Schreibweise. Die Konjunktion wird durch den Maloperator (\cdot), die Disjunktion durch das Plus (+) und die Negation durch einen Überstrich ($\bar{\quad}$) dargestellt. Die Formel

$$(p \wedge q) \vee (p \wedge q \wedge r) \vee q$$

schreibt sich dann folgendermaßen:

$$p \cdot \bar{q} + \bar{p} \cdot q \cdot r + q$$

bzw. noch einfacher

$$p\bar{q} + \bar{p}qr + q,$$

weil der Malpunkt weggelassen wird.

Wir wollen diese Schreibweise hier jedoch nicht weiter verfolgen.

Die logischen Verknüpfungen werden technisch durch sogenannte *Gatter* realisiert. Dabei werden an einem oder mehreren Eingängen Spannungszustände angelegt. Meist entspricht die logische Konstante 0 (falsch) einer geringen Spannung und die logische Konstante 1 (wahr) einer hohen Spannung.

Logikgatter können auf vielfältige Weise realisiert werden. Die ersten Logikgatter, die Charles Babbage¹ in seiner *Analytical Engine* verbaute, waren mechanischer Natur. Später wurden elektromagnetische Relais verwendet, und heutzutage werden Gatter elektronisch implementiert, können aber auch optisch oder auf Molekularebene realisiert werden.

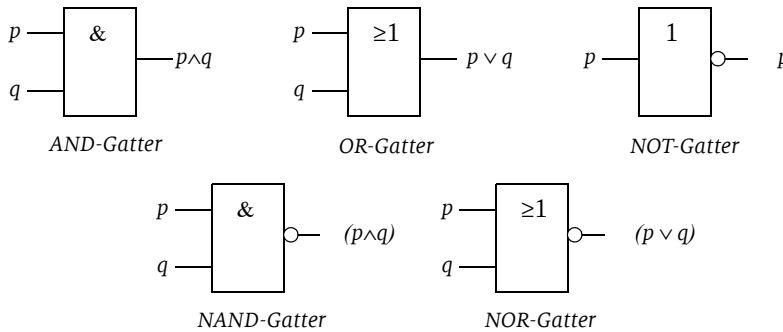


Abb. 1-1
Einige Logikgatter

In Schaltplänen werden Gatter durch ihre Schaltsymbole dargestellt. Die Schaltsymbole einiger gebräuchlicher Gatter sind in Abbildung 1-1 dargestellt. Besonders wichtig ist das NAND-Gatter, denn mit ihm kann jede komplexe Schaltung aufgebaut werden. Das Gleiche gilt für das NOR-Gatter. Den Beweis dafür haben wir auf Seite 19 geführt (► Tabelle 1-1), wo wir die Konjunktion, Disjunktion und Negation durch den Sheffer- bzw. den Peirce-Operator komplett ersetzt haben.

Die Grundgatter können zu komplexen Schaltungen zusammgebaut werden. In Schaltplänen symbolisiert man vor- oder nachgelagerte NOT-Gatter meist mit einem Kreis am Ein- oder Ausgang (► Abbildung 1-2).

Beispiel 7 Das folgende Beispiel zeigt, wie die Addition zweier Binärzahlen *x* und *y* mit logischen Gattern realisiert werden kann.

Nehmen wir zunächst an, es handelt sich um einstellige Binärzahlen *x* und *y*. Es gibt 4 Möglichkeiten:

$$0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 10.$$

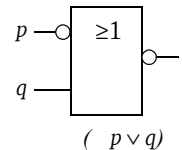


Abb. 1-2
Vereinfachte Darstellung vor- und nachgelagerter NOT-Gatter

1. Charles Babbage (1791–1871), englischer Mathematiker, Philosoph und Erfinder

Sind beide Zahlen gleich 1, so entsteht ein Übertrag. Die Schaltung benötigt deshalb zwei Ausgänge: Einen Ausgang s für das Summenbit (das niedrigerwertige Bit) und einen Ausgang \ddot{u} für das Übertragsbit (das höherwertigere Bit). Die Wahrheitstafel sieht dann folgendermaßen aus:

x	y	\ddot{u}	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Die disjunktive Normalform lässt sich direkt ablesen:

$$\ddot{u} = x \wedge y$$

$$s = (x \wedge \bar{y}) \vee (\bar{x} \wedge y).$$

Diese Formeln lassen sich mit der KV-Methode nicht mehr vereinfachen. Die Realisierung des sogenannten *Halbaddierers* mit Gattern ist in Abbildung 1-3 links dargestellt.

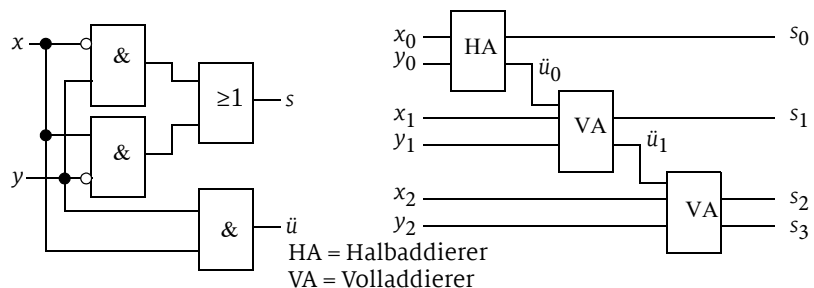
Der Halbaddierer hat zwei Eingänge, x und y , und zwei Ausgänge, s und \ddot{u} . Für die Addition mehrstelliger Binärzahlen benötigt man sogenannte *Volladdierer*, die den Übertrag der niedrigerwertigen Stelle mit verarbeiten. Der Volladdierer benötigt daher einen zusätzlichen Eingang \ddot{u}_{Ein} für den Übertrag der niedrigwertigeren Stelle.

Für die Addition n -stelliger Binärzahlen benötigt man einen Halbaddierer für die niedrigstwertige Stelle und $n-1$ Volladdierer für die restlichen Stellen. Die Realisierung ist in Abbildung 1-3 rechts dargestellt.

Aufgaben zu 1.3

- 1 Bringen Sie die „zwei aus drei“-Formel in konjunktive und in disjunktive Normalform und vereinfachen Sie anschließend mithilfe eines KV-Diagramms. Die Formel besagt, dass von drei Aussagevariablen p , q und r mindestens zwei wahr sind.
- 2 Realisieren Sie die Schaltung eines Volladdierers.
- 3 Realisieren Sie einen Halbaddierer unter ausschließlicher Verwendung des NAND-Gatters.

Abb. 1-3
Schaltung eines
Halbaddierers (links)
Addierwerk für
3-stellige Binärzahlen
(rechts)



4 Realisieren Sie die Konjunktion unter ausschließlicher Verwendung des OR- und des NOT-Gatters (vereinfachte Darstellung mit Kreisen an Ein- bzw. Ausgängen).

5 Erstellen Sie möglichst einfache Formeln aus den folgenden KV-Diagrammen:

	q	q	q	q	
p	1				r
p	1	1	1	1	r
p	1	1	1	1	r
p	1				r
	s	s	s	s	

	q	q	q	q	
p	1				r
p	1		1		r
p		1	1	1	r
p			1	1	r
	s	s	s	s	

6 Arno, Britta, Carl und Dörte überlegen, heute Abend auf eine Party zu gehen.

- Arno: „Wenn Carl kommt, komme ich auch!“
- Britta: „Wenn Arno kommt, dann gehe ich auf gar keinen Fall dorthin. Aber wenn er nicht hingehet, dann bin ich mit dabei.“
- Carl: „Ich komme nur, wenn Dörte und Britta auch kommen.“
- Dörte: „Ohne Carl gehe ich auf keinen Fall dorthin.“
- Arno und Dörte wurden noch nie zusammen auf einer Party gesehen.
- Heute Abend ist auf jeden Fall Britta oder Dörte anwesend.

Wer ist auf der Party, wer nicht?

7 Inspektor Quak hat wieder drei Männer, A , B und C , unter Verdacht auf einen Einbruch verhaftet. Fest steht, dass mindestens einer der Drei an der Tat beteiligt war, und dass außer den Dreien niemand als Täter infrage kommt. Quak weiß:

- Wenn A schuldig ist, dann ist auch B schuldig.
- C arbeitet nie alleine.
- A arbeitet nie mit C zusammen.

Wer war's?

8 Dieses Mal hat Inspektor Quak vier Männer, A , B , C und D verhaftet. Mindestens einer der Vier war an der Tat beteiligt, und sonst kommt niemand als Täter infrage.

- B arbeitet nie alleine.
- A und C arbeiten nie zusammen.
- Wenn C schuldig ist, dann ist auch D schuldig.
- D arbeitet nie alleine, sondern immer nur mit A zusammen.

Wer war's?