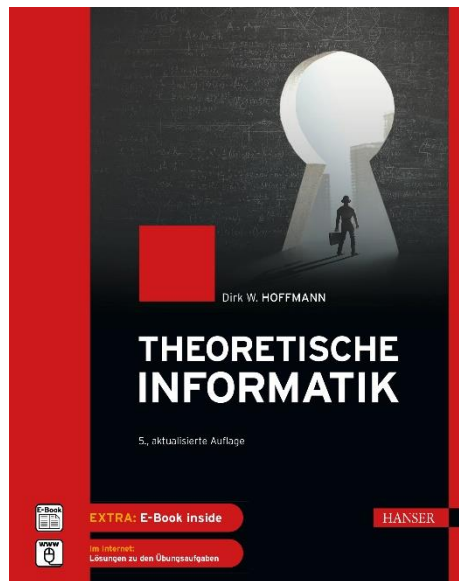


HANSER



Leseprobe

zu

Theoretische Informatik

von Dirk W. Hoffmann

Print-ISBN 978-3-446-47029-3
E-Book-ISBN 978-3-446-47256-3

Weitere Informationen und Bestellungen unter
<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446470293>

sowie im Buchhandel

© Carl Hanser Verlag, München



Vorwort

Für die meisten Menschen ist die Informatik fest mit der Entstehungsgeschichte des Computers verbunden; einer Technik, die von außen betrachtet keinen Grenzen zu unterliegen scheint. Wir erleben seit Jahren eine schier ungebremste Entwicklung und sind längst daran gewöhnt, dass der Computer von heute schon morgen überholt ist. Dass sich hinter der Computertechnik eine tiefgründige Wissenschaft verbirgt, die all die großen Erfolge erst möglich macht, bleibt vielen Menschen verborgen. Die Rede ist von der theoretischen Informatik.

In der Grundlagenausbildung hat die theoretische Informatik ihren festen Platz eingenommen. Viele Studierende begegnen ihr mit gemischten Gefühlen und von manchen wird sie gar als bedrohlich empfunden. Mitverantwortlich für diese Misere sind die historischen Wurzeln der theoretischen Informatik. Entstanden aus der Mathematik, wird sie häufig in einer Präzision dargestellt, die in der Informatik ihresgleichen sucht. Manch ein Leser verirrt sich schnell in einem Gewirr aus Definitionen, Sätzen und Beweisen, das die Sicht auf die eigentlichen Konzepte und Methoden unfreiwillig verdeckt. Dass die theoretische Informatik weder schwer noch trocken sein muss, versuche ich mit diesem Buch zu beweisen.

Die folgenden Kapitel werden von zwei Leitmotiven getragen. Zum einen möchte ich die grundlegenden Konzepte, Methoden und Ergebnisse der theoretischen Informatik vermitteln, ohne diese durch einen zu hohen Abstraktionsgrad zu vernebeln. Hierzu werden die Problemstellungen durchweg anhand von Beispielen motiviert und die Grundideen der komplizierteren Beweise an konkreten Probleminstanzen nachvollzogen. Zum anderen habe ich versucht, den Lehrstoff in vielerlei Hinsicht mit Leben zu füllen. An zahlreichen Stellen werden Sie Anmerkungen und Querbezüge vorfinden, die sich mit der historischen Entwicklung dieser einzigartigen Wissenschaftsdisziplin beschäftigen.

Bei allen Versuchen, einen verständlichen Zugang zu der nicht immer einfachen Materie zu schaffen, war es mir ein Anliegen, keinen Verlust an Tiefe zu erleiden. Das Buch ist für den Bachelor-Studiengang konzipiert und deckt die typischen Lehrinhalte ab, die im Grundstudium an den hiesigen Hochschulen und Universitäten unterrichtet werden.



Inhaltsverzeichnis

1	Einführung	11
1.1	Was ist theoretische Informatik?	11
1.2	Zurück zu den Anfängen	14
1.2.1	Die Mathematik in der Krise	14
1.2.2	Metamathematik	18
1.2.3	Die ersten Rechenmaschinen	22
1.2.4	Der Computer wird erwachsen	24
1.2.5	Berechenbarkeit versus Komplexität	26
1.3	Theoretische Informatik heute	32
1.4	Übungsaufgaben	34
2	Mathematische Grundlagen	37
2.1	Grundlagen der Mengenlehre	38
2.1.1	Der Mengenbegriff	38
2.1.2	Mengenoperationen	41
2.2	Relationen und Funktionen	44
2.3	Die Welt der Zahlen	52
2.3.1	Natürliche, rationale und reelle Zahlen	52
2.3.2	Von großen Zahlen	55
2.3.3	Die Unendlichkeit begreifen	57
2.4	Rekursion und induktive Beweise	65
2.4.1	Vollständige Induktion	66
2.4.2	Strukturelle Induktion	68
2.5	Übungsaufgaben	70
3	Logik und Deduktion	81
3.1	Aussagenlogik	82
3.1.1	Syntax und Semantik	82
3.1.2	Normalformen	91
3.1.3	Beweistheorie	96
3.1.3.1	Hilbert-Kalkül	98
3.1.3.2	Resolutionskalkül	104
3.1.3.3	Tableaukalkül	109
3.1.4	Anwendung: Hardware-Entwurf	112
3.2	Prädikatenlogik	117
3.2.1	Syntax und Semantik	118

3.2.2	Normalformen	122
3.2.3	Beweistheorie	124
3.2.3.1	Resolutionskalkül	130
3.2.3.2	Tableaukalkül	135
3.2.4	Anwendung: Logische Programmierung	138
3.3	Logikerweiterungen	145
3.3.1	Prädikatenlogik mit Gleichheit	146
3.3.2	Logiken höherer Stufe	147
3.3.3	Typentheorie	149
3.4	Übungsaufgaben	150
4	Formale Sprachen	161
4.1	Sprache und Grammatik	162
4.2	Chomsky-Hierarchie	168
4.3	Reguläre Sprachen	170
4.3.1	Definition und Eigenschaften	170
4.3.2	Pumping-Lemma für reguläre Sprachen	172
4.3.3	Satz von Myhill-Nerode	174
4.3.4	Reguläre Ausdrücke	176
4.4	Kontextfreie Sprachen	179
4.4.1	Definition und Eigenschaften	179
4.4.2	Normalformen	179
4.4.2.1	Chomsky-Normalform	179
4.4.2.2	Backus-Naur-Form	181
4.4.3	Pumping-Lemma für kontextfreie Sprachen	182
4.4.4	Entscheidungsprobleme	186
4.4.5	Abschlusseigenschaften	188
4.5	Kontextsensitive Sprachen	191
4.5.1	Definition und Eigenschaften	191
4.5.2	Entscheidungsprobleme	192
4.5.3	Abschlusseigenschaften	193
4.6	Phrasenstruktursprachen	193
4.7	Übungsaufgaben	195
5	Endliche Automaten	201
5.1	Begriffsbestimmung	202
5.2	Deterministische Automaten	204
5.2.1	Definition und Eigenschaften	204
5.2.2	Automatenminimierung	206
5.3	Nichtdeterministische Automaten	208
5.3.1	Definition und Eigenschaften	208
5.3.2	Satz von Rabin, Scott	210
5.3.3	Epsilon-Übergänge	212
5.4	Automaten und reguläre Sprachen	216

5.4.1	Automaten und reguläre Ausdrücke	217
5.4.2	Abschlusseigenschaften	218
5.4.3	Entscheidungsprobleme	220
5.4.4	Automaten und der Satz von Myhill-Nerode	221
5.5	Kellerautomaten	223
5.5.1	Definition und Eigenschaften	223
5.5.2	Kellerautomaten und kontextfreie Sprachen	226
5.5.3	Deterministische Kellerautomaten	228
5.6	Transduktoren	230
5.6.1	Definition und Eigenschaften	230
5.6.2	Automatenminimierung	231
5.6.3	Automatensynthese	233
5.6.4	Mealy- und Moore-Automaten	234
5.7	Petri-Netze	238
5.8	Zelluläre Automaten	243
5.9	Übungsaufgaben	246
6	Berechenbarkeitstheorie	253
6.1	Berechnungsmodelle	254
6.1.1	Loop-Programme	254
6.1.2	While-Programme	260
6.1.3	Goto-Programme	264
6.1.4	Primitiv-rekursive Funktionen	269
6.1.5	Turing-Maschinen	277
6.1.5.1	Einband-Turing-Maschinen	277
6.1.5.2	Einseitig und linear beschränkte Turing-Maschinen	285
6.1.5.3	Mehrspur-Turing-Maschinen	286
6.1.5.4	Mehrband-Turing-Maschinen	286
6.1.5.5	Maschinenkomposition	288
6.1.5.6	Universelle Turing-Maschinen	289
6.1.5.7	Zelluläre Turing-Maschinen	293
6.1.6	Alternative Berechnungsmodelle	295
6.1.6.1	Registermaschinen	296
6.1.6.2	Lambda-Kalkül	300
6.2	Church'sche These	302
6.3	Entscheidbarkeit	309
6.4	Akzeptierende Turing-Maschinen	312
6.5	Unentscheidbare Probleme	319
6.5.1	Halteproblem	319
6.5.2	Satz von Rice	322
6.5.3	Reduktionsbeweise	325
6.5.4	Das Post'sche Korrespondenzproblem	326
6.5.5	Weitere unentscheidbare Probleme	330
6.6	Übungsaufgaben	333

7	Komplexitätstheorie	341
7.1	Algorithmische Komplexität	342
7.1.1	O-Kalkül	349
7.1.2	Rechnen im O-Kalkül	352
7.2	Komplexitätsklassen	356
7.2.1	P und NP	359
7.2.2	PSPACE und NPSPACE	365
7.2.3	EXP und NEXP	367
7.2.4	Komplementäre Komplexitätsklassen	369
7.3	NP-Vollständigkeit	371
7.3.1	Polynomielle Reduktion	371
7.3.2	P-NP-Problem	372
7.3.3	Satz von Cook	373
7.3.4	Reduktionsbeweise	380
7.4	Übungsaufgaben	386
A	Notationsverzeichnis	397
B	Abkürzungsverzeichnis	401
C	Glossar	403
	Literaturverzeichnis	419
	Namensverzeichnis	423
	Sachwortverzeichnis	425

1 Einführung

„Wir müssen wissen. Wir werden wissen.“

David Hilbert

1.1 Was ist theoretische Informatik?

Kaum eine andere Technologie hat unsere Welt so rasant und nachhaltig verändert wie der Computer. Unzählige Bereiche des täglichen Lebens werden inzwischen von Bits und Bytes dominiert – selbst solche, die noch vor einigen Jahren als elektronikfreie Zone galten. Die Auswirkungen dieser Entwicklung sind bis in unser gesellschaftliches und kulturelles Leben zu spüren und machen selbst vor der deutschen Sprache keinen Halt. Vielleicht haben auch Sie heute schon *gemailt*, *gesimst* oder *gegoogelt* (Abbildung 1.1). Die Digitalisierung unserer Welt ist in vollem Gange und eine Abschwächung der eingeschlagenen Entwicklung zumindest mittelfristig nicht abzusehen.

Die in der Retrospektive einzigartige Evolution der Computertechnik ist eng mit der Entwicklung der Informatik verbunden. Als naturwissenschaftliche Fundierung der Computertechnik untersucht sie die Methoden und Techniken, die eine digitale Welt wie die unsere erst möglich machen. In der gleichen Geschwindigkeit, in der Computer die Welt eroberten, konnte sich die Informatik von einer Nischendisziplin der Mathematik und Elektrotechnik zu einer eigenständigen Grundlagenwissenschaft entwickeln. War sie zu Anfang auf wenige Kernbereiche beschränkt, so präsentiert sich die Informatik mittlerweile als eine breit gefächerte Wissenschaftsdisziplin. Heute existieren Schnittstellen in die verschiedensten Bereiche wie die Biologie, die Medizin und sogar die bildenden Künste.

In Abbildung 1.2 sind die vier Säulen dargestellt, von denen die Informatik gegenwärtig getragen wird. Eine davon ist die theoretische Informatik. Sie beschäftigt sich mit den abstrakten Konzepten und Methoden, die sich hinter den Fassaden moderner Computersysteme verbergen. Die theoretische Informatik ist vor der technischen Informatik die

down|load|den <engl.> (EDV - Daten von einem Computer, aus dem Internet herunterladen); ich habe downgeloadet

googeln (mit Google im Internet suchen); ich goog[e]le;

mail|en <engl.> (als E-Mail senden); gemailt

sim|sen (ugs. für eine SMS versenden)



Abbildung 1.1: Die zunehmende Technisierung des Alltagslebens macht auch vor der deutschen Sprache keinen Halt. Im Jahr 2004 schaffte es das neudeutsche Verb *googeln* in den Duden [106]. Dort finden sich auch die Worte *mailen*, *simsen* und *downloaden* wieder.

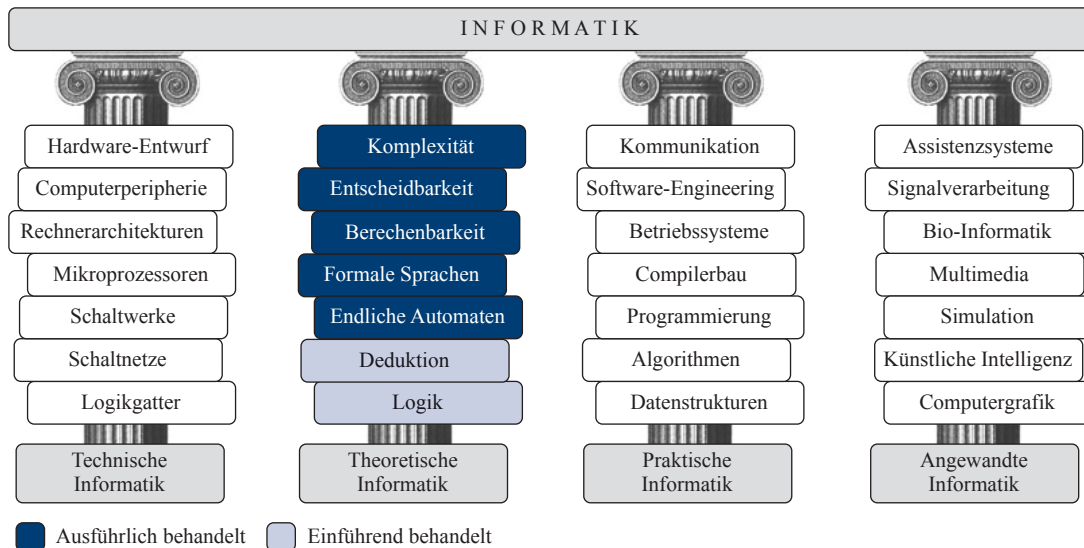


Abbildung 1.2: Die vier Säulen der Informatik

älteste Kernsäule und hat ihren direkten Ursprung in der Mathematik. Trotz ihres relativen Alters hat dieser Zweig der Informatik nichts von seiner ursprünglichen Bedeutung verloren. Er bildet das konzeptionelle Fundament, auf dem die anderen Bereiche der Informatik solide ruhen und aus dessen Wissensfundus sie schöpfen.

Betrachten wir die inhaltlichen Themen der modernen theoretischen Informatik genauer, so lassen sich diese in die folgenden Teilgebiete untergliedern (vgl. Abbildung 1.3):

■ Logik und Deduktion (Kapitel 3)

Die Logik beschäftigt sich mit grundlegenden Fragestellungen mathematischer Theorien. Im Mittelpunkt steht die Untersuchung *formaler Systeme (Kalküle)*, in denen Aussagen aus einer kleinen Menge vorgegebener Axiome durch die Anwendung fest definierter Schlussregeln abgeleitet werden. Die Logik spielt nicht nur in der theoretischen Informatik, sondern auch in der technischen Informatik und der Software-Entwicklung eine Rolle. Mit der Aussagenlogik gibt sie uns ein Instrumentarium an die Hand, mit dem wir jede erdenkliche Hardware-Schaltung formal beschreiben und analysieren können. Ferner lässt sich mit der Prädikatenlogik und den Logiken höherer Stufe das Verhalten komplexer Hardware- und Software-Systeme exakt spezifizieren und in Teilen verifizieren.

■ Formale Sprachen (Kapitel 4)

Die Theorie der formalen Sprachen beschäftigt sich mit der Analyse, der Klassifikation und der generativen Erzeugung von Wortmengen. Künstliche Sprachen sind nach festen Regeln aufgebaut, die zusammen mit dem verwendeten Symbolvorrat eine formale Grammatik bilden. Die zugrunde liegende Theorie gibt uns die Methoden und Techniken an die Hand, die für den systematischen Umgang mit modernen Programmiersprachen und dem damit zusammenhängenden Compilerbau unabdingbar sind. Viele Erkenntnisse aus diesem Bereich haben ihre Wurzeln in der Linguistik und stoßen dementsprechend auch außerhalb der Informatik auf reges Interesse.

■ Automatentheorie (Kapitel 5)

Hinter dem Begriff des endlichen Automaten verbirgt sich ein abstraktes Maschinenmodell, das sich zur Modellierung, zur Analyse und zur Synthese zustandsbasierter Systeme eignet. Auf der obersten Ebene untergliedern sich endliche Automaten in Akzeptoren und Transduktoren. Erstere zeigen einen engen Bezug zu den formalen Sprachen, Letztere spielen im Bereich des Hardware-Entwurfs eine dominierende Rolle. Sie sind das mathematische Modell, mit dem sich das zeitliche Verhalten synchron getakteter Digitalschaltungen exakt beschreiben und analysieren lässt.

■ Berechenbarkeitstheorie (Kapitel 6)

Die Berechenbarkeitstheorie beschäftigt sich mit grundlegenden Untersuchungen über die algorithmische Lösbarkeit von Problemen. Die Bedeutung dieses Teilgebiets der theoretischen Informatik ist zweigeteilt. Zum einen wird das gesamte Gebiet der Algorithmentechnik durch die Definition formaler Berechnungsmodelle auf einen formalen Unterbau gestellt. Zum anderen ermöglicht uns die systematische Vorgehensweise, die Grenzen der prinzipiellen Berechenbarkeit auszuloten.

■ Komplexitätstheorie (Kapitel 7)

Während die Berechenbarkeitstheorie Fragen nach der puren Existenz von Algorithmen beantwortet, versucht die Komplexitätstheorie die Eigenschaften einer Lösungsstrategie quantitativ zu erfassen. Algorithmen werden anhand ihres Speicherplatzbedarfs und Zeitverbrauchs in verschiedene Komplexitätsklassen eingeteilt, die Rückschlüsse auf deren praktische Verwertbarkeit zulassen. Die Ergebnisse dieser Theorie beeinflussen den gesamten Bereich der modernen Software- und Hardware-Entwicklung.

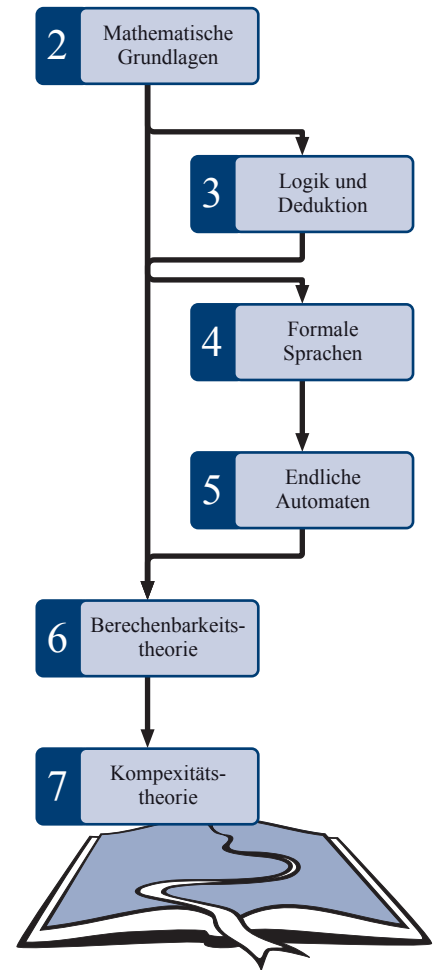


Abbildung 1.3: Kapitelübersicht. Die Pfeile deuten an, wie die einzelnen Kapitel inhaltlich zusammenhängen.

1. Axiom:

„Zu zwei Punkten gibt es genau eine Gerade, auf der sie liegen.“



2. Axiom:

„Jede gerade Strecke zwischen zwei Punkten lässt sich eindeutig verlängern.“



3. Axiom:

„Zu einem Punkt und einer Strecke kann man genau einen Kreis konstruieren.“



4. Axiom:

„Alle rechten Winkel sind gleich.“



5. Axiom:

„Zu einer Geraden und einem Punkt außerhalb der Geraden gibt es genau eine Gerade, die durch den Punkt geht und parallel zur ersten Geraden ist.“



Euklid von Alexandria

(ca. 365 v. Chr. – ca. 300 v. Chr.)

Abbildung 1.4: Die euklidischen Axiome

1.2 Zurück zu den Anfängen

Bevor wir uns ausführlich mit den Begriffen und Methoden der theoretischen Informatik beschäftigen, wollen wir in einem historischen Streifzug herausarbeiten, in welchem Umfeld ihre Teilgebiete entstanden sind und in welchem Zusammenhang sie heute zueinander stehen.

1.2.1 Die Mathematik in der Krise

Die theoretische Informatik hat ihre Wurzeln in der Mathematik. Ihre Geschichte beginnt mit der *Grundlagenkrise*, die Anfang des zwanzigsten Jahrhunderts einen Tiefpunkt in der mehrere tausend Jahre alten Historie der Mathematik markierte. Um die Geschehnisse zu verstehen, wollen wir unseren Blick zunächst auf das achtzehnte Jahrhundert richten. Zu dieser Zeit war die Mathematik schon weit entwickelt, jedoch noch lange nicht die abstrakte Wissenschaft, wie wir sie heute kennen. Fest in der realen Welt verankert, wurde sie vor allem durch Problemstellungen der physikalischen Beobachtung vorangetrieben. Zahlen waren nichts weiter als Messgrößen für reale Objekte und weit von den immateriellen Gedankengebilden der modernen Zahlentheorie entfernt. So wenig wie die Mathematik als eigenständige Wissenschaft existierte, so wenig gab es den reinen Mathematiker.

Im neunzehnten Jahrhundert änderte sich die Sichtweise allmählich in Richtung einer abstrakteren Mathematik. Zahlen und Symbole wurden von ihrer physikalischen Interpretation losgelöst betrachtet und entwickelten sich langsam, aber beharrlich zu immer abstrakter werdenden Größen. Mit der geänderten Sichtweise war es nun möglich, eine Gleichung der Form

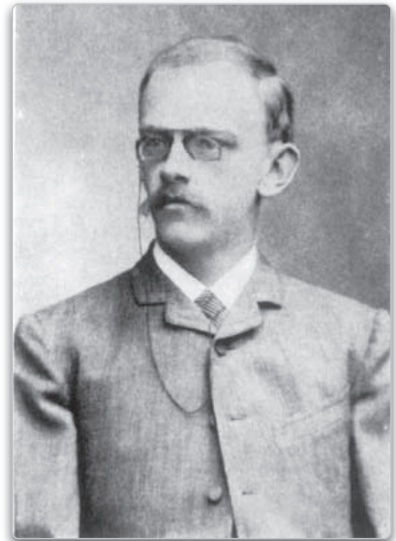
$$c^2 = a^2 + b^2$$

völlig unabhängig von ihrer pythagoreischen Bedeutung zu betrachten. In ihrer abstraktesten Interpretation lässt sie sich als mathematisches Spiel begreifen, das uns erlaubt, die linke Seite durch die rechte zu ersetzen. Die Variablen a , b und c degradieren in diesem Fall zu inhaltsleeren Größen, die in keinerlei Bezug mehr zu den Seitenlängen eines rechtwinkligen Dreiecks stehen.

Dass es richtig war, das mathematische Gedankengerüst von seiner physikalischen Interpretation zu trennen, wurde durch die Physik selbst untermauert. So machte die zu Beginn des zwanzigsten Jahrhunderts auf-

„Wenn es sich darum handelt, die Grundlagen einer Wissenschaft zu untersuchen, so hat man ein System von Axiomen aufzustellen, welche eine genaue und vollständige Beschreibung derjenigen Beziehungen enthalten, die zwischen den elementaren Begriffen jener Wissenschaft stattfinden. Die aufgestellten Axiome sind zugleich die Definitionen jener elementaren Begriffe und jede Aussage innerhalb des Bereiches der Wissenschaft, deren Grundlagen wir prüfen, gilt uns nur dann als richtig, falls sie sich mittelst einer endlichen Anzahl logischer Schlüsse aus den aufgestellten Axiomen ableiten lässt. Bei näherer Betrachtung entsteht die Frage, ob etwa gewisse Aussagen einzelner Axiome sich untereinander bedingen und ob nicht somit die Axiome noch gemeinsame Bestandteile enthalten, die man beseitigen muss, wenn man zu einem System von Axiomen gelangen will, die völlig voneinander unabhängig sind.

Vor allem aber möchte ich unter den zahlreichen Fragen, welche hinsichtlich der Axiome gestellt werden können, dies als das wichtigste Problem bezeichnen, zu beweisen, dass dieselben untereinander widerspruchlos sind, d.h., dass man aufgrund derselben mittelst einer endlichen Anzahl von logischen Schlüssen niemals zu Resultaten gelangen kann, die miteinander in Widerspruch stehen.“



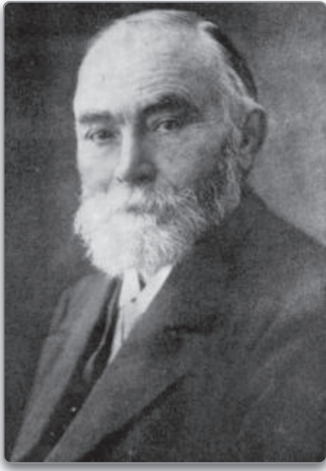
David Hilbert
(1862 – 1943)

Abbildung 1.5: Auszug aus Hilberts historischer Jahrhundertrede auf dem internationalen Kongress der Mathematiker in Paris

keimende Quantenmechanik deutlich, dass die damals wie heute merkwürdig anmutenden Effekte der Elementarteilchenphysik nur mithilfe abstrakter Modelle präzise erfasst werden können. Viele mathematische Konstrukte wie z. B. der *Hilbertraum* oder die *abstrakte Gruppe* konnten nachträglich zur Beschreibung der Natur eingesetzt werden, obwohl diese nichts mit unserer makroskopischen Anschauung gemeinsam haben.

Die zunehmende Abstraktion ließ Raum für Fragen zu, die sich in einer physikalisch gedeuteten Mathematik nicht stellen. Interpretieren wir z. B. die *euklidischen Axiome* (Abbildung 1.4) ausschließlich im Sinne der klassischen Geometrie, so erscheinen sie als reine Selbstverständlichkeit. Sie decken sich mit den Erfahrungen, die wir in der makroskopischen Welt tagtäglich machen und kaum jemand würde auf die Idee kommen, an ihnen zu zweifeln. Entsprechend lange galten die Axiome als unantastbar.

Die Situation ändert sich, sobald wir die Mathematik als ein abstraktes Wechselspiel von Symbolen und Regeln betreiben. Lösen wir uns von der intuitiven Interpretation der euklidischen Axiome, so stellt sich



Gottlob Frege
(1848 – 1925)

Abbildung 1.6: Gottlob Frege. Der im mecklenburgischen Wismar geborene Mathematiker zählt zu den Mitbegründern der mathematischen Logik und der analytischen Philosophie. Im Jahr 1879 eröffnete Frege mit seiner berühmten *Begriffsschrift* einen axiomatischen Zugang zur Logik [35]. Er führt darin die grundlegenden Konzepte und Begriffe ein, die wir auch heute noch in der Prädikatenlogik (Abschnitt 3.2) und den Logiken höherer Stufe (Abschnitt 3.3.2) verwenden. Sein Begriffsgestützte war deutlich weiter entwickelt als die Syllogismen des Aristoteles – der bis dato präzisesten Form des logischen Schließens. Die meiste Zeit seines Lebens war Frege ein überzeugter Verfechter des *Logizismus*. Er vertrat die Auffassung, dass die Mathematik ein Teil der Logik sei. In diesem Sinne müssen sich alle Wahrheiten auf eine Menge von Axiomen zurückführen lassen, die nach Freges Worten „*eines Beweises weder fähig noch bedürftig*“ seien. Er stand damit in einer Gegenposition zu anderen Mathematikern seiner Zeit, von denen viele die Logik als isoliertes Teilgebiet der Mathematik begriffen.

die Frage, ob diese ein vollständiges und widerspruchsfreies Gebilde ergeben. Im Jahr 1899 gelang es David Hilbert, diese Frage positiv zu beantworten. Er postulierte ein Axiomensystem, aus dem sich alle Sätze der euklidischen Geometrie ableiten lassen, ohne die verwendeten Symbole mit einer speziellen Interpretation zu versehen [46].

Inspiziert von den Anfangserfolgen stand die Mathematik um die Jahrhundertwende vollends im Zeichen der axiomatischen Methode. Das Führen eines Beweises wurde als der Prozess verstanden, Sätze durch die Anwendung wohldefinierter Schlussregeln aus einer kleinen Menge vorgegebener, a priori als wahr definierter Axiome abzuleiten. Eine spezielle Interpretation der Symbole war hierzu nicht erforderlich und im Grunde genommen auch gar nicht angestrebt. Die Mathematik wurde zu einem Spiel, das nach starren Regeln funktionierte, und das Führen eines Beweises zu einem mechanischen Prozess werden ließ.

Der deutsche Mathematiker David Hilbert war kein Unbekannter. Bereits zu Lebzeiten wurde er als Ikone gefeiert und beeinflusste wie kein anderer die Mathematik des beginnenden zwanzigsten Jahrhunderts. Im Jahr 1900 hielt Hilbert auf dem internationalen Kongress der Mathematiker in Paris eine wegweisende Rede, an der sich die weitere Stoßrichtung der gesamten Mathematik über Jahre hinweg orientieren sollte (vgl. Abbildung 1.5). In seiner Ansprache trug er 23 ungelöste Probleme vor, die für die Mathematik von immenser Wichtigkeit, aber bis dato ungelöst waren.

Bereits an zweiter Stelle forderte Hilbert die Weltgemeinschaft dazu auf, einen Beweis für die Widerspruchsfreiheit der arithmetischen Axiome zu liefern. Das Problem, das Hilbert hier ansprach, war von immenser Wichtigkeit für die gesamte Mathematik, schließlich adressieren die arithmetischen Axiome den vitalen Kern, auf dem alle Teilbereiche dieser Wissenschaft aufbauen. Solange es nicht gelingt, die Widerspruchsfreiheit formal zu beweisen, kann nicht mit Sicherheit ausgeschlossen werden, dass sich z. B. neben dem Theorem $1 + 1 = 2$ auch das Theorem $1 + 1 \neq 2$ aus den Axiomen ableiten lässt. Das verästelte Gebäude der Mathematik würde auf einen Schlag in Trümmern vor uns liegen.

Bereits wenige Jahre nach Hilberts Rede sollte die Wissenschaftsgemeinde erleben, wie real eine solche Gefahr wirklich war. Der deutsche Mathematiker Gottlob Frege (Abbildung 1.6) spürte sie am eigenen Leib, als er 1902 ein formales Axiomensystem für ein Teilgebiet der Mathematik aufstellte, das auf den ersten Blick so intuitiv und einfach erscheint wie kaum ein anderes. Die Rede ist von der *Mengenlehre*. Der zweite Band seiner *Grundgesetze der Arithmetik* schließt mit dem folgenden Nachwort [33, 34]:

„Einem wissenschaftlichen Schriftsteller kann kaum etwas Unerwünschteres begegnen, als dass ihm nach Vollendung einer Arbeit eine der Grundlagen seines Baues erschüttert wird.“

Doch wodurch wurde Freges Arbeit so grundlegend erschüttert, dass er sein gesamtes Werk gefährdet sah? Die Antwort ist in einem Brief von Bertrand Russell zu finden, den er im Jahr 1902 an Frege schickte – just zu der Zeit, als dieser sein mathematisches Werk vollendete. Aufbauend auf den Begriffen der naiven Mengenlehre definierte Russell die Menge aller Mengen, die sich nicht selbst als Element enthalten:

$$M := \{M' \mid M' \notin M'\}$$

Die Definition von M ist mit der damals verwendeten Mengendefinition von Georg Cantor vereinbar, führt bei genauerer Betrachtung jedoch unweigerlich zu einem Widerspruch. Da für jedes Element a und jede Menge M entweder $a \in M$ oder $a \notin M$ gilt, muss auch M entweder in sich selbst enthalten sein oder nicht. Die Definition von M offenbart uns jedoch das folgende erstaunliche Ergebnis:

$$M \in M \Rightarrow M \notin M, \quad M \notin M \Rightarrow M \in M$$

Der als *Russell'sche Antinomie* bekannte Widerspruch entlarvte den Cantor'schen Mengenbegriff als in sich widersprüchlich und lies Freges Gedankengerüst wie ein Kartenhaus in sich zusammenstürzen. Die Geschehnisse unterstrichen nachhaltig, wie wichtig eine widerspruchsfreie Fundierung der Mathematik tatsächlich war.

Zu den ersten, die sich der neugeborenen Herausforderung stellten, gehörten die britischen Mathematiker Bertrand Russell und Alfred North Whitehead. Sie starteten den Versuch, ein widerspruchsfreies Fundament zu errichten, auf dem die Mathematik für alle Zeiten einen sicheren Halt finden sollte. Nach zehn Jahren intensiver Arbeit war das Ergebnis greifbar: Die *Principia Mathematica* waren fertiggestellt (vgl. Abbildung 1.7). In einem dreibändigen Werk unternahmen Russell und Whitehead den Versuch, weite Bereiche der Mathematik mit den Mitteln der elementaren Logik formal herzuleiten. Ein großer Teil des Werks ist der *Typentheorie* gewidmet; einer widerspruchsfreien Konstruktion des Mengenbegriffs, mit dem die Art von Selbstbezug vermieden wird, die wenige Jahre zuvor die Mathematik in ihre größte Krise stürzte. Heute gilt die Typentheorie der *Principia* als überholt. An ihre Stelle tritt der formale axiomatische Aufbau der Mengenlehre durch Ernst Zermelo und Abraham Fraenkel, der die Russell'sche Antinomie ebenfalls beseitigt [32, 111].

Die mathematische Widerspruchsfreiheit ist eine unabdingbare Eigenschaft des mathematischen Schließens. Fehlt sie, so verkommt jedes formale System zu einem wertlosen Gedankengebilde. Warum dies so ist, wollen wir im Folgenden kurz begründen. Nehmen wir an, es gebe eine Aussage R , für die sich sowohl R als auch ihre Negation $\neg R$ innerhalb des Kalküls ableiten lassen. Die Situation erscheint wenig bedrohlich, wenn es sich um eine Aussage handelt, die uns nicht weiter interessiert. Eventuell ist R eine Aussage der Russell'schen Art, die uns ohnehin suspekt erscheint. Können wir den Kalkül vielleicht trotzdem sinnvoll einsetzen, wenn wir Aussagen dieser Form schlicht außen vor lassen?

Dass sich widersprüchliche Aussagen in einem Kalkül nicht isoliert betrachten lassen, liegt an den Schlussregeln der klassischen Logik. In Kapitel 3 werden Sie erlernen, wie sich das Theorem

$$\neg P \rightarrow (P \rightarrow Q)$$

aus den elementaren Axiomen der Logik herleiten lässt. In Worten besagt es: Ist P falsch, so folgt aus der Wahrheit von P die Wahrheit von Q . Substituieren wir R für P , so erhalten wir das Theorem

$$\neg R \rightarrow (R \rightarrow Q).$$

Da $\neg R$ eine wahre Aussage ist, lässt sich mithilfe der *Abtrennungsregel (Modus ponens)* das Theorem

$$R \rightarrow Q$$

herleiten. Nach Voraussetzung ist R ebenfalls wahr, so dass eine erneute Anwendung der Abtrennungsregel das Theorem Q hervorbringt. Da die Wahl von Q keinen Einschränkungen unterliegt, können wir eine beliebige Aussage für Q substituieren. Kurzum: In einem widersprüchlichen Kalkül lassen sich ausnahmslos alle Aussagen als wahr beweisen.

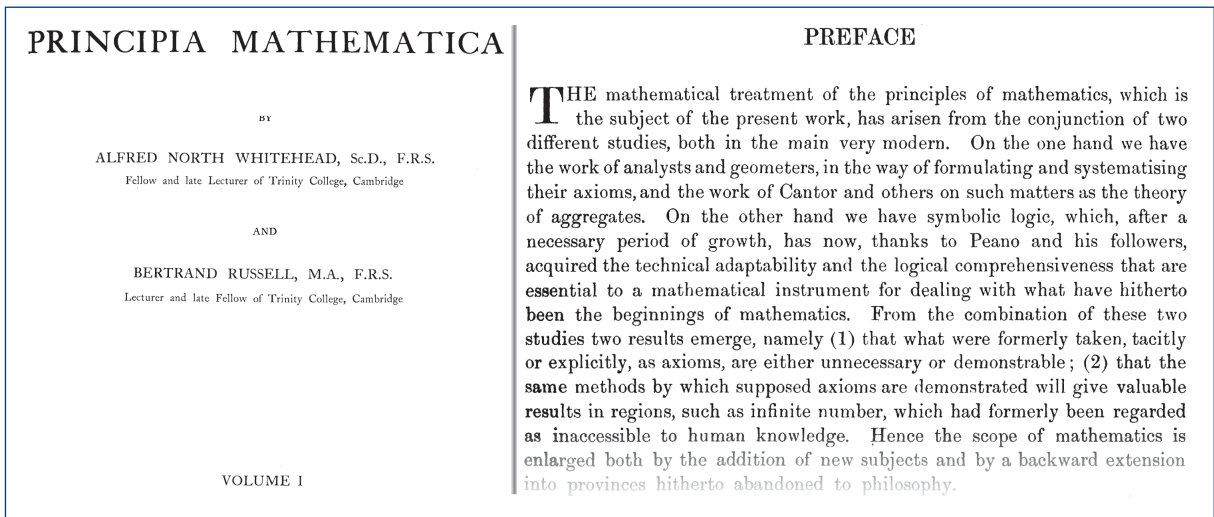


Abbildung 1.7: Die *Principia Mathematica*, erstmals erschienen in den Jahren 1910 bis 1913, ist eines der berühmtesten mathematischen Werke unserer Geschichte. Auf über 1800 Seiten, verteilt auf 3 Bände, unternahmen die Autoren den Versuch, alle mathematischen Erkenntnisse aus einer kleinen Menge von Axiomen systematisch herzuleiten.

Die Principia war in puncto Präzision jedem anderen Werk ihrer Zeit weit voraus. Sie fasste einen mathematischen Beweis als eine Folge von Regelanwendungen auf, durch die eine Aussage in endlich vielen Schritten aus einer festgelegten Menge von Axiomen abgeleitet wurde.

1.2.2 Metamathematik

Durch die zunehmende Beschäftigung mit den verschiedensten formalen Systemen entstand im Laufe der Zeit eine Metamathematik, die sich nicht mit der Ableitung von Sätzen *innerhalb* eines Kalküls beschäftigt, sondern mit Sätzen, die Aussagen *über* den Kalkül treffen. Drei Fragestellungen rückten in das Zentrum des Interesses:

■ Vollständigkeit

Ein formales System heißt *vollständig*, wenn jede wahre Aussage, die in der Notation des Kalküls formuliert werden kann, innerhalb desselben beweisbar ist. Mit anderen Worten: Für jede wahre Aussage P muss es eine endliche Kette von Regelanwendungen geben, die P aus den Axiomen deduziert. Beachten Sie, dass uns ein vollständiger Kalkül nicht preisgeben muss, wie eine solche Kette zu finden ist. Die Vollständigkeit garantiert lediglich deren Existenz.

■ Widerspruchsfreiheit

Ein formales System heißt *widerspruchsfrei*, wenn für eine Aussage P niemals gleichzeitig P und die Negation von P (geschrieben als $\neg P$) abgeleitet werden kann. Auf die immense Bedeutung der Widerspruchsfreiheit eines Kalküls sind wir weiter oben bereits eingegangen. Erfüllt ein formales System diese Eigenschaft nicht, so könnte es kaum wertloser sein. Es würde uns gestatten, jede beliebige Aussage zu beweisen.

■ Entscheidbarkeit

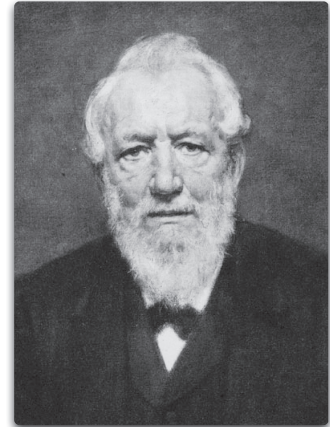
Ein formales System heißt *entscheidbar*, wenn ein systematisches Verfahren existiert, mit dem für jede Aussage entschieden werden kann, ob sie innerhalb des Kalküls beweisbar ist. Hinter der Eigenschaft der Entscheidbarkeit verbirgt sich nichts Geringeres als der Wunsch nach einer mechanisierten Mathematik. Wäre z. B. die Zahlentheorie vollständig und entscheidbar, so ließe sich für jede wahre zahlentheoretische Aussage auf maschinellem Wege ein Beweis konstruieren. Der Traum eines jeden Mathematikers würde wahr.

Hilbert war überzeugt, dass eine vollständige, widerspruchsfreie und entscheidbare Axiomatisierung der Mathematik möglich sei. Im Jahr 1929 wurden seine Hoffnungen durch die Arbeiten des jungen Mathematikers Kurt Gödel zusätzlich genährt, als dieser in seiner Promotionschrift die Vollständigkeit der Prädikatenlogik erster Stufe bewies [36]. Es war also möglich, einen Kalkül zu konstruieren, in dem sich jede wahre prädikatenlogische Formel in endlich vielen Schritten aus den Axiomen ableiten lässt. In diesen Tagen schien es nur eine Frage der Zeit zu sein, bis aus Hilberts Vermutungen Gewissheit werden würde.

1930 war das Jahr, in dem die Entwicklung eine abrupte Kehrtwende nehmen sollte. Am 8. September bekräftigte Hilbert vor der Versammlung Deutscher Naturforscher und Ärzte in seiner Heimatstadt Königsberg seine tiefe Überzeugung, dass es in der Wissenschaft keine unlösbaren Probleme gibt. Ein Auszug aus seiner Rede wurde in Form einer Radioansprache ausgestrahlt. Sie schließt mit den berühmten Worten:

„Wir dürfen nicht denen glauben, die heute mit philosophischer Miene und überlegenem Tone den Kulturuntergang prophezeien und sich in dem Ignorabimus gefallen. Für uns gibt es kein Ignorabimus, und meiner Meinung nach auch für die Naturwissenschaft überhaupt nicht. Statt des törichten Ignorabimus heiße im Gegenteil unsere Lösung: Wir müssen wissen, wir werden wissen.“

„Ignoramus et ignorabimus.“
(Wir wissen es nicht und wir werden es niemals wissen)



Emil Heinrich Du Bois-Reymond
(1818 – 1896)

„Für uns gibt es kein Ignorabimus.“ Mit diesem Satz bekräftigte David Hilbert seine Haltung, dass es in den Naturwissenschaften keine unbeweisbaren Wahrheiten gibt. Der Begriff *Ignorabimus* wurde durch den deutschen Gelehrten Emil Heinrich Du Bois-Reymond geprägt. Durch seine Leipziger Rede vor der Versammlung Deutscher Naturforscher und Ärzte löste er im Jahr 1872 einen Richtungstreit aus, der auf Jahre hinweg zu kontroversen Diskussionen in der Wissenschaftsgemeinde führen sollte. Er vertrat die Meinung, dass Begriffe wie das Bewusstsein niemals mit naturwissenschaftlichen Methoden erklärbar sein werden. Kurzum: Die Wissenschaft besitzt unüberwindbare Grenzen. *„Ich werde jetzt, wie ich glaube, in sehr zwingender Weise dartun, dass nicht allein bei dem heutigen Stand unserer Kenntnis das Bewusstsein aus seinen materiellen Bedingungen nicht erklärbar ist, was wohl jeder zugibt, sondern dass es auch der Natur der Dinge nach aus diesen Bedingungen nicht erklärbar sein wird.“* [8].

Der Unvollständigkeitsbeweis ist nicht nur aufgrund seiner inhaltlichen Tragweite von Bedeutung. Auch die trickreiche Beweisführung, mit der Gödel sein Resultat erzielte, zeugt von der Tiefgründigkeit des Ergebnisses. Gödel konnte zeigen, dass mathematische Schlussregeln, die Aussagen *über* Zahlen machen, selbst als Zahl verstanden werden konnten. Damit war es möglich, die Ebene der Zahlentheorie mit ihren Metaebenen zu vermischen. Aussagen sind nichts anderes als Zahlen, die selbst Aussagen über Zahlen tätigen. Auf diese Weise gelang es Gödel, Metaaussagen wie „*Aussage XYZ ist beweisbar*“ innerhalb des Systems zu codieren.

Um die Unvollständigkeit zu beweisen, wandte Gödel einen Trick an. Er konstruierte Aussagen, die auf sich selbst Bezug nehmen und so eine Metaaussage über sich selbst beinhalten. Auf diese Weise gelang es ihm, eine Formel zu konstruieren, die der Metaaussage „*Diese Formel ist nicht beweisbar*“ entspricht. Ist die Formel wahr, so lässt sie sich nicht beweisen und das zugrunde liegende Axiomensystem ist unvollständig. Ist sie falsch, so würde ein Beweis für eine falsche Aussage existieren und das Axiomensystem wäre nicht widerspruchsfrei. Mit anderen Worten: Erfüllt ein Axiomensystem die Eigenschaft der Widerspruchsfreiheit, so ist es zwangsläufig unvollständig.

Der Unvollständigkeitssatz zeigte zudem, dass die Widerspruchsfreiheit eines hinreichend aussagekräftigen formalen Systems nicht *innerhalb* des Systems selbst bewiesen werden kann. Gödel nutzte aus, dass in einem widersprüchlichen System alle Aussagen wahr sind, d. h., ein Kalkül ist genau dann widerspruchsfrei, wenn es eine einzige Aussage gibt, die nicht bewiesen werden kann. Gödel konnte jedoch zeigen, dass eine Aussage der Form „*es existiert eine unbeweisbare Aussage*“ ebenfalls nicht innerhalb des Systems bewiesen werden kann.

Die Rede ist im Originalton erhalten und ein unschätzbare Dokument der Zeitgeschichte. Sie zeigt nachdrücklich, wie überzeugt Hilbert von der Durchführbarkeit seines ehrgeizigen Programms wirklich war.

Zum Zeitpunkt seiner Rede wusste Hilbert noch nichts von den Ereignissen, die sich am Vortag an anderer Stelle in Königsberg abspielten. Es war die große Stunde eines vierundzwanzigjährigen Mathematikers, der mit der Präsentation seines Unvollständigkeitsatzes die Mathematik aus den Angeln hob. Derselbe Kurt Gödel, der kurze Zeit zuvor die Vollständigkeit der Prädikatenlogik bewies, konnte zeigen, dass die Arithmetik aus fundamentalen Überlegungen heraus unvollständig sein musste. Sein Ergebnis war so allgemein, dass es auf jedes axiomatische System angewendet werden konnte, das ausdrucksstark genug ist, um die Zahlentheorie zu formalisieren. Damit war nicht nur gezeigt, dass der logische Apparat der Principia Mathematica unvollständig war, sondern auch, dass jeder Versuch, die Principia oder ein ähnliches System zu vervollständigen, von Grund auf zum Scheitern verurteilt ist. Gödel versetzte dem Hilbert'schen Programm einen schweren Schlag, von dem es sich nie erholen sollte.

Gödels Arbeit verwies die Mathematik zweifelsohne in ihre Grenzen, ließ jedoch Hilberts dritte Vermutung außen vor. Auch wenn wir nicht in der Lage sind, einen widerspruchsfreien und zugleich vollständigen Kalkül für die Theorie der Zahlen zu konstruieren, so könnte die Frage nach der Entscheidbarkeit eines Kalküls dennoch positiv beantwortet werden. Der Unvollständigkeitsbeweis schließt nicht aus, dass ein systematisches Verfahren existiert, das für jede Aussage bestimmt, ob es innerhalb des Systems beweisbar ist oder nicht.

Die Hoffnung, dass zumindest diese letzte Frage positiv beantwortet werden könnte, wurde im Jahr 1936 vollends zerstört, als der britische Mathematiker Alan Turing seine grundlegende Arbeit *On computable numbers, with an application to the Entscheidungsproblem* der Öffentlichkeit präsentierte (Abbildung 1.8). Turings Arbeit ist für die theoretische Informatik aus zweierlei Gründen von Bedeutung. Zum einen gelang es ihm als einem der Ersten, die Grenzen der Berechenbarkeit formal zu erfassen und das Entscheidungsproblem negativ zu beantworten; die Jagd nach dem mathematischen Perpetuum Mobile war zu Ende. Zum anderen konstruierte Turing für seinen Beweis ein abstraktes Maschinenmodell, das dem Funktionsprinzip moderner Computer bereits sehr nahe kam. Aus heutiger Sicht bildet das gedankliche Gebilde der *Turing-Maschine* die Nahtstelle zwischen der abstrakten Mathematik des frühen zwanzigsten Jahrhunderts und der Welt der realen Rechenmaschinen. In gewissem Sinne ersann Turing den *missing link*, der die Mathematik in Form des Computers zum Leben erweckte.

Sachwortverzeichnis

Symbole

μ -Operator, 274
 μ -Rekursion, 274
 μ -rekursive Funktion, 275, 411
3SAT, 380, 403
4er-Nachbarschaft, 244
8er-Nachbarschaft, 244

A

Abbildung, 50
Ableitungsrelation, 96
Abschwächungsregel, 98
Absolute Adressierung, 296
Abstraktion, 300
Abtrennungsregel, 17
Abzählbare Sprache, 310
Abzählbarkeit, **59**, 310, 403
Ackermann-Funktion, 56, 403
Addierer
 Carry-look-ahead-, 116
 Carry-ripple-, 114
Adressierung
 absolute, 296
 indirekte, 296
 unmittelbare, 296
Äquivalenz, 87
Akkumulator, 296
AKS-Algorithmus, 33
Akzeptierende Turing-Maschine, 312
Akzeptierender Automat, 203, 403
Akzeptor, 203, 403
 Minimierung, 206
 Turing-, 312
Algorithmische Komplexität, 342
Algorithmus
 CYK-, 186, 405
 effektiver, 27

effizienter, 27
Gilmore-, 128, 407
Las-Vegas-, 32
Monte-Carlo-, 32
randomisierter, 32
rekursiver, 271
Robinson-, 131, 414
Strassen-, 391
Allgemeingültigkeit, 85, 403
 prädikatenlogische, 122
Allgemeinster Unifikator, 131, 403
Alphabet, 162
Antinomie, 403
 Russell'sche, **17**, 39, 415
Antivalenzoperator, 83
Äquivalenz, 87
 -klasse, 44
 -operator, 83
 -problem, 163, 403
 -relation, 49
Arbeitsband, 292
Asymptotische Komplexität, 350
Asymptotisches Wachstum, 403
Atomare Aussage, 82, 404
Atomare Formel, 83
Aufzählbare Sprache, 310
Aufzählbarkeit, 310
Ausdruck
 regulärer, 26, **176**, 217, 414
Ausgabealphabet
 von Transduktoren, 230
Ausgabeband, 296
Ausgabefunktion, 230
Ausgabeschaltnetz, 233
Aussage
 atomare, 82, 404
Aussagenlogik, 23, **82**, 404
 Normalformen, 91
Auswahlaxiom, 39

Automat

äquivalenter, 206
akzeptierender, 203, 403
DEA, 204, 405
deterministischer, 204
endlicher, 25, **201**, 406
Keller-, 223, 409
linearer, 244
Mealy-, 203, **234**, 411
Moore-, 203, **234**, 411
NEA, 209, 411
nichtdeterministischer, 208
Potenzmengen-, 211, 413
Produkt-, 219
reduzierter, 206
übersetzender, 203, 230
zellulärer, **243**, 252, 418
Automatenminimierung, 404
 von Akzeptoren, 206
 von Transduktoren, 231
Automatensynthese, 233, 404
Automatentheorie, 25, 201
Axiom, 35, 404
Axiome
 von Peano, 52

B

Backtracking, 145
Backus-Naur-Form, 181, 404
 erweiterte, 181
Bandalphabet
 von Turing-Maschinen, 279
Bandplatzfunktion, 365
Bar-Hillel-Theorem, 185
Barbier-Paradoxon, 34, 404
Basis, 306
BCD-Code, 249
Befehlszähler, 296

- Belegung, 84
 Berechenbarkeit, **254**, 302, 404
 Goto-, 266
 Loop-, 256
 Turing-, 281, 417
 While-, 261
 Berechenbarkeitstheorie, 253, 404
 Berechnungsmodell, 254, 404
 Beweis
 direkter, 66
 durch Widerspruch, 66
 induktiver, 65
 Beweistheorie
 aussagenlogische, 96
 prädikatenlogische, 124
 Biberfunktion, 337
 Bijektive Funktion, 51
 Bild, 51
 Binärbaum, 68
 balancierter, 68
 saturierter, 68
 Binäre Codierung, 282
 Binäre Suche, 387
 Binomialkoeffizient, 79
 Binomischer Lehrsatz, 79
 Bisimulation, **206**, **231**, 405
 Blättermenge, 68
 Blank-Symbol, 279
 Boolesche Algebra, 43
 Boolesche Funktion, 85
 Brute-Force-Methode, 363
- C**
- Cantor'sche Paarungsfunktion, 61, 75
 Cantor-Maschine, 312
 Carry bit, 114
 Carry-look-ahead-Addierer, 116
 Carry-ripple-Addierer, 114
 CD, 248
 Charakteristische Funktion, 309, 405
 Chomsky-Hierarchie, 25, **168**, 405
 Chomsky-Normalform, 179, 405
 Church'sche These, 254, **302**, 308, 405
 Church-Rosser-Eigenschaft, 301
 CLIQUE, 383
 Co-Komplexität, 369, 405
- COBOL, 24
 Code
 einschrittiger, 231
 Codierung
 binäre, 282
 unäre, 282
 Collatz-Funktion, 78
 Cook
 Satz von, **373**, 383
 Cook, Satz von, 415
 CYK-Algorithmus, 186, 405
- D**
- Datenspur, 287
 DEA, 204, 405
 Deadlock, 242
 Dedekind'scher Schnitt, 54
 Deduktion, 81
 Deduktionsbeweis, 66
 Definition
 rekursive, 65
 Definitionsmenge, 50
 Deklarative Programmierung, 138
 Deterministischer Automat, 204
 Diagonalisierung, 38, 405
 Diagonalisierungsargument, 62
 Diagonalsprache, 338
 Differenzmenge, 42
 Dirichlet'sches Schubfachprinzip, 90, 406
 Disjunktion, 82
 Disjunktive Form, 406
 Disjunktive Minimalform, 95
 Disjunktive Normalform, 95, 406
 kanonische, 93
 Distributivgesetz, 42, 98
 Divide and conquer, 356
 DNA computing, 308
 DNF, 94
 DVD, 248
 Dyck-Sprache, 165, 227
 Dynamische Logik, 295
 Dynamische Programmierung, **186**, **346**, 406
- E**
- Ebene
 Meta-, 82
 Objekt-, 82
 Einband-Turing-Maschine, 277
 Eingabealphabet
 von ε -Automaten, 213
 von DEAs, 204
 von Kellerautomaten, 224
 von NEAs, 209
 von Transduktoren, 230
 von Turing-Maschinen, 279
 Eingabeband, 296
 Einschrittiger Code, 231
 Element, 38
 Elementaroperatoren, 89
 Endlicher Automat, 25, **201**, 406
 Endlichkeitsproblem, 162, 406
 Endrekursion, 272
 Endzustand
 von ε -Automaten, 213
 von DEAs, 204
 von NEAs, 209
 von Turing-Maschinen, 279
 Enigma, 23
 Entscheidbare Sprache, 313
 Entscheidbarkeit, 19, **309**, 406
 Semi-, 309, 416
 Epsilon-Übergang, 212, 213, 406
 Erfüllbarkeit, 406
 aussagenlogische, 85
 prädikatenlogische, 122
 Erfüllbarkeitsäquivalenz, 123, 406
 Erreichbarkeitsanalyse, 241
 Euklidische Axiome, 15
 Euler-Kreis, 29
 EXP, 367, 406
- F**
- Faktorisierungsregel, 134
 Faktum
 in Prolog, 138
 Ferritkernspeicher, 24
 Fibonacci-Folge, 390
 Finalzustand

- von ε -Automaten, 213
- von DEAs, 204
- von NEAs, 209
- von Turing-Maschinen, 279

Fixpunkt, 208

- operator, 301

Fleißiger Biber, 337

Flipflop, 233

Formale Sprache, 25, **162**, 406

Formales System, 12, 35, 407

Formel

- aussagenlogische, 82

- bereinigte, 119

- erfüllbarkeitsäquivalente, 123

- geschlossene, 119

- prädikatenlogische, 119

Formulario-Projekt, 53

FORTRAN, 24

Funktion, **44**, 50

- μ -rekursive, 275, 411

- Ackermann-, 56, 403

- bijektive, 51

- boolesche, 85

- charakteristische, 309, 405

- injektive, 51

- partielle, **51**, 258, 412

- platzkonstruierbare, 366

- primitiv-rekursive, 269, 413

- surjektive, 51

- totale, 51, 417

- unberechenbare, 319

- zeitkonstruierbare, 361

Funktions

- variable, 147

Funktionsstabelle, 85

Funktionswert, 51

G

Gegenbeispiel, 109

Generative Grammatik, 25

Gilmore-Algorithmus, 128, 407

Gleichheitsrelation, 47

Gödelisierung, 291, 407

Gödelnummer, **291**, 321, 339, 407

Goto-Berechenbarkeit, 266

Goto-Programm, 264, 407

Goto-Sprache, 264, 407

Grad

- von Polynomen, 353

Grammatik, 162–164, 407

- eindeutige, 167

- generative, 25

- kontextfreie, 168, **179**, 409

- kontextsensitive, 168, **191**, 410

- mehrdeutige, 167

- rechtslineare, 170

- reguläre, 168, **170**, 414

Gray-Code, 231

Greibach-Normalform, 197, 407

Grundinstanz, 127, 407

Grundlagenkrise, 14

Grundmenge, 120

Grundsubstitution, 120

H

Halteproblem, 21, **319**, 407

- allgemeines, 320

- auf leerem Band, 322, 408

- spezielles, 339, 416

Hamilton-Kreis, 30

Hamilton-Problem, 364, 408

Hardware-Entwurf, 112

Haskell, 300

Head recursion, 272

Herbrand

- Satz von, 126, 415

Herbrand-Interpretation, 126, 408

Herbrand-Modell, 126, 408

Herbrand-Universum, 125, 408

Hilbert-Kalkül, 98, 408

Hilbert-Wüste, 76

Hilberts Hotel, 61

Horn-Formel, 143

Hülle

- Kleene'sche, 162

- reflexiv-transitive, 47, 48

- transitive, 47

Huffman-Normalform, 234

I

Identität, 47

Ignorabimus, 19

Imaginäre Einheit, 71

Implikation, 82

Indirekte Adressierung, 296

Individuenbereich, 120

Induktion

- strukturelle, 68, 416

- vollständige, **66**, 418

Induktionsaxiom, 408

Induktiver Beweis, 65

Injektive Funktion, 51

Instanzen, 99

Interpretation, **84**, **120**, 408

- Herbrand-, 126, 408

Inverses Element, 42

Inzidenzmatrix, 239, 408

Irrationale Zahl, 54

Isomorphie, 212

Iterationslemma, 185

K

Kalkül, 12, 35, **96**, 408

- Hilbert-, 98, 408

- Lambda-, 300

- Resolutions-, **104**, 130, 414

- Tableau-, **109**, 135, 416

- Widerspruchs-, 97, 418

Kapazität, 241

Kardinalität, 38, 58

Kardinalzahl, 64, 408

Kartesisches Produkt, 44

KDNF, 93

Kelleralphabet, 224

Kellerautomat, 223, 409

- deterministischer, 228, 229

Kellerspeicher, 409

Kettenregel, 180

KKNF, 93

Klausel, 95, 409

- leere, 95

Klauseldarstellung, 95

Kleene

- Satz von, 264, 415

Kleene'sche Hülle, 162

Kleene'sche Normalform, **268**, 304, 409

KNF, 94

- Königsberger Brückenproblem, 29, 409
 Kollisionsfreiheit, 135
 Kommutativgesetz, 42
 Komplementärautomat, 218
 Komplementärmenge, 42
 Komplexe Zahl, 71
 Komplexitätsklasse, 29, **356**, 409
 komplementäre, 369, 405
 Komplexitätstheorie, 341, 409
 Komposition, 270
 Konfiguration, 409
 globale, 244
 lokale, 244
 von ϵ -Automaten, 213
 von DEAs, 205
 von Kellerautomaten, 225
 von NEAs, 210
 von Petri-Netzen, 239
 von Turing-Maschinen, 280
 Konjunktion, 82
 Konjunktive Form, 409
 Konjunktive Minimalform, 95
 Konjunktive Normalform, 95, 409
 kanonische, 93
 Konklusion, 98
 Kontextfreie Grammatik, 168, **179**, 409
 Kontextfreie Sprache, **179**, 226, 410
 Kontextsensitive Grammatik, 168, **191**,
 410
 Kontextsensitive Sprache, 191, 410
 Kontinuum, 64
 Kontinuumshypothese, 64
 Kontraposition, 98
 Konversionsregel, 300
 Kopfrekursion, 272
 Kopfzelle, 293
 Korrektheit, 96
 Korrespondenzproblem, 326, 413
 binäres, 340
 modifiziertes, 328
- L**
- Lambda-Kalkül, 300
 Lambda-Term, 300
 Landau-Symbole, 349, 410
 Las-Vegas-Algorithmus, 32
 Last-In-First-Out, 224
 Latch, 233
 Laufzeitfunktion, 359
 LBA-Problem
 erstes, 318
 zweites, 318
 Lebendigkeit
 von Petri-Netzen, 241
 Leere Klausel, 95
 Leere Menge, 39
 Leerheitsproblem, 162, 410
 LIFO, 224
 Lineare Rekursion, 272
 Lineare Suche, 387
 Linearer Automat, 244
 Linksableitung, 165, 410
 Literal, 92, 410
 Logik, 81, 410
 Aussagenlogik, 23, **82**, 404
 dynamische, 295
 höherer Stufe, 147, 410
 Prädikatenlogik, 117, 413
 Logikminimierung, 95
 Logische Folgerung, 87
 Logische Programmierung, 138
 Logizismus, 16, 35
 Loop-Berechenbarkeit, 256
 Loop-Programm, 254, 410
 Loop-Sprache, 254, 410
- M**
- Mächtigkeit, 58, 410
 Makro, 257
 Marke, 238, 412
 Markenindex, 265
 Markierungsgleichung, 241
 Maschinenkomposition, 288
 Matrix, 122
 Maxterm, 92, 411
 Mealy-Automat, 203, **234**, 411
 Mehrband-Turing-Maschine, 286
 Mehrdeutigkeitsproblem, 411
 Mehrspur-Turing-Maschine, 286
 Menge, 38
 disjunkte, 41
 leere, 39
 Null-, 40
 unentscheidbare, 319
 wohlgeordnete, 72
 Mengenalgebra, 42
 Mengenlehre, 16
 axiomatische, 39
 Cantor'sche, 38
 Fraenkel-, 39
 Zermelo-Fraenkel-, 39
 Mengenoperation, 41
 Metaebene, 82
 Millennium-Probleme, 32
 Minimalform, 95
 disjunktive, 95
 konjunktive, 95
 Minimierung, 404
 Logik-, 95
 von Akzeptoren, 206
 von Transduktoren, 231
 Minterm, 92, 411
 Miranda, 300
 ML, 300
 Modell, **84**, **121**, 411
 Herbrand-, 126, 408
 Modellrelation, **84**, 120, 121
 Modus barbara, 150
 Modus ponens, 17, **98**, 411
 Monte-Carlo-Algorithmus, 32
 Moore-Automat, 203, **234**, 411
 Moore-Nachbarschaft, 244
- N**
- Nachbarschaft
 Moore-, 244
 Von-Neumann-, 244
 Nachbarschaftsfunktion
 von zellulären Automaten, 243
 Natürliche Zahl, 39, 52
 NEA, 209, 411
 Negation, 82
 Negationsnormalform, 122, 411
 Nerode-Relation, 175
 Neutrales Element, 42
 NEXP, 367, 411
 Nichtdeterministischer Automat, 208
 Nichtterminal, 163

- Nonterminal, 163, 164
 Normalform, 92, 179
 aussagenlogische, 91
 Chomsky-, 179, 405
 disjunktive, 406
 Greibach-, 197, 407
 Huffman-, 234
 kanonische
 disjunktive, 93
 konjunktive, 93
 Kleene'sche, **268**, 304, 409
 konjunktive, 409
 Negations-, 122, 411
 prädikatenlogische, 122
 NP, 359, 411
 NP-hart, 372, 412
 NP-vollständig, 31, **371**, 412
 NPSPACE, 365, 412
 Nullmenge, **40**
- O**
- O-Kalkül, 349
 O-Notation, 349, 412
 Obermenge, 41
 Objektenebene, 82
 ODER-Operator, 82
 Ogdens Lemma, 185
 Operator, 51, 55
 Operatorensystem
 vollständiges, 89
 Orakel, 364
 Ordnung
 lineare, 50
 partielle, 50
 totale, 50
 Ordnungsrelation, 50
- P**
- P, 359, 412
 P-NP-Problem, 372, 412
 Paarungsfunktion, **61**, 258, 412
 Cantor'sche, 61, 75
 Palindromsprache, 196, 226
 Parikh-Vektor, 239, 412
 Paritätsbit, 246
 Paritätscode, 246
 Partielle Funktion, **51**, 258, 412
 Partition, 44
 Peano-Axiome, 52, 412
 Petri-Netz, 238, 412
 Pfad
 geschlossener, 109
 offener, 109
 vollständiger, 110
 widerspruchsfreier, 109
 widersprüchlicher, 109
 Phrasenstrukturgrammatik, 168
 Phrasenstruktursprache, 193
 Pigeonhole principle, 90, 406
 Platzkonstruierbare Funktion, 366
 Polynom, 353
 Polynomielle Reduktion, 371, 413
 Pop-Operation, 224
 Positionsspur, 287
 Post'sche Tag-Maschine, 295
 Post'sches Korrespondenzproblem, 326, 413
 binäres, 340
 modifiziertes, 328
 Potenzmenge, 44
 Potenzmengenautomat, 211, 413
 Prädikat, 117
 -variable, 147
 Prädikatenlogik, 117, 413
 Normalformen, 122
 zweiter Stufe, 147
 Prämisse, 98
 Pränex-Form, 122, 413
 Primitiv-rekursive Funktion, 269, 413
 Primitive Rekursion, 269, 413
 Principia Mathematica, 17, 18
 Problem
 unentscheidbares, 319
 Produktautomat, 219
 Produktion, 164
 Programm, 296
 Goto-, 264, 407
 Loop-, 254, 410
 While-, 260, 418
 Programmierung
 deklarative, 138
 dynamische, **186**, **346**, 406
 logische, 138
 Prolog, 413
 PSPACE, 365, 414
 Pumping-Lemma, 185, 414
 für kontextfreie Sprachen, 182
 für reguläre Sprachen, 172
 Push-Operation, 224
- Q**
- Quantenrechner, 308
- R**
- Rabin und Scott
 Satz von, 210, 415
 Rad des Theodoros, 65
 Random access machine, 296
 Randomisierter Algorithmus, 32
 Rationale Zahl, 53
 Rechtsableitung, 165, 414
 Rechtslineare Grammatik, 170
 Reduktion, 414
 polynomielle, 371, 413
 Reduktionsbeweis, 325, 380
 Reelle Zahl, 54
 Regel, 35, 164
 in Prolog, 138
 Registermaschine, 296, 414
 verallgemeinerte, 296
 Reguläre Grammatik, 168, **170**, 414
 Reguläre Sprache, **170**, 216, 414
 Regulärer Ausdruck, 26, **176**, 217, 414
 Rekurrenzgleichung, 390
 Rekursion, 65
 μ -, 274
 lineare, 272
 primitive, 269, 413
 verschachtelte, 272
 verzweigende, 272
 wechselseitige, 272
 Rekursiv aufzählbare Sprache, 310
 Rekursive Definition, 65
 Relation, 44
 Ableitungs-, 96
 Äquivalenz-, 49
 inverse, 47
 Ordnungs-, 50

Relationenattribut, 45
 Relationenprodukt, 47
 Resolutionsbaum, 105
 Resolutionskalkül, 414
 aussagenlogisches, 104
 prädikatenlogisches, 130
 Resolutionsregel, 105
 Resolvente, 105
 Rice
 Satz von, 322, 415
 Robinson-Algorithmus, 131, 414
 Rucksackproblem, 186, **342**, 415
 Russell'sche Antinomie, **17**, 39, 415

S

SAT, 415
 Satz
 von Cantor, 64, 415
 von Cook, **373**, 383, 415
 von Herbrand, 126, 415
 von Kleene, 264, 415
 von Myhill-Nerode, 174, 221
 von Rabin und Scott, 210, 415
 von Rice, 322, 415
 von Savitch, 367, 415
 Savitch
 Satz von, 367, 415
 Schaltnetz, 233
 Ausgabe-, 233
 Übergangs-, 233
 Schaltwerk, 25, 233
 Scheme, 300
 Schleife
 While-, 260
 Schleifensatz, 185
 Schlussregel, 416
 Schnittmenge, 41
 Schubfachprinzip, 90, 406
 Selbstabbildung, 51
 Semantik, 82, 416
 Semi-entscheidbare Sprache, 313
 Semi-entscheidbarkeit, 313
 Semi-Entscheidbarkeit, 309, 416
 Semi-Thue-System, 169
 Sicherheit
 von Petri-Netzen, 241

Sierpinski-Dreieck, 245, 252
 Skolem-Form, 123, 416
 Speicher, 296
 Speichervektor, 255
 Spezielles Halteproblem, 339, 416
 Sprache
 abzählbar, 310
 Diagonal-, 338
 entscheidbare, 313
 formale, 25, **162**, 406
 Goto-, 264, 407
 inhärent mehrdeutige, 167
 kontextfreie, **179**, 226, 410
 kontextsensitive, 191, 410
 Loop-, 254, 410
 Palindrom-, 196, 226
 Phrasenstruktur-, 193
 reguläre, **170**, 216, 414
 rekursiv aufzählbare, 310
 semi-entscheidbare, 313
 unentscheidbare, 319
 While-, 260, 418
 Stack, 224, 259
 Stapel, 224, 259
 Startkonfiguration, 245
 Startsymbol, 164
 Startvariable, 164
 Startzustand
 von ε -Automaten, 213
 von DEAs, 204
 von Kellerautomaten, 224
 von NEAs, 209
 von Transduktoren, 230
 von Turing-Maschinen, 279
 Stirling-Zahl, 73
 Strassen-Algorithmus, 391
 Strukturelle Induktion, 68, 416
 Substitution, 119
 Substitutionstheorem, 89
 Suche
 binäre, 387
 lineare, 387
 Surjektive Funktion, 51
 Syllogismus, 150
 Syntax, 82, 416
 Syntaxbaum, 167, 416
 Synthese
 von Automaten, 233, 404, 416

T

Tableau
 geschlossenes, 110
 offenes, 110
 vollständiges, 110
 widerspruchsfreies, 110
 Tableukalkül, 416
 aussagenlogisches, 109
 prädikatenlogisches, 135
 Tag-Maschine, 295
 Tail recursion, 272
 Taubenschlagprinzip, 90, 406
 Tautologie, 416
 aussagenlogische, 85
 prädikatenlogische, 122
 Teile-und-herrsche-Prinzip, 356
 Teilformel, 83
 Teilmenge, 41
 Terminal, 163
 Terminalalphabet, 164
 Terminierungsmenge, 261
 Thue-System, 169
 Totale Funktion, 51, 417
 Totalordnung, 50
 Trägermenge, 41
 Transduktor, 203, **230**, 248, 417
 Minimierung, 231
 Transistor, 24
 Turing-Akzeptor, 312
 Turing-Berechenbarkeit, 281, 417
 Turing-Bombe, 23
 Turing-Maschine, 20, **277**, 417
 akzeptierende, 312
 Einband-, 277
 einseitig beschränkte, 285
 Komposition, 288
 linear beschränkte, 285
 Mehrband-, 286
 Mehrspur-, 286
 universelle, 289, 418
 zelluläre, 293
 Turing-Test, 279

U

Überabzählbarkeit, 59, 417

Übergangsfunktion, 255
 Übergangsrelation, 314
 Übergangsschaltnetz, 233
 Übersetzender Automat, 203, 230
 Umkehrabbildung, 51
 Unäre Codierung, 282
 Unberechenbarkeit, 319, 417
 UND-Operator, 82
 Unendlichkeit, 57
 Unentscheidbarkeit, 319, 417
 Unerfüllbarkeit, 85, 417
 Unifikation, 130, 417
 Unifikator, 130, 418
 allgemeinster, 131, 403
 Universalmenge, 41
 Universelle Turing-Maschine, 289, 418
 Universum, 120
 Herbrand-, 125, 408
 Unmittelbare Adressierung, 296
 Untermenge, 41
 Up-Arrow-Notation, 56, 418
 Urbild, 51

V

Variable, 82, 164
 Funktions-, 147
 Prädikat-, 147
 Venn-Diagramm, 41, 42
 Vereinigungsmenge, 41
 Verklemmungsfreiheit
 von Petri-Netzen, 242
 Verschachtelte Rekursion, 272
 Verzweigende Rekursion, 272

Vollständige Induktion, **66**, 418
 Vollständiges Operatorensystem, 89
 Vollständigkeit, 18, 96
 Von-Neumann-Nachbarschaft, 244

W

Wahrheitstabelle, 85
 Wahrheitstafel, 85
 Wechselseitige Rekursion, 272
 While-Berechenbarkeit, 261
 While-Programm, 260, 418
 While-Schleife, 260
 While-Sprache, 260, 418
 Widerspruchsbeweis, 66
 Widerspruchsfreiheit, 19
 Widerspruchskalkül, 97, 418
 Wohlordnung, 72
 Wort, 162
 Wortproblem, 162, 418
 Wurzelschnecke, 65

X

XOR-Operator, 83

Z

Z3, 22
 Zahl
 ganze, 39, 53
 große, 55
 irrationale, 54
 natürliche, 39, 52

 nichtnegative, 39
 positive, 39
 rationale, 53
 reelle, 54
 Zeichen, 162
 Zeitkonstruierbare Funktion, 361
 Zelle, 243
 Zellmenge, 243
 Zelluläre Turing-Maschine, 293
 Zellulärer Automat, **243**, 252, 418
 Zentraleinheit, 296
 Zermelo-Fraenkel-Mengenlehre, 39
 Zermelo-Mengenlehre, 39
 Zielmenge, 50
 Zustand, 202
 äquivalenter, 206
 Zustandsband, 292
 Zustandsmenge
 von ε -Automaten, 213
 von DEAs, 204
 von Kellerautomaten, 224
 von NEAs, 209
 von Transduktoren, 230
 von Turing-Maschinen, 279
 von zellulären Automaten, 243
 Zustandsübergangsdigramm, 202
 Zustandsübergangsfunktion
 von ε -Automaten, 213
 von DEAs, 204
 von Kellerautomaten, 224
 von NEAs, 209
 von Transduktoren, 230
 von Turing-Maschinen, 279
 von zellulären Automaten, 243