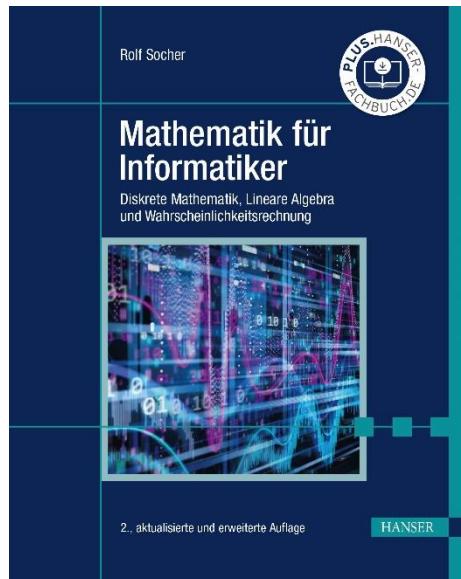


HANSER



Leseprobe

zu

Mathematik für Informatiker

von Rolf Socher

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

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

Weitere Informationen und Bestellungen unter

<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446467477>

sowie im Buchhandel

© Carl Hanser Verlag, München

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 Mausklickpunkt 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 Orlowski für viele nützliche Hinweise. Ferner danke ich dem Carl Hanser Verlag, allen voran Frau Fritzsch und Frau Wulst für die gewohnt gute Zusammenarbeit.

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.hanser-fachbuch.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	Injective, 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 Ebene193
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 Raum218
10.1 Vektoren im Raum	218
10.2 Ebenen	221
10.3 Spatprodukt, lineare Unabhängigkeit von 3 Vektoren, Basen	230
11 Lineare und affine Abbildungen233
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 Vektorräume258
12.1 Einführung	258
12.2 Vektorräume und Unterräume	261
12.3 Basis, Dimension und lineare Unabhängigkeit	265
13 Lineare Abbildungen und Matrizen275
13.1 Lineare Abbildungen	275
13.2 Matrizen zur Darstellung linearer Abbildungen	282
14 Der Gauß-Algorithmus291
14.1 Berechnung des Rangs einer Matrix	291
14.2 Berechnung der Inversen einer Matrix	296
14.3 Lösen linearer Gleichungssysteme	298
15 Fehlerkorrigierende Codes306
15.1 Grundbegriffe	306
15.2 Lineare Codes	311
15.3 Konstruktion linearer Codes	313
Symbolverzeichnis318
Sachwortverzeichnis320

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 ausgleiche. 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 $\mid\mid$ benutzt. Die offensichtlich wahre Aussage „6 ist durch 3 oder durch 4 teilbar“ wird dargestellt durch:

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

bzw. in Javanesisch:

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

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:

$$(4|6)$$

bzw. in Javanesisch:

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

oder noch einfacher durch $6 \% 4 != 0$.

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 → 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

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

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 $(p \wedge q) \vee (\neg p \wedge q)$:

p	q	p	q	$p \wedge q$	$\neg p \wedge q$	$(p \wedge q) \vee (\neg p \wedge q)$
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 $p = \text{„Es ist Freitag“}$ und $q = \text{„Ich gehe ins Kino“}$.

- 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 $p \vee p$:

p	p	$p \vee p$
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 Wahrheitstafeln 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 Junktor. 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 p : Wenn die Aussage p falsch ist, dann ist die Formel $p \rightarrow q$ wahr, bzw. $p \Rightarrow p \rightarrow q$.

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 Wahrheitstafeln 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)	
$F \leftrightarrow G \Leftrightarrow (\neg F \vee G) \wedge (\neg G \vee F)$	(12)	
$F \mid G \Leftrightarrow (F \wedge G)$	(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 \neg 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 Wahrheitstafeln 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.

Sachwortverzeichnis

A

- Abbildung 62
 - affine 251
 - flächentreue 239, 243
 - identische 239, 246
 - inverse 256
 - invertierbare 256
 - lineare 236, 275
- Abel, Niels Henrik 116
- Abstand 205
- Additionssatz 169
- adjazent 130
- Adjazenzliste 130
- Adjazenzmatrix 130
- Algorithmus 92, 291
 - erweiterter euklidischer 95
 - euklidischer 92
 - Gauß- 291 ff
 - Greedy- 140
 - RSA- 111 ff
 - ungarischer 152
 - von Hierholzer 136
 - von Kruskal 147
 - zum Test auf Zusammenhang eines Graphen 135
 - zur Färbung eines Graphen 139
 - zur Konstruktion eines Gerüsts 144
- Äquivalenz, logische 16
- Äquivalenzklasse 57, 99, 134
- Äquivalenzrelation 56, 58, 99, 134
 - induzierte 58
- ASCII-Codierung 62, 69
- Assoziativgesetz 66, 116
- Aussage 10
- Aussageform 10, 42

B

- Babbage, Charles 29
- Basis 269, 270
 - des R2 231
 - des R3 231
 - eines Vektorraums 266
 - kanonische 231, 266
- Basisoperationen des Gauß-Algorithmus 293

Basiswechsel 287 ff

Baum 141

aufspannender 144

Binär- 143

Wurzel- 143

Baumdiagramm 169

bedingte Wahrscheinlichkeit 185

Bestensuche 146

Betrag eines Vektors 196, 219

Beweis

direkter 32 ff

durch Fallunterscheidung 34 ff

durch vollständige Induktion 37 ff

indirekter 35 ff

Widerspruchs- 36

Bézout, Étienne 94

Bézout-Koeffizienten 94

Biimplikation 13

Bild 277

Binärbaum 143

Binomialsatz 84

Binomialverteilung 179

Erwartungswert 180

Varianz 180

Binomialzahl 82

Blatt 143

Blockcode 308

Breitensuche 145

C

Cantor, Georg 40

Cäsar-Code 61, 69, 75 ff, 88

chromatische Zahl 139, 149

Code 308

linearer 311

perfekter 310, 312

D

Definitionsmenge 62

Descartes, René 50

Determinante 203, 230, 243, 249

Diagonalsierungsverfahren 74

Differenzmenge 46

Dimension 273

Dimensionssatz 279

disjunkt 46

- Disjunktion 11
- Drehmatrix 237, 245
- Drehung 235
- Dreiecksmatrix 292
- Dreieckszahlen 38
- Dualisieren 18
- E**
 - Ebene 221 ff
 - Ebenendarstellung 221 ff
 - funktionale Form 221
 - implizite Form 221
 - Parameterform 224
 - Elementarer eig nisse 162
 - elementare Zeilenumformungen 293
 - Elementarereignis 163
 - Endknoten 131
 - Endomorphismus 275
 - Ereignis 163
 - sicheres 163
 - unmögliches 163
 - Ergebnis 162
 - Ergebnismenge 162, 162
 - Erwartungswert 174
 - erweiterte Dreiecksform 292
 - Euklid von Alexandria 92
 - Euler, Leonhard 107
 - F**
 - Faktor 127
 - Faktorielle
 - fallende 80
 - steigende 80
 - Fakultät 79
 - Falk'sches Schema 241
 - Fermat, Pierre de 109
 - Funktion 62
 - bijektive 68, 79
 - boolesche 28
 - Darstellung von 63 ff
 - identische 62
 - injektive 67
 - inverse 69
 - invertierbare 69
 - mit mehreren Argumenten 64 ff
 - surjektive 68
 - umkehrbare 69
 - Funktionswert 62
 - G**
 - Galois, Évariste 123
 - Gatter 29
 - Gauß, Carl Friedrich 292
 - Gauß-Algorithmus 291 ff
 - Geburtstagsparadoxon 161, 170
 - Gegenereignis 163
 - Geheimtext 61
 - Generatormatrix 313
 - geometrische Verteilung
 - Erwartungswert 181
 - Varianz 181
 - Gerade 209 ff
 - Gerdendarstellung
 - explizite Form 210
 - implizite Form 210
 - Parameterform 211
 - Gerüst 144
 - Gewicht 312
 - Gleichungssystem
 - homogenes 298 ff
 - inhomogenes 298, 302 ff
 - lineares 298 ff
 - Grad
 - eines Knotens 131
 - eines Polynoms 124
 - Graph 130
 - bipartiter 149
 - eulerscher 136
 - Färbungen 138 ff
 - gewichteter 146
 - planarer 139
 - vollständiger 130
 - vollständiger bipartiter 149
 - zusammenhängender 134
 - Greedy-Algorithmus 140, 146
 - größter gemeinsamer Teiler 91, 127
 - Gruppe 116, 122
 - abelsche 116, 261
 - isomorphe 119
 - Guthrie, Francis 139
 - H**
 - Halbaddierer 30
 - Hall, Philip 150
 - Hamming, Richard W. 308
 - Hamming-Abstand 308
 - Hamming-Code 316
 - Hamming-Matrix 316
 - Häufigkeit
 - absolute 157

- relative 157
- Häufigkeitsverteilung 158
- Höhe
 - eines Baumes 143
 - eines Knotens 143
- homogene Koordinaten 252
- Homomorphismus
 - von Gruppen 119
 - von Vektorräumen 275
- hypergeometrische Verteilung 182
 - Erwartungswert 183
 - Varianz 183
- I**
- Implikation 12
- injektiv 278
- Inverse
 - modulare 104
 - multiplikative 107
- inverses Element in einer Gruppe 116
- invertierbar 107
 - modulo m 104
- Involution 120
- incident 130
- ISBN-10-Code 101
- isomorph 119, 132, 276
- Isomorphismus 276, 287
 - von Graphen 132
 - von Gruppen 119
 - von Vektorräumen 276
- J**
- Junktor 9 ff
- K**
- Kabinettprojektion 248
- Kante 130
- Kantenzug 134
 - geschlossener 134
 - offener 134
- kartesisches Produkt 49
- Kavalierprojektion 248
- Kern 277
- Kern einer linearen Abbildung 277
- Klartext 61
- Klein, Felix 118
- kleinsche Vierergruppe 118, 120, 121
- Knoten 130
 - End- 131
 - gerader 132, 136
 - isolierter 131
- ungerader 132
- verbundene 134
- Knotenfärbung 139
- Koeffizienten
 - einer Matrix 282
 - eines Polynoms 124
- kollinear 204, 223
- Kolmogorow-Axiome 163
- Kommutativgesetz 116
- Komplementmenge 46
- Komposition
 - von Funktionen 65
 - von Relationen 53
- kongruent ... modulo 98
- König, Dénes 152
- Königsberger Brückenproblem 136
- Konjunktion 10
- Konsequenz, logische 17
- Kontradiktion 15
- Körper 123
- Kosinusformel 203
- Kreis 134
 - einfacher 134
 - eulerscher 136
 - hamiltonscher 137
- Kreuzprodukt 219
- Kruskal, Joseph 146
- KV-Diagramm 25 ff
- L**
- Länge eines Vektors 196
- Lemma von Bézout 94
 - für Polynome 127
- linear abhängig 230, 268
- linear unabhängig 223, 230, 268
- lineare Abbildung 275
 - Bild 277
 - Kern 277
 - Rang 277
- lineare Hülle 263
- Linearkombination 263
- Linkssystem 249
- Literal 24
 - komplementäres 24
- logisch äquivalent 16
- logische Junktoren 10
- Logische Schaltungen 28 ff
- Lösungsmenge 43

M

- Mächtigkeit einer Menge 40
- Matching 149
 - maximales 150
 - vollständiges 150
- Matrix 237, 282
 - Anwendung auf einen Vektor 238
 - Dreh- 237
 - einer linearen Abbildung 237, 245
 - Generator- 313
 - inverse 256, 286, 296 ff
 - invertierbare 256, 286
 - Prüf- 313
 - quadratische 282
 - Rang 291
- Matrixprodukt 285 ff
- Matrizenprodukt 241, 285
- Maximalgrad eines Graphen 131
- Median 159
- Menge
 - abzählbare 72
 - leere 43
 - überabzählbare 72
- Merkmal 156
- metalogische Symbole 16 ff
- Minimalabstand 308
- Minimalgerüst 146 ff
- Minimalgewicht 312
- Minterm 24
 - vollständiger 24
- Mittelwert 158
- Modul 99
- monoalphabetische Substitution 75
- Multiplikation
 - skalare 199, 261
- Multiplikationssatz 169

N

- NAND-Gatter 29
- Negation 11
- neutrales Element einer Gruppe 116
- NOR-Gatter 29
- Normalenvektor 225
- Normalform
 - disjunktive 24
 - hessesche 228
 - konjunktive 24
- nullteilerfrei 124
- Nullvektor 261

O

- Ordnung einer Gruppe 116
- Ordnungsrelation 56
 - strikte 56
 - wohlfundierte 56
- orthogonal 204
- Ortsvektor 195
- P**
- Parallelprojektion 247 ff
 - orthogonale 247
 - schiefe 247
- Parameterform
 - einer Ebene 224
 - einer Geraden 211
- Paritätsprüfung 101
- Partition 49
- Pascal, Blaise 84
- Peirce-Operator 13, 19
- Permutation 79
- Phi-Funktion, eulersche 107
- Pivot-Element 294
- Pivot-Spalte 294
- Pivot-Zeile 294
- Poissonverteilung 183
 - Erwartungswert 183
 - Varianz 183
- Polynom 124, 262
 - division 125 ff
 - normiertes 124
- Potenzmenge 44
- Primzahl 40, 96, 107, 109, 111, 117, 123
- Prinzip des nächsten Nachbarn 307
- Produkt von Matrizen 285
- Produktregel 76
- Projektion 205, 238
 - Kabinett 248
 - Kavalier- 248
- Prüfmatrix 313
- Prüfziffern 101 ff
- Public-Key-Kryptografie 111
- Q**
- Quak, Jonathan 23, 31, 35, 39
- Quersumme 100
 - alternierende 101
- R**
- Rang 277
 - einer Matrix 291
- Rechtssystem 220, 249

- Relation 52
 - Äquivalenz- 56
 - asymmetrische 54
 - inverse 53
 - reflexive 54
 - symmetrische 54
 - transitive 54
 - Umkehr- 53
- Restklasse 99
- Richtungsvektor 211, 224
- Ring 122
 - kommutativer 122
- Rotation 235, 239, 246
- RSA-Algorithmus 111 ff
- RSA-Verfahren 92, 97
- Russell, Bertrand 41
- S**
 - Sarrus
 - Schema von 249
 - Satz
 - Binomial- 84
 - des Pythagoras 196
 - Dimensions- 279
 - Vier-Farben- 139
 - von Bayes 188
 - von der totalen Wahrscheinlichkeit 186
 - von Euklid 97
 - von Euler 107 ff
 - von Fermat 109
 - von Hall 150
 - von Steinitz 272
 - von Thales 205
 - von Varignon 200
 - Schaltjahr 9, 19, 47
 - Scherung 235, 238
 - Schnittmenge 46
 - Schubfachprinzip 71 ff
 - Sheffer-Operator 13, 19
 - Sichtbarkeitsbestimmung 225
 - Sinusformel 203
 - Skalar 195, 261
 - skalare Multiplikation 219
 - Skalarprodukt 203, 219, 282
 - Skalierung 234, 238, 246
 - Spaltenrang 291
 - Spaltenvektor 195, 282
 - Spatprodukt 230
- Spiegelung 234, 238, 246
- Stack 145
- Standardabweichung 160, 177
- Steigungswinkel 196
- Steinitz'scher Austauschsatz 272
- Stochastische Unabhängigkeit 189
- Stützvektor 211, 224
- Suchbaum, binärer 142
- Sudoku-Eigenschaft 107
- Summe von Vektoren 198
- Summenformel 76
- surjektiv 278
- Symmetriegruppe des Dreiecks 122
- Symmetrietransformation 121
- T**
 - Tautologie 15
 - teilbar 89
 - Teiler 89, 127
 - größter gemeinsamer 91, 127
 - teilerfremd 91, 107
 - Teilmenge 43
 - Ternärbaum 143
 - Tiefensuche 145
 - Transformation
 - 2-D- 233 ff
 - 3-D- 245 ff
 - Translation 233
 - transponiert 195
- U**
 - umkehrbar 69
 - Umkehrfunktion 69
 - Umkehrrelation 53
 - Unabhängigkeit
 - Stochastische 189
- Unterraum 262
 - trivialer 263
- V**
 - Varianz 177
 - lineare 160
 - quadratische 160
 - Varignon, Pierre de 200
 - Vektor 194 ff
 - Betrag 196
 - Länge 196
 - linear abhängig 230
 - linear unabhängig 230
 - Normalen- 225
 - normierter 196

- Null- 261
- Orts- 195
- Richtungs- 211
- Spalten- 195
- Stütz- 211
- transponierter 195
- Zeilen- 195
- Vektoren
 - kollineare 200
- Vektorraum 261
- Venn-Diagramm 44, 45
- Vereinigungsmenge 46
- Verkettung 65, 241
- Verschiebung 233
- Verteilung
 - Binomial- 179
 - geometrische 181
 - Hypergeometrische 182
 - Poisson- 183
- Vielfaches 89
- Vier-Farben-Satz 139
- Vierfeldertafel 188
- Volladdierer 30
- W**
 - Wahrheitstafel 10 ff
 - Wahrheitswert 10
 - Wahrscheinlichkeit
 - bedingte 185
 - Wahrscheinlichkeitsfunktion 163, 186
 - Wahrscheinlichkeitsverteilung 173
 - kumulative 173
 - Weg 134
 - alternierender 151
 - einfacher 134
 - eulerscher 136
 - hamiltonscher 137
 - Wertebereich 62
 - Wertemenge 62
 - Wilder, Billy 148
 - Wurzel 143
 - Wurzelbaum 143
- Z**
 - Zeilenrang 291
 - Zeilenvektor 195, 282
 - Zoom 234, 238
 - Zufallsvariable 173
 - Zusammenhangskomponente 134
 - zyklische Verschiebung 61