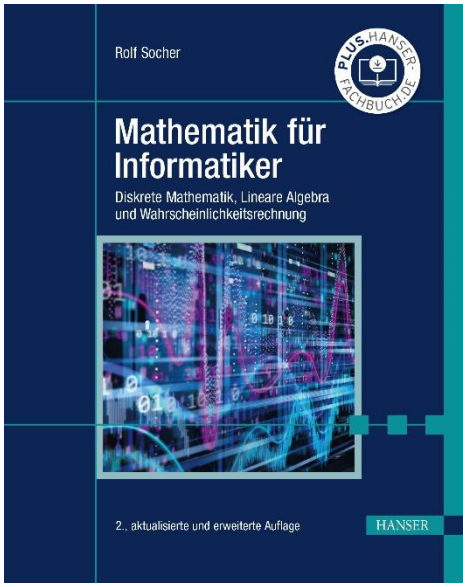
HANSER



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

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.

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.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

Lösungen zu den Aufgaben auf https://plus.hanser-fachbuch.de/

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 ausgliche. 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:

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

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 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 p ist genau dann wahr, wenn p falsch ist:

p	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 ausgeprochen 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 $p \mid q$ 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	$p \mid q$	$p \downarrow q$
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

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

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 $((\neg p) \vee q) \rightarrow r$ dasselbe wie $\neg p \vee q \rightarrow r$, 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 $(p \wedge \neg q) \vee (\neg p \wedge q)$:

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

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*.

Definition Tautologie, Kontradiktion

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.

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 $p \to (p \to q)$:

p	q	p	$p \to q$	$p \to (p \to q)$
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 $(p \to q) \to q$:

p	q	$p \to q$	$(p \to q) \to q$
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 $(p \to q) \to q$ ist dieselbe wie der Disjunktion $p \vee q$. Wir sagen, die beiden Formeln $(p \to q) \to q$ 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: $(p \to q) \to q \Leftrightarrow p \vee q$ (▶ 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:

Satz

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

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.

Definition Konsequenz

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 $F \Rightarrow G$.

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		
$F \to G$	$\Leftrightarrow$	$F \vee G$	⑪		
$F \leftrightarrow G$	$\Leftrightarrow$	$(F \vee G) \wedge (G \vee F)$	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."

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

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 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